МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
ІНСТИТУТ КІБЕРНЕТИКИ імені В.М. ГЛУШКОВА
НАН УКРАЇНИ
ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО
“ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ”

А.Д. Кожухівський, І.Д. Горбенко, В.К. Задірака,
Ю.І. Горбенко, Г.І. Гайдур, О.А. Кожухівська, В.В. Марченко

ВСТУП ДО КВАНТОВОЇ КРИПТОЛОГІЇ

Рекомендовано до друку Вченою радою
Державного університету
інформаційно-комунікаційних технологій,
протокол № 8 від 18.12.2023 р.

Київ - Харків – 2024

УДК 004.056.5 (075.8)
DOI : 10.31673/IQC092026
ISBN : 978-617-8580-53-7

Рецензенти:
О.Г. КОРЧЕНКО – д-р, техн. наук, професор, завідувач кафедри безпеки інформаційних технологій Національного авіаційного університету, Заслужений діяч науки і техніки України, лауреат Державної премії України в галузі науки і техніки;
В.М. МАКСИМОВИЧ – д-р, техн. наук, професор, завідувач кафедри безпеки інформаційних технологій Національного університету "Львівська політехніка", лауреат Державної премії України в галузі науки і техніки.
А.М. ОЛЕКСІЙЧУК – професор Спеціальної кафедри № 1 ІСЗЗІ КПІ ім. Ігоря Сікорського д-р, техн. наук, професор, лауреат Державної премії України в галузі науки і техніки.

Вступ до квантової криптології [Текст]: Навчальний посібник (Для студентів техн. спец. вищ. навч. закл.) / [А.Д. Кожухівський, І.Д. Горбенко, В.К. Задірака і ін.]; М-во освіти і науки України, Державний університет інформаційно - комунікаційних технологій.- К.: ТОВ "Тропеа", 2024. – 597 с.

У посібнику розглянуті питання з квантової інформатики та квантової криптографії. В першій частині посібника розглянуті основні принципи квантової інформатики. Безпека сучасних інформаційних систем та технологій грунтується на стійкості криптографічних перетворень, які вони використовують для криптографічної обробки інформації. Використовуючи квантові алгоритми Шора та Гровера, певні математичні задачі можна розв'язувати навіть з поліноміальною складністю. Ідеї використання потужностей квантового середовища висунули Пауль і Фейман. Важливим є не тільки сам факт побудови квантового комп'ютера, а й технічні характеристики, якими володітиме такий комп'ютер.

Зміст посібника відповідає спеціальності 125 – Кібербезпека і може бути використаний студентами, магістрами і здобувачами наукового ступеня доктора філософії всіх кафедр Навчально-наукового інституту Захисту інформації Державного університету інформаційно-комунікаційних технологій, Харківського Національного університету імені В.Н. Каразіна, а також споріднених спеціальностей інших вищих навчальних закладів України.

Ключові слова: Квантова інформатика, телепортація, алгоритм Шора, алгоритм Гровера, квантова криптографія, електронний цифровий підпис.

УДК 004.056.5 (075.8)

В авторській редакції
© Кожухівський А.Д., Горбенко І.Д.,
© Задірака В.К.,
© Горбенко Ю.І., Гайдур Г.І.,
© Кожухівська О.А., Марченко В.В. 2024

ЗМІСТ

ПЕРЕДМОВА 8
1 ОСНОВНІ ПРИНЦИПИ КВАНТОВОЇ ІНФОРМАТИКИ 13
1.1 Кубіт 13
1.2 Однокубітові логічні елементи 15
2 ПРИКЛАДИ ОДНОКУБІТОВИХ ЕЛЕМЕНТІВ 19
2.1 Логічний елемент NOT 19
2.2 Логічний елемент Z 21
2.3 Логічний елемент Адамара H 23
2.4 Логічний елемент Y 24
2.5 Логічний елемент $\frac{\pi}{8}$ 25
2.6 Логічний елемент S 27
2.7 Логічний елемент Ф 29
3 ДВОКУБІТОВІ СИСТЕМИ 32
3.1 Двокубітові стани і оператори 32
3.2 Двокубітовий квантовий логічний елемент CNOT 33
3.3 Можливість копіювання невідомого біта за допомогою двокубітового квантового елемента CNOT 37
4 КВАНТОВІ СХЕМИ 40
4.1 Однокубітові квантові схеми 40
4.2 Однокубітові квантові схеми, які побудовані із однакових елементів 41
4.3 Однокубітові квантові схеми, які побудовані із різних елементів 44
4.4 Двокубітові квантові схеми 44
4.5 Квантова схема, що створює стан Белла 47
5 ПРИКЛАДНІ АСПЕКТИ КВАНТОВОЇ ІНФОРМАТИКИ 51
5.1Алгоритм розпізнавання функцій (алгоритм Дойча) 51
5.2 Зверхущільнене кодування 54
5.3 Використання стану Белла в квантовій телепортації 55
6 ІНФОРМАЦІЙНА ЕНТРОПІЯ 58
6.1 Класична інформаційна ентропія 58
6.2 Квантова інформаційна ентропія 58
7 КВАНТОВА ТЕОРІЯ ОБЧИСЛЕНЬ 62
7.1 Наочне уведення в квантові представлення 62
7.2 Елементи квантової механіки 64
7.2.1 Принципи класичної фізики 64
7.2.2 Протиріччя класичної механіки 67
7.2.3 Стан в квантовій механіці 70
7.2.4 Спостережуваність 75
7.2.5 Еволюція 85
7.2.6 Вимірювання 88

7.3 Квант інформації – кубіт 100
7.4 Елементарні квантові операції 103
7.5 Універсальні квантові схеми 110
7.6 Паралельність квантових обчислень 113
7.7 Неможливість клонування квантового стану 114
7.8 Щільне кодування 116
7.9 Квантові машини Т'юринга 118
7.10 Квантовий алгоритм пошуку в несортованій базі даних 119
7.11 Квантовий алгоритм факторизації чисел Шора 127
7.12 Квантовий комп'ютер 132
8 КВАНТОВА ТЕЛЕПОРТАЦІЯ 137
8.1 Телепортація квантового стану по класичному каналу 137
8.2 Експериментальна телепортація 140
8.3 Квантова однокубітова телепортація 143
8.4 Квантова телепортація переплутаності 145
8.5 "Класична" телепортація 147
9 КВАНТОВІ КЛЮЧІ І ЇХ ПЕРЕДАЧА 150
9.1 Класичні коди, що виправляють помилки 152
9.2 Квантові коди, що виправляють помилки 155
9.3 Квантовий ефект Зенона і пригнічення помилок 160
9.4 Стійкі квантові гейти 162
10 ВСТУП В ТЕОРІЮ ТА ПРАКТИКУ КВАНТОВИХ ОБЧИСЛЕНЬ 166
10.1 Основні поняття і сутність квантової криптографії 174
10.2 Протоколи квантового поширення ключа 178
10.2.1 Протокол BB84 178
10.2.2 Приклад шифрування протоколом BB84 179
10.2.3 Зниження рівня помилок і збільшення таємності ключа 179
10.2.4 Протокол Екерта 180
10.2.5 Практична реалізація протоколу BB84 181
10.2.6 Протокол B92 182
10.2.7 Практичні аспекти квантової криптографії 184
10.3 Квантові протоколи узгодження ключів 184
10.3.1 Квантовий протокол узгодження криптографічного ключа BB84 185
10.3.2 Стан робіт у галузі квантової криптографії та протоколів 196
11 ФІЗИЧНІ ОСНОВИ КВАНТОВОЇ КРИПТОГРАФІЇ 201
11.1 Протокол квантового щільного кодування 228
12 ПРАКТИЧНА РЕАЛІЗАЦІЯ КВАНТОВИХ АЛГОРИТМІВ 241
12.1 Алгоритм факторизації цілих чисел 241
12.2 Методи перехоплення інформації у системах квантової криптографії 245
12.3 Алгоритм факторизації Шора 254
12.3.1 Приклад факторизації за алгоритмом Шора 255
12.3.2 Опис програми факторизації 257

12.4 Квантовий алгоритм Гровера 259
12.5 Особливості методу Гровера при використанні в несортованій базі даних 261
12.6 Практична реалізація методу Гровера 278
12.7 Сутність та особливості реалізації методу Гровера на класичному комп'ютері для симетричного криптоаналізу 279
12.7.1 Криптоаналіз з використанням методу Гровера 280
12.7.2 Модель та реалізація методу Гровера на класичному комп'ютері.....280
12.8 Аналіз можливостей та особливості програмування задач криптології на квантовому комп'ютері 283
12.9 Особливості та можливості квантового комп'ютера..... 284
12.10 Аналіз можливостей та наявності забезпечення для вирішення задач криптоаналізу 286
12.11 Особливість квантового програмування для методу Гровера 286
12.11.1 Квантовий Оракул 287
12.11.2 Реалізація методу Гровера на квантовому комп'ютері 288
12.12 Схеми застосування методу Гровера на квантовому комп'ютері для квантових регістрів з використанням хмарних сервісів 291
13 АНАЛІЗ СКЛАДНОСТІ ТА МОЖЛИВОСТІ РЕАЛІЗАЦІЇ КВАНТОВОЇ АТАКИ СТОСОВНО ПОСТКВАНТОВОГО АЛГОРИТМУ ЕЛЕКТРОННОГО ПІДПИСУ RAINBOW..... 298
13.1 Параметри RAINBOW 298
13.2 Атаки на Rainbow 299
14 ПОСТКВАНТОВА КРИПТОГРАФІЯ ТА МЕХАНІЗМИ ЇЇ РЕАЛІЗАЦІЇ... 303
14.1 Аналіз основних проблем створення квантового комп'ютера та розробки математичного забезпечення..... 304
14.1.1 Аналіз досягнень в побудові квантового комп'ютера..... 304
14.2 Аналіз основних загроз відносно криптографічних перетворень у постквантовий період 307
14.3 Вимоги NIST щодо можливих кандидатів постквантових криптопримітивів 310
14.3.1 Вимоги до квантово-захищених алгоритмів ЕП ETSI..... 312
14.4 Порівняльний аналіз перспективних механізмів ЕП..... 316
14.5 Сутність механізму ЕП на основі перетворення на алгебраїчних решітках (LB-криптографія) 320
14.5.1 Сутність механізму ЕП на основі мультиваріативного квадратичного перетворення (MQ-криптографія)..... 321
14.5.2 Порівняльний аналіз ЕП для постквантового періоду 323
14.6 Модель порушника систем електронних цифрових підписів в умовах постквантового криптоаналізу 327
14.6.1 Загальна модель порушника та загроз ЕЦП 327
14.6.2 Модель порушника та загроз ЕЦП в умовах появи квантового комп'ютера 331

14.6.3 Оцінка загроз схеми ЕЦП для класичних та квантових атак 339
15 МЕТОДИ ТА МЕХАНІЗМИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ ПОСТКВАНТОВИХ КРИПТОСИСТЕМ..... 345
15.1 Порівняльний аналіз одноразових підписів на базі геш-функцій..... 345
15.2 Методи обчислення системних параметрів для електронного підпису «CRYSTALS- DILITHIUM» 128, 256, 384 та 512 біт рівнів безпеки 336
15.2.1 Алгоритми реалізації перспективного ЦП CRYSTALS-DILITHIUM..... 387
15.2.2 Вибір параметрів схеми цифрового підпису DILITHIUM..... 397
15.3 Несиметричні криптосистеми на алгебраїчних кодах для постквантового періоду 401
15.3.1 Алгебраїчні коди для несиметричних криптосистем 402
15.3.2 Несиметричне шифрування на основі кодів (теоретико-кодові схеми) 406
15.4 Реалізація постквантового алгоритму електронно-цифрового підпису..... 423
15.5 Методи побудови та властивості логарифмічних підписів 429
15.6 Властивості багатовимірного алгоритму RAINBOW та його здатність протистояти різноманітним методам криптоаналізу і атаці сторонніми каналами 436
15.6.1 Загальні положення щодо схеми ЕП RAINBOW..... 436
15.6.2 Застосування різноманітних методів криптоаналізу проти алгоритму RAINBOW..... 437
15.7 Постквантовий алгоритм інкапсуляції ключів CLASSIC MCELIECE 443
16 МЕТОДИ АНАЛІЗУ ТА СИНТЕЗУ ПЕРСПЕКТИВНИХ КРИПТОПЕРЕТВОРЕНЬ ДЛЯ ПОСТКВАНТОВОГО ПЕРІОДУ 469
16.1 Особливості побудування та аналіз електронних підписів 5 рівня безпеки для постквантового періоду на алгебраїчних решітках..... 469
16.2 Основні положення та результати порівняння властивостей електронних підписів постквантового періоду на алгебраїчних решітках..... 487
16.2.1 Основні параметри та ключі ЕП Falcon та Dilithium..... 489
16.3 Основні положення щодо моделі безпеки для асиметричних перетворень типу ЕП з урахуванням вимог та загроз постквантового періоду..... 508
16.4 Аналіз та дослідження алгоритму цифрового підпису PICNIC 519
16.5 Аналіз стійкості постквантового електронного підпису DILITHIUM до атак на помилки 526
16.6 Аналіз апаратних реалізацій алгоритмів електронного підпису QTESLA, CRYSTALS-DILITIUM і MQDSS на різних рівнях безпеки..... 534
16.7 Моделі та методи створення криптосистем, заснованих на використанні ізогеній еліптичних кривих..... 548

16.8 Процеси та методи вибору загальносистемних параметрів та аналізу стійкості до атак сторонніх каналів для ключового механізму інкапсуляції ДСТУ 8961:2019 557
СУТНІСТЬ СТАНДАРТИЗАЦІЇ ТА ВПРОВАДЖЕННЯ КВАНТОВО - СТІЙКИХ СТАНДАРТІВ У США ТА УКРАЇНІ 566
ВИСНОВКИ 567
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 572
ПРЕДМЕТНИЙ ПОКАЖЧИК 595

ПЕРЕДМОВА

Наразі світова цивілізація робить суттєві кроки в науці та практиці, що пов'язані з квантовими обчисленнями. Робляться суттєві кроки щодо досягнення конкурентної переваги країн в галузі квантової інформаційної науки та практики впровадження квантових технологій. Наукові та практичні дослідження направлені перш за все на зниження ризиків, що пов'язані з квантовими комп'ютерами щодо кібербезпеки, економічної та національної безпеки. Визначаються конкретні дії, які мають зробити технологічно розвинені держави, що розпочинають багаторічний процес переведення вразливих комп'ютерних систем на квантовостійку криптографію.

Робляться спроби стимулювання інновацій економіки в сферах від матеріалознавства та фармацевтики до фінансів та енергетики. Хоча повний спектр застосування квантових комп'ютерів ще невідомий, проте очевидно, що подальше технологічне та наукове лідерство держав залежатиме принаймні частково від здатності країни підтримувати конкурентну перевагу в галузі квантових обчислень та квантової інформаційної науки.

Але, поряд з потенційними перевагами, квантові обчислення також становлять значні ризики для економічної та національної безпеки. Зокрема, квантовий комп'ютер достатньої потужності, що відомий як криптоаналітично значущий, зможе зламати існуючі стандарти асиметричних криптографічних перетворень - криптографії з відкритим ключем.

Наведемо переваги та недоліки фізичних реалізацій квантових комп'ютерів. Різні фізичні реалізації квантових комп'ютерів мають переваги та недоліки щодо таких факторів, як:

• масштабованість, тобто, можливість створення та управління все більшими і більшими квантовими пристроями із все більшою кількістю кубітів, використовуючи фізичні та інженерні ресурси, які управляються керованим способом;

• сумісність з різними обчислювальними моделями та простота їх реалізації;

• типовий час декогерентності (тобто, скільки часу залишаються збереженими та використаними в працездатному стані характеристики, а також можуть бути використані квантові особливості, такі як суперпозиції);

• швидкість і точність, з якою вентилі можуть бути застосовані. Наразі є проблеми стосовно вибору фізичної реалізації квантових комп'ютерів:

• Високорівнева класифікація перспективних фізичних реалізацій квантових комп'ютерів, основними з них є такі:

- Квантова оптика, коли інформація зберігається та захищається в станах квантів світла на основі поляризації або станах з певним числом фотонів та може бути реалізована в чіпі за допомогою інтегрованої оптики;

- Надпровідні системи, коли інформація зберігається та обробляється (захищається) в електричних ланцюгах, які використовують властивості надпровідних матеріалів;

- Топологічні системи, коли інформація зберігається та захищається з використанням деяких топологічних властивостей, тобто, властивостей, які залежать від «глобальних» (геометричних) властивостей, нечутливих до «локальних» змін квантових систем;

- Іонні пастки, коли інформація зберігається (захищається) та маніпулюється з використанням властивостей іонів (атомів із незникаючим повним електричним зарядом), які обмежені електромагнітними полями;

- Квантові спінові системи, коли інформація зберігається та захищається (маніпулюється) у внутрішній степені свободи, яка називається квантовим спіном. Такі системи можуть бути реалізовані в кремнії, як стандартні мікрочіпи, або в менш звичайних системах, як алмази з точковими дефектами, відомі як азотно-заміщена (NV) вакансія;

- Гази холодних атомів, де нейтральні атоми (а не іони) охолоджуються до значення, близького до абсолютного нуля. У той час як іони відштовхуються один від одного через свій електричний заряд, нейтральні атоми можуть бути захоплені і організовані в дуже регулярні масиви за допомогою лазерних променів, що створюють так звані оптичні решітки. Атомами можна керувати аж до рівня окремих ділянок в решітці.

Стан створення та можливості вирішення задач криптоаналізу:

1) IBM розробила та представила квантовий 127-кубітний процесор Eagle. Він прийшов на зміну 65-кубітному квантовому процесору Hummingbird, що відповідає дорожній карті квантових технологій IBM.

2) Є відомості про наміри IBM представити 433-кубітний процесор Osprey в 2022 році, а 1121-кубітний процесор Condor в 2023.

3) IBM повідомляє про наміри щодо побудови на основі покращених чіпів нової інтегрованої квантової обчислювальної системи IBM Quantum System Two замість існуючої системи IBM Quantum System One.

4) Компанія D-Wave, що відома розробками псевдоквантових (гібридних) комп'ютерів з великою загальною кількістю кубітів понад 2000 та понад 5000 кубітів сьогодні, повідомила про наміри представити машину понад 7000 кубітів (2023-2024) р.

Таким даним і розробкам, що рекламуються, можна довіряти, проте фактичний стан розроблення та застосування потужних квантових комп'ютерів є закритим.

Також до створення квантових комп'ютерів, по суті з випередженням, потрібно розробити математику та програмне забезпечення на відповідній основі. Для цього потрібне зовсім інше сприйняття та мислення як з математичних, так і програмних основ, а також на практиці.

Таким чином, зрозуміло, що потрібні інші підходи до підготовки спеціалістів математиків, фізиків, алгоритмістів, програмістів тощо. Навіть використовуючи класичну математику для розробки криптографічно стійких постквантових стандартів, необхідно застосовувати інші математичні методи для асиметричних криптографічних перетворень. Перелік таких методів може зводитися до використання: математичних решіток та математичних кодів,

математики ізогеній еліптичних кривих, криптографічних перетворень в квадратичних полях (багатовимірних перетворень в квадратичних полях) тощо.

До основних задач, які можуть бути вирішені на квантовому комп'ютері, необхідно віднести такі:

1) квантовий алгоритм факторизації Шора;

2) квантовий алгоритм Гровера пошуку елемента в несортованій базі;

3) квантовий алгоритм Шора для розв'язку дискретного логарифму в скінченному полі;

4) квантовий алгоритм розв'язку дискретного логарифму в групі точок ЕК Шора тощо.

Окрім класичної криптографії, яка базується на математичних алгоритмах, нині активно розвивається квантова криптографія. Створення квантових комп'ютерів відкриє принципіально нові можливості для людства, але при цьому існуючі методи захисту інформації втратять свою ефективність. Не дивлячись на те, що поки що квантові комп'ютери тільки виходять за межі лабораторій, необхідність у використанні квантово-безпечної або, як її ще називають, постквантової криптографії є уже сьогодні.

Безпека сучасних інформаційних систем та технологій грунтується на стійкості криптографічних перетворень, які вони використовують для криптографічної обробки інформації. Криптографічна стійкість базується на складності розв'язання певних математичних задач (факторизації великого цілого числа, розв'язку дискретного алгоритму тощо), для таких задач характерна субекспоненційна або експоненційна складність розв'язання на сучасних (класичних) комп'ютерах. Проте, використовуючи квантові алгоритми Шора та Гровера, певні математичні задачі можна розв'язувати навіть з поліноміальною складністю. Ідеї використання потужностей квантового середовища висунули Пауль і Фейман. Важливим стало розроблення у 1992 р. Дойчем та іншими першого квантового алгоритму, можливості якого значно перевищували можливості звичайних комп'ютерів. У випадку появи квантового комп'ютера, на якому може бути запущений квантовий алгоритм криптоаналізу Шора або алгоритм пошуку в неупорядкованій базі даних Гровера, можуть виникнути великі загрози у інформаційній сфері відносно забезпечення криптографічної стійкості як для асиметричних криптоперетворень, так і для певних симетричних. Важливим є не тільки сам факт побудови такого комп'ютера, а й технічні характеристики, якими володітиме квантовий комп'ютер.

В квантовому комп'ютері ключ шифрування передається за допомогою елементарних часток світла – фотонів. Будь-який прилад, за допомогою якого третя сторона спробує перехватити дані, обов'язково вплине на стан фотона, і ключ буде не дійсним. Передавати фотони можна за допомогою виділених оптоволоконних ліній, на обох кінцях такої лінії необхідні спеціальні шифрувальні пристрої. Недоліком квантової криптографії є те, що вона вимагає великих затрат на інфраструктуру. При цьому поки що ключі удається передавати на відстань (50-100 км), швидкість їх генерації досить низька, а на передачу фотонів впливає маса зовнішніх факторів.

Постквантова (квантово – безпечна) криптографія, як і класична, основана на розв'язку математичних задач. Проте нові алгоритми шифрування повинні бути іншими, щоб їх не могли розв'язати за допустимий час не тільки звичайні, але і квантові комп'ютери.

Грунтуючись на зазначеному вище, у навчальний посібник включені наступні науково - навчальні матеріали та результати досліджень авторів навчального посібника та інші, на які ми робимо коректні посилання.

В першій частині посібника розглядаються елементи і схеми квантової інформатики. Уводиться поняття кубіта. Це квантова система, яка може знаходитися в двох станах $|0\rangle$ і $|1\rangle$. Ці позначення уведені Діраком. Прикладом такої системи є фотон з двома можливими поляризаціями або електрон з двома можливими напрямками спіну. В цілому стан такої системи задається хвильовою функцією. Хвильова функція розглядається як вектор у двовимірному лінійному просторі. Представлено геометричне зображення кубіта для випадку дійсних коефіцієнтів. Геометричне зображення кубіта для випадку комплексних коефіцієнтів представлено сферою Блоха. Розглядаються однокубітові логічні елементи. Приведені приклади двокубітових станів і операторів. Розглянуті квантові схеми, а також квантова схема, що створює стан Белла. Розглянуто використання стану Белла в квантовій телепортації. Уведені поняття інформаційної ентропії.

У другій частині розглядається квантова теорія обчислень. Приводиться наочне уведення її в квантові представлення. Розглянуті основні принципи класичної фізики і протиріччя класичної механіки. Наведені елементарні квантові операції. Детально розглянуті квантова телепортація, а також квантові ключі і їх передача. Приведені вступ в теорію та практику квантових обчислень, протоколи квантового поширення ключа, квантові протоколи узгодження ключів, фізичні основи квантової криптографії, практична реалізація квантових алгоритмів, алгоритм факторизації цілих чисел, методи перехоплення інформації у системах квантової криптографії. Приведені приклади факторизації за алгоритмом Шора, квантовий алгоритм Гровера та особливості методу Гровера при використанні в несортованій базі даних, наведений приклад практичної реалізації методу Гровера, розглянутий парадокс Ейнштейна-Подольські-Розена (ЕПР- парадокс). Розглянутий приклад квантового перетворення Фур'є. Введено поняття принципу невизначеності Гейзенберга. Ідеї Стівена Віснера сформували фундамент нового напряму науки – **квантової інформаційної безпеки**. **Суть цього напряму** – використання фундаментальних квантових властивостей мікросвіту для захисту інформації. **Основна перевага** – теоретична стійкість квантових протоколів та способів захисту. **Основний недолік** – складність практичної реалізації. Практичні роботи у галузі квантової криптографії виконують такі компанії як IBM, Mitsubishi, Toshiba, лабораторія GAP-Optique, Національна лабораторія в Лос-Аламосі, Каліфорнійський технологічний інститут, MagiQ, холдинг QinetiQ.

Тим не менше, не так давно, у світі з'явилися загрози, які поставили під сумнів цілісну картину криптографічних технологій, що успішно використовувалися до цього часу. Мова йде про квантові комп'ютери та

сукупність алгоритмів для них, практичне впровадження яких сприятиме повному перегляду сучасних концепцій криптографічного захисту інформації.

Рух по цьому шляху вже розпочато. Так, у квітні 2016 року Національний інститут стандартів і технологій США (NIST) випустив документ NISTIR 8105 DRAFT Report on Post-Quantum Cryptography, в якому оцінив розвиток квантового комп'ютингу та його вплив на стан криптографічного захисту інформації в недалекому майбутньому. Висновки, до яких прийшли автори документу, скорочено подані у навчальному посібнику.

Як видно з наведених висновків, квантовий комп'ютинг (звичайно, при впровадженні масштабованих квантових комп'ютерів) завдає відчутного удару по всіх напрямах криптографії. Зокрема, НІСТ вважає, що для збереження потрібного рівня захисту симетричні криптоалгоритми на зразок AES-256 потребують збільшення довжини криптографічного ключа, функції хешування SHA-256 або SHA-3 будуть стійкими до зламу за допомогою квантових алгоритмів при збільшенні довжини хеш-образу. Слід зауважити, що вплив квантового комп'ютингу не залежить (або слабо залежить) від архітектури функції хешування, оскільки SHA-256 та SHA-3 ґрунтуються на різних підходах: структурі Меркля - Дамгарда та "криптографічної губки" відповідно.

Згідно до документу NISTIR 8105 DRAFT Report on Post-Quantum Cryptography ми стоїмо перед надзвичайно відповідальним етапом розвитку криптографічної науки, який так і називають у Національному інституті стандартів і технологій США - постквантовою криптографією. Згаданий документ було видано у квітні 2016 року, а пізніше, в грудні того ж року НІСТ оголосив конкурс криптоалгоритмів, стійких до атак за допомогою квантових комп'ютерів.

Сьогодні конкурс ще триває, проходить "півфінал", в якому кількість учасників скорочено до 26 алгоритмів. Ці алгоритми, на думку математиків НІСТ та спеціалістів з криптографії, вважаються найсильнішими кандидатами для розробки стандартів постквантової криптографії. Як повідомляють на сайті НІСТ (https://www.nist.gov/news-events/news/2019/01/nist-reveals-26-algorithms-advancing-post-quantum-crypto-semifinals), в подальшому увагу експертів буде зосереджено на продуктивності алгоритмів, які мають бути ефективними як для настільних комп'ютерів, так і для обчислювачів з обмеженими ресурсами (смартфони, кредитні картки, вбудована техніка, "розумний будинок") тощо.

У третій частині посібника розглянута постквантова криптографія та механізми її реалізації. Розглянуті процеси та методи вибору загальносистемних параметрів та аналізу стійкості до атак сторонніх каналів для ключового механізму інкапсуляції ДСТУ 8961:2019. В ДСТУ 8961:2019 приведені стан розробки та стандартизації ключових протоколів інкапсуляції на міжнародному рівні та в Україні.

Розглянута і проаналізована сутність стандартизації та впровадження квантово стійких стандартів у США та Україні. Приведена порівнювальна таблиця квантово стійких асиметричних стандартів.

1 ОСНОВНІ ПРИНЦИПИ КВАНТОВОЇ ІНФОРМАТИКИ

1.1 Кубіт

Кубітом називається квантова система, яка може знаходитися в двох станах $|0\rangle$ і $|1\rangle$ (позначення Дірака). Прикладом такої системи є фотон з двома можливими поляризаціями або електрон з двома можливими напрямками спіну. В цілому стан такої системи задається хвильовою функцією виду

$$|\psi\rangle = a|0\rangle + \beta|1\rangle, \tag{1.1}$$

де α і β - комплексні коефіцієнти. При вимірюванні стану системи з хвильовою функцією (1.1) ймовірність її виявлення в стані $|0\rangle$ дорівнює $|a|^2$, а ймовірність її виявлення в стані $|1\rangle$ дорівнює $|\beta|^2$. Сума цих ймовірностей дорівнює одиниці:

$$|a|^2 + |\beta|^2 = 1. \tag{1.2}$$

Хвильову функцію $|\psi\rangle$ будемо розглядати як вектор у двовимірному лінійному просторі. При цьому хвильові функції $|0\rangle$ і $|1\rangle$ відіграють роль основи цього простору. Поряд з вектором $|\psi\rangle$ введемо ермітово-пов'язаний вектор $\langle\psi| \equiv |\psi\rangle^+$. Тоді скалярний добуток двох векторів $|\psi_1\rangle$ і $|\psi_2\rangle$ запишемо у вигляді $\left\langle \psi_1 \middle| \psi_2 \right\rangle$. Будемо вважати основні стани $|0\rangle$ і $|1\rangle$ ортонормованими:

$$\langle 0|0\rangle = 1, \langle 1|1\rangle = 1, \langle 0|1\rangle = 0. \tag{1.3}$$

Тоді з умови нормування хвильового вектора кубіту $\langle\psi|\psi\rangle = 1$ відразу слідує вимога (1.2) на коефіцієнти α і β. Кубіт допускає геометричне зображення. Розглянемо спочатку випадок дійсних коефіцієнтів α і β.

В цьому випадку зручно використовувати тригонометричне представлення:

$$\alpha = \cos\phi, \ \beta = \sin\phi. \tag{1.4}$$

Тоді умова (1.2) виконується автоматично. На площині (α,β) умова (1.2) задає одиничне коло з центром на початку координат (див. Рис. 1.1.) При $\phi = 0$, у відповідності до ((1.1), (1.4)), ми отримуємо $|\psi\rangle = |0\rangle$, а при $\phi = \frac{\pi}{2}$, $|\psi\rangle = |1\rangle$.

В загальному випадку комплексних коефіцієнтів α і β вони можуть бути представлені у вигляді:

$$a = e^{i\gamma}\cos\frac{\vartheta}{2}, \ \beta = e^{i\lambda}\sin\frac{\vartheta}{2}.$$

Тоді кубіт (1.1) приймає вигляд:

$$|\psi\rangle = e^{i\gamma}\cos\frac{\vartheta}{2}|0\rangle + e^{i\lambda}\sin\frac{\vartheta}{2}|1\rangle$$

або

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\vartheta}{2}|0\rangle + e^{i(\lambda-\gamma)}\sin\frac{\vartheta}{2}|1\rangle\right).$$

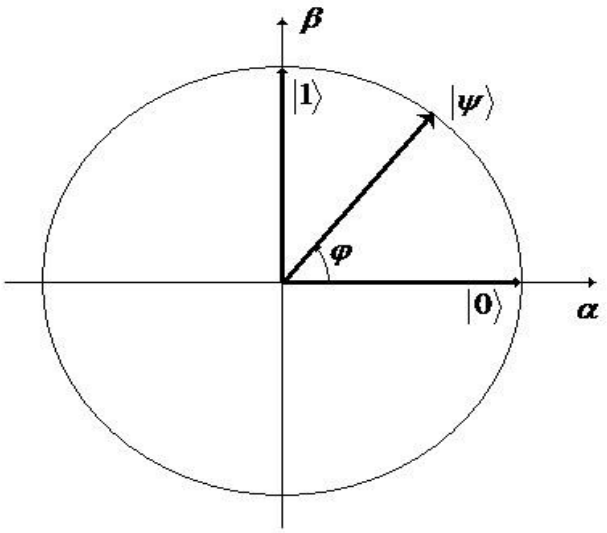


Рис.1.1. Геометричне зображення кубіта |ψ⟩=α|0⟩ +β|1⟩ для випадку дійсних коефіцієнтів α і β

Фазовий множник в багатьох випадках виявляється несуттєвим. Опускаючи цей множник і позначаючи

$$\phi = \lambda - \gamma,$$

представимо кубіт у вигляді:

$$|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle. \qquad (1.5)$$

Таким чином, отримане представлення кубіта має два параметри - ϑ і ϕ. Будемо інтерпретувати їх як кути сферичної системи координат (див. Рис. 1.2).

Кубіт представляється вектором одиничної довжини в тривимірному просторі. Таке геометричне зображення кубіту називається його представленням на сфері Блоха. У відповідності з (1.5) при $\vartheta = 0$ отримаємо базисний |0⟩, а при $\vartheta = \pi$, $|\psi\rangle = |1\rangle$ (із точністю до фазового множника $e^{i\phi}$) (див. підрозділ 7.3).

На сфері Блоха нескінченно багато точок. Відповідно, кубіти можна знайти в одному з нескінченної множини станів. Здавалося б, використовуючи один кубіт, можна зберігати нескінченно багато інформації. Однак, в дійсності це не так. При вимірюванні стану кубіту його можна зустріти в одному з двох можливих станів - |0⟩ або |1⟩.

Відповідно, ми можемо отримати один біт інформації з кубіта. Якби ми могли отримати багато однакових копій нашого кубіту і провести вимірювання стану кожної копії, то з імовірністю $|a|^2$ ми б знайшли кубіт в стані |0⟩, а з імовірністю $|\beta|^2$ – в стані |1⟩. Число $|a|^2$ може приймати нескінченно багато значень. Таким чином, ми б отримали нескінченно багато інформації, що закодована в одному кубіті.

Для цього ми повинні були б, на початку, отримати якомога більше копій нашого кубіту. Однак існує теорема про неможливість клонування кубіту в

невідомому стані, яка буде викладена нижче. Таким чином, з одного кубіту можна отримати лише один біт інформації.

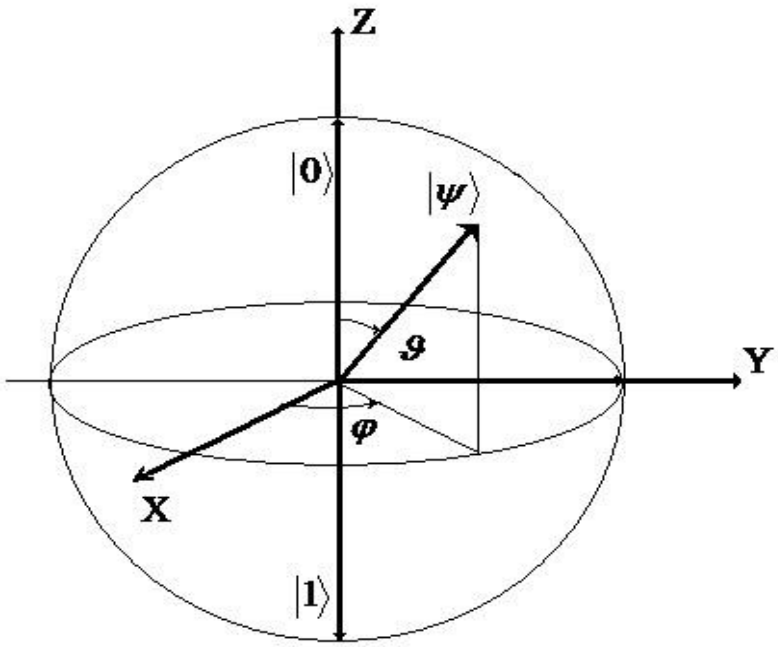


Рис. 1.2. Геометричне зображення кубіта $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ для випадку комплексних коефіцієнтів α і β – сфера Блоха

1.2 Однокубітові логічні елементи

Нагадаємо, що кубіт представляє собою квантову систему, стан якої задається хвильовим вектором $|\psi\rangle$ (1.1). Фізичний вплив на цю систему перетворює її в інший стан $|\tilde{\psi}\rangle$:

$$|\tilde{\psi}\rangle = \tilde{\alpha} |0\rangle + \tilde{\beta} |1\rangle. \quad (1.6)$$

У квантовій механіці будь-який вплив на систему описується лінійним оператором $\hat{U}$, що діє на вектор стану системи:

$$|\tilde{\psi}\rangle = \hat{U} |\psi\rangle. \quad (1.7)$$

Лінійність оператора $\hat{U}$ витікає із лінійності рівняння Шредінгера. Дійсно, нехай $|\psi\rangle$ - залежний від часу вектор стану системи. Рівняння Шредінгера з гамільтоніаном $\hat{H}$ запишемо у вигляді (див. підрозділ 7.2.5):

$$ih \frac{\partial |\psi\rangle}{\partial t} = \hat{H}|\psi\rangle. \quad (1.8)$$

Нехай, для простоти, гамільтоніан системи не залежить від часу. Тоді розв'язок рівняння (1.8) з початковою умовою:

$$|\psi\rangle_{t=0} = |\psi\rangle$$

може бути записаний у вигляді:

$$|\tilde{\psi}\rangle = \exp(-i\,\hat{H}t\,/h)| \qquad (1.9)$$

Порівняння (1.9) з (1.7) дозволяє зв'язати лінійний оператор $\hat{U}$ з гамільтоніаном $\hat{H}$, що описує зовнішній вплив на кубіт:

$$\hat{U} = \exp(-i\,\hat{H}t\,/h). \qquad (1.10)$$

Оператор Гамільтона повинен бути ермітовим:

$$\widehat{H^+}\ \widehat{H} = \mathrm{I}\ (\mathrm{I} - \text{одиничний оператор})$$

для того, щоб його власні числа - допустимі значення енергії системи, були б дійсними. Тоді з (1.10) випливає, що оператор $\hat{U}$ повинен бути унітарним:

$$\hat{U} + \hat{U} = \exp(-i\,\hat{H}t\,/h)\ \exp(i\hat{H}^+\,t\,/h) = \mathrm{I}. \qquad (1.11)$$

Унітарність оператора $\hat{U}$ означає, що якщо початковий стан системи нормований

$$\langle\psi|\psi\rangle = 1, \qquad (1.12)$$

то і стан, в який система переходить після взаємодії, також є нормованим.

Дійсно, відповідно до (1.7), (1.11) та (1.12), ми отримуємо

$$\langle\tilde{\psi}|\tilde{\psi}\rangle = \langle\psi|\hat{U}^+\hat{U}|\psi\rangle = \langle\psi|\psi\rangle = 1. \qquad (1.13)$$

Унітарний оператор завжди є оборотним:

$$\hat{U}^+\hat{U} = \mathrm{I} \implies \hat{U}^+\ \hat{U}\,\hat{U}^{-1} = \hat{U}^{-1} \implies \hat{U}^{-1} = \hat{U}^+ \qquad (1.14)$$

У подальшому вплив на кубіт (або на систему кубітів) ми будемо розглядати, як процес обчислення. При цьому вектор $|\psi\rangle$ відіграє роль вхідного сигналу, оператор $\hat{U}$ визначає обчислювальний процес, а вектор $|\tilde{\psi}\rangle$ представляє собою результат обчислення (див. Рис. 1.3).

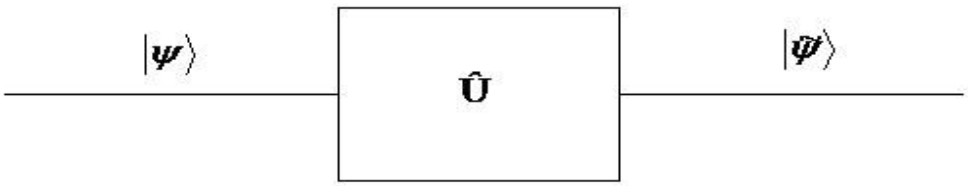


Рис. 1.3. Квантовий логічний елемент $\hat{U}$ перетворює вектор $|\psi\rangle$ у вектор $|\tilde{\psi}\rangle$

Згідно (1.14), кожний квантовий обчислювальний процес оборотний. Тобто, якщо в результаті обчислювального процесу $\hat{U}$ вхідний сигнал $|\psi\rangle$ був перетворений у вихідний сигнал $|\tilde{\psi}\rangle$, то існує обернений квантовий обчислювальний процес $\hat{U} - 1$, який здійснює обернене перетворення (див. Рис. 1.4). Відмітимо, що класичні обчислювальні процеси не завжди оборотні.

У подальшому ми будемо використовувати *матричне представлення операторів* $\widehat{\mathbf{U}}$**.** Опишемо відповідні конструкції. Розглянемо дію оператора на кубіт. Внаслідок лінійності оператора

$$\widehat{\mathrm{U}}|\psi\rangle = \tilde{\alpha}\,|0\rangle + \tilde{\beta}\,|1\rangle = \widehat{\mathrm{U}}\,(a|0\rangle + \beta|1\rangle) = a\widehat{\mathrm{U}}|0\rangle + \beta\widehat{\mathrm{U}}|1\rangle. \qquad (1.15)$$

Таким чином, дія оператора на кубіт визначається його дією на базові вектори $|0\rangle$ і $|1\rangle$. Ці вектори утворюють ортонормований базис в двовимірному гільбертовому просторі.

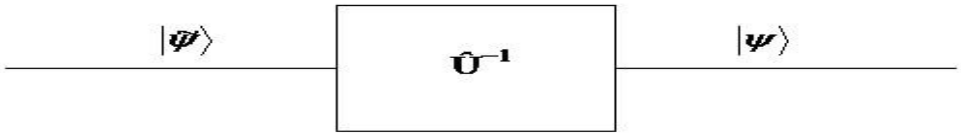


Рис. 1.4. Квантовий логічний елемент $\widehat{\mathrm{U}} - 1$ перетворює вектор $|\widetilde{\psi}\rangle$ у вектор $|\psi\rangle$

Це означає, що будь-який вектор цього простору може бути розкладений за базисними векторами. Запишемо розкладання векторів $\widehat{\mathrm{U}}|0\rangle$і $\widehat{\mathrm{U}}|1\rangle$ за базисом $|0\rangle, |1\rangle$:

$$\widehat{\mathrm{U}}|0\rangle = U_{00}|0\rangle + U_{10}|1\rangle,$$
$$\widehat{\mathrm{U}}|1\rangle = U_{01}|0\rangle + U_{11}|1\rangle. \qquad (1.16)$$

Згідно з (1.16), коефіцієнти розкладання U_{mn} можуть бути обчислені як

$$U_{00} = \langle 0|\widehat{\mathrm{U}}|0\rangle, U_{10} = \langle 1|\widehat{\mathrm{U}}|0\rangle,$$
$$U_{01} = \langle 0|\widehat{\mathrm{U}}|1\rangle, U_{11} = \langle 1|\widehat{\mathrm{U}}|1\rangle. \qquad (1.17)$$

Підставляючи (1.16) в (1.15), отримуємо

$$\tilde{\alpha}\,|0\rangle + \tilde{\beta}\,|1\rangle = a(U_{00}|0\rangle + U_{10}|1\rangle) + \beta(U_{01}|0\rangle + U_{11}|1\rangle)$$

або

$$|\widetilde{\psi}\rangle = \tilde{\alpha}\,|0\rangle + \tilde{\beta}\,|1\rangle = (U_{00}a + U_{01}\beta)|0\rangle + (U_{10}a + U_{11}\beta)|1\rangle.$$

Таким чином, коефіцієнти розкладання вектора $|\widetilde{\psi}\rangle$ виражені через коефіцієнти розкладання початкового вектора $|\psi\rangle$ у вигляді:

$$\tilde{\alpha} = U_{00}\,a + U_{01}\beta,\ \tilde{\beta} = U_{10}a + U_{11}\beta. \qquad (1.18)$$

Уведемо вектори

$$\psi = \begin{pmatrix} a \\ \beta \end{pmatrix}, \widetilde{\psi} = \begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix},$$

які відповідають початковому і кінцевому станам кубіту, та матрицю

$$\mathrm{U} = \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix}. \tag{1.19}$$

Тоді вирази (1.18), що зв’язують нові і старі компоненти векторів, можна записати у вигляді:

$$\begin{pmatrix} \tilde{\mathrm{A}} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix} \tag{1.20}$$

або

$$\tilde{\psi} = \mathrm{U}\psi. \tag{1.21}$$

Матриця U називається матричним представленням оператора $\hat{\mathrm{U}}$. Властивість (1.11) унітарності оператора $\hat{\mathrm{U}}$ приводить до вимоги унітарності його матриці:

$$\mathrm{U}^{+}\mathrm{U} = \begin{pmatrix} U_{00}^{*} & U_{10}^{*} \\ U_{10}^{*} & U_{11}^{*} \end{pmatrix} \begin{pmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{pmatrix} = \mathrm{I}. \tag{1.22}$$

2 ПРИКЛАДИ ОДНОКУБІТОВИХ ЕЛЕМЕНТІВ

2.1 Логічний елемент NOT

Позначимо квантовий логічний елемент NOT через $\hat{X}$. Визначимо спочатку дію цього оператора на базисні вектори. Будемо вимагати, щоб він переводив $|0\rangle$ в $|1\rangle$, а $|1\rangle$ в $|0\rangle$:

$$\hat{X}\,|0\rangle = |1\rangle,$$
$$\hat{X}\,|1\rangle = |0\rangle. \qquad (2.1)$$

Тим самим квантовий оператор NOT стає природним узагальненням класичного оператора NOT. Використовуючи лінійність оператора $\hat{X}$, визначимо дію оператора на довільний кубіт:

$$\hat{X}|\psi\rangle = \hat{X}\,(\alpha|0\rangle + \beta|1\rangle) = a\hat{X}\,|0\rangle + \beta\hat{X}\,|1\rangle.$$

Використовуючи (1.3), отримаємо

$$\hat{X}|\psi\rangle = \alpha\,|1\rangle + \beta\,|0\,\rangle = \beta\,|0\,\rangle + \alpha\,|1\rangle. \qquad (2.2)$$

Таким чином, оператор $\hat{X}$ міняє місцями коефіцієнти при базисних векторах $|0\,\rangle$ і $|1\rangle$. Порівнюючи (2.1) з (1.16), находимо матричні елементи X_{mn} оператора $\hat{X}$:

$$X_{00} = 0,\ X_{10} = 1,$$
$$X_{01} = 1,\ X_{11} = 0. \qquad (2.3)$$

Запишемо матрицю X оператора X_{mn}:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \qquad (2.4)$$

Подіємо матрицею (2.4) на вектор вхідного кубіту. Тоді отримуємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix} \tilde{a} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ a \end{pmatrix} \qquad (2.5)$$

у відповідності з (2.2).

Отримана матриця X є унітарною. Дійсно

$$X^{+}X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{+} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$$

Дію оператора $\hat{X}$ на кубіт

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\,\rangle$$

для дійсних α і β легко інтерпретувати геометрично. У випадку дійсних значень ми використовуємо тригонометричне представлення:

$$a = \cos\phi\ ,\ \beta = \sin\phi\ ,\ \tilde{\alpha} = \cos\tilde{\phi},\ \tilde{\beta} = \sin\tilde{\varphi}\ . \qquad (2.6)$$

Тоді, згідно (2.5), (2.6)

$$\tilde{\phi} = \frac{\pi}{2} - \phi. \qquad (2.7)$$

Таким чином, оператор $\hat{X}$ повертає одиничний вектор, що зображує кубіт на одиничному колі, відображаючи його від бісектриси першого і третього координатних кутів (див. Рис. 2.1).

Знайдемо власні числа λ та власні вектори $|\psi\rangle_\lambda$ оператора $\hat{X}$:

$$\hat{X}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda$$

або у матричному вигляді:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = \lambda \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda.$$

Тоді:

$$\begin{pmatrix} -\lambda & 1 \\ 1 & -\lambda \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = 0.$$

Власні числа матриці X визначаються із рівняння:

$$\begin{pmatrix} -\lambda & 1 \\ 1 & -\lambda \end{pmatrix} = \lambda^2 - 1 = 0.$$

Звідси:

$$\lambda_{1,2} = \pm 1.$$

У свою чергу, власні вектори матриці X визначаються із умови:

$$-\lambda\alpha + \beta = 0$$

або

$$\beta = \lambda\alpha.$$

Вважаючи $\lambda_{1,2} = \pm 1$, маємо:

$$\beta_1 = \lambda\alpha,\ \beta_2 = -\alpha_2.$$

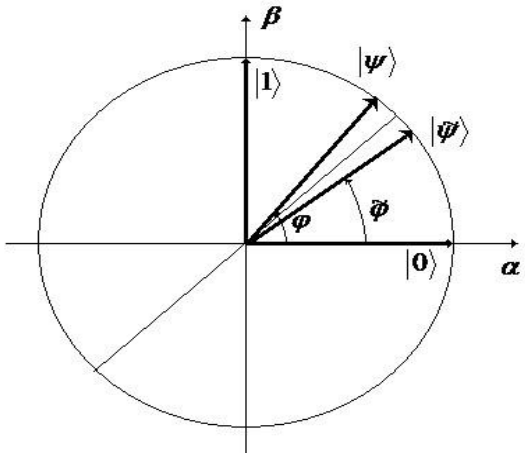


Рис. 2.1. Геометричне зображення перетворювання NOT для випадку дійсних коефіцієнтів α і β

Вибираючи $\alpha_1 = \alpha_2 = 1$, отримаємо $\beta_{1,2} = \pm 1$. Власні вектори мають вигляд:

$$|\psi\rangle_{1,2} = \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix}$$

або в нормованому вигляді:

$$|\tilde{\psi}\rangle_{1,2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix}.$$

2.2 Логічний елемент Z

Визначимо спочатку дію оператора $\hat{Z}$ на базисні вектори. Будемо вимагати, щоб він не змінював $|0\rangle$, а $|1\rangle$ переводив в $-|1\rangle$.

$$\hat{Z}\,|0\rangle = |0\rangle,$$
$$\hat{Z}\,|1\rangle = -|1\rangle. \qquad (2.8)$$

Використовуючи лінійність оператора $\hat{Z}$, визначимо дію оператора на довільний кубіт:

$$\hat{Z}\,|\psi\rangle = \hat{Z}\,(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\hat{Z}\,|0\rangle + \beta\hat{Z}\,|1\rangle.$$

Використовуючи (2.8), отримаємо:

$$\hat{Z}\,|\psi\rangle = \alpha\,|0\rangle - \beta\,|1\rangle. \qquad (2.9)$$

Таким чином, оператор $\hat{Z}$ не змінює коефіцієнт при базисному векторі $|0\rangle$ і змінює знак коефіцієнта при базисному векторі $|1\rangle$. Порівнюючи (2.8) з (1.16), знаходимо матричні елементи Z_{mn} оператора $\hat{Z}$:

$$Z_{00} = 1, Z_{10}=0,$$
$$Z_{01} = 0, Z_{11}= -1. \qquad (2.10)$$

Запишемо матрицю Z оператора Z_{mn}:

$$\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \qquad (2.11)$$

Подіємо матрицею (2.11) на вектор вхідного кубіту. Тоді отримуємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \qquad (2.12)$$

у відповідності з (2.9).

Отримана матриця Z є унітарною. Дійсно,

$$Z^{+}\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^{+} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathrm{I}.$$

Дію оператора $\hat{Z}$ на кубіт

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

для дійсних α і β легко інтерпретувати геометрично. В випадку дійсних величин використовуємо тригонометричне представлення

$$a = \cos\phi,\ \beta = \sin\phi\ ,\ \tilde{\alpha} = \cos\tilde{\phi},\ \tilde{\beta} = \sin\tilde{\phi}. \qquad (2.13)$$

Дійсно, згідно (2.6), (2.7)

$$\tilde{\phi} = -\phi. \qquad (2.14)$$

Таким чином, оператор $\hat{Z}$ повертає одиничний вектор, що зображує кубіт на одиничному колі, відображаючи його від осі абсцис (див. Рис. 2.2).

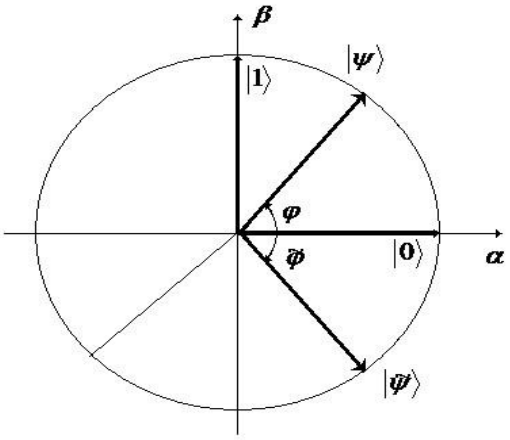


Рис. 2.2. Геометричне зображення перетворювання $\hat{Z}$ для випадку дійсних коефіцієнтів α і β

В загальному випадку комплексних α і β кубіт можна представити у наступному вигляді (1.5):

$$\cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle.$$

Оператор $\hat{Z}$ змінює знак коефіцієнта при базисному векторі $|1\rangle$. Це відповідає заміні $\vartheta \to -\vartheta$. На сфері Блоха це відповідає відображенню від горизонтальної площини (див. Рис. 2. 3).

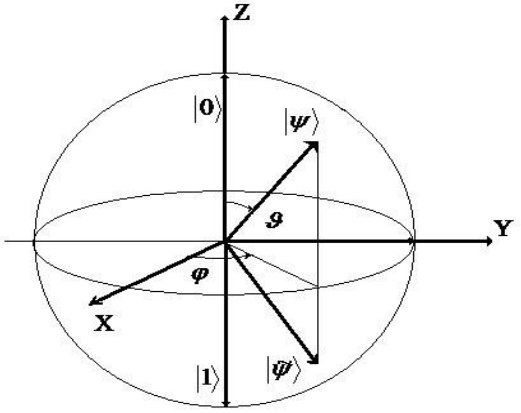


Рис. 2.3. Геометричне зображення перетворювання $\hat{Z}$ на сфері Блоха для випадку комплексних коефіцієнтів α і β

2.3 Логічний елемент Адамара H

Елемент Адамара задається матрицею

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}. \tag{2.15}$$

Подіємо матрицею (2.15) на вектор вхідного кубіту. Тоді отримуємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}a\\ \beta\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\alpha+\beta\\ \alpha-\beta\end{pmatrix}. \tag{2.16}$$

Відповідний оператор $\hat{\mathrm{H}}$ діє на кубіт за правилом:

$$\hat{\mathrm{H}}|\psi\rangle = \frac{\alpha+\beta}{\sqrt{2}}\,|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}\,|1\rangle. \tag{2.17}$$

Матриця H є унітарною. Дійсно,

$$H^{+}\mathrm{H} = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}^{+}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix} = \mathrm{I}.$$

Із унітарності матриці оператора виходить унітарність самого оператора. Знайдемо власні числа λ і власні вектори $|\psi\rangle_\lambda$ оператора $\hat{\mathrm{H}}$:

$$\hat{\mathrm{H}}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda$$

або в матричному вигляді:

$$\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}a\\ \beta\end{pmatrix}_\lambda = \lambda\begin{pmatrix}a\\ \beta\end{pmatrix}_\lambda.$$

Тоді

$$\begin{pmatrix}1-\lambda & 1\\ 1 & -1-\lambda\end{pmatrix}\begin{pmatrix}a\\ \beta\end{pmatrix}_\lambda = 0.$$

Власні числа матриці H визначаються із рівняння

$$\begin{pmatrix}1-\lambda & 1\\ 1 & -1-\lambda\end{pmatrix} = \lambda^2 - 2 = 0.$$

Звідси

$$\lambda_{1,2} = \pm\sqrt{2}\,.$$

В свою чергу, власні вектори матриці H визначаються із умови

$$(1-\lambda)\alpha + \beta = 0,$$

або

$$\beta = (\lambda - 1)\alpha\,.$$

Вважаючи $\lambda_{1,2} = \pm\sqrt{2}$, маємо

$$\beta_1 = (\sqrt{2}\text{-}1)\,\alpha_1\,,\ \beta_2 = (-\sqrt{2}\text{-}1)\,\alpha_2\,.$$

Вибираючи $\alpha_1 = \alpha_2 = 1$, отримаємо $\beta_{1,2} = \pm\sqrt{2} - 1$. Відповідні власні вектори мають вигляд:

$$|\psi\rangle_{1,2} = \begin{pmatrix} 1 \\ \pm\sqrt{2} - 1 \end{pmatrix}$$

або в нормованому вигляді:

$$|\widehat{\psi}\rangle_1 = \frac{1}{\sqrt{4-2\sqrt{2}}} \begin{pmatrix} 1 \\ \sqrt{2} - 1 \end{pmatrix},$$

$$|\widehat{\psi}\rangle_2 = \frac{1}{\sqrt{4+2\sqrt{2}}} \begin{pmatrix} 1 \\ -\sqrt{2} - 1 \end{pmatrix}.$$

2.4 Логічний елемент Y

Елемент Y задається матрицею

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (2.18)$$

На відміну від попередніх розглянутих нами елементів елемент Y є комплексним.

Подіємо матрицею Y на вектор вхідного кубіту. Тоді отримаємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix} = \begin{pmatrix} -i\beta \\ i\alpha \end{pmatrix}. \quad (2.19)$$

Відповідний оператор $\widehat{Y}$ діє на кубіт за правилом

$$\widehat{Y}\,|\psi\rangle = -i\,\beta|0\rangle + i\alpha\,|1\rangle. \quad (2.20)$$

Матриця Y є унітарною. Дійсно,

$$Y^{+}Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}^{+} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$$

Із унітарності матриці оператора слідує унітарність самого оператора. Знайдемо власні числа λ і власні вектори $|\psi\rangle_\lambda$ оператора $\widehat{Y}$:

$$\widehat{Y}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda,$$

або в матричному вигляді:

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = \lambda \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda.$$

Тоді

$$\begin{pmatrix} -\lambda & -i \\ i & -\lambda \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = 0.$$

Власні числа матриці Y визначаються із рівняння

$$\begin{pmatrix} -\lambda & -i \\ i & -\lambda \end{pmatrix} = \lambda^2 - 1 = 0.$$

Звідси

$$\lambda_{1,2} = \pm 1.$$

В свою чергу, власні вектори матриці Y визначаються із умови

$$\lambda\alpha + i\beta = 0,$$

або

$$\beta = i\lambda\,\alpha.$$

Вважаючи $\lambda_{1,2} = \pm 1$, маємо:

$$\beta_{1,2} = \pm i\,\alpha_{1,2}\,.$$

Вибираючи $\alpha_1 = \alpha_2 = 1$, отримаємо $\beta_{1,2} = \pm i$. Відповідні власні вектори мають вигляд:

$$|\psi\rangle_{1,2} = \begin{pmatrix} 1 \\ \pm i \end{pmatrix}$$

або в нормованому вигляді:

$$|\hat{\psi}\rangle_{1,2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i \end{pmatrix}.$$

2.5 Логічний елемент $\frac{\pi}{8}$

Розглянемо ще один комплексний логічний елемент T, який часто позначається $\frac{\pi}{8}$. Елемент T задається матрицею

$$\mathrm{T} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix}. \qquad (2.21)$$

Подіємо матрицею T на вектор вхідного кубіту. Тоді отримаємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \exp\left(i\frac{\pi}{8}\right) \end{pmatrix}. \qquad (2.22)$$

Відповідний оператор $\hat{\mathrm{T}}$ діє на кубіт за правилом

$$\hat{\mathrm{T}}\,|\psi\rangle = \hat{\mathrm{T}}\,(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,|0\rangle + \beta \exp\left(i\frac{\pi}{8}\right)|1\rangle. \qquad (2.23)$$

Таким чином, оператор $\hat{\mathrm{T}}$ не змінює коефіцієнт при базисному векторі $|0\rangle$ і змінює фазу коефіцієнта при базисному векторі $|1\rangle$.

Матриця T є унітарною. Дійсно,

$$T^{+}\mathrm{T} = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(-i\frac{\pi}{8}\right) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathrm{I}.$$

Із унітарності матриці оператора слідує унітарність самого оператора. Представимо кубіт у вигляді (1.5):

$$|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle.$$

Оператор $\hat{\mathrm{T}}$ змінює фазу коефіцієнта при базисному векторі $|1\rangle$. На сфері Блоха це відповідає повороту на кут $\frac{\pi}{8}$ навколо осі Z (див. Рис. 2.4). Знайдемо власні числа λ і власні вектори $|\psi\rangle_\lambda$ оператора $\hat{\mathrm{T}}$:

$$\hat{\mathrm{T}}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda,$$

або в матричному вигляді:

$$\begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = \lambda \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda.$$

Тоді

$$\begin{pmatrix} 1-\lambda & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right)-\lambda \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = 0.$$

Власні числа матриці T визначаються із рівняння:

$$\begin{vmatrix} 1-\lambda & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right)-\lambda \end{vmatrix} = (1-\lambda)\left(\exp\left(i\frac{\pi}{8}\right)-\lambda\right) = 0.$$

Звідси

$$\lambda_1 = 1, \lambda_2 = \exp\left(i\frac{\pi}{8}\right).$$

У свою чергу, власні вектори матриці T визначаються згідно однієї із двох умов:

$$(1-\lambda)a + 0\,\beta = 0,$$

або

$$0a + \left(\exp\left(i\frac{\pi}{8}\right)-\lambda\right)\beta = 0.$$

Вважаючи $\lambda = \lambda_1 = 1$, бачимо, що перше із рівнянь задовольняється тотожно. Згідно другому рівнянню,

$$\beta = \beta_1 = 0.$$

Вважаючи $\lambda = \lambda_2 = \exp\left(i\frac{\pi}{8}\right)$, бачимо, що друге із рівнянь задовольняється тотожно. Згідно першому рівнянню,

$$a = a_1 = 0.$$

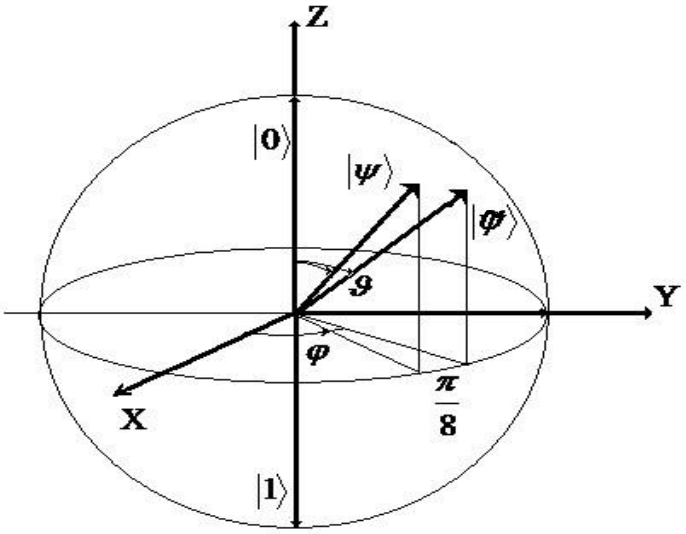


Рис. 2.4. Геометричне зображення перетворювання T = $\frac{\pi}{8}$ на сфері Блоха

Вибираючи $a_1 = \beta_1 = 1$, отримаємо:

$$|\psi\rangle_1 = \begin{pmatrix}1\\0\end{pmatrix}, |\psi\rangle_2 = \begin{pmatrix}0\\1\end{pmatrix}.$$

Таким чином, власні вектори оператора $\hat{T}$ співпадають з базисними векторами.

2.6 Логічний елемент S

Визначимо спочатку дію оператора $\hat{S}$ на базисні вектори. Будемо вимагати, щоб він не змінював $|0\rangle$, а $|1\rangle$ переводив в $i|1\rangle$.

$$\hat{S}\,|0\rangle = |0\rangle,$$
$$\hat{S}\,|1\rangle = i|1\rangle. \qquad (2.24)$$

Використовуючи лінійність оператора $\hat{S}$, визначимо дію оператора на довільний кубіт:

$$\hat{S}\,|\psi\rangle = \hat{S}\,(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,\hat{S}|0\rangle + \beta\hat{S}|1\rangle.$$

Використовуючи (2.24), отримаємо:

$$\hat{S}|\psi\rangle = \alpha\,|0\rangle + i\,\beta\,|1\rangle. \qquad (2.25)$$

Порівнюючи (2.25) з (1.16), знаходимо матричні елементи S_{mn} оператора $\hat{S}$:

$$S_{00} = 1,\ S_{10} = 0,$$
$$S_{01} = 0,\ S_{11} = i. \qquad (2.26)$$

Запишемо матрицю S оператора S_{mn}:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \quad (2.27)$$

Подіємо матрицею (2.27) на вектор вхідного кубіту. Тоді отримуємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix} = \begin{pmatrix} a \\ i\beta \end{pmatrix} \quad (2.28)$$

у відповідності з (2.25).

Отримана матриця S є унітарною. Дійсно,

$$S^{+}S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}^{+} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$$

Із унітарності матриці оператора слідує унітарність самого оператора.

Представимо кубіт у вигляді (1.5).

$$|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle.$$

Оператор $\hat{S}$ змінює фазу коефіцієнта при базисному векторі $|1\rangle$. На сфері Блоха це відповідає повороту на кут $\frac{\pi}{4}$ (див. Рис. 2.5).

Знайдемо власні числа λ та власні вектори $|\psi\rangle_\lambda$ оператора $\hat{S}$:

$$\hat{S}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda$$

або у матричному вигляді:

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = \lambda \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda.$$

Тоді:

$$\begin{pmatrix} 1-\lambda & 0 \\ 0 & i-\lambda \end{pmatrix} \begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = 0.$$

Власні числа матриці S визначаються із рівняння

$$\begin{pmatrix} 1-\lambda & 0 \\ 0 & i-\lambda \end{pmatrix} = (1-\lambda)(i-\lambda) = 0.$$

Звідси

$$\lambda_1 = 1, \lambda_2 = i.$$

В свою чергу, власні вектори матриці S визначаються згідно однієї із двох умов

$$(1-\lambda)\alpha + 0\beta = 0,$$

або

$$0\alpha = (1-\lambda)\beta = 0.$$

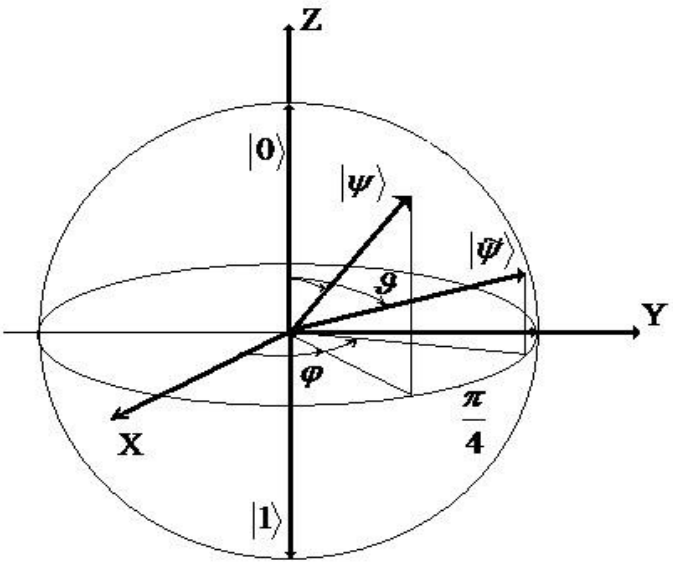


Рис. 2.5. Геометричне зображення перетворювання S на сфері Блоха

Вважаючи $\lambda = \lambda_1 = 1$, бачимо, що перше із рівнянь задовольняється тотожно. Згідно другого рівняння

$$\beta = \beta_1 = 0\,.$$

Вважаючи $\lambda = \lambda_2 = i$, бачимо, що друге із рівнянь задовольняється тотожно. Згідно першого рівняння

$$\alpha = \alpha_2 = 0.$$

Вибираючи $\alpha_1 = \beta_2 = 1$, отримаємо

$$|\psi\rangle_1 = \begin{pmatrix}1\\0\end{pmatrix}, |\psi\rangle_2 = \begin{pmatrix}0\\1\end{pmatrix}.$$

Таким чином, власні вектори оператора $\hat{S}$ співпадають з базисними векторами.

2.7 Логічний елемент Ф

Розглянемо ще один комплексний логічний елемент, який позначимо Ф. Елемент Ф задається матрицею

$$\Phi = \begin{pmatrix}1 & 0\\ 0 & \exp(i\phi_0)\end{pmatrix}. \qquad (2.29)$$

Подіємо матрицею Ф на вектор вхідного кубіту. Тоді отримаємо вектор вихідного кубіту у вигляді:

$$\begin{pmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & \exp(i\phi_0)\end{pmatrix}\begin{pmatrix}a\\ \beta\end{pmatrix} = \begin{pmatrix}\alpha\\ \beta\exp(i\phi_0)\end{pmatrix}. \qquad (2.30)$$

Відповідний оператор $\hat{\Phi}$ діє на кубіт за правилом

$$\hat{\Phi}\,|\psi\rangle = \hat{\Phi}\,(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,|0\rangle + \beta\exp(i\phi_0)|1\rangle. \qquad (2.31)$$

Таким чином, оператор $\widehat{\Phi}$ не змінює коефіцієнт при базисному векторі $|0\rangle$ і змінює фазу коефіцієнта при базисному векторі $|1\rangle$. Матриця Ф є унітарною.

Дійсно,

$$\Phi^{+}\Phi = \begin{pmatrix} 1 & 0 \\ 0 & \exp(-i\phi_0) \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \exp(i\phi_0) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathrm{I}.$$

Із унітарності матриці оператора слідує унітарність самого оператора. Представимо кубіт у вигляді (1.5)

$$|\psi\rangle = \cos\frac{\vartheta}{2}|0\rangle + e^{i\phi}\sin\frac{\vartheta}{2}|1\rangle.$$

Оператор $\widehat{\Phi}$ змінює фазу коефіцієнта при базисному векторі $|1\rangle$. На сфері Блоха це відповідає повороту на кут ϕ_0 навколо осі Z (див. рис. 2.6).

Знайдемо власні числа λ і власні вектори $|\psi\rangle_\lambda$ оператора $\widehat{\Phi}$:

$$\widehat{\Phi}\,|\psi\rangle_\lambda = \lambda\,|\psi\rangle_\lambda$$

або в матричному вигляді:

$$\begin{pmatrix} 1 & 0 \\ 0 & \exp(i\phi_0) \end{pmatrix}\begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = \lambda\begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda.$$

Тоді

$$\begin{pmatrix} 1-\lambda & 0 \\ 0 & \exp(i\phi_0)-\lambda \end{pmatrix}\begin{pmatrix} a \\ \beta \end{pmatrix}_\lambda = 0.$$

Власні числа матриці Ф визначаються із рівняння:

$$\begin{vmatrix} 1-\lambda & 0 \\ 0 & \exp(i\phi_0)-\lambda \end{vmatrix} = (1-\lambda)(\exp(i\phi_0)-\lambda) = 0.$$

Звідси

$$\lambda_1 = 1, \lambda_2 = \exp(i\phi_0).$$

У свою чергу, власні вектори матриці Ф визначаються згідно одній із двох умов:

$$(1-\lambda)a + 0\,\beta = 0$$

або

$$0a + (\exp(i\phi_0)-\lambda)\beta = 0.$$

Вважаючи $\lambda = \lambda_1 = 1$, бачимо, що перше із рівнянь задовольняється тотожно. Згідно другому рівнянню,

$$\beta = \beta_1 = 0.$$

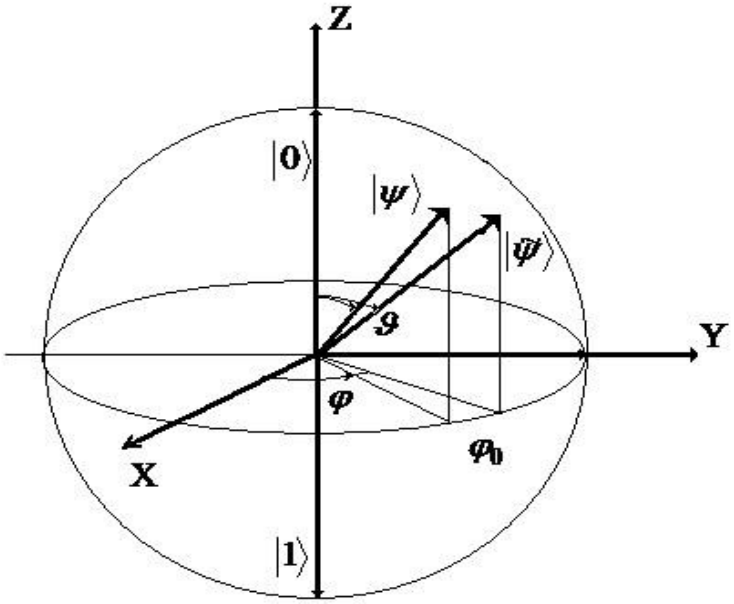


Рис. 2.6. Геометричне зображення перетворювання Ф
на сфері Блоха

Вважаючи $\lambda = \lambda_2 = \exp(i\phi_0)$, бачимо, що друге із рівнянь задовольняється тотожно. Згідно першому рівнянню

$$a = a_2 = 0.$$

Вибираючи $a_1 = \beta_2 = 1$, отримаємо:

$$|\psi\rangle_1 = \begin{pmatrix}1\\0\end{pmatrix},\ |\psi\rangle_2 = \begin{pmatrix}0\\1\end{pmatrix}.$$

Таким чином, власні вектори оператора $\hat{\Phi}$ співпадають з базисними векторами.

3 ДВОКУБІТОВІ СИСТЕМИ

3.1 Двокубітові стани і оператори

Розглянемо квантову систему, яка складається із двох кубітів:

$$|\psi_1\rangle = a_1|0_1\rangle + \beta_1|1_1\rangle,$$
$$|\psi_2| = a_2|0_2\rangle + \beta_2|1_2\rangle, \qquad (3.1)$$

де $|0_1\rangle, |1_1\rangle, |0_2\rangle, |1_2\rangle$ – базисні стани першого і другого кубіту відповідно, $a_1, \beta_1, a_2, \beta_2$ – комплексні числа. Як і раніше, у випадку однокубітової системи, імовірність виявити m-й кубіт (m=1,2) в стані $|0_m\rangle$ дорівнює $|a_m|^2$, а імовірність виявити його в стані $|1_m\rangle$ дорівнює $|\beta_m|^2$. Сума цих імовірностей дорівнює одиниці.

$$|a_m|^2 + |\beta_m|^2 = 1. \qquad (3.2)$$

Разом із вектором $|\psi_m\rangle$ уведемо ермітово-пов'язаний вектор $\langle\psi_m| = |\psi_m\rangle^+$. Тоді скалярний добуток двох векторів $|\psi_m\rangle$ і $|\widetilde{\psi}_m\rangle$ запишемо у вигляді $\langle\psi_m|\widetilde{\psi}_m\rangle$. Будемо вважати основні стани $|0_m\rangle$ і $|1_m\rangle$ ортонормованими:

$$\langle 0_m|0_m\rangle = 1, \langle 1_m|1_m\rangle = 1, \langle 0_m|1_m\rangle = 0. \qquad (3.3)$$

Тоді із умови нормованості хвильового вектора m-го кубіту $\langle\psi_m|\psi_m\rangle = 1$ відразу слідує вимога (3.2) на коефіцієнти a_m і β_m. Відмітимо, що кубіти $|\psi_1\rangle$ і $|\psi_2\rangle$ (а також і відповідні базисні вектори $|0_1\rangle, |1_1\rangle, |0_2\rangle, |1_2\rangle$ відносяться до різних векторних просторів – назвемо їх H_1 і H_2 :

$$|\psi_1\rangle \in H_1, |\psi_2\rangle \in H_2 .$$

Побудуємо векторний простір, елементами якого є пари векторів, перший із яких належить простору H_1, а другий – простору H_2. Такий простір називається тензорним добутком просторів H_1 і H_2. Він позначається $H_1 \otimes H_2$. Елементи його позначимо $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. Базисні вектори цього простору представляють собою тензорні добутки базисних векторів із просторів H_1 і H_2 :

$$|00\rangle = |0_1\rangle \otimes |0_2\rangle, \ |01\rangle = |0_1\rangle \otimes |1_2\rangle,$$
$$|10\rangle = |1_1\rangle \otimes |0_2\rangle, \ |11\rangle = |1_1\rangle \otimes |1_2\rangle. \qquad (3.4)$$

Ми будемо вважати, що двокубітова система знаходиться в стані $|00\rangle$, якщо її перший кубіт знаходиться в стані $|0_1\rangle$, а другий кубіт – в стані $|0_2\rangle$ і т. д. Пов'язані вектори в тензорному добутку просторів представляють собою тензорні добутки пов'язаних векторів із просторів співмножників:

$$\langle\psi| = \langle\psi_1| \otimes \langle\psi_2|.$$

Скалярний добуток визначається покомпонентно. Нехай

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle, \qquad |\widetilde{\psi}\rangle = |\widetilde{\psi}_1\rangle \otimes |\widetilde{\psi}_2\rangle .$$

Тоді скалярний добуток цих векторів в $H_1 \otimes H_2$ визначається як

$$\langle\psi|\tilde{\psi}\rangle_{H_1\otimes H_2} = \langle\psi_1|\tilde{\psi}_1\rangle_{H_1}\,\langle\psi_2|\tilde{\psi}_2\rangle_{H_2}. \quad (3.5)$$

У відповідності з (3.5) базисні вектори тензорного добутку просторів є ортонормованими:

$$\langle ij|mn\rangle = \langle i|m\rangle\langle j|n\rangle = \delta_{im}\delta_{jn}\,.$$

Розкладемо двокубітовий стан за базисними векторами:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = (a_1|0_1\rangle + \beta_1|I_1\rangle) \otimes (a_2|0_2\rangle + \beta_2|I_2\rangle),$$

$$|\psi\rangle = a_1a_2|0_1\rangle \otimes |0_2\rangle + a_1\beta_2|0_1\rangle \otimes |I_2\rangle +$$

$$+ \beta_1 a_2|I_1\rangle \otimes |0_2\rangle + \beta_1\beta_2|I_1\rangle \otimes |I_2\rangle.$$

Остаточно

$$|\psi\rangle = \gamma_{00}|00\rangle + \gamma_{01}|01\rangle + \gamma_{10}|10\rangle + \gamma_{11}|11\rangle, \quad (3.6)$$

де

$$\gamma_{mn} = \langle mn|\psi\rangle.$$

Як і у випадку одного кубіту, $|\gamma_{mn}|^2$ представляє собою імовірність знайти двокубітову систему в стані $|mn\rangle$. Сума таких імовірностей повинна дорівнювати одиниці:

$$|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2 = 1. \quad (3.7)$$

Це твердження також випливає із нормованості вектора $|\psi\rangle$:

$$\langle\psi|\psi\rangle = 1.$$

Із (3.7) виходить, що система двох кубітів описується вектором одиничної довжини. В чотирьохвимірному комплексному просторі коефіцієнтів кінець вектора $|\psi\rangle$ лежить на одиничній сфері (3.6). Проте багатовимірність простору коефіцієнтів не дозволяє використовувати цю обставину для наочного представлення стану двох кубітів за аналогією зі сферою Блоха. Оператори, які визначені в H_1 і H_2, діють в тензорному добутку просторів $H_1 \otimes H_2$ покомпонентно:

$$\left(\hat{U}_1 \otimes \hat{U}_2\right)(|\psi_1\rangle \otimes |\psi_2\rangle) = \left(\hat{U}_1|\psi_1\rangle\right) \otimes \left(\hat{U}_2|\psi_2\rangle\right).$$

3.2 Двокубітовий квантовий логічний елемент CNOT

Найбільш важливим двокубітовим оператором являється CNOT або керований NOT. Розглянемо базис тензорного добутку просторів $H_1 \otimes H_2$ (див. 3.4).

$$|mn\rangle = |m\rangle \otimes |n\rangle,$$

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

В цьому базисі матриця оператора CNOT має вигляд:

$$U_{\text{CN}} = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}. \tag{3.8}$$

Розглянемо дію цієї матриці на базисні вектори:

$$\begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

або

$$U_{\text{CN}}\,|00\rangle = |00\rangle,\;\; U_{\text{CN}}\,|01\rangle = |01\rangle,$$
$$U_{\text{CN}}|10\rangle = \;|11\rangle,\;\; U_{\text{CN}}\,|11\rangle = |10\rangle. \tag{3.9}$$

Таким чином, якщо перший кубіт знаходиться в стані $|0\rangle$, а другий кубіт знаходиться в одному із базових станів $|0\rangle$ або $|1\rangle$, то внаслідок дії оператора CNOT стан такої двокубітової системи не змінюється. Якщо ж перший кубіт знаходиться в стані $|1\rangle$, а другий – в одному із базових станів, то оператор CNOT переводить другий кубіт в інший базовий стан. Стан же першого кубіту не змінюється:

$$U_{\text{CN}}\,|0\rangle \otimes |0\rangle = |0\rangle \otimes |0\rangle,\;\; U_{\text{CN}}\,|0\rangle \otimes |1\rangle = |0\rangle \otimes |1\rangle, \tag{3.10}$$
$$U_{\text{CN}}\,|1\rangle \otimes |0\rangle = |1\rangle \otimes |1\rangle,\;\; U_{\text{CN}}\,|1\rangle \otimes |1\rangle = |1\rangle \otimes |0\rangle.$$

Будемо зображати оператор CNOT у вигляді квантової схеми (див. Рис. 3.1).

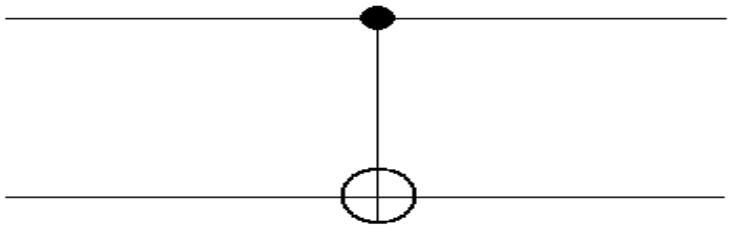


Рис. 3.1. Квантова схема, що зображує оператор CNOT

За двома провідниками, зліва направо, до схеми підводяться два кубіти. По двом правим провідникам від схеми відводяться два кубіти – результат дії схеми. На рис. 3.2 – 3.5 показані дії схеми на базисні двокубітові стани у відповідності з (3.10).

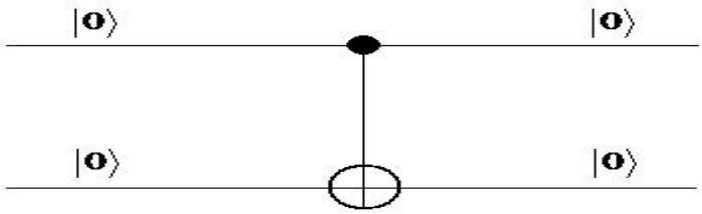


Рис. 3.2. Дія CNOT на базисний стан $U_{\mathrm{CN}}\,|0\rangle \otimes |0\rangle = |0\rangle \otimes |0\rangle$

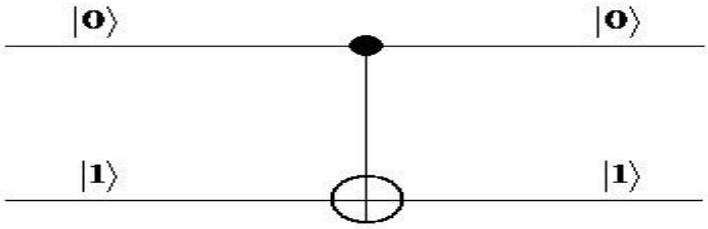


Рис. 3.3. Дія CNOT на базисний стан $U_{\mathrm{CN}}\,|0\rangle \otimes |1\rangle = |0\rangle \otimes |1\rangle$

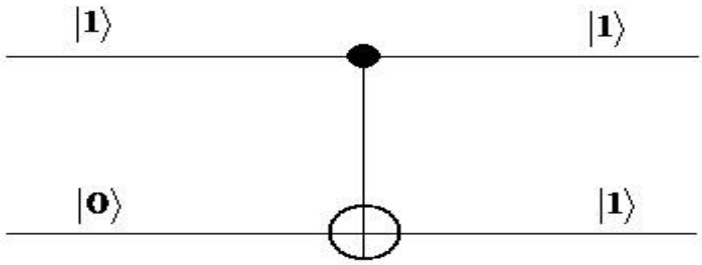


Рис. 3.4. Дія CNOT на базисний стан $U_{\mathrm{CN}}\,|1\rangle \otimes |0\rangle = |1\rangle \otimes |1\rangle$

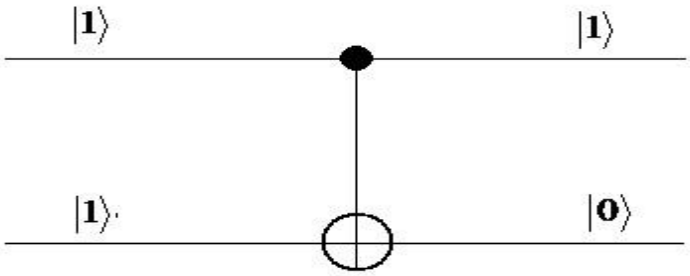


Рис. 3.5. Дія CNOT на базисний стан $U_{\mathrm{CN}}\,|1\rangle \otimes |1\rangle = |1\rangle \otimes |0\rangle$

Відмітимо, що діючи на базисний стан, квантовий оператор поводить себе як класичний. Для CNOT класичним прототипом є оператор XOR. На вхід класичного оператора XOR поступають два біти – керуючий і керований. Якщо керуючий біт знаходиться в стані 0, то керований біт не змінюється. Якщо керуючий біт знаходиться в стані 1, то керований біт змінюється на протилежний: 0 перетворюється в 1, а 1 в 0. Керуючий біт у всіх випадках залишається незмінним. Таке перетворювання можна представити у вигляді класичної схеми (див. Рис. 3.6).

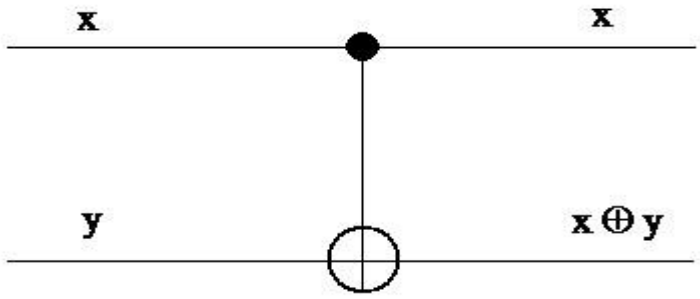


Рис. 3.6. Класичний елемент XOR

На вхід схеми подаються два біти x і y. На виході ми отримуємо незмінний керуючий біт *x*, при цьому керований біт перетворюється в x $\oplus$ y , тобто, в суму за модулем 2 керуючого і керованого бітів. Ця сума працює так:

$$0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0. \qquad (3.11)$$

Порівнюючи (3.11) з (3.10), бачимо, що квантовий елемент CNOT дійсно є узагальненням класичного елемента XOR. На базисних векторах CNOT веде себе так же, як XOR.

Розглянемо двокубітовий стан, в якому перший – керуючий кубіт – знаходиться в стані $|0_1\rangle$, а другий – керований кубіт – в довільному стані $a|0_2\rangle + \beta|1_2\rangle$. Вектор, який описує такий стан, має вигляд:

$$|\Psi\rangle = 1|0_1\rangle \otimes (a|0_2\rangle + \beta|I_2\rangle) = a|0_1\rangle \otimes |0_2\rangle + \beta|0_1\rangle \otimes |I_2\rangle$$

або

$$|\Psi\rangle = a|00\rangle + \beta|01\rangle + 0|10\rangle + 0|11\rangle = \begin{pmatrix} a \\ \beta \\ 0 \\ 0 \end{pmatrix}.$$

Діючи оператором CNOT на такий вектор стану, отримаємо

$$|\tilde{\psi}\rangle = U_{CN}\,|\Psi\rangle = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}\begin{pmatrix} a \\ \beta \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ \beta \\ 0 \\ 0 \end{pmatrix},$$

$$|\tilde{\psi}\rangle = |0_1\rangle \otimes (a|0_2\rangle + \beta|I_2\rangle) = |\Psi\rangle.$$

Отже, якщо керуючий кубіт дорівнює $|0_1\rangle$, то який би не був керований кубіт, він залишається незмінним (разом з керованим кубітом) внаслідок дії оператора CNOT.

Розглянемо тепер двокубітовий стан, в якому перший – керуючий кубіт – знаходиться в стані $|I_1\rangle$, а другий – керований кубіт – в довільному стані $a|0_2\rangle + \beta|1_2\rangle$. Вектор, який описує такий стан, має вигляд:

$$|\Psi\rangle = 1|1_1\rangle \otimes (a|0_2\rangle + \beta|1_2\rangle) = a|1_1\rangle \otimes |0_2\rangle + \beta|1_1\rangle \otimes |I_2\rangle$$

або

$$|\Psi\rangle = 0|00\rangle + 0\,|01\rangle + a|10\rangle + \beta|11\rangle = \begin{pmatrix} 0 \\ 0 \\ a \\ \beta \end{pmatrix}.$$

Діючи оператором CNOT на такий вектор стану, отримаємо:

$$|\tilde{\psi}\rangle = U_{\text{CN}}\,|\Psi\rangle = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ a \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ a \\ \beta \end{pmatrix},$$

$$|\tilde{\psi}\rangle = |1_1\rangle \otimes (\beta|0_2\rangle + a|\mathrm{I}_2\rangle).$$

Отже, якщо керуючий кубіт дорівнює $|1_1\rangle$, то дія оператора CNOT на керований кубіт еквівалентна дії оператора NOT. Якраз ця властивість дозволяє називати оператор, що розглядається, керованим оператором NOT.

3.3 Можливість копіювання невідомого біта за допомогою двокубітового квантового елемента CNOT

Покажемо, що класичний елемент CNOT (XOR) можна використати для копіювання невідомого біта. Для цього подаємо на керуючий вхід невідомий біт *x*, а на керований вхід подаємо 0. Тоді, у відповідності з рис. 3.6, керуючий вхід не зміниться, а керований перетворимо в $x \oplus 0$. Використовуючи властивості додавання за модулем 2, бачимо, що

$$x \oplus 0 = x.$$

Отже, керуючий біт став рівним копії керованого біта (див. Рис. 3.7).

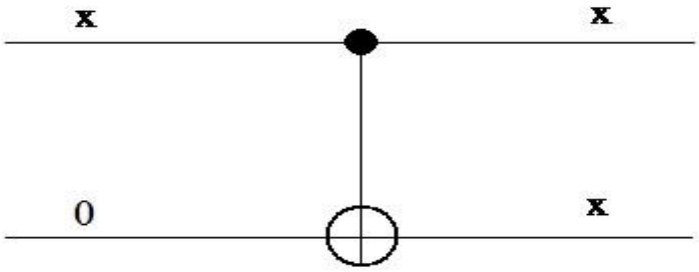


Рис. 3.7. Копіювання невідомого біта за допомогою класичного елемента CNOT

Спробуємо тепер використати квантовий оператор CNOT для копіювання кубіту. Для цього на керуючий вхід подаємо кубіт

$$|\psi\rangle = a|0\rangle + \beta|1\rangle,$$

а на керований вхід подаємо нульовий базисний вектор $|0\rangle$. Розрахуємо вхідний двокубітовий стан:

$$|\Psi\rangle = (a|0_1\rangle + \beta|I_1\rangle) \otimes |0_2\rangle$$

або

$$|\Psi\rangle = a|0_1\rangle \otimes |0_2\rangle + \beta|I_1\rangle \otimes |0_2\rangle = a|00\rangle + 0|01\rangle + \beta|10\rangle + 0|11\rangle.$$

Таким чином, в базисі (3.4) вхідний вектор може бути представлений у вигляді стовпця коефіцієнтів

$$|\Psi\rangle = \begin{pmatrix} a \\ 0 \\ \beta \\ 0 \end{pmatrix}.$$

Тоді вихідний вектор $|\tilde{\psi}\rangle = U_{\text{CN}} |\Psi\rangle$,

$$|\tilde{\psi}\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ 0 \\ \beta \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ 0 \\ 0 \\ \beta \end{pmatrix}. \qquad (3.12)$$

Для того, щоб CNOT копіював кубіт, на виході у нього повинно було б бути дві копії вхідного кубіту. Тоді двокубітовий вихідний стан мав би вигляд:

$$|\tilde{\tilde{\psi}}\rangle = (a|0_1\rangle + \beta|I_1\rangle) \otimes (a|0_2\rangle + \beta|I_2\rangle)$$

або

$$|\tilde{\tilde{\psi}}\rangle = a^2|0_1\rangle \otimes |0_2\rangle + a\beta|0_1\rangle \otimes |I_2\rangle + a\beta|1_1\rangle \otimes |0_2\rangle + \beta^2|1_1\rangle \otimes |I_2\rangle,$$

$$|\tilde{\tilde{\psi}}\rangle = a^2|00\rangle + a\beta|01\rangle + a\beta|10\rangle + \beta^2|11\rangle.$$

Представимо $|\tilde{\tilde{\psi}}\rangle$ у векторній формі

$$|\tilde{\tilde{\psi}}\rangle = \begin{pmatrix} a^2 \\ a\beta \\ a\beta \\ \beta^2 \end{pmatrix}. \qquad (3.13)$$

Прирівнюючи (3.12) і (3.13), отримаємо умови копіювання кубіту у вигляді системи рівнянь:

$$\begin{cases} a^2 = a \\ a\beta = 0 \\ \beta^2 = \beta \end{cases}. \qquad (3.14)$$

У системи (3.14) є три можливих розв’язки:

$$\begin{aligned} &1)\ a = 0,\ \beta = 0, \\ &2)\ a = 1,\ \beta = 0, \\ &3)\ a = 0,\ \beta = 1. \end{aligned} \qquad (3.15)$$

Таким чином, копіювання кубіту загального вигляду за допомогою оператора CNOT неможливе. Отримані випадки (3.15), в яких копіювання можливе, відповідають класичним станам.

Неможливість копіювання (як кажуть, клонування) невідомого кубіту доведена для довільного унітарного оператора. Як уже відмічалося вище, багаторазове копіювання кубіту (якби воно було можливе) дозволило б витягти із одного кубіта скільки завгодно багато інформації за допомогою багатократних вимірювань стану копій початкового кубіту (див. підрозділ 7.7).

4 КВАНТОВІ СХЕМИ

4.1 Однокубітові квантові схеми

Розглянемо два послідовно включені однокубітові квантові логічні елементи. На вхід першого елемента поступає кубіт, який описується хвильовим вектором $|\Psi\rangle$.

Дія першого елемента на кубіт описується оператором $\hat{U}_1$. Результатом дії оператора $\hat{U}_1$ на вектор $|\Psi\rangle$ є хвильовий вектор $|\Psi_1\rangle$ (див. Рис. 4.1).

Розглянемо тепер другий однокубітовий квантовий логічний елемент $\hat{U}_2$. На його вхід поступає хвильовий вектор $|\Psi_1\rangle$ – результат дії логічного елемента $\hat{U}_1$. На виході логічного елемента $\hat{U}_2$ з'являється хвильовий вектор $|\Psi_2\rangle$ (див. Рис. 4.2).

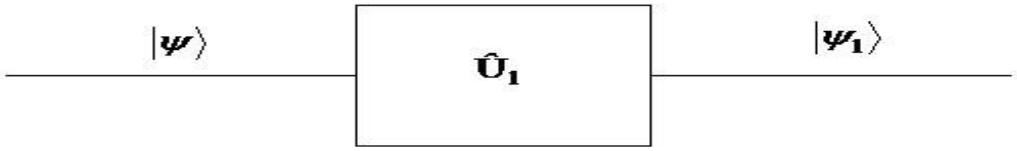


Рис. 4.1. Дія квантового логічного елемента $\hat{U}_1$

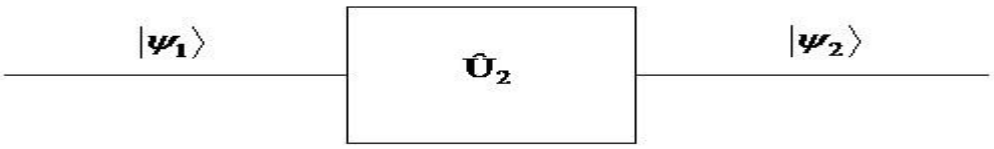


Рис. 4.2. Дія квантового логічного елемента $\hat{U}_2$

Послідовна дія операторів $\hat{U}_1$ і $\hat{U}_2$ представляється квантовою схемою, що зображена на рис. 4.3.

Таким чином, перетворювання початкового вектора $|\Psi\rangle$ в кінцевий вектор $|\Psi_2\rangle$ за рахунок послідовної дії операторів $\hat{U}_1$ і $\hat{U}_2$ можна записати так:

$$|\Psi_2\rangle = \hat{U}_2 |\Psi_1\rangle = \hat{U}_2\hat{U}_1 |\Psi\rangle. \quad (4.1)$$

В матричному представленні квантова схема, що зображена на рис. 4.3, описується матрицею U, яка дорівнює добутку матриць $\hat{U}_1$ і $\hat{U}_2$:

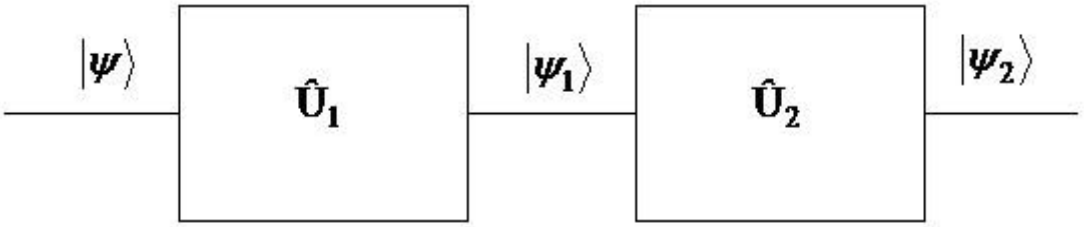


Рис. 4.3. Послідовна дія квантового логічного елемента $\hat{U}_1$ і квантового логічного елемента $\hat{U}_2$

$$U = \hat{U}_2 \hat{U}_1 . \quad (4.2)$$

Добуток унітарних матриць є унітарним. Тому послідовна дія двох унітарних перетворювань являється унітарним перетворюванням.

Описаний алгоритм конструювання однокубітових квантових логічних схем узагальнюється на загальний випадок послідовної дії *n* логічних елементів. На рис. 4.4 представлений ланцюжок із *n* послідовно з'єднаних елементів $\hat{U}_1$, …, $\hat{U}_n$.

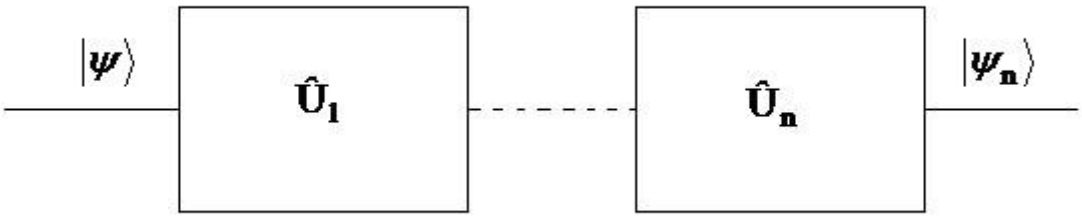


Рис. 4.4. Послідовна дія квантових логічних елементів $\hat{U}_1$, …, $\hat{U}_n$

На вхід ланцюжка поступає хвильовий вектор $|\Psi\rangle$, на виході з'являється хвильовий вектор $|\Psi_n\rangle$. Перетворювання початкового вектора $|\Psi\rangle$ в кінцевий вектор $|\Psi_n\rangle$ внаслідок послідовної дії операторів $\hat{U}_1$, …, $\hat{U}_n$ можна записати так:

$$|\Psi_n\rangle = \hat{U}_n|\Psi_{n-1}\rangle = \ldots = \hat{U}_n\hat{U}_{n-1}\ldots\hat{U}_1 |\Psi\rangle . \quad (4.3)$$

В матричному представленні квантова схема, що зображена на рис. 4.4, описується матрицею U, яка дорівнює добутку матриць $\hat{U}_n\hat{U}_{n-1}\ldots\hat{U}_1$:

$$U = \hat{U}_n\hat{U}_{n-1}\ldots\hat{U}_1. \quad (4.4)$$

Добуток унітарних матриць є унітарним. Тому послідовна дія *n* унітарних перетворювань являється унітарним добутком.

4.2 Однокубітові квантові схеми, які побудовані із однакових елементів

Розглянемо найпростіший випадок послідовної дії однакових дійсних симетричних квантових елементів. Матриці, що описують такі елементи,

симетричні і мають дійсні коефіцієнти. Прикладом таких елементів є логічні елементи NOT (X), Z і елемент Адамара H.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{4.5}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{4.6}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4.7}$$

Як і будь-які матриці, які описують квантові логічні елементи, матриці (4.5) – (4.7) унітарні, тобто,

$$U^{+}U = I. \tag{4.8}$$

Дійсні симетричні матриці володіють додатковою властивістю

$$U^{+} = U. \tag{4.9}$$

Із (4.8) і (4.9) слідує рівність

$$UU = I. \tag{4.10}$$

Таким чином, послідовна дія двох однакових дійсних симетричних квантових елементів еквівалентна одиничному оператору. Отже, послідовна дія парного числа дійсних симетричних квантових елементів також еквівалентна одиничному оператору.

$$\underbrace{U\ U \ldots U}_{2n} = I. \tag{4.11}$$

Послідовна дія непарного числа дійсних симетричних квантових елементів U еквівалентна дії одного елемента U:

$$\underbrace{U\ U \ldots U}_{2n+1} = U. \tag{4.12}$$

Розглянемо тепер випадок послідовної дії однакових комплексних квантових елементів.

Логічний квантовий елемент Y

Елемент Y задається матрицею

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{4.13}$$

Специфіка цього комплексного логічного елементу в тому, що (як і у симетричних дійсних елементів) квадрат цього елементу дорівнює одиниці:

$$Y^2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = 1. \tag{4.14}$$

Логічний квантовий елемент S

Елемент **S** задається матрицею

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \quad (4.15)$$

Особливість цього комплексного логічного елементу в тому, що квадрат цього елементу дорівнює квантовому логічному елементу Z**:**

$$S^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} = Z. \quad (4.16)$$

Можливість виразити один квантовий логічний елемент через інші дозволяє поставити питання про мінімальний набір елементів, через які можна виразити усі інші. Зокрема, уже зараз видно, що в цей мінімальний набір можна включити S, але не включати Z.

Логічний квантовий елемент $\boldsymbol{\frac{\pi}{8}}$

Розглянемо ще один комплексний логічний елемент T, який часто позначається $\boldsymbol{\frac{\pi}{8}}$.

Елемент T задається матрицею

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix}. \quad (4.17)$$

Так як цей елемент здійснює поворот навколо осі **Z** (на сфері Блоха) на кут $\boldsymbol{\frac{\pi}{8}}$, природно припустити, що четвертий степінь цього елементу здійснює поворот на кут $\boldsymbol{\frac{\pi}{2}}$. Дійсно

$$T^4 = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{8}\right) \end{pmatrix}^4 = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\frac{\pi}{2}\right) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = S. \quad (4.18)$$

Таким чином, четвертий степінь елемента T співпадає з елементом S.

Нагадаємо, що квадрат елемента S співпадає з елементом Z. Таким чином, при побудові мінімального набору елементів в нього можна включити T, але не включати S і Z.

Логічний елемент Ф

Елемент Ф задається матрицею

$$Ф = \begin{pmatrix} 1 & 0 \\ 0 & \exp(iФ) \end{pmatrix}. \quad (4.19)$$

Цей елемент здійснює поворот навколо осі Z (на сфері Блоха) на кут Ф.

Частковими випадками цього елементу є: T (поворот на кут $\frac{\pi}{8}$ на сфері Блоха), елемент S (поворот на кут $\frac{\pi}{2}$) і елемент Z (поворот на кут $\frac{\pi}{2}$). Для кутів повороту $Ф = 2\pi\frac{m}{n}$, n – а степінь цього оператора співпадає з одиничним оператором.

4.3 Однокубітові квантові схеми, які побудовані із різних елементів

На множині елементів X, Y, Z, H виконуються наступні тотожності:

$$HXH = Z, \tag{4.20}$$

$$HYH = -Y, \tag{4.21}$$

$$HZH = X. \tag{4.22}$$

Дійсно:

$$HXH = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} = Z,$$

$$HYH = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = -\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix} = -Y,$$

$$HZH = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} = X.$$

4.4 Двокубітові квантові схеми

Розглянемо квантову систему, що складається із двох кубітів:

$$|\psi_1\rangle = a_1|0_1\rangle + \beta_1|I_1\rangle ,$$

$$|\psi_2\rangle = a_2|0_2\rangle + \beta_2|I_2\rangle, \tag{4.23}$$

де $|0_1\rangle, |I_1\rangle, |0_2\rangle, |I_2\rangle$ – базисні стани першого і другого кубітів відповідно, $a_1, \beta_1, a_2, \beta_2$ – комплексні числа.

Відмітимо, що кубіти $|\psi_1\rangle$ і $|\psi_2\rangle$ (а також і відповідні базисні вектори $|0_1\rangle, |I_1\rangle, |0_2\rangle, |I_2\rangle$) відносяться до різних векторних просторів – назвемо їх H_1 і H_2:

$$|\psi_1\rangle \in H_1 , |\psi_2\rangle \in H_2 .$$

Побудуємо векторний простір, елементами якого являються пари векторів, перший із яких належить простору H_1, а другий – простору H_2. Такий простір називається тензорним добутком просторів H_1 і H_2. Він позначається $H_1 \otimes H_2$. Елементи його позначимо $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. Базисні вектори цього простору представляють собою тензорний добуток базисних векторів із просторів H_1 і H_2:

$$|00\rangle = |0_1\rangle \otimes |0_2\rangle, \ |01\rangle = |0_1\rangle \otimes |1_2\rangle,$$

$$|10\rangle = |1_1\rangle \otimes |0_2\rangle, \ |11\rangle = |1_1\rangle \otimes |1_2\rangle. \tag{4.24}$$

Оператори, які визначені в H_1 і H_2, діють в тензорному добутку просторів $H_1 \otimes H_2$ покомпонентно:

$$\left(\hat{U}_1 \otimes \hat{U}_2\right)(|\psi_1\rangle \otimes |\psi_2\rangle) = \left(\hat{U}_1|\psi_1\rangle\right) \otimes \left(\hat{U}_2|\psi_2\rangle\right). \tag{4.25}$$

В базисах просторів H_1 і H_2 вектори $|\psi_1\rangle$ і $|\psi_2\rangle$ представляються у вигляді стовпців:

$$|\psi_1\rangle = \begin{pmatrix} a_1 \\ \beta_1 \end{pmatrix}, \quad |\psi_2\rangle = \begin{pmatrix} a_2 \\ \beta_2 \end{pmatrix}, \tag{4.26}$$

а оператори $\hat{U}^{(1)}$ і $\hat{U}^{(2)}$, що діють в просторах H_1 і H_2 , відповідно – у вигляді матриць:

$$U^{(1)} = \begin{pmatrix} U_{00}^{(1)} & U_{01}^{(1)} \\ U_{10}^{(1)} & U_{11}^{(1)} \end{pmatrix},$$

$$U^{(2)} = \begin{pmatrix} U_{00}^{(2)} & U_{01}^{(2)} \\ U_{10}^{(2)} & U_{11}^{(2)} \end{pmatrix}. \tag{4.27}$$

Для того, щоб представити хвильові вектори і оператори тензорного добутку просторів $H_1 \otimes H_2$ в базисі (4.24), уведемо поняття тензорного (Кронекерова) добутку матриць.

Нехай A – матриця $m \times n$, B – матриця $r \times s$. Добуток Кронекера матриць A і B визначається як матриця $(m \cdot r) \times (n \cdot s)$

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \cdots & \cdots & \cdots & \cdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{pmatrix}. \tag{4.28}$$

Представимо базисні вектори просторів H_1 і H_2 у вигляді:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

$$|00\rangle = |0_1\rangle \otimes |0_2\rangle = \begin{pmatrix} 1 & .\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 & .\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$|01\rangle = |0_1\rangle \otimes |1_2\rangle = \begin{pmatrix} 1 & .\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 & .\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = |1_1\rangle \otimes |0_2\rangle = \begin{pmatrix} 0 & .\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1 & .\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \tag{4.29}$$

$$|11\rangle = |1_1\rangle \otimes |1_2\rangle = \begin{pmatrix} 0 & .\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 & .\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Запишемо тепер двокубітовий стан $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$

$$|\psi\rangle = \begin{pmatrix} a_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} a_2 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} a_1 & \begin{pmatrix} a_2 \\ \beta_2 \end{pmatrix} \\ \beta_1 & \begin{pmatrix} a_2 \\ \beta_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_1 & \beta_2 \\ \beta_1 & a_2 \\ \beta_1 & \beta_2 \end{pmatrix}.$$

Нарешті, тензорний (Кронекерів) добуток матричних представлень операторів $\widehat{\mathrm{U}}^{(1)}$ і $\widehat{\mathrm{U}}^{(2)}$ мають вигляд:

$$\mathrm{U}^{(1)} \otimes \mathrm{U}^{(2)} = \begin{pmatrix} \mathrm{U}_{00}^{(1)} & \begin{pmatrix} \mathrm{U}_{00}^{(2)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{10}^{(2)} & \mathrm{U}_{11}^{(2)} \end{pmatrix} & \mathrm{U}_{01}^{(1)} & \begin{pmatrix} \mathrm{U}_{00}^{(2)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{10}^{(2)} & \mathrm{U}_{11}^{(2)} \end{pmatrix} \\ \mathrm{U}_{10}^{(1)} & \begin{pmatrix} \mathrm{U}_{00}^{(2)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{10}^{(2)} & \mathrm{U}_{11}^{(2)} \end{pmatrix} & \mathrm{U}_{11}^{(1)} & \begin{pmatrix} \mathrm{U}_{00}^{(2)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{10}^{(2)} & \mathrm{U}_{11}^{(2)} \end{pmatrix} \end{pmatrix}$$

або

$$\mathrm{U} = \mathrm{U}^{(1)} \otimes \mathrm{U}^{(2)} =$$

$$= \begin{pmatrix} \mathrm{U}_{00}^{(1)} & \mathrm{U}_{00}^{(2)} & \mathrm{U}_{00}^{(1)} & \mathrm{U}_{01}^{(2)} & \mathrm{U}_{01}^{(1)} & \mathrm{U}_{00}^{(2)} & \mathrm{U}_{01}^{(1)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{00}^{(1)} & \mathrm{U}_{10}^{(2)} & \mathrm{U}_{00}^{(1)} & \mathrm{U}_{11}^{(2)} & \mathrm{U}_{01}^{(1)} & \mathrm{U}_{10}^{(2)} & \mathrm{U}_{01}^{(1)} & \mathrm{U}_{11}^{(2)} \\ \mathrm{U}_{10}^{(1)} & \mathrm{U}_{00}^{(2)} & \mathrm{U}_{10}^{(1)} & \mathrm{U}_{01}^{(2)} & \mathrm{U}_{11}^{(1)} & \mathrm{U}_{00}^{(2)} & \mathrm{U}_{11}^{(1)} & \mathrm{U}_{01}^{(2)} \\ \mathrm{U}_{10}^{(1)} & \mathrm{U}_{10}^{(2)} & \mathrm{U}_{10}^{(1)} & \mathrm{U}_{11}^{(2)} & \mathrm{U}_{11}^{(1)} & \mathrm{U}_{10}^{(2)} & \mathrm{U}_{11}^{(1)} & \mathrm{U}_{11}^{(2)} \end{pmatrix}. \qquad (4.30)$$

Як і раніше, у випадку однокубітової системи, імовірність виявити m –й кубіт (m = 1,2) в стані $|0_m\rangle$ дорівнює $|a_m|^2$, а імовірність виявити його в стані $|1_m\rangle$ дорівнює $|\beta_m|^2$. Сума цих імовірностей дорівнює одиниці:

$$|a_m|^2 + |\beta_m|^2 = 1. \qquad (4.31)$$

Поряд з вектором $|\psi_m\rangle$ уведемо ермітово-пов'язаний вектор $\langle\psi_m| = |\psi_m\rangle^*$. Тоді скалярний добуток двох векторів $|\psi_m\rangle$ і $|\tilde{\psi}_m\rangle$ представимо у вигляді $\langle\psi_m|\tilde{\psi}_m\rangle$. Будемо вважати основні стани $|0_m\rangle$ і $|1_m\rangle$ ортонормованими:

$$\langle 0_m|0_m\rangle = 1,\ \langle 1_m|1_m\rangle = 1,\ \langle 0_m|1_m\rangle = 0. \qquad (4.32)$$

Тоді із умови нормування хвильового вектора m –го кубіту $\langle\psi_m|\psi_m\rangle = 1$ відразу випливає вимога (4.31) на коефіцієнти a_m і β_m.

Ми будемо говорити, що двокубітова система знаходиться в стані $|00\rangle$, якщо її перший кубіт знаходиться в стані $|0_1\rangle$, а другий кубіт – в стані $|0_2\rangle$ і т. д. Зв'язані вектори в тензорному добутку просторів представляють собою тензорний добуток зв'язаних векторів із просторів – співмножників:

$$\langle\psi| = \langle\psi_1| \otimes \langle\psi_2|.$$

Скалярний добуток визначається покомпонентно. Нехай

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle, |\tilde{\psi}\rangle = |\tilde{\psi}_1\rangle \otimes |\tilde{\psi}_2\rangle.$$

Тоді скалярний добуток цих векторів $\mathrm{H}_1 \otimes \mathrm{H}_2$ визначається як:

$$\langle\psi|\tilde{\psi}\rangle_{\mathrm{H}_1\otimes\mathrm{H}_2} = \langle\psi_1|\tilde{\psi}_1\rangle_{\mathrm{H}_1}\langle\psi_2|\tilde{\psi}_2\rangle_{\mathrm{H}_2}. \quad (4.33)$$

У відповідності з (4.33) базисні вектори тензорного добутку просторів являються ортонормованими:

$$\langle ij|mn\rangle = \langle i|m\rangle\langle j|n\rangle = \delta_{im}\delta_{jn}.$$

Розкладемо двокубітовий стан за базисними векторами:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = (a_1|0_1\rangle + \beta_1|1_1\rangle) \otimes (a_2|0_2\rangle + \beta_2|1_2\rangle),$$
$$|\psi\rangle = a_1a_2|0_1\rangle \otimes |0_2\rangle + a_1\beta_2|0_1\rangle \otimes |1_2\rangle +$$
$$+\beta_1a_2|1_1\rangle \otimes |0_2\rangle + \beta_1\beta_2|1_1\rangle \otimes |1_2\rangle.$$

Остаточно

$$|\psi\rangle = \gamma_{00}|00\rangle + \gamma_{01}|01\rangle + \gamma_{10}|10\rangle + \gamma_{11}|11\rangle, \quad (4.34)$$

де

$$\gamma_{mn} = \langle mn|\psi\rangle.$$

Як і у випадку одного кубіту, $|\gamma_{mn}|^2$ представляє собою імовірність знайти двокубітову систему в стані $|mn\rangle$. Сума таких імовірностей повинна дорівнювати одиниці:

$$|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2 = 1. \quad (4.35)$$

Це твердження також випливає із нормованості вектора $|\psi\rangle$ -

$$\langle\psi|\psi\rangle = 1.$$

Із (4.35) випливає, що система двох кубітів описується вектором одиничної довжини. В чотирьохвимірному комплексному просторі коефіцієнтів кінець вектора $|\psi\rangle$ лежить на одиничній сфері (4.35). Проте багатовимірність простору коефіцієнтів не дозволяє використовувати цю обставину для наочного представлення стану двох кубітів за аналогією зі сферою Блоха.

4.5 Квантова схема, що створює стан Белла

Стан Белла або ЕПР- стан названий так на честь Ейнштейна, Подольського і Розена, які першими вказали на незвичайні властивості цих станів [1].

Таких станів чотири:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}|00\rangle + |11\rangle, |\beta_{01}\rangle = \frac{1}{\sqrt{2}}|01\rangle + |10\rangle,$$
$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}|00\rangle - |11\rangle, |\beta_{11}\rangle = \frac{1}{\sqrt{2}}|01\rangle - |10\rangle. \quad (4.36)$$

Для отримання станів Белла можна використати квантову схему, що приведена на рис. 4.5.

Схема складається із квантового логічного елемента CNOT, на керуючий вхід якого включений елемент Адамара H. Оператор, який описує цю квантову схему, має вигляд:

$$L = (I \otimes H) \cdot CNOT. \tag{4.37}$$

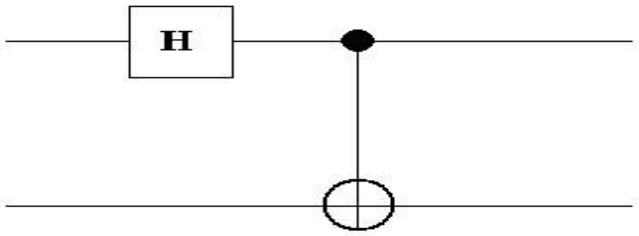


Рис. 4.5. Квантова схема, що створює стан Белла

Розглянемо дію цієї схеми, коли на її вхід подаються базисні елементи.

1. Нехай на обидва входи схеми подаються нульові базисні вектори $|0\rangle$. Знайдемо стан вектора після проходження елемента Адамара:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix}$$

або

$$H|0\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle).$$

Ми отримали вектор, який поступає на керуючий вхід елемента CNOT.

На керований вхід елемента CNOT поступає базовий вектор $|0\rangle$. Таким чином, на вхід елемента CNOT поступає вектор $|\psi\rangle$, який дорівнює:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |0\rangle$$

або

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 0\\ 1\\ 0\end{pmatrix}.$$

Розрахуємо дію оператора CNOT на такий вхідний вектор:

$$|\tilde{\psi}\rangle = U_{CN}|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1\\ 0\\ 1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 0\\ 0\\ 1\end{pmatrix}.$$

Таким чином,

$$|\beta_{00}\rangle = L|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{4.38}$$

2. Нехай на керуючий вхід схеми подається нульовий базисний вектор $|0\rangle$, а на керований вхід схеми – базисний вектор $|1\rangle$. Знайдемо стан вектора після проходження елемента Адамара:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix}$$

або

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle + \ |1\rangle).$$

Ми отримали вектор, який поступає на керуючий вхід елемента CNOT.

На керований вхід елемента CNOT поступає базовий вектор $|1\rangle$. Таким чином, на вхід елемента CNOT поступає вектор $|\psi\rangle$, який дорівнює:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle + \ |1\rangle) \otimes \ |1\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes \ |1\rangle + \ \frac{1}{\sqrt{2}}|1\rangle \otimes \ |1\rangle$$

або

$$|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\1\end{pmatrix}.$$

Розрахуємо дію оператора CNOT на такий вхідний вектор:

$$|\tilde{\psi}\rangle = \ \mathrm{U_{CN}}|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}0\\1\\0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\\0\end{pmatrix}.$$

Таким чином,

$$|\beta_{01}\rangle = L|00\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \qquad (4.39)$$

3. Нехай на керуючий вхід схеми подається одиничний базисний вектор $|1\rangle$, а на керований вхід схеми – базисний вектор $|0\rangle$. Знайдемо стан вектора $|1\rangle$ після проходження елемента Адамара:

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\1&-1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

або

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle - |1\rangle).$$

Ми отримали вектор, який поступає на керуючий вхід елемента CNOT. На керований вхід елемента CNOT поступає базовий вектор $|0\rangle$. Таким чином, на вхід елемента CNOT поступає вектор $|\psi\rangle$, який дорівнює:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle - \ |1\rangle) \otimes \ |0\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes \ |0\rangle - \ \frac{1}{\sqrt{2}}|1\rangle \otimes \ |0\rangle$$

або

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|10\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\-1\\0\end{pmatrix}.$$

Розрахуємо дію оператора CNOT на такий вхідний вектор:

$$|\tilde{\psi}\rangle = \ \mathrm{U_{CN}}|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\0\\-1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\-1\end{pmatrix}.$$

Таким чином,

$$|\beta_{10}\rangle = L|10\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle). \quad (4.40)$$

4. Нехай на керуючий вхід схеми подається одиничний базисний вектор $|1\rangle$, а на керований вхід схеми – базисний вектор $|1\rangle$. Знайдемо стан вектора $|1\rangle$ після проходження елемента Адамара:

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ -1\end{pmatrix}$$

або

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle - |1\rangle).$$

Ми отримали вектор, який поступає на керуючий вхід елемента CNOT. На керований вхід елемента CNOT поступає базовий вектор $|1\rangle$. Таким чином, на вхід елемента CNOT поступає вектор $|\psi\rangle$, який дорівнює:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\ (|0\rangle - \ |1\rangle) \otimes \ |1\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes \ |1\rangle - \ \frac{1}{\sqrt{2}}|1\rangle \otimes \ |1\rangle$$

або

$$|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|11\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\ 1\\ 0\\ -1\end{pmatrix}.$$

Розрахуємо дію оператора CNOT на такий вхідний вектор:

$$|\tilde{\psi}\rangle = \ \mathrm{U_{CN}}|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\\ 0\\ -1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix}.$$

Таким чином,

$$|\beta_{11}\rangle = L|00\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \quad (4.41)$$

Усі чотири випадки (4.38) – (4.41) можна коротко записати у вигляді:

$$|\beta_{ab}\rangle = \mathrm{L}|ab\rangle \equiv (\mathrm{I} \otimes \mathrm{H}) \cdot \mathrm{CNOT}\ |ab\rangle, \quad a,b=0,1. \quad (4.42)$$

5 ПРИКЛАДНІ АСПЕКТИ КВАНТОВОЇ ІНФОРМАТИКИ

5.1 Алгоритм розпізнавання функцій (алгоритм Дойча)

Розглянемо множину, яка складається із двох базисних векторів $|0\rangle$ і $|1\rangle$ і усі можливі функції, які відображають цю множину в себе. Таких функцій існує чотири. Дві функції f_1 і f_2, які називаються постійними, приймають одне і те ж значення при будь-яких значеннях аргументу:

$$f_1\ |0\rangle = |0\rangle, f_1\ |1\rangle = |0\rangle, \text{ або } f_1\ |x\rangle = |0\rangle, x = 0,1,$$

$$f_2\ |0\rangle = |1\rangle, f_2\ |1\rangle = |1\rangle, \text{ або } f_2\ |x\rangle = |1\rangle, x = 0,1.$$

Назвемо збалансованими дві інші функції f_3 і f_4 , які визначаються рівностями:

$$f_3\ |x\rangle = |x\rangle, f_4\ |x\rangle = NOT|x\rangle, x = 0,1.$$

Легко бачити, що перераховані функції вичерпують усі можливі відображення f : $\{|0\rangle, |1\rangle\} \rightarrow \{|0\rangle, |1\rangle\}$. Необхідно узнати, до якого із двох класів належить невідома функція *f*. В класичному комп'ютері для цього вимагається дві операції – необхідно розрахувати f $|0\rangle$ і f $|1\rangle$. В квантовому комп'ютері достатньо однієї операції. У випадку 2^m значень аргумента класичному комп'ютеру необхідно 2^m операцій, квантовому – m операцій. Відповідний квантовий алгоритм називається алгоритмом Дойча. Для реалізації цього алгоритму побудуємо двокубітовий унітарний лінійний оператор U_f , який діє на базисні елементи $|0\rangle$ і $|1\rangle$ за правилом:

$$U_f\ (|x\rangle\ \otimes\ |y\rangle) = |x\rangle \otimes (|y\rangle \oplus \text{f}\ |x\rangle),$$

де знаком $\oplus$ позначено додавання за модулем 2. Подивимося, як цей оператор виглядає для різних функцій *f*.

1) Враховуючи, що $f_1\ |x\rangle = |0\rangle$ і позначаючи $U_1\ \equiv\ U_{f_1}$, отримаємо

$$\text{U}_1(|x\rangle\ \otimes\ |y\rangle) = |x\rangle \otimes (|y\rangle \oplus f_1\ |x\rangle) = |x\rangle \otimes (|y\rangle \oplus |0\rangle) = |x\rangle\ \otimes\ |y\rangle.$$

Це означає, що U_1 є одиничним оператором:

$$\text{U}_1 = 1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (5.1)$$

2) Аналогічно для оператора U_2 маємо:

$$\text{U}_2(|x\rangle\ \otimes\ |y\rangle) = |x\rangle \otimes (|y\rangle \oplus f_2|x\rangle) = |x\rangle \otimes (|y\rangle \oplus |1\rangle) = |x\rangle\ \otimes\ \text{NOT}|y\rangle.$$

Використовуючи визначення тензорного добутку операторів (4.30), отримаємо:

$$\text{U}_2(|x\rangle\ \otimes\ |y\rangle) = (\text{I} \otimes\ \text{NOT}\ )\ (|x\rangle \otimes\ |y\rangle)\ .$$

$$U_2 = I \otimes NOT = \begin{pmatrix} NOT & 0 \\ 0 & NOT \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.2)$$

3) Для оператора U_3 отримуємо:

$$U_3(|x\rangle \otimes |y\rangle) = |x\rangle \otimes (|y\rangle \oplus f_3\ |x\rangle) = |x\rangle \otimes (|y\rangle \oplus |x\rangle). \quad (5.3)$$

Порівнюючи (5.3) з визначенням оператора CNOT, бачимо, що

$$U_3 = CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.4)$$

4) Нарешті, для оператора U_4 отримаємо

$$U_4(|x\rangle \otimes |y\rangle) = |x\rangle \otimes (|y\rangle \oplus f_4\ |x\rangle) = |x\rangle \otimes (|y\rangle \oplus NOT|x\rangle). \quad (5.5)$$

Порівнюючи (5.5) з (5.3), (5.4), бачимо:

$$U_4 = CNOT \cdot (NOT \otimes I).$$

З іншої сторони,

$$NOT \otimes I = \begin{pmatrix} 0 \cdot 1 & 1 \cdot 1 \\ 1 \cdot 1 & 0 \cdot 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Остаточно:

$$U_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (5.6)$$

Розглянемо тепер квантову схему, що реалізує алгоритм Дойча (див. Рис. 5.1).

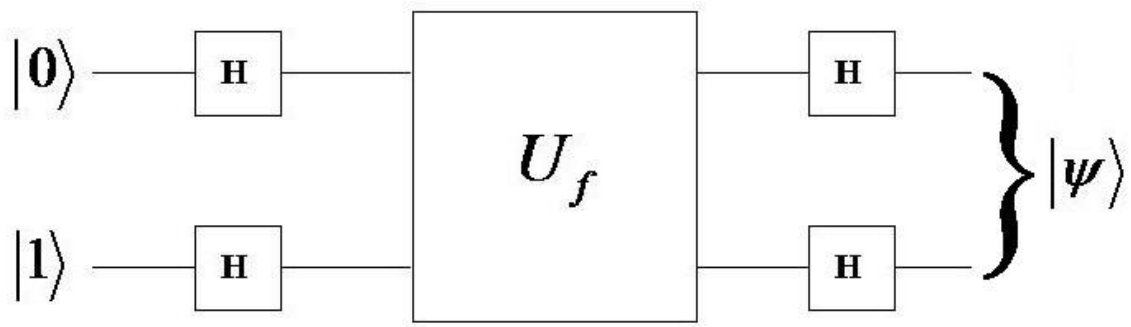


Рис. 5.1. Квантова схема, що реалізує алгоритм Дойча

Розпізнавання функції f замінюється на розпізнавання оператора U_f. На вхід схеми завжди подається двокубітовий стан:

$$|0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}.$$

Ця хвильова функція опрацьовується оператором H ⊗ H:

$$\text{H} \otimes \text{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix}.$$

В результаті отримується стан

$$|\chi\rangle = (\text{H} \otimes \text{H})(|0\rangle \otimes |1\rangle) = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix}\begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix},$$

який і поступає на вхід невідомого логічного оператора U_f , а потім на вхід оператора H ⊗ H. Отже, сигнал на виході схеми має вигляд (H ⊗ H) · U_f · $|\chi\rangle$.

Розглянемо чотири випадки:

1) U_1= 1.

$$|\psi_1\rangle = (\text{H} \otimes \text{H}) \cdot |\chi\rangle = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix} \cdot \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\1\end{pmatrix} = |0\rangle \otimes |1\rangle. \qquad (5.7)$$

Цей же результат можна отримати простіше. Робота схеми в загальному вигляді описується оператором L_f = (H ⊗ H) · U_f · (H ⊗ H), який в даному випадку зводиться до одиничного оператора:

$$L_1 = (\text{H} \otimes \text{H}) \cdot (I \otimes I) \cdot (\text{H} \otimes \text{H}) = (\text{H} \cdot \text{H}) \otimes (\text{H} \cdot \text{H}) = (I \otimes I).$$

Отже, $|\psi_1\rangle = L_1|0\rangle \otimes |1\rangle = |0\rangle \otimes |1\rangle$.

2) $$\text{U}_2 = \begin{pmatrix}0 & 1 & 0 & 0\\1 & 0 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}.$$

Тоді $\text{U}_2\,|\chi\rangle = \frac{1}{2}\begin{pmatrix}0 & 1 & 0 & 0\\1 & 0 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix} = -|\chi\rangle$.

Використовуючи (5.7), отримаємо

$$|\psi_2\rangle = -\,(\text{H} \otimes \text{H}) \cdot \text{w} = -\,|0\rangle \otimes |1\rangle. \qquad (5.8)$$

2) $\mathbf{U_3}$ = CNOT.

$$|\psi_3\rangle = (\text{H} \otimes \text{H}) \cdot \text{CNOT} \cdot (\text{H} \otimes \text{H})|\chi\rangle.$$

$$|\psi_3\rangle = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1-1 & 1 & -1 \\ 1 & 1 & -1-1 \\ 1-1-1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |1\rangle \otimes |1\rangle. \quad (5.9)$$

3) U4 = CNOT· (NOT ⊗ 1). Використовуючи (5.6), отримаємо

$$|\psi_4\rangle = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1-1 & 1 & -1 \\ 1 & 1 & -1-1 \\ 1-1-1 & 1 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |1\rangle \otimes |1\rangle. \quad (5.10)$$

Отже, перший канал вихідного сигналу для перших двох випадків, тобто, для постійних функцій, містить $|1\rangle$. Для другої пари випадків (для збалансованих функцій) перший канал вихідного сигналу містить $|0\rangle$, у другому каналі вихідного сигналу у всіх чотирьох випадках знаходиться $|1\rangle$.

5.2 Зверхущільнене кодування

Ідея методу полягає в передачі одного кубіту для кодування двох класичних бітів. Аліса і Боб створюють декілька екземплярів станів Белла $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \equiv \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}}$. Боб забирає по одному кубіту із кожної пари. Аліса кодує кожну пару послідовних бітів ab оператором L_{ab} і діє цим оператором на свій кубіт пари.

Парі кубітів 00 вона співставляє одиничний оператор L_{00} = 1,

парі 01 – оператор L_{01}= X = $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$,

парі 10 – оператор L_{10}= Z = $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$,

парі 11 – оператор L_{11}= iY = $i\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$.

Аліса може впливати тільки на свій кубіт пари, тому її дії на усю пару описуються оператором $\mathbf{L_{ab} \otimes I}$.

Враховуючи властивості операторів L_{ab}:

$L_{00}|0\rangle = |0\rangle$, $L_{00}|1\rangle = |1\rangle$,

$L_{01}|0\rangle = |1\rangle$, $L_{01}|1\rangle = |0\rangle$,

$L_{10}|0\rangle = |0\rangle$, $L_{10}|1\rangle = -|1\rangle$,

$L_{11}|0\rangle = -|1\rangle$, $L_{11}|1\rangle = |0\rangle$,

бачимо, що

$$(\mathrm{L}_{00} \otimes \mathrm{I})|\beta_{00}\rangle = (\mathrm{I} \otimes \mathrm{I})\ \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}} = \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}},$$

$(\mathrm{L}_{00} \otimes \mathrm{I})|\beta_{00}\rangle = |\beta_{00}\rangle$,

$$(\mathrm{L}_{01} \otimes \mathrm{I})|\beta_{00}\rangle = (L_{01} \otimes \mathrm{I})\ \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}} = \frac{|1\rangle\otimes|0\rangle+|0\rangle\otimes|1\rangle}{\sqrt{2}},$$

$$(\mathrm{L}_{01} \otimes \mathrm{I})|\beta_{00}\rangle = |\beta_{01}\rangle,$$

$$(\mathrm{L}_{10} \otimes \mathrm{I})|\beta_{00}\rangle = (L_{10} \otimes \mathrm{I})\ \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}} = \frac{|0\rangle\otimes|0\rangle-|0\rangle\otimes|1\rangle}{\sqrt{2}},$$

$$(\mathrm{L}_{10} \otimes \mathrm{I})|\beta_{00}\rangle = |\beta_{10}\rangle,$$

$$(\mathrm{L}_{11} \otimes \mathrm{I})|\beta_{00}\rangle = (L_{11} \otimes \mathrm{I})\ \frac{|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle}{\sqrt{2}} = \frac{-|1\rangle\otimes|0\rangle+|0\rangle\otimes|1\rangle}{\sqrt{2}},$$

$$(\mathrm{L}_{11} \otimes \mathrm{I})|\beta_{00}\rangle = |\beta_{11}\rangle.$$

Таким чином, $(\mathrm{L}_{\mathrm{ab}} \otimes \mathrm{I})|\beta_{00}\rangle = |\beta_{ab}\rangle$. Після кодування Аліса пересилає свій кубіт Бобу. Якщо Аліса хотіла переслати Бобу послідовність кубітів *ab*, то у Боба виявляється переплутана пара кубітів в стані Белла $|\beta_{ab}\rangle$. Тепер цей сигнал необхідно розкодувати.

Нагадаємо, що стан Белла можна отримати за допомогою квантової схеми, що зображена на рис. 5.2:

$$|\beta_{ab}\rangle = (\mathrm{I} \otimes \mathrm{H}) \cdot \mathrm{CNOT}\ |ab\rangle,\ ab = 0,1.$$

Для розкодування необхідно сконструювати схему, яка здійснює обернену операцію:

$$|ab\rangle \equiv |a\rangle \otimes |b\rangle = ((\mathrm{I} \otimes \mathrm{H}) \cdot CNOT)^{-1}|\beta_{ab}\rangle\ .$$

Використовуючи властивості унітарних операторів, отримаємо

$$((\mathrm{I} \otimes \mathrm{H}) \cdot CNOT)^{-1} = ((\mathrm{I} \otimes \mathrm{H}) \cdot CNOT)^{+} = (\mathrm{I} \otimes\ \mathrm{H})^{+} CNOT^{+}\ .$$

Остаточно

$$((\mathrm{I} \otimes \mathrm{H}) \cdot CNOT)^{-1} = (\mathrm{I} \otimes \mathrm{H}) \cdot CNOT. \qquad (5.11)$$

Оператору (5.11) відповідає декодуюча схема (див. Рис. 5.2).

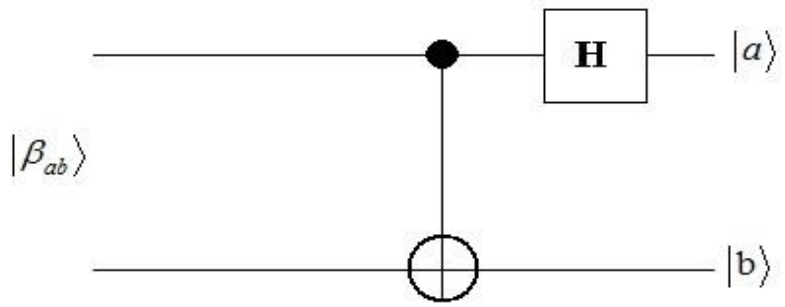


Рис. 5.2. Декодуюча схема методу зверхущільненого кодування

5.3 Використання стану Белла в квантовій телепортації

Будемо розуміти під квантовою телепортацією перенесення невідомого квантового стану із однієї системи на іншу з використанням квантового каналу зв'язку. Розглянемо одну із можливих схем квантової телепортації (див. Рис. 5.3).

Аліса хоче передати Бобу невідомий квантовий стан

$$|\psi\rangle = a\left|\begin{matrix}\mathrm{A}\\0\end{matrix}\right\rangle + \beta\left|\begin{matrix}\mathrm{A}\\1\end{matrix}\right\rangle.$$

Аліса і Боб приготовляють стан

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left[\left|\begin{matrix}\mathrm{BC}\\00\end{matrix}\right\rangle + \left|\begin{matrix}\mathrm{BC}\\11\end{matrix}\right\rangle\right].$$

Тут і далі ми записуємо над кубітом назву квантового провідника, по якому він передається. Боб забирає один кубіт з ЕПР пари, другий кубіт залишається у Аліси. На вхід схеми подається трьохкубітовий стан.

$$|\varphi_1\rangle = |\psi\rangle|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\ a\left[\left|\begin{matrix}\mathrm{A}\\0\end{matrix}\right\rangle\left[\left|\begin{matrix}\mathrm{BC}\\00\end{matrix}\right\rangle + \left|\begin{matrix}\mathrm{BC}\\11\end{matrix}\right\rangle\right] + \beta\left|\begin{matrix}\mathrm{A}\\1\end{matrix}\right\rangle\left[\left|\begin{matrix}\mathrm{BC}\\00\end{matrix}\right\rangle + \left|\begin{matrix}\mathrm{BC}\\11\end{matrix}\right\rangle\right]\right].$$

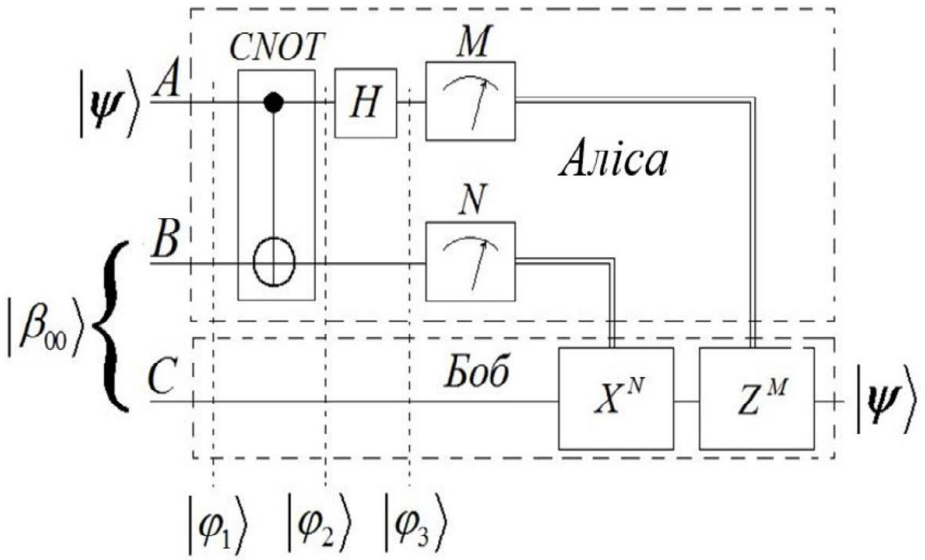


Рис. 5.3. Схема квантової телепортації. Одинарні лінії зображають квантові канали, подвійні – класичні

Оператор CNOT переплутує кубіт$|\psi\rangle$, що передається, з одним із кубітів ЕПР пари. При цьому виникає стан

$$|\varphi_2\rangle = (CNOT \otimes 1)|\varphi_1\rangle,$$

$$|\varphi_2\rangle = \frac{1}{\sqrt{2}}\left[a\left|\begin{matrix}\mathrm{A}\\0\end{matrix}\right\rangle\left[\left|\begin{matrix}\mathrm{BC}\\00\end{matrix}\right\rangle + \left|\begin{matrix}\mathrm{BC}\\11\end{matrix}\right\rangle\right] + \beta\left|\begin{matrix}\mathrm{A}\\1\end{matrix}\right\rangle\left[\left|\begin{matrix}\mathrm{BC}\\10\end{matrix}\right\rangle + \left|\begin{matrix}\mathrm{BC}\\01\end{matrix}\right\rangle\right]\right].$$

Стан $|\varphi_2\rangle$ступає на вхід оператора H ⊗ I ⊗ I. В результаті виникає стан $|\varphi_3\rangle$. Враховуючи властивості оператора H.

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

Бачимо, що

$$|\varphi_3\rangle = (H \otimes I \otimes I)\,|\varphi_2\rangle =$$

$$= \frac{1}{\sqrt{2}}\left[a\left[\left|\begin{matrix}A\\0\end{matrix}\right\rangle + \left|\begin{matrix}A\\1\end{matrix}\right\rangle\right]\left[\left|\begin{matrix}BC\\00\end{matrix}\right\rangle + \left|\begin{matrix}BC\\11\end{matrix}\right\rangle\right] + \beta\left[\left|\begin{matrix}A\\0\end{matrix}\right\rangle - \left|\begin{matrix}A\\1\end{matrix}\right\rangle\right]\left[\left|\begin{matrix}BC\\10\end{matrix}\right\rangle + \left|\begin{matrix}BC\\01\end{matrix}\right\rangle\right]\right].$$

Остаточно

$$|\varphi_3\rangle = \frac{1}{2}\left[\left|\begin{matrix}AB\\00\end{matrix}\right\rangle a\left[\left|\begin{matrix}C\\0\end{matrix}\right\rangle + \beta\left|\begin{matrix}C\\1\end{matrix}\right\rangle\right] + \left|\begin{matrix}AB\\01\end{matrix}\right\rangle a\left[\left|\begin{matrix}C\\1\end{matrix}\right\rangle + \beta\left|\begin{matrix}C\\0\end{matrix}\right\rangle\right] +$$

$$+ \left|\begin{matrix}AB\\10\end{matrix}\right\rangle a\left[\left|\begin{matrix}C\\0\end{matrix}\right\rangle - \beta\left|\begin{matrix}C\\1\end{matrix}\right\rangle\right] + \left|\begin{matrix}AB\\11\end{matrix}\right\rangle a\left[\left|\begin{matrix}C\\1\end{matrix}\right\rangle - \beta\left|\begin{matrix}C\\0\end{matrix}\right\rangle\right]\right]. \qquad (5.12)$$

Згідно (5.12), якщо по провідникам A і B розповсюджується стан $\left|\begin{matrix}AB\\00\end{matrix}\right\rangle$, то по провіднику C розповсюджується кубіт $|\psi\rangle$. В інших випадках кубіт, що розповсюджується по провіднику C, відрізняється від $|\psi\rangle$, проте може бути перетворений в $|\psi\rangle$ за допомогою коректуючих операторів. Для розпізнавання сигналів, які розповсюджуються по провідникам A і B, служать вимірювальні оператори M і N. При попаданні на них двокубітового оператора $\left|\begin{matrix}AB\\MN\end{matrix}\right\rangle$, вони видають пару чисел (MN).

Коректуючий оператор має вигляд $X^M Y^N$. Розглянемо його роботу. Згідно (5.12):

1) якшо по провідникам A і B розповсюджується стан $\left|\begin{matrix}AB\\00\end{matrix}\right\rangle$, то по провіднику *C* розповсюджується кубіт $a\left|\begin{matrix}C\\0\end{matrix}\right\rangle + \beta\left|\begin{matrix}C\\1\end{matrix}\right\rangle = |\psi\rangle$. В цьому випадку M=0, N=0 і коректуючий оператор $X^0Z^0=1$.

2) якшо по провідникам A і B розповсюджується стан $\left|\begin{matrix}AB\\01\end{matrix}\right\rangle$, то по провіднику C розповсюджується кубіт $a\left|\begin{matrix}C\\1\end{matrix}\right\rangle + \beta\left|\begin{matrix}C\\0\end{matrix}\right\rangle$. В цьому випадку M=0, N=1 і коректуючий оператор $X^1Z^0=X$.

3) якшо по провідникам A і B розповсюджується стан $\left|\begin{matrix}AB\\10\end{matrix}\right\rangle$, то по провіднику C розповсюджується кубіт $a\left|\begin{matrix}C\\0\end{matrix}\right\rangle - \beta\left|\begin{matrix}C\\1\end{matrix}\right\rangle$. В цьому випадку M=1, N=0 і коректуючий оператор $X^0Z^1= Z$.

4) якшо по провідникам A і B розповсюджується стан $\left|\begin{matrix}AB\\11\end{matrix}\right\rangle$, то по провіднику C розповсюджується кубіт $a\left|\begin{matrix}C\\1\end{matrix}\right\rangle - \beta\left|\begin{matrix}C\\1\end{matrix}\right\rangle$. В цьому випадку M=1, N=1 і коректуючий оператор $X^1Z^1= XZ$.

6 ІНФОРМАЦІЙНА ЕНТРОПІЯ

6.1 Класична інформаційна ентропія

Нехай класична система може знаходитися в одному із n станів, причому в стані k система знаходиться із імовірністю p_k. Визначимо, згідно Шеннону, інформаційну ентропію S такою класичною системою співвідношень:

$$S = -\sum_k p_k log_2 p_k. \quad (6.1)$$

Згідно (6.1), інформаційна ентропія системи, що знаходиться з імовірністю 1 в одному із можливих станів, дорівнює нулю. Протилежний крайній випадок – система, яка рівноймовірно знаходиться в будь-якому із n станів. Згідно (6.1), інформаційна ентропія такої системи дорівнює $log_2 n$ (див. Рис. 6.1).

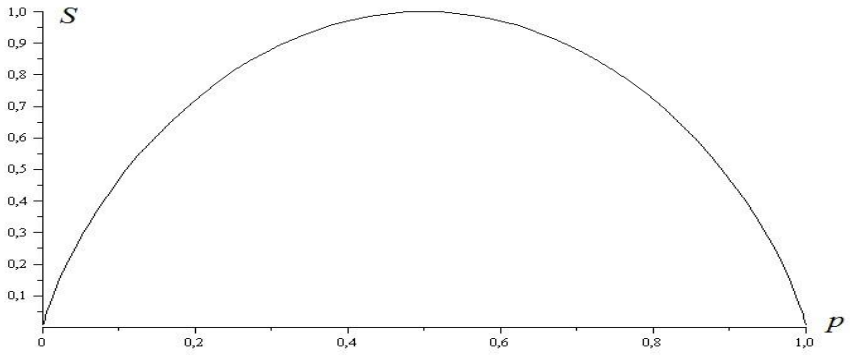


Рис. 6.1. Залежність класичної інформаційної ентропії дворівневої системи від імовірності реалізації одного із станів

Розглянемо, для прикладу, систему, яка може знаходитися всього в двох можливих станах. Імовірність першого стану дорівнює p, імовірність другого 1-p. Тоді, згідно (6.1), інформаційна ентропія такої системи дорівнює:

$$S(p) = - p \log_2 p - (1-p) \log_2 (1-p). \quad (6.2)$$

Із (6.2) видно, що S(0) = S(1) =0, S(0.5) = $\log_2 2$=1. Графік залежності інформаційної ентропії від імовірності p представлений на рис. 6.1. Із малюнка видно, що ентропія являється функцією, яка симетрична відносно значення аргумента $p = \frac{1}{2}$.

Відмітимо, що при p = 0 ентропія розраховується як ліміт (6.2) при $p \to 0$. Те саме зауваження справедливе і для загальної формули (6.1).

6.2 Квантова інформаційна ентропія

Перед тим, як уводити квантову ентропію, визначимо функцію від оператора. Нехай $\hat{L}$ – лінійний оператор, λ_k і $|\psi_k\rangle$ – його власні числа і власні вектори. Тоді оператор можна представити у вигляді:

$$\hat{L} = \sum_n \lambda_n |\psi_n\rangle \langle\psi_n|. \quad (6.3)$$

Визначимо функцію від оператора $\hat{L}$ співвідношенням:

$$\mathrm{f}(\hat{L}) = \sum_n \mathrm{f}(\lambda_n) |\psi_n\rangle \langle\psi_n|. \quad (6.4)$$

Розглянемо функцію

$$\mathrm{f}(x) = \begin{cases} -x \log_2 x, & x > 0 \\ 0, & x = 0 \end{cases}.$$

Уведемо ще слід від матриці оператора $\hat{A}$ співвідношенням

$$\mathrm{Tr}\,\hat{A} = \sum_n \langle\psi_n|\hat{A}|\psi_n\rangle. \quad (6.5)$$

Нехай $\hat{\rho}$ – матриця щільності системи, що розглядається. Квантовою ентропією (ентропією фон-Неймана) називається величина

$$\mathrm{S}(\hat{\rho}) = \mathrm{Tr}\,(\mathrm{f}(\hat{\rho})). \quad (6.6)$$

Використовуючи (6.3) – (6.6), отримаємо:

$$\hat{\rho} = \sum_n \lambda_n |\psi_n\rangle \langle\psi_n|, \quad (6.7)$$

$$\mathrm{S}(\hat{\rho}) = \mathrm{Tr}\left(\sum_n \mathrm{f}(\lambda_n) |\psi_n\rangle \langle\psi_n|\right)$$

або

$$\mathrm{S}(\hat{\rho}) = \left(\sum_{n,k} \mathrm{f}(\lambda_n) \langle\psi_k|\psi_n\rangle \langle\psi_n|\psi_k\rangle\right) = \sum_{n,k} \mathrm{f}(\lambda_n)\, \delta_{kn}.$$

Остаточно

$$\mathrm{S}(\hat{\rho}) = \sum_n \mathrm{f}(\lambda_n) = -\sum_n \lambda_n \log_2(\lambda_n). \quad (6.8)$$

Розрахуємо квантову ентропію декількох систем.

Приклад 6.1. Розглянемо спочатку один кубіт $|\psi\rangle = a|0\rangle + \beta|1\rangle$. Його матриця щільності

$$\hat{\rho} = |\psi_n\rangle \langle\psi_n| \quad (6.9)$$

або

$$\hat{\rho} = (a|0\rangle + \beta|1\rangle)(a^*\langle 0| + \beta^*\langle 0|) =$$

$$= aa^*|0\rangle\langle 0| + a\beta^*|0\rangle\langle 1| + \beta a^*|1\rangle\langle 0| + \beta\beta^*|1\rangle\langle 1|. \quad (6.10)$$

Таким чином, матриця щільності кубіту має вигляд

$$\hat{\rho} = \begin{pmatrix} |a|^2 & a\beta^* \\ a^*\beta & |\beta|^2 \end{pmatrix}. \quad (6.11)$$

Власні числа матриці щільності являються коренями рівняння

$$\begin{vmatrix} |a|^2 - \lambda & a\beta^* \\ a^*\beta & |\beta|^2 - \lambda \end{vmatrix} = 0.$$

Враховуючи умову нормування $|a|^2 + |\beta|^2 = 1$, знаходимо власні числа $\lambda_1 = 0$, $\lambda_2 = 1$. Згідно (6.8),

$$\mathrm{S}(\hat{\rho}_A) = -\lambda_1 \log_2(\lambda_1) - \lambda_2 \log_2(\lambda_2) = 0. \quad (6.12)$$

Приклад 6.2. Розглянемо тепер переплутаний двокубітовий стан - один із станів Белла:

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|0_A\rangle|1_B\rangle - |1_A\rangle|0_B\rangle).$$

Відповідна матриця щільності має вигляд:

$$\hat{\rho}_{AB} = |\beta_{11}\rangle\langle\beta_{11}| = \frac{1}{2}(|0_A\rangle|1_B\rangle - |1_A\rangle|0_B\rangle)(\langle 0_A|\langle 1_B| - \langle 1_A|\langle 0_B|). \quad (6.13)$$

Побудуємо тепер матрицю щільності підсистеми А, усереднивши матрицю щільності АВ за усіма можливими станами підсистеми В:

$$\hat{\rho}_A = \langle 0_B|\hat{\rho}_{AB}|0_B\rangle + \langle 1_B|\hat{\rho}_{AB}|1_B\rangle.$$

Використовуючи (6.13), отримаємо

$$\hat{\rho}_A = \frac{1}{2}|0_A\rangle\langle 0_A| + \frac{1}{2}|1_A\rangle\langle 1_A|. \quad (6.14)$$

Таким чином, матриця щільності має вигляд:

$$\hat{\rho} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}. \quad (6.15)$$

Власні числа матриці щільності являються коренями рівняння

$$\begin{vmatrix} \frac{1}{2} - \lambda & 0 \\ 0 & \frac{1}{2} - \lambda \end{vmatrix} = 0. \quad (6.16)$$

Таким чином, матриця щільності має двократний корінь $\lambda_1 = \lambda_2 = \frac{1}{2}$.

Згідно (6.8)

$$S(\hat{\rho}_A) = 1. \quad (6.17)$$

Приклад 6.3. Розглянемо тепер двокубітовий стан загального вигляду:

$$|\psi\rangle = a|0_A\rangle|0_B\rangle + b|0_A\rangle|1_B\rangle + c|1_A\rangle|0_B\rangle + d|1_A\rangle|1_B\rangle$$

з умовою нормування

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1.$$

Відповідна матриця щільності має вигляд:

$$\hat{\rho}_{AB} = |\psi\rangle\langle\psi|. \quad (6.18)$$

Побудуємо тепер матрицю щільності підсистеми А, усереднивши матрицю щільності АВ за усіма можливими станами підсистеми В:

$$\hat{\rho}_A = \langle 0_B|\hat{\rho}_{AB}|0_B\rangle + \langle 1_B|\hat{\rho}_{AB}|1_B\rangle.$$

Використовуючи (6.18), отримаємо:

$$\hat{\rho}_A = (|a|^2 + |b|^2)|0_A\rangle\langle 0_A| + (ac^* + bd^*)|0_A\rangle\langle 1_A| + (a^*c + b^*d)|1_A\rangle\langle 0_A| + (|c|^2 + |d|^2)|1_A\rangle\langle 1_A|. \quad (6.19)$$

Таким чином, матриця щільності має вигляд:

$$\hat{\rho} = \begin{pmatrix} |a|^2 + |b|^2 & ac^* + bd^* \\ a^*c + b^*d & |c|^2 + |d|^2 \end{pmatrix}. \qquad (6.20)$$

Власні числа матриці щільності являються коренями рівняння

$$\begin{vmatrix} |a|^2 + |b|^2 - \lambda & ac^* + bd^* \\ a^*c + b^*d & |c|^2 + |d|^2 - \lambda \end{vmatrix} = 0$$

або

$$\lambda^2 - \lambda + |a|^2|d|^2 + |b|^2|c|^2 - ab^*c^*d - a^*b\,c\,d^* = 0. \qquad (6.21)$$

Згідно (6.8),

$$S(\hat{\rho}_A) = -\lambda_1 \log_2(\lambda_1) - \lambda_2 \log_2(\lambda_2) . \qquad (6.22)$$

Зауваження. Попередні приклади показують, що значення квантової ентропії $S(\hat{\rho}_A)$ можна використовувати для того, щоб характеризувати степінь переплутаності стану AB. Проте існує і більш прямий метод оцінювання степені переплутаності двокубітового стану. Розглянемо наряду із загальною хвильовою функцією двокубітового стану

$$|\psi\rangle = a\,|0_A\rangle|0_B\rangle + b|0_A\rangle|1_B\rangle + c\,|1_A\rangle|0_B\rangle + d\,|1_A\rangle|1_B\rangle \qquad (6.23)$$

прямий добуток двох кубітів:

$$|\psi\rangle = (a|0\rangle + \beta\,|1\rangle)(\gamma|0\rangle + \delta|1\rangle) =$$

$$= a\gamma|00\rangle + a\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle. \qquad (6.24)$$

Прирівнюючи коефіцієнти в розкладаннях (6.23) і (6.24), знаходимо

$$a = a\gamma, \quad b = a\delta, \quad c = \beta\gamma, \quad d = \beta\delta. \qquad (6.25)$$

Із (6.25) випливає необхідна і достатня умова на коефіцієнти довільного двокубітового стану, при виконанні якого двокубітовий стан являється тензорним добутком двох однокубітових станів:

$$ad - bc = 0. \qquad (6.26)$$

При виконанні умови (6.26) двокубітовий стан являється абсолютно не переплутаним. Отже, величину

$$P = ad - bc \qquad (6.27)$$

можна розглядати як міру переплутаності двокубітового стану.

7 КВАНТОВА ТЕОРІЯ ОБЧИСЛЕНЬ

Обчислення нерозривно пов’язані з реальними фізичними системами.

Строго кажучи, ніякі обчислення не можуть бути виконані без реальних фізичних систем. Існують навіть різні форми тезису Черча, що легалізують цей тісний зв’язок. Так, Дойч запропонував версію тезису Черча у вигляді: “Кожна фізична система, яка кінцево реалізується, може бути повністю промодельована універсальною обчислювальною машиною, що діє кінцевими засобами”.

До переваг формулювання тезису Дойча-Черча можна віднести її фізичність і велику визначеність. В принципі, цей тезис стає фізичним принципом, із якого природним методом виходить необхідність переходу в квантову область. Дослідження енергетичних витрат на проведення обчислень стимулювали спроби більш глибокого аналізу природи обчислень, яка використовує квантову механіку. Фейнман, аналізуючи обмеження, мабуть вперше сформулював питання про можливість квантового комп’ютера [2]. Правда, уведення такого комп’ютера або алгоритму вимагає, перш за все, створення мікроскопічної квантової теорії обчислень. Як і в класичній теорії обчислень, це означає уведення елементарних квантових “атомів” інформації або квантів інформації і елементарних (“атомарних”) операцій. У якості таких атомів інформації фігурує квантовий біт або кубіт.

7.1 Наочне уведення в квантові представлення

Перш за все, чим обговорювати таку мікроскопічну теорію квантових обчислень, необхідно затриматися на квантовій механіці. Як відмічалося вище, квантова теорія більш фундаментальна, чим класична. Зокрема, це означає, що багато чого, що складно пояснюється в класичній фізиці, а інколи зовсім не пояснюється в класичній фізиці, стає зрозумілим і прозорим у квантовій фізиці. Це стосується і реальних обчислень. Дійсно, спробуємо “пояснити” обчислення людини, що розв’язує деяку проблему [3]. Природно, має сенс говорити про деяку тривіальну проблему, яка має два розв’язки або тільки дві відповіді “ТАК” або “НІ”. Наприклад, їй (людині) треба прийняти рішення: вивчати квантовий комп’ютер, чи ні. Позначимо варіанти відповіді “НІ” як 0 і “ТАК” як 1. Почнемо з того, що поставимо перед нею цю проблему. Природно, до отримання відповіді людині властиві роздуми, її думка буде змінюватися в процесі роздумів. Тому природно увести деяку кількісну характеристику, яка описує ці зміни. Позначимо її величиною ψ. Природа цієї величини поки що невідома. Природно, коливання думок до прийняття рішення можуть змінюватися оборотним чином від 0 до 1. Людина, що розглядається нами, нічим не виділена і на її рішення, крім роздумів (які і самі носять випадковий характер), можуть впливати випадкові фактори. Тому для розуміння смислу величини ψ звернемося до уявного експерименту. Візьмемо N людей і влаштуємо опитування думок з проблеми, що розв’язується. Зрозуміло, що частина людей N_1 відповість на задане питання позитивно, а частина N_0 - негативно. Таким чином, із

експерименту можна мати очікувану імовірність відповіді “ТАК” – $W_1 = N_1/\mathrm{N}$ і очікувану імовірність відповіді “НІ” – $W_0 = N_0/\mathrm{N}$. Умова нормування імовірності $W_1 + W_0 = 1$ виконується. Цю умову можна виконати автоматично, якщо вибрати ці імовірності у вигляді $W_1 = cos^2(\theta)$ і $W_0 = sin^2(\theta)$. Тепер уведемо вектор $\psi = (C_1, C_2) = (\cos(\theta), \sin(\theta))$, квадрати проєкцій якого співпадають з імовірностями відповідних розв’язків. Так, $W_1 = C_1^2$, а $W_0 = C_2^2$ і, відповідно, вектор ψ одиничної довжини із-за умови нормування. На рис. 7.1 показаний процес еволюції хвильової функції ψ в процесі прийняття рішення. Відмітимо, що двовимірний вектор завжди можна розглядати як комплексну функцію і далі будемо використовувати як той, так і інший термін. Уводячи позначення Дірака для базисних векторів відповідей $|0\rangle$ і $|1\rangle$, вектор ψ представимо як $\psi = C_1|1\rangle + C_2|0\rangle$. Тепер стан людини на етапі роздумів описується вектором ψ, а квадрати модулів комплексних коефіцієнтів або координат цього вектора описують імовірність відповідного рішення. Фактично ψ характеризує схильність людини до тієї чи іншої відповіді. Очевидно, що ця схильність змінюється у процесі обчислень.

Коливання в процесі розв’язку проблеми тепер можна інтерпретувати як поворот одиничного вектора ψ в двовимірному просторі.

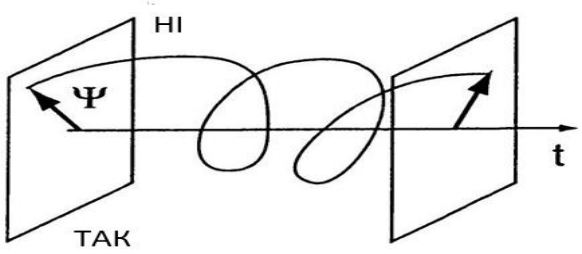


Рис. 7.1. Процес еволюції хвильової функції $\boldsymbol{\psi}$
при прийнятті рішення

Це єдина можливість еволюції ψ із-за умови нормування. У квантовій механіці ця функція називається хвильовою функцією або вектором стану. З іншої сторони, коливання ψ на етапі обчислення можна розглядати як змінювання хвильової функції під дією деякого оператора, що дає інший стан людини на етапі прийняття рішення або іншу хвильову функцію ψ'.

$$\psi' = \widehat{U}\,\psi.$$

Так як коливання у прийнятті рішення відповідають повороту вектора ψ, то оператор $\widehat{U}$ не може бути довільним. Зрозуміло, що коливання думки оборотні (до прийняття рішення), тому існує обернений оператор $\widehat{U}^{-1}$, дія якого на ψ' повертає ψ в початковий стан

$$\psi = \widehat{U}^{-1}\,\psi'.$$

Крім цього, використовуючи визначення скалярного добутку векторів і збереження довжини векторів, легко довести, що оператори $\widehat{U}$ унітарні. Таким чином, стан проблеми на етапі обчислення характеризується хвильовою

функцією, а її змінювання в процесі обчислень відповідає змінюванням під дією унітарних операторів. Відмітимо, що хвильова функція нам принципово невідома до прийняття рішення людиною. Це є важливим елементом квантової механіки. Прийняття рішення або закінчення обчислень супроводжується незворотним процесом переходу від ψ до відповідей "НІ" (0) або "ТАК" (1). Цей процес легко інтерпретувати у термінах вимірювань або оператора проєктування $\hat{P}$ ($\hat{P}^2 = \hat{P}$). Вимірювання отриманого результату відповідає проєктуванню вектора ψ на один із базисних станів $|0\rangle$ або $|1\rangle$. Зрозуміло, що при цьому стан людини миттєво змінюється, переходячи із стану коливань ψ, як хотілося б надіятися, в стан $|1\rangle$. Надмірно говорити, що стан після вимірювання відрізняється від стану системи до вимірювання. Фактично на цьому прикладі ми обговорили на якісному рівні як основні елементи квантової механіки, так і грубі риси квантових обчислень. Нижче уточнимо деякі важливі деталі, які упущені в розглянутій спрощеній схемі.

7.2 Елементи квантової механіки

В цьому підрозділі дуже коротко згадаємо основні математичні структури квантової механіки. За більш детальним викладенням необхідно звернутися до підручників з квантової фізики. Досить часто у фізиці робляться спроби пояснити складні фізичні явища без формул або, як прийнято говорити, на пальцях. В дійсності існує декілька типів таких пояснень. До першого типу, зрозуміло, умовно можна віднести пояснення уже існуючі і також загальноприйняті. В цьому випадку пояснюючий пробує визвати в пам'яті відповідні асоціації, які у свою чергу визивають потрібні фрагменти фізичної і математичної картини. Наступний тип пояснень на пальцях апелює до асоціацій між відомим і невідомим, пробуючи встановити між ними зв'язок і нові властивості невідомого. І нарешті, можливо, самий цікавий тип, який пробує створити інтуїтивне представлення про невідоме, формулюючи принципи, які дозволяють просунутися в розумінні сенсу за допомогою оцінок, які заміняють точні знання про невідоме. Усі ці типи якісних пояснень і будуть використані нижче.

7.2.1 Принципи класичної фізики

Перед тим, як перейти до обговорення квантової фізики, зручно нагадати принципи класичної фізики. В цьому підрозділі основну увагу зосередимо не на законах класичної фізики, а на її принципах, які, як правило, навіть не обговорюються в більшості підручників із фізики. Мабуть, самий загальний принцип класичної фізики можна назвати *принципом абсолютного реалізму*. Під ним розуміється абсолютизація фізичного процесу або явища. Вважається, що процес об'єктивно існує і відбувається абсолютно незалежно від спостерігача. Іншими словами, у класичній фізиці представлення розігрується незалежно, присутні глядачі на цьому спектаклі або відсутні, або переживання глядачів не впливає на події спектаклю. Цей гордий принцип виявляється виключно природним. Однак, це всього лише принцип. Наскільки далеко його можна

розповсюдити – питання експериментальної перевірки. Чи відповідає він Природі, обговоримо далі.

Наступний принцип – це принцип абсолютизації стану системи. Згідно цьому принципу стан фізичної системи може бути вивчено абсолютно точним образом. Саме на цьому базується горда заява Лапласа: "Дайте мені початкові умови і я передбачу еволюцію Світу". Зрозуміло, з цим зв'язаний детермінізм класичної фізики. Іще більш важливо, що до цього принципу прив'язується найважливіше фізичне поняття замкненої фізичної системи. Дійсно, при роздумі над цим принципом стає зрозумілим абстрактність його застосування по відношенню до усього нашого Всесвіту. Тут на допомогу приходить природне для людей бажання усе розділити на більш прості частини і вивчити їх окремо. В цьому проявляється виправдавший себе підхід Homo Sapiens. Поняття ізольованої системи у фізиці появляється як абстракція цього бажання виділити частину Всесвіту, на яку не впливають інші її частини. Частина, звичайно, простіше цілого і її вивчення спрощується. Так, розкладаючи Всесвіт на частини і припускаючи універсальність фізичних законів, досягається прогрес у пізнанні усього Всесвіту. Виділена частина і буде замкнутою фізичною системою, стан якої уже стає цілком осяжним. Зрозуміло, у фізиці зустрічаються і замкнуті системи, які більш складно співвідносяться з оточуючим світом. Так, наприклад, в деяких випадках вплив оточення усе-таки може враховуватися. В якості прикладу можна привести систему у контакті з термостатом. В цьому випадку відбувається обмін температурою, а не частками системи. У всіх випадках вплив оточення враховується усереднено і головне – відсутній зворотний зв'язок. Після виділення замкнутої системи, яка складається, наприклад, із набору часток, стан такої системи визначається значеннями положень і імпульсами часток, що входять до неї. Далі, для простоти, часто будуть використовуватися приклади із області класичної фізики. Зрозуміло, що це не означає, що усе сказане про принципи обмежується класичною фізикою. В дійсності ці принципи відносяться до усіх областей класичної фізики. Таким чином, множина станів замкненої системи створює простір, який називається фазовим простором. Часто у фізиці використовується і його підпростір – простір положень системи. Цей підпростір називається конфігураційним простором.

Приклад 7.1. *Цікаво нагадати, що апорія Зенона або парадокс Зенона, що ставить питання про відмінність стану руху від стану спокою, якраз і передбачав поняття фазового простору. По викладу Арістотеля, що дійшов до нас, в парадоксі про стрілу, що летить, Зенон стверджує: будь-яка річ або рухається, або стоїть на місці. Проте, ніщо не може перебувати в русі, займаючи простір, який рівний йому по протяжності. У визначений момент тіло, що рухається (в даному випадку стріла), постійно знаходиться на одному місці. Отже, стріла, що летить, не рухається. В коментарях Сімплікія формулює цей парадокс в більш стисненій формі: "Предмет, що летить, завжди займає простір, рівний собі, але те, що завжди займає рівний собі простір, не рухається. Отже, він покоїться". Дійсно, цей парадокс демонстрував недостатність конфігураційного простору для вивчення руху. З точки зору конфігураційного простору неможливо відрізнити стан спокою*

стріли від стану польоту. Тут ми маємо справу з проєкцією фазового простору на конфігураційний. Світ в конфігураційному просторі - це всього лише світ тіней або проєкцій.

Визначення стану системи означає визначення положення системи у фазовому просторі.

Наступний принцип класичної фізики - це принцип *ідеальної вимірюваності фізичних систем*. Іншими словами, стан класичної системи можна скільки завгодно точно і повністю визначити в результаті експериментів. Строго кажучи, вимірювання над системою порушує її замкненість, тому в дійсності в класичній фізиці вимірювання зводиться до спостереження. Це означає, що спостереження не робить ніякого впливу на досліджувану систему. В більш складних випадках вважається, що якщо такий вплив присутній, то його дія може строго контролюватися і враховуватися при обробці експерименту. Якраз з цим принципом зв'язані часто уявні експерименти, що використовуються в класичній фізиці. На рис. 7.2 приведена схема уявного експерименту Галілея.

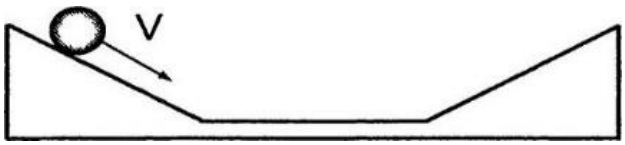


Рис. 7.2. Схема уявного експерименту Галілея

Приклад 7.2. *Найбільш ранній уявний експеримент у фізиці провів Галілео Галілей, встановивши, що тіла, не схильні до дії сили, зберігають вічно стан руху з постійною швидкістю. Слід нагадати, що у той час це суперечило канонізованому вченню Арістотеля і спростування цього вчення могло коштувати життя. Згідно Арістотелю, тіло рухається з постійною швидкістю, якщо на нього діє постійна сила. Відмітимо, що натурні експерименти, які проводилися, добре підтверджували саме Арістотеля. Для того, щоб установити новий принцип відносності, Галілей не збирався проводити такий експеримент натурно. Для цього не хватило б йому часу. Да і релізувати відсутність дії сил на тіло також було виключно складно і навіть неможливо. Тому Галілей звернувся до уявного експерименту. В цей час він вже знав, що куля, що скачується з похилої Дошки, збільшує свою швидкість, а коли закочується на неї, швидкість зменшується. Далі Галілей міркував таким чином. Давайте представимо собі установку, як на рис. 7.2. Зрозуміло, що при скачуванні кулі швидкість зростає, а при закочуванні вверх по похилій площині швидкість зменшується, тоді як при руху по горизонтальній ділянці повинна зберігатися. По безперервності між збільшенням і зменшенням є тільки незмінність. А, отже …*

Помітимо, що роль експериментатора в цьому експерименті зводиться до пасивного спостереження за подіями без жодного впливу на них. Звичайно, все не так просто. При спостереженні все-таки вплив на систему є присутнім, а отже, система стає незамкнутою. Можна спробувати повернути їй статус замкнутості, включаючи спостерігача в цю систему. Зрозуміло, слід зробити декілька

роздумів про статус спостерігача, наприклад, про його віддільність і тому подібне. Проте, все одно залишається можливість здійснення уявних експериментів над такою розширеною системою. Іншими словами, залишається можливість спостереження за спостерігачем. Залишаючи ці тонкощі, відмітимо, що в цьому принципі домінує інтуїція, яка базується на спостереженнях за макроскопічним світом, в якому вплив спостереження на макроскопічні об'єкти може бути малим. Зі зменшенням розмірів досліджуваних об'єктів зовсім не очевидно, що вимірюванню можна надати статус спостереження. Окрім скільки завгодно точного вимірювання - спостереження, допустимого в класичній фізиці, слід підкреслити ще раз властивість повноти. Це означає, що вимірювання дозволяє встановити або виміряти усі характеристики замкнутої системи. Ясно, що це не означає існування приладів для вимірювання усіх мислимих характеристик класичних систем. Це не означає і відсутність величезних труднощів при створенні вимірювальної апаратури. Просто існує принципова можливість повного вивчення характеристик класичної системи.

Нарешті, існує принцип *лапласівського детермінізму*, який означає, що еволюція системи повністю і однозначно визначається початковим станом класичної системи. Існує принцип фаталізму класичної фізики. Останній принцип навіть в класичному сенсі погано відповідає поведінці Світу, що оточує нас. Для узгодження його раніше притягувалася "слабка" незамкнутість систем. Вважалося, що випадкова поведінка пояснюється слабким неврахованим впливом оточення. Таке пояснення не зовсім задовільне. Як бути з усім Всесвітом, як замкнутою системою? Цікаво відмітити, що питання, чи є Всесвіт замкнутою системою, не є тривіальним.

7.2.2 Протиріччя класичної механіки

Для того, щоб відмовитися від досягнень класичної фізики і навіть не відмовитися, а взятися за їх перебудову, для фізиків були потрібні виключно вагомі підстави. Таким першим сигналом послужила робота Планка по дослідженню випромінювання чорного тіла. Чорним тілом природно назвати тіло, від якого не відбивається світло. В якості чорного тіла можна представити порожнину, наповнену електромагнітними хвилями, що знаходяться в рівновазі із стінками порожнини. Випромінювання, що виходить через отвір в порожнині, можна вважати випромінюванням чорного тіла. Експериментатально закон випромінювання чорного тіла до цього часу був добре відомий. Проте, теоретичний висновок цього закону, виходячи із законів класичної фізики, представляв проблему, що не піддається вирішенню. Макс Планк, займаючись цією проблемою, після багатьох безуспішних спроб вирішив її, використовуючи на ті часи незвичайну гіпотезу. Він припустив, що світло або електромагнітна хвиля є ансамблем корпускул або часток, енергія яких визначається частотою хвилі

$$E = h\omega. \qquad (7.1)$$

Постійна $h=\frac{h}{2\pi}$ пов'язана з фундаментальною постійною h, яка зараз називається постійна Планка. Квант дії $h = 6.626 \cdot 10^{-27}$ерг-сек введений Планком в 1900 р. Ці елементарні порції електромагнітної хвилі були названі фотонами. У певному значенні цю дату і можна прийняти за дату народження квантової фізики. Цікаво відмітити, що сам Планк, будучи швидше консерватором, ніж новатором, залишався незадоволеним цим виведенням. Він опублікував свої результати тільки після численних безуспішних спроб отримати цей закон по іншому.

Приклад 7.3. *У 1922 році Комптон виявив цікавий і на ті часи абсолютно незрозумілий ефект. Розсіюючи рентгенівські промені на парафіні, він спостерігав збільшення довжини хвилі рентгенівського випромінювання. У той час, виходячи з класичних уявлень про електромагнітні хвилі, вважалося, що при розсіянні довжина хвилі не повинна мінятися. Теорія цього явища була запропонована Комптоном і незалежно Дебаєм (рис. 7.3).*

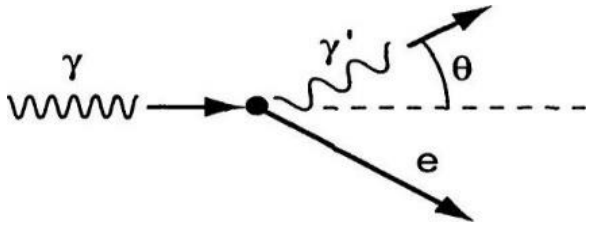


Рис. 7.3. Символічно показаний процес комптонівського розсіяння рентгенівського випромінювання на електроні, що покоїться

Основна ідея, яка дозволила їм пояснити це явище, це представлення рентгенівських променів у вигляді фотонів Планка або корпускулярна точка зору. Так, енергія фотона $E = h\omega$, *а його імпульс* $\vec{p} = h\,\vec{k}$. *Тут* $\vec{k}$ *– хвильовий вектор. Дисперсійне співвідношення визначає зв'язок частоти з хвильовим вектором* $\omega = ck$ (*c - швидкість світла).*

Для того, щоб пояснити ефект Комптона, використовуємо закон збереження енергії і імпульсу при розсіянні фотона на електроні речовини.

$$\varepsilon + E = \varepsilon' + E',$$

$$h\,\vec{k} + \vec{p} = h\,\vec{k}' + \vec{p}'.$$

Тут ε, $\vec{k}$ *і* ε', $\vec{k}'$ *енергія і хвильові вектори фотона до і після зіткнення. Відповідно, E,* $\vec{p}$ *і* $E', \vec{p}'$ *- енергія і імпульси електрона до і після зіткнення. Для простоти припустимо, що до зіткнення електрон покоївся, тобто,* $E = mc^2$ *і* $\vec{p} = 0$ *(див. Рис. 7.3). Тоді отримана вище система рівнянь спрощується*

$$\varepsilon + mc^2 = \varepsilon' + E',$$

$$h\,\vec{k} = h\,\vec{k}' + \vec{p}'.$$

Тепер врахуємо, що енергія електрона визначається співвідношенням

$$E = \sqrt{m^2c^4 + c^2p^2}\ .$$

Використовуючи зв'язок енергії електрона з його імпульсом і дисперсійне співвідношення у вигляді $\vec{k} = \frac{\omega}{c}\vec{n}$ *(*$\vec{n}$*- одиничний вектор у напрямі поширення фотона), виключимо характеристики електрона.*

Після простих перетворень отримаємо

$$\omega' = \frac{\omega}{1+\frac{\hbar\omega}{mc^2}(1-cos(\theta))}. \qquad (7.2)$$

Тут θ *- кут між напрямами поширення фотона до і після зіткнення з електроном. Переходячи від частот до довжин хвиль* $\lambda = \frac{2\pi c}{\omega}$ *, можна перетворити співвідношення (7.2)*

$$\lambda' - \lambda = 4\pi\lambda_c sin^2\left(\frac{\theta}{2}\right), \qquad (7.3)$$

де λ_c*= h/mc називається комптонівською довжиною хвилі і для електрона* $\lambda_c \approx 2.426 \cdot 10^{-10}$ *см. Ця формула показує, наскільки збільшується довжина хвилі. Для рентгенівських хвиль збільшення досить помітне. Легко оцінити, що відносне збільшення для рентгенівського діапазону довжин хвиль* $\lambda \simeq 10^{-9}$*см*

$$\frac{\Delta\lambda}{\lambda} \lesssim 0.1.$$

Це складає помітну величину близько 10%. Зрозуміло, для видимого діапазону світла таке збільшення незначне і його важко виявити.

У наданні електромагнітній хвилі статусу частки містився зародок корпускулярно - хвильового дуалізму квантових об'єктів. Наступним кроком послужило поширення корпускулярно – хвильового дуалізму на матеріальні частки. У 1923 році Луї де Бройль висловив чудову ідею, що частки володіють хвильовими властивостями, як фотони володіють властивостями часток. Іншими словами, він увів хвилі матерії. З кожною часткою, по де Бройлю, пов'язана хвиля з довжиною хвилі λ

$$\lambda = \frac{\mathrm{h}}{\mathrm{p}}, \qquad (7.4)$$

де h – постійна Планка, а p - імпульс частки. Інша форма цього співвідношення зв'язує імпульс з хвильовим вектором хвилі $k = 2\pi/\lambda$

$$\vec{p} = h\,\vec{k}. \qquad (7.5)$$

Енергія частки пов'язана з частотою цієї хвилі згідно формули Планка (7.1). Зрозуміло, ця хвиля не пов'язана з наявністю деякого поля як електромагнітна хвиля. Як з'ясувалося пізніше, їй можна надати сенс імовірнісної хвилі. Використовуючи співвідношення (7.4), можна отримати багато результатів квантової фізики на простому якісному рівні. Чудова здогадка Луї де Бройля була підтверджена експериментально в 1927 році. Девіссон і Джермер поставили дослід по розсіянню електронів на кристалах нікелю. Міняючи прискорюючий потенціал $\boldsymbol{U}$, вони вивчили залежність інтенсивності відображених електронів

від нього і від кута відображення. В результаті вони спостерігали типову картину дифракційних мінімумів і максимумів. Дуже схожа дифракційна картина спостерігалася при розсіянні ренгенівських променів на кристалах. За цими даними вдалося підтвердити і залежність (7.4) довжини хвилі електрона від імпульсу. З того часу корпускулярно - хвильовий дуалізм набув офіційного статусу.

Одна з глобальних проблем, що виникли в класичній фізиці на тлі її значних успіхів в поясненні багатьох явищ природи, стала планетарна модель атома. Цю модель запропонував чудовий фізик Резерфорд, виходячи з експериментальних даних, в 1911 році. Виглядає його модель приблизно як планетарна система (див. Рис. 7.4). У центрі її розміщується компактне позитивно заряджене ядро, а електрони обертаються по еліптичних орбітах навколо нього. На перший погляд така модель виглядає дуже привабливо. Дійсно, легко встановити основні характеристики атома водню. Ясно, що електрон обертається по орбіті, на якій сила кулонівського тяжіння до ядра урівноважується відцентровою силою. Проте, з класичної електродинаміки було добре відомо, що такий прискорений рух електрона повинен призводити до електромагнітного випромінювання. Це означає, що електрон повинен втрачати свою кінетичну енергію і радіус обертання його повинен зменшуватися. Іншими словами, через деякий час електрон повинен впасти на ядро і атом припинить своє існування. Проста оцінка часу життя атома давала $r \approx 10^{-10}$ сек. Ясно, що такий час життя класичного атома не відповідає дійсності і суперечить їх чудовій стійкості. Тому класична теорія не могла пояснити стійкість атомів і знадобилася радикальна перебудова основ фізики.

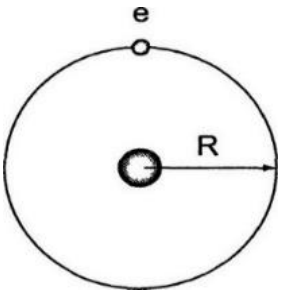


Рис. 7.4. Планетарна модель атома Резерфорда. У центрі позитивно заряджене ядро, навколо якого обертаються електрони. Зображений простий атом водню. Пропорції на цьому рисунку не дотримані, інакше ні електрона, ні ядра побачити б не вдалося. Власне атом, як не парадоксально, складається з порожнечі

7.2.3 Стан в квантовій механіці

Стан фізичної системи в квантовій механіці описується комплексним вектором (хвильовою функцією) в гільбертовому просторі. Елементи цього простору, слідуючи Діраку, позначатимемо як $|\psi\rangle$ ket і $\langle\varphi|$ bra вектори. Строго кажучи, ket вектори мають абсолютно іншу природу, чим bra вектори. Наприклад, складати ket і bra вектори безглуздо. Bra вектор є дуальним вектором

до відповідного ket вектора. Це означає, що на ket векторах можна визначити лінійні функції, коефіцієнти цих лінійних функцій і можна розглядати як дуальні до ket векторів bra вектори.

Приклад 7.4. *Мнемонічне правило, що дозволяє запам'ятати ці назви, досить цікаво. Воно пов'язане з введенням цих понять і зайвий раз демонструє глибоку продуманість навіть дрібних деталей квантової механіки. Походження цих назв пов'язане з англійським словом* ***bracket****, яке означає дужки. Скористаємося тепер і дужками, і словом* ***(bracket)****, що їх позначає. Розділимо отриману конструкцію на частини* $\langle \boldsymbol{bra}|\boldsymbol{c}|\boldsymbol{ket}\rangle$. *Це ділення виділяє дві компоненти* $\langle \boldsymbol{bra}|$ ***i*** $|\boldsymbol{ket}\rangle$, *що відповідають* ***bra i ket*** *векторам. Далі можна виявити, що таке правило має ще глибший сенс.*

У гільбертовому просторі визначений скалярний добуток векторів $\langle\varphi|\psi\rangle$. Цей скалярний добуток набуває комплексного значення і задовольняє умовам:

- $\langle\varphi|\varphi\rangle > 0$,
- $\langle\varphi|\,(a|\psi\rangle + \beta|\phi\rangle) = a\langle\varphi|\psi\rangle + \beta\,\langle\varphi|\phi\rangle$,
- $\langle\varphi|\psi\rangle^* = \langle\psi|\varphi\rangle$.

Тут a і β довільні комплексні числа. Перша властивість означає позитивність норми вектору, друга - лінійність скалярного добутку і третя визначає дію комплексного зв’язування, яке можна розуміти як $|\psi\rangle^* = \langle\psi|$. Звернемо увагу на це співвідношення. З нього виходить досить незвичайний висновок. Як відзначалося вище, ket вектор набуває комплексних значень. Проте, представлення його у вигляді реальної і чисто уявної частини немає фізичного сенсу. Причина цього в тому, що, наприклад, реальна частина є сума ket вектора і комплексно зв'язаного. Але комплексно зв'язаний до нього bra вектор. Іншими словами, величина абсолютно іншої природи. Складати bra і ket вектори не можна.

Природно, що наявність скалярного добутку дозволяє визначити властивість ортогональності векторів як $\langle\varphi|\psi\rangle = 0$. Стан фізичної системи описується векторами одиничної довжини або норми $\langle\varphi|\varphi\rangle = 1$. Це означає, що стани $|\varphi\rangle$ і $e^{i\theta}\,|\varphi\rangle$ визначають один і той же стан.

Важливою властивістю квантових систем є принцип суперпозиції квантових станів. З точки зору Фейнмана цей принцип є "серце квантової механіки” [5]. Так, якщо квантова система має стани $|\varphi\rangle$ і $|\psi\rangle$, то вона може знаходитися і в станах

$$|\phi\rangle = a|\varphi\rangle + \beta|\psi\rangle. \qquad (7.6)$$

Враховуючи обмеження на норму станів, довільні комплексні коефіцієнти a *і* β задовольняють умові

$$|a|^2 + |\beta|^2 = 1.$$

Додатково залучається імовірнісна інтерпретація таких станів. Вважається, що $|a|^2$ співпадає з вірогідністю виявити стан $|\varphi\rangle$, $|\beta|^2$ - з вірогідністю виявлення стану $|\psi\rangle$. Величини a *і* β часто називаються амплітудами станів $|\varphi\rangle$ *і* $|\psi\rangle$

відповідно. Необхідно підкреслити істотні відмінності принципу суперпозиції квантової механіки і знайомого принципу суперпозиції, скажімо, лінійних хвиль. Передусім нагадаємо, що звичайні хвилі переносять імпульс і енергію. Хвильова функція не є звичайним полем і не переносить ні імпульсу, ні енергії. Її можна розглядати тільки як переносника інформації і тому вважати інформаційним полем. Окрім цього, суперпозиція хвильових функцій не призводить до збільшення амплітуди, як у випадку звичайних хвиль. Причину цього легко побачити в збереженні норми хвильових функцій. Забігаючи вперед, відмітимо ще одну важливу відмінність квантової суперпозиції. Розпочнемо з обговорення окремих станів системи. Припустимо, що якщо система знаходиться в стані $|\varphi\rangle$, то вимірювання її стану дає значення a. Аналогічно, якщо вона знаходиться тільки в стані $|\psi\rangle$, то результат вимірювання дає β. Тепер нехай система може знаходитися в стані (7.6) суперпозиції цих двох станів. Виникає запитання, яким буде результат вимірювання такого стану? Грунтуючись на класичній інтуїції, ми могли б припустити, що спостерігатиметься значення щось середнє між a і β. І, звичайно, помилилися б. У квантовій фізиці спостереження дасть тільки а або β з вірогідністю їх спостереження, пов'язаною з a і β. Ніяких інших значень спостережень не виникне.

Важливі особливості квантової фізики вже можна продемонструвати навіть на цій стадії.

Приклад 7.5. *Ще Шредінгер [6] для демонстрації цих особливостей розглянув уявний експеримент, що дістав потім назву Шредінгерівського кота. Незвичність квантових станів (7.6) можна виявити в уявному експерименті, слідуючи Шредінгеру. Нехай в закритому ящику знаходиться живий кіт і пристрій, що складається з лічильника Гейгера, ампули з отрутою і механізму, що руйнує ампулу при спрацьовуванні лічильника Гейгера. Тоді при розпаді одного атома спрацює лічильник Гейгера і зруйнується оболонка ампули, в результаті кіт загине під дією отрути. Тепер згадаємо, що стан радіоактивного атома є суперпозиція тих станів, що розпався і не розпався. Тоді ясно, що стан Шредінгерівського кота буде суперпозицією живого $|Live\rangle$ і мертвого $|Dead\rangle$ кота*

$$|\psi\rangle = a|Live\rangle + \beta|Dead\rangle.$$

Звичайно, для мікроскопічної квантової системи це виглядає не парадоксалъно, але для макроскопічної системи "кіт" такий стан не лише парадоксальний, але і не спостерігається. Дійсно, реалізується стан живого і мертвого кота одночасно. Зрозуміло, для руйнування такого стану існує певна причина, пов'язана з декогеренцією станів. Явище декогеренції обговоримо нижче.

Поняття фазового простору класичної системи зазнає в квантовій фізиці радикальні зміни. Так, фазовий простір квантової системи - це безліч променів в лінійному комплексному гільбертовому просторі із скалярним добутком. Такий погляд на фазовий простір і стан квантової системи містить досить велику інформацію. Лінійність означає, що для станів справедливий принцип суперпозиції. Задавання стану у вигляді вектора, що направляє промінь, означає,

що він визначає стан з точністю до перемноження на комплексне число. Якщо навіть в якості такого вектора вибрати вектор одиничної норми, то він визначає стан з точністю до множення на $\varepsilon^{i\theta}$. Це означає, що фаза θ квантових станів неспостережувана. Більш складними властивостями володіють стани квантової системи, що складається з декількох, наприклад, двох невзаємодіючих підсистем. Нехай кожна з цих підсистем може знаходитися у базисних станах $|0\rangle$ і $|1\rangle$. Тоді система може знаходитися в станах:

$$|0\rangle|1\rangle \equiv |0,1\rangle,$$

а також в станах $|0,0\rangle, |1,0\rangle$ і $|1,1\rangle$. За домовленістю, вважатимемо, що на першому місці відмічений стан "першої", а на другому -"другої" підсистеми. Це дозволить уникнути позначень типу $|0\rangle_1|1\rangle_2$, де нижній індекс вказує на відповідну підсистему. Стани системи, які можна представити у вигляді добутку станів підсистем, називаються чисті стани. Проте, система, згідно з принципом суперпозиції, може знаходитися і в станах типу

$$|\psi\rangle = a_{00}|0,0\rangle + a_{01}|0,1\rangle + a_{10}|1,0\rangle + a_{11}|1,1\rangle.$$

Такі стани називаються переплутаними станами, оскільки стан не є добутком хвильових функцій підсистем. Саме із переплутаними станами пов'язаний ряд специфічних рис квантової фізики. Зокрема, переплутані стани

$$\begin{aligned}
|\Phi_+\rangle &= \frac{1}{\sqrt{2}}\,|0,0\rangle + \frac{1}{\sqrt{2}}\,|1,1\rangle,\\
|\Phi_-\rangle &= \frac{1}{\sqrt{2}}\,|0,0\rangle - \frac{1}{\sqrt{2}}\,|1,1\rangle,\\
|\psi_+\rangle &= \frac{1}{\sqrt{2}}\,|0,1\rangle + \frac{1}{\sqrt{2}}\,|1,0\rangle,\\
|\psi_-\rangle &= \tfrac{1}{\sqrt{2}}\,|0,1\rangle - \tfrac{1}{\sqrt{2}}\,|1,0\rangle
\end{aligned} \qquad (7.7)$$

називаються базисними станами Белла. Ці стани володіють сильною кореляцією. Дійсно, якщо система знаходиться, наприклад, в стані $|\psi_+\rangle$, то стани підсистем протилежні і рівноймовірні. Помітимо, що тут можна виявити також незвичайну властивість – повне знання про систему в квантовій фізиці не тягне повне знання про підсистеми. Повертаючись до стану $|\psi_+\rangle$, легко зрозуміти, що якщо при вимірюванні виявлена перша підсистема в стані $|0\rangle$, то друга підсистема виявиться в стані $|1\rangle$ з вірогідністю 1. Саме ці стани послужили основою для "парадоксу" Ейнштейна, Подільського, Розена. Нагадаємо, що в класичній фізиці зв'язки або кореляції між частинами системи виникають тільки завдяки взаємодії або обміну енергією між ними (див. Рис. 7.5).

Як було відмічено вище, в квантовій фізиці кореляції можуть виникати і по інших механізмах (див. Рис. 7.6). Детальніше парадокс Ейнштейна – Подільського - Розена обговоримо нижче після обговорення основних елементів квантової механіки.

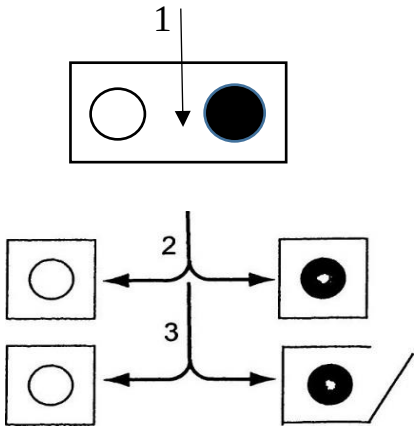


Рис. 7.5. Класичні кореляції.

У верхньому ящику знаходяться чорна і біла кулі. На першому кроці вставляємо перегородку і розділяємо ящик на два. Потім на другій стадії видаляємо їх один від одного на дуже велику відстань. На третьому етапі заглядаємо в правий ящик і по спостереженню кольору кулі однозначно визначаємо, яка куля в другому ящику

Приклад 7.6. *Із станами невзаємодіючих систем тісно пов'язано питання про стани тотожних часток. Для розуміння цього обговоримо властивості тотожних часток. Нагадаємо, що частки в квантовій фізиці володіють спіном. Тому вектор стану однієї частки залежить від координат* (x_1, x_2, x_3) *і спінової змінної* σ*, тобто, від чотирьох узагальнених координат* $\vec{x} = (x_1, x_2, x_3, \sigma)$*. При цьому спінова координата приймає ряд дискретних значень. Якщо спін частки* s*, то спінова змінна міняється від* $-s$ *до* s *і приймає, відповідно,* $2s+1$ *значень:* $-s, 1-s, \ldots, 0, 1, \ldots, s$*. Розпочнемо з розгляду простої системи двох незалежно діючих часток. Вектор стану* $|\psi\rangle$ *або хвильова функція залежатимуть від двох наборів узагальнених координат, що відповідають першій* $\vec{x}_1$ *і другій частці* $\vec{x}_2$*. Іншими словами,* $|\psi\rangle = |\vec{x}_1, \vec{x}_2\rangle$*. В силу тотожності часток їх перестановка і, відповідно, перестановка аргументів вектора станів не повинні міняти стан системи. Отже, ці стани можуть відрізнятися тільки множником* $\boldsymbol{a}$

$$|\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2\rangle = a\,|\vec{\mathrm{x}}_2, \vec{\mathrm{x}}_1\rangle.$$

Проробимо перестановку аргументів ще раз і отримаємо

$$|\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2\rangle = a^2\,|\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2\rangle.$$

Ясно, що $a^2 = 1$*. Тому існує тільки дві можливості* $a = \pm 1$*. Це означає, що перестановка часток може давати при* $a=+1$ ***симетричний відносно перестановок вектор стану, а при*** $a = -1$

$$|\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2\rangle = |\vec{\mathrm{x}}_2, \vec{\mathrm{x}}_1\rangle$$

антисиметричний вектор стану

$$|\vec{\mathrm{x}}_1, \vec{\mathrm{x}}_2\rangle = -|\vec{\mathrm{x}}_2, \vec{\mathrm{x}}_1\rangle\,. \qquad (7.8)$$

Властивість симетрії при перестановках часток зберігається з часом. Звідси можна зробити висновок, що вона є властивістю, яка визначається тільки природою часток. Саме величина спіна частки визначає властивість симетрії вектору станів системи тотожних часток. Так, якщо спін частки цілий, то вектор станів системи тотожних часток симетричний відносно їх перестановки. Якщо спін частки напівцілий, то вектор станів системи тотожних часток антисиметричний відносно їх попарної перестановки. Частки з цілим спіном називаються бозонами, а з напівцілим – ферміонами. Так, електрони, позитрони, протони і нейтрони – це ферміони, а фотони і мезони – це бозони. Для ферміонів виконується принцип Паулі, який стверджує, що два ферміони не можуть знаходитися в однакових станах. Дійсно, повертаючись до співвідношення (7.8), яке визначає симетрію вектора станів двох ферміонів, припустимо, що вони знаходяться в однаковому стані з квантовими числами $\vec{x}_1 = \vec{x}_2 = \vec{x}$. *Тоді*

$$|\vec{x}, \vec{x}\rangle = -|\vec{x}, \vec{x}\rangle.$$

Звідки виходить, що

$$|\vec{x}, \vec{x}\rangle = 0.$$

Проте, 0 не є вектором стану квантової системи і, отже, такий стан не існує. Іншими словами, два ферміони не можуть знаходитися в однаковому стані. Принцип Паулі виключно важливий для опису властивостей ансамблів ферміонів. Наприклад, для опису властивостей атомів або твердих тіл.

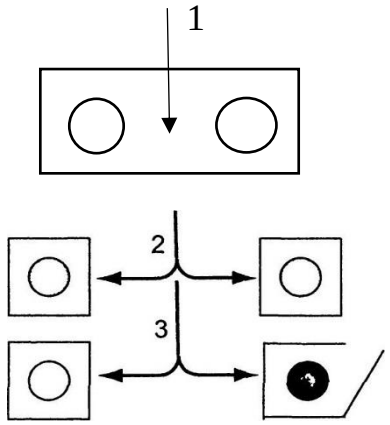


Рис. 7.6. Відмінність квантового випадку в тому, що до вимірювання у будь - якому ящику реалізується стан суперпозиції чорної і білої куль. Розділення кольору відбувається тільки у момент вимірювання кольору кулі

7.2.4 Спостережуваність

Спостережуваними величинами в класичній механіці були координати і імпульси, а також функції цих величин. У квантовій механіці поняття спостережуваних зазнає радикальних змін. Спостережувані величини в квантовій механіці відповідають лінійним ермітовим операторам. Ці оператори, діючи на хвильову функцію, ставлять у відповідність одному вектору гільбертового простору інший вектор

$$A|\psi\rangle \rightarrow |\psi'\rangle.$$

Лінійність цього оператора означає, що

$$A(a|\psi\rangle + \beta\,|\varphi\rangle) = aA|\psi\rangle + \beta A|\varphi\rangle\,.$$

Зв'язаний до A оператор A^+ визначається рівністю

$$\langle\varphi|A|\psi\rangle^* = \langle\psi|A^+|\varphi\rangle.$$

Оператор A називають самозв’язаним або ермітовим, якщо

$$A = A^+.$$

Оператори A і B комутують один з одним, якщо $[A, B] = - BA = 0$. У разі загального положення оператори не комутативні. Некомутативність спостережуваності квантової механіки, що виявлена Гейзенбергом, свого часу настільки уразила його, що тільки підтримка Нільса Бора допомогла йому продовжити дослідження в цьому напрямку. Причина цього пов'язана із спостереженням чисел в класичній фізиці і, звичайно, добуток чисел завжди комутативний, наприклад, 2 х 5 = 5 х 2 = 10. Некомутативність спостережуваності відразу означала, що спостережувані оператори в квантовій фізиці мають складнішу природу, ніж числа. Це досить дивно, оскільки, вимірюючи деяку динамічну величину, ми вимірюємо врешті-решт деяке число. Для того, щоб зрозуміти, як це можливо в квантовій механіці, звернемося до понять власних значень і власних векторів. Для цього розглянемо рівняння

$$A\,|\varphi_\lambda\rangle = \lambda|\varphi_\lambda\rangle\,.$$

Тут A деякий лінійний оператор, λ - число. При вивченні цього рівняння, зазвичай, відомий тільки оператор A. Число λ , як і вектор $|\varphi_\lambda\rangle$, не відомі. Сенс цього рівняння в тому, що дія оператора на власний вектор $|\varphi_\lambda\rangle$ виключно проста і зводиться до множення на відповідне власне число. Власне, це рівняння і служить для визначення власних векторів і власних чисел лінійних операторів. Зрозуміло, дія оператора A на довільний вектор не зводиться до множення на число. Таким чином, легко зрозуміти, що якщо система знаходиться в стані $|\varphi_\lambda\rangle$, то результатом вимірювання спостережуваності, що відповідає оператору A, може служити вже число λ. Таким чином, стани системи, що відповідають власним векторам, виключно важливі. Зрозуміло, в квантовій механіці основну роль відіграють лінійні оператори, власні значення яких дійсні. Тому далі основну увагу приділимо саме таким операторам. Легко переконатися, що такі оператори самозв’язані. Дійсно, розглянемо дію самозв’язаного оператора на власний вектор

$$A\,|\varphi_\lambda\rangle = \lambda|\varphi_\lambda\rangle.$$

Перемножимо це рівняння на відповідний bra вектор

$$\langle\varphi_\lambda|A|\varphi_\lambda\rangle = \lambda\langle\varphi_\lambda|\varphi_\lambda\rangle. \qquad (7.9)$$

Тепер виконаємо комплексне зв’язування і використаємо властивість добутку, що приведена вище, тоді

$$\langle\varphi_\lambda|A|\varphi_\lambda\rangle^* = \langle\varphi_\lambda|A^+|\varphi_\lambda\rangle = \langle\varphi_\lambda|A|\varphi_\lambda\rangle.$$

Легко довести, що права частина рівняння (7.9) дійсна, враховувуючи, що норма вектора $\langle\varphi_\lambda|\varphi_\lambda\rangle$ дійсна за визначенням з рівняння (7.9). Ясно, що власне число самозв'язаного оператора дійсне $\lambda^* = \lambda$. З цього результату виходить, що власні числа самозв'язаного оператора для власного bra вектора

$$\langle\varphi_\lambda|\, A = \lambda\, \langle\varphi_\lambda|$$

співпадають з власними числами відповідного власного ket вектора.

Далі будемо говорити про самозв'язані оператори. Зрозуміло, лінійні самозв'язані оператори можуть мати багато власних векторів і, відповідно, багато власних значень. Важливо довести, що власні вектори оператора з різними власними значеннями ортогональні. Для цього запишемо рівняння, які визначають такі власні вектори:

$$A|\varphi_{\lambda_1}\rangle = \lambda_1|\varphi_{\lambda_1}\rangle\,,$$
$$A|\varphi_{\lambda_2}\rangle = \lambda_2|\varphi_{\lambda_2}\rangle.$$

Перемножимо тепер перше рівняння на $\langle\varphi_{\lambda_2}|$, а друге на $\langle\varphi_{\lambda_1}|$, отримаємо:

$$\langle\varphi_{\lambda_2}|A|\varphi_{\lambda_1}\rangle = \lambda_1\langle\varphi_{\lambda_2}|\varphi_{\lambda_1}\rangle,$$
$$\langle\varphi_{\lambda_1}|A|\varphi_{\lambda_2}\rangle = \lambda_2\langle\varphi_{\lambda_1}|\varphi_{\lambda_2}\rangle.$$

Враховуючи дійсність правих частин

$$\langle\varphi_{\lambda_2}|A|\varphi_{\lambda_1}\rangle=\langle\varphi_{\lambda_1}|A|\varphi_{\lambda_2}\rangle\,,$$

віднімемо одне рівняння з іншого

$$(\lambda_1 - \lambda_2)\,\langle\varphi_{\lambda_1}|\varphi_{\lambda_2}\rangle = 0.$$

Так як за умовою $(\lambda_1 - \lambda_2) \neq 0$, тоді із цієї рівності виходить $\langle\varphi_{\lambda_1}|\varphi_{\lambda_2}\rangle$=0 є ортогональність власних векторів з різними власними значеннями.

Приклад 7.7. *Використовуючи корпускулярно-хвильовий дуалізм, спробуємо встановити вид такого важливого оператора, як оператор імпульсу. Нагадаємо, що згідно де Бройлю, з вільною часткою пов'язана хвиля. Тепер врахуємо, що усі лінійні хвилі виглядають однаково. Таким чином, з вільною часткою пов'язана лінійна хвиля, яку в термінах вектору станів або хвильової функції можна записати як*

$$|\varphi\rangle = a e^{i\,(\vec{k}\vec{x} - \omega t)}.$$

Використовуючи співвідношення (7.1), (7.5), запишемо її як

$$|\varphi\rangle = a\, e^{\frac{i}{h}\,(\vec{p}\vec{x} - Et)}. \qquad (7.10)$$

Тут a - амплітуда хвилі. Це співвідношення в точності співпадає з видом хвильової функції для вільної частки,отриманої з рішення рівняння Шредінгера. Більше того, використовуючи вид вектора станів вільної частки, легко встановити лінійне рівняння, якому вона задовольняє,

$$\mathrm{i}\,\frac{\partial|\varphi\rangle}{\partial \mathrm{t}} = -\frac{\mathrm{h}^2}{2\mathrm{m}}\,\triangle\,|\varphi\rangle.$$

Тут $\triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ *- оператор Лапласа. Це і є рівняння Шредінгера для вільної частки. Для нас цікаво, використовуючи вид вектора станів, встановити вид оператора імпульсу. Основну ідею, яку можна використати, це те, що спостережуваний імпульс є власне значення оператора імпульсу, тобто,*

$$\hat{p}\,|\varphi\rangle = p|\varphi\rangle.$$

Вільна частка володіє імпульсом p і легко зрозуміти, що оператор диференціювання по координаті діятиме як оператор імпульсу з точністю до числового коефіцієнта. Потрібний коефіцієнт легко підібрати і встановити вид оператора імпульсу

$$\hat{p} = \frac{h}{i}\frac{\partial}{\partial \vec{x}} \quad . \tag{7.11}$$

Зв'язок оператора імпульсу з диференціюванням по координаті має глибокий сенс. У класичній фізиці фундаментальні закони збереження пов'язані з певними симетріями. Так, закон збереження імпульсу пов'язаний з однорідністю простору. Це означає, що усі точки простору рівноправні. У квантовій фізиці це означає, що вектор станів вільної частки при зрушенні координат не міняється, точніше, змінює тільки фазу, яка не впливає на стан частки

$$|\varphi\rangle(\vec{\mathrm{x}} + \vec{\varepsilon}) = \mathrm{e}^{\mathrm{i}\theta}\,|\varphi\rangle(\vec{\mathrm{x}}).$$

Фаза $i\theta = i\frac{\vec{e}\vec{p}}{h}$. *Зрозуміло, що стан не залежить від цієї фази. Передбачаючи малим зміщення* $\vec{e}$*, запишемо розкладання в ряд Тейлора вектора станів*

$$|\varphi\rangle(\vec{x} + \vec{\varepsilon}) = |\varphi\rangle(\vec{x}\,) + \vec{\varepsilon}\,\frac{\partial}{\partial \vec{x}}\,|\varphi\rangle(\vec{x}\,) + \ldots$$

Це розкладання можна розглядати і як розкладання за фазою $i\theta$

$$|\varphi\rangle(\vec{x} + \vec{\varepsilon}) = |\varphi\rangle(\vec{x}\,) + \frac{i}{h}\,\vec{\varepsilon}\frac{h}{i}\,\frac{\partial}{\partial \vec{x}}\,|\varphi\rangle(\vec{x}\,) + \ldots$$

Порівнюючи отримане співвідношення з видом розкладання за фазою, також легко відновити вид оператора імпульсу (7.11). Іншими словами, оператор імпульсу є інфінітезимальним оператором зрушення уздовж класичної траєкторії. Окрім цього, використовуючи оператор імпульсу, можна побудувати і інший важливий оператор моменту імпульсу

$$\vec{M} = \vec{x} \times \vec{p}.$$

Тут знак $\times$ *означає векторний добуток відповідних векторів. Це співвідношення класичної фізики повинне зберігатися як співвідношення між операторами в квантовій фізиці. Тому оператор моменту імпульсу можна записати як*

$$M_x = yp_z - zp_y = y\hat{p}_z - z\hat{p}_y = y\frac{h}{i}\frac{\partial}{\partial z} - z\frac{h}{i}\frac{\partial}{\partial y},$$

$$M_y = zp_x - xp_z = z\hat{p}_x - x\hat{p}_z = z\frac{h}{i}\frac{\partial}{\partial x} - x\frac{h}{i}\frac{\partial}{\partial z},$$

$$M_z = xp_y - yp_x = x\hat{p}_y - y\hat{p}_x = x\frac{h}{i}\frac{\partial}{\partial y} - y\frac{h}{i}\frac{\partial}{\partial x}.$$

У класичній фізиці збереження моменту імпульсу частки пов'язане з ізотропністю простору. Тому вид цього оператора можна отримати як інфінітезимальний оператор повороту навколо вибраної осі, не опираючись на вигляд оператора імпульсу. Так давайте здійснимо поворот відносно осі на кут ϑ. Тоді залежність вектору станів від кута φ зміниться на $|\varphi\rangle(\varphi + \vartheta)$. Знову, передбачаючи поворот малим, отримаємо:

$$|\varphi\rangle(\varphi + \vartheta) = |\varphi\rangle(\varphi) + \vartheta\frac{\partial}{\partial\varphi}|\varphi\rangle(\varphi) + \dots .$$

Такий поворот в силу ізотропності простору не міняє стан вільної частки і, отже, еквівалентний перемноженню вектора станів на фазовий множник $e^{i\theta}$. Фаза θ в цьому випадку пропорційна моменту M_z, перемноженому на кут повороту ϑ. Звідси, як і для імпульсу, легко встановити, що

$$M_z = \frac{h}{i}\frac{\partial}{\partial\varphi}.$$

До цього виразу можна перейти і просто, виконуючи в $M_z = x\frac{h}{i}\frac{\partial}{\partial y} - y\frac{h}{i}\frac{\partial}{\partial x}$ перехід до циліндричних координат з $x = \rho\cos\varphi$ і $y = \rho\sin\varphi$. Таким чином, проєкція оператора моменту імпульсу $M_к$ на відповідну ось пропорційна інфінітезимальному оператору повороту відносно к – ої вісі.

Досі ми передбачали існування власних векторів і обговорювали їх властивості. Проте, слід торкнутися питання їх існування і як їх знаходити. Зрозуміло, в цьому розділі обмежимося швидше ілюстративними прикладами, а не загальними складними рецептами. В якості простого випадку розглянемо оператори, які самі задовольняють деяким алгебраїчним рівнянням. З прикладами таких операторів ми вже зустрічалися і раніше. До таких операторів відносився оператор проєктування, що задовольняє рівнянню $P^2 = P$. В більш загальному випадку оператор може задовольняти складнішому рівнянню виду

$$A^n + b_n A^{n-1} + b_{n-1} A^{n-2} + \dots + b_1 = 0.$$

Коефіцієнти b_i *з i =1,…, n,* що входять в це рівняння, є числами. Для таких операторів доведемо декілька важливих властивостей. Розпочнемо з кількості власних векторів і власних значень таких операторів. Для цього представимо ліву частину у вигляді:

$$A^n + b_n A^{n-1} + b_{n-1} A^{n-2} + \dots + b_1 = (A - c_1)(A - c_2)\dots(A - c_n).$$

Величини $c_1, c_2, \dots, c_n$ - деякі числа, серед яких можуть зустрічатися і однакові. Запис у вигляді добутку можливий з міркувань чисто алгебри. Усі величини, що входять в рівняння, комутують одна з іншою і в цьому сенсі не відрізняються від чисел. Для цього добутку зручно ввести позначення:

$$(A\text{-}\ c_1\ )(A-\ c_2\ )\ldots\ (A\text{-}\ c_n)\ =\ (A\text{-}\ c_i\ )\ B_i\ .$$

Тут B_i – частка від ділення добутку на $(A - c_i)$. Особливо слід підкреслити відсутність в цьому виразі підсумовування за значеннями індексу і. Правило Ейнштейна - підсумовування за індексами, що повторюються, до цього співвідношення не застосовується. Подіємо тепер на довільний вектор $|\psi\rangle$ цим операторним виразом:

$$(A\text{-}\ c_i\ )\ B_i\ |\psi\rangle=0.$$

Ясно, що $B_i\ |\psi\rangle \neq 0$ на довільному векторі. Інакше операторне рівняння для А було б більш низької степені. Позначаючи результат дії $B_i\ |\psi\rangle$ як $|\varphi_i\rangle$, отримаємо:

$$(A\text{-}\ c_i\ )\ |\varphi_i\rangle = 0.$$

Переписавши це рівняння в зручній для аналізу формі

$$A\ |\varphi_i\rangle\ =\ c_i\ |\varphi_i\rangle,$$

легко упізнати в ньому рівняння для власних векторів і власних значень. Ясно, що це виконується для $i=1\ ,\ldots,\ n$. Таким чином, кожне c_i є власним значенням оператора A. Іншими словами, оператор A має рівно n власних значень. Набір дискретних власних значень називається дискретним спектром оператора. Залишається переконатися, що усі власні значення різні. Припустимо, що це не так і існує $< n$ однакових власних значень. Тоді

$$(A\text{-}\ c_1\ )\ (A\text{-}\ c_2\ )\ldots\ (A\text{-}\ c_n)=\ (A-c)^k\ (A\text{-}\ c_{k+1}\ )\ (A\text{-}\ c_{k+2}\ )\ldots\ (A\text{-}\ c_n).$$

Тут c - чисельне значення однакових власних значень. Знову подіємо цим оператором на довільний *вектор* $|\psi\rangle$ і позначимо $|\phi\rangle\ =\ (A\ -\ -\ c_{k+1}\ )\ (A\text{-}\ c_{k+2}\ )\ldots\ (A\text{-}\ c_n)|\psi\rangle\ \neq 0$. Тоді

$$(A-c)^k|\phi\rangle = 0.$$

Проте, якщо c власне значення оператора А, тоді

$$(A-c)|\phi\rangle\ =\ 0.$$

Це означає, що рівняння для оператора A може бути зведено до рівняння нижчої степені $n - k + 1$, що суперечить визначенню оператора А рівнянням n-ої степені. Таким чином, усі n власних значень оператора A різні.

Тепер обговоримо повноту власних векторів оператора A. Під цим будемо розуміти, що будь-який вектор може бути представлений у вигляді суперпозиції власних векторів оператора A. Ясно, що це виключно важлива властивість. Нагадаємо, що раніше вводилося позначення $B_i\ (A)\ =(A\ -\ c_1\ )\ \ldots\ (A-\ c_{i-1}\ )(A\text{-}\ c_{i+1})\ \ \ldots(A\ -\ c_n)$, тобто, в добутку відсутній множник $(A\text{-}\ c_i)$. Цей добуток набуває ненульового значення, якщо оператор A замінити в ньому на власне значення $c_i(B_i\ (c_i)\ \neq\ 0)$. Нагадаємо, що усі власні значення різні. Для виявлення повноти розглянемо вираз:

$$\sum_i \frac{B_i\ (A)}{B_i\ (c_i)}\ .$$

При підстановці в цю суму замість оператора A будь-якого з власних значень $c_1, c_2, \ldots, c_n$, наприклад c_k, усі доданки з індексом $i \neq k$ перетворяться на нуль, а доданок з індексом $i = k$ дає одиницю.

Таким чином, на власних значеннях $c_1, c_2, \ldots, c_n$ виконується рівність

$$1 = \sum_i \frac{B_i(A)}{B_i(c_i)}.$$

Проте аналізований вище вираз є поліномом $n - 1$ степені від A і, отже, ця рівність виконується тотожно. Тоді, діючи цією рівністю на довільний вектор $|\psi\rangle$, отримаємо

$$|\psi\rangle = \sum_i \frac{B_i(A)|\psi\rangle}{B_i(c_i)} = \sum_i \frac{1}{B_i(c_i)} |\varphi_i\rangle. \quad (7.12)$$

Таким чином, довільний вектор $|\psi\rangle$ може бути представлений у вигляді суперпозиції власних векторів $|\varphi_i\rangle$. Це доводить, що власні вектори такого оператора утворюють повний базис, по якому можна розкласти довільний вектор станів.

Обговоримо тепер фізичні погляди на спостережувані величини в квантовій механіці. Якщо фізична система знаходиться в стані, співпадаючому з власним вектором спостережуваної A, то в результаті її вимірювання буде отримано дійсне число, співпадаюче з власним числом цього власного вектора. Виконується і обернене твердження. Якщо в результаті вимірювання отримуємо однозначно власне значення, то система знаходиться в стані, що відповідає власному вектору цієї спостережуваної величини. Природним розширенням цієї ідеології є перехід до станів квантової системи у вигляді суперпозиції власних векторів. Вважається, що в цьому випадку вимірювання дасть одне з власних чисел, що входять в суперпозицію власних векторів. Вірогідність того або іншого власного значення визначається амплітудами вірогідності суперпозиції. Тепер стає ясним фізичний сенс власних чисел. Це значення, які може приймати спостережувана величина. З цієї точки зору стає важливим, чи утворюють власні вектори повний набір, по якому може бути розкладений будь-який стан, чи ні. Якщо власні вектори не утворюють повний набір, то така величина не може бути виміряна, а, отже, немає підстав надавати їй статус спостережуваної. Таким чином, спостережуваними називатимемо тільки такі ермітові оператори, власні вектори яких утворюють повний набір.

Приклад 7.8. *Давайте обговоримо властивості атома водню, виходячи з квантової механіки. Основне питання пов'язане з енергіями атома і радіусами орбіт електрона. Для з'ясування цих характеристик використовуємо хвильові властивості електрона. Так, будемо допускати, що стаціонарні орбіти електрона в атомі відповідають стоячим хвилям. Отже, довжина хвилі електрона λ повинна укладатися ціле число разів уздовж його орбіти.*

$$\frac{2\pi R}{\lambda} = n,$$

де n - ціле число, R - радіус орбіти електрона. Враховуючи, що довжина хвилі де Бройля визначається співвідношенням (7.4), отримаємо

$$\frac{2\pi Rp}{h} = n.$$

Тепер виразимо імпульс електрона через його швидкість p = mv, вважаючи його нерелятивістським

$$\frac{mRv}{h} = n.$$

Зручно це співвідношення записати у вигляді

$$mRv = nh,$$

звідки відразу слідує важлива властивість квантових систем – квантування моменту імпульсу $\vec{M} = \vec{R} \times \vec{p}$ *. Враховуючи, що* $\vec{R}$ *ортогонально* $\vec{p}$*, для величини моменту імпульсу отримаємо*

$$M = nh. \qquad (7.13)$$

Поки що ми отримали тільки умову квантування. Ця умова означає, що момент імпульсу може приймати тільки ряд дискретних значень. Якщо розглянути можливі значення проєкцій моменту імпульсу на вибрану ось, то проєкції так само можуть приймати тільки ряд дискретних значень. При фіксованому значенні моменту імпульсу n (у одиницях h) їх число дорівнює 2n +1. Тепер перейдемо до з'ясування радіусів стаціонарних орбіт електрона в атомі водню. Для цього знову використовуємо баланс сил, що діють на електрон

$$\frac{mv^2}{R} = \frac{e^2}{R^2}.$$

Цю рівність сил запишемо як

$$\frac{m^2 v^2 R^2}{mR} = e^2.$$

Тепер використаємо правило квантування моменту імпульсу

$$\frac{(hn)^2}{mR} = e^2,$$

звідки отримуємо радіуси стаціонарних орбіт електронів

$$R_n = \frac{(hn)^2}{me^2}. \qquad (7.14)$$

Тут n = 1, 2,3, ... набуває цілих значень. Таким чином, електрони в атомі можуть знаходитися на дискретному наборі орбіт. Використовуючи це, отримаємо енергію атомів. Повна енергія - це сума кінетичної енергії електрона і його потенційної енергії

$$E = \frac{mv^2}{2} - \frac{e^2}{R} = \frac{e^2}{2R} - \frac{e^2}{R} = -\frac{e^2}{2R}.$$

Тут використана і умова рівності сил на стаціонарній орбіті. Залишається підставити в повну енергію знайдений раніше радіус орбіт

$$E_n = -\frac{me^4}{2(hn)^2} \qquad (7.15)$$

і отримати оцінку дискретних рівнів енергії атома. Цікаво відмітити, що отримані значення енергії в точності співпадають з рівнями енергії, які можуть бути отримані з розв'язків рівняння Шредінгера.

Таким чином, енергія атома може набувати тільки певних дискретних значень. Ці значення називаються спектром енергії або рівнями енергії атома. Для макроскопічних тіл така дискретність стає непомітною із-за крихти "відстаней" між рівнями енергії в порівнянні з енергіями мікроскопічних тіл. Ця формула зіграла значну роль в становленні квантової механіки. Дійсно, до появи квантової механіки було відомо, що випромінювання атомів відбувається на певних частотах. Зокрема, для водню ще в 1885 році Бальмер отримав, що в області видимого світла частоти підкоряються наступній закономірності

$$\nu = R\left(\frac{1}{4} - \frac{1}{n^2}\right),$$

де R - постійна, яка називається постійна Рідберга. Величину цієї постійної уперше встановив Пашен $\boldsymbol{R \approx 1.09678 \cdot 10^5}$ ***1/см*** *ще в 1916 році. Кожному цілому числу n ≥ 2 відповідала спектральна лінія. Ця серія частот згодом була названа серією Бальмера. Були відомі і інші серії частот. Наприклад, серія Лаймана в ультрафіолетовій області, серія Пашена і серія Брекетта в інфрачервоному діапазоні. Існування таких серій частот випромінювання атома не знаходило пояснень в класичній фізиці. Пояснення з точки зору квантової фізики встановив Бор. Він зрозумів, що енергія випромінюваного фотона повинна відповідати переходам електрона з однієї стаціонарної орбіти на іншу, а частоти випромінювання визначаються енергією фотона. Враховуючи зв'язок енергії фотона з частотою, легко отримати*

$$h\omega_{n\to j} = \frac{me^4}{2(h)^2}\left(\frac{1}{j^2} - \frac{1}{n^2}\right).$$

Тут $\omega_{n\to j}$ *- кутова частота фотона під час переходу електрона з n-го радіусу орбіти на j-й радіус. З цієї формули виходили усі відомі серії частот. Так, серія Лаймана відповідала j = 1, серія Бальмера j = 2, а серії Пашена і Брекетта j=3 і j=4. Окрім цього, легко бачити, що постійна Рідберга тепер виражається через фундаментальні постійні і її значення може бути легко обчислено. Це і було з успіхом виконано Бором. З цієї миті почався грандіозний розвиток атомної фізики.*

У випадку загального стану власні числа спостережуваної координати можуть складатися з дискретної послідовності чисел і з чисел, що займають цілий інтервал дійсної осі. У першому випадку говорять про дискретний спектр, у другому – про безперервний спектр оператора. Дискретна послідовність може бути кінцевою або нескінченною. В цьому випадку будь-який стан системи повинен виражатися через суму по дискретним власним векторам і інтегралу по безперервній компоненті власних векторів

$$\sum_i a_i \ |\varphi_i\rangle + \int a(v)\,|v\rangle\, dv. \qquad (7.16)$$

Тут v "номерує" безперервну компоненту.

Приклад 7.9. *Спостережуваною, власні числа якої можуть співпадати з будь - яким дійсним числом, може служити одновимірна координата квантової частки. Природно, що вимірювання положення в просторі може виявити її у будь-якому місці дійсної осі. Оператор координат співпадає з множенням на відповідну координату* $\hat{x} = x$, $\hat{y} = y$, $\hat{z} = z$ *в координатному представленні.*

У квантовій фізиці, окрім скалярного добутку ket і bra векторів, визначений ще один добуток між ними. Ця операція записується як $|\varphi\rangle\langle\varphi|$. Сенс цього виразу стане ясним, якщо ми визначимо його дію на довільний вектор. Передусім помітимо, що цю величину можна розглядати як лінійний оператор, дія якого переводить довільний вектор $|\psi\rangle$ у вектор $|\varphi\rangle$ згідно з наступним співвідношенням

$$|\psi\rangle \xrightarrow{|\varphi\rangle\langle\varphi|} |\varphi\rangle.$$

Дію цього оператора визначимо співвідношенням

$$|\varphi\rangle\langle\varphi|\,(\,|\psi\rangle\,) = |\varphi\rangle\langle\varphi|\psi\rangle.$$

Також легко довести, що це оператор проєктування. Дійсно, піднесемо цей оператор в квадрат:

$$(|\varphi\rangle\langle\varphi|\,)^2 = (|\varphi\rangle\langle\varphi|)(\,|\varphi\rangle\langle\varphi|) = |\varphi\rangle\langle\varphi|\varphi\rangle\langle\varphi|.$$

Враховуючи, що стан $|\varphi\rangle$ нормований на одиницю, тобто, $\langle\varphi|\varphi\rangle = 1$, отримуємо:

$$(|\varphi\rangle\langle\varphi|\,)^2 = |\varphi\rangle\langle\varphi|.$$

Це є операторне рівняння алгебри, що визначає оператор проєктування. Власні числа такого оператора легко встановити в сенсі так, як у попередніх розглядах складніших операторних рівнянь. Виявляється, що власні числа цього оператора 1 і 0. Одиниці відповідає власний вектор $|\varphi\rangle$, а нулю будь-який ортогональний до $|\varphi\rangle$ вектор. Вимірювання цієї спостережуваної фактично визначає, чи знаходиться ця спостережувана в стані $|\varphi\rangle$ або ні. На кінець цього розділу підкреслимо декілька важливих елементів квантових спостережуваних. Передусім, існують квантові спостережувані, такі, що не мають класичних аналогів.

Приклад 7.10. *Зокрема, такою спостережуваною є спін частки. Виникнення спіна в якості спостережуваної було зумовлене експериментальними фактами. Передусім, дослідами Штерна і Герлаха. Для того, щоб виконувався фундаментальний закон збереження моменту імпульсу, потрібно було увести власний момент часток, що покояться, який назвали спіном. Деякою теоретичною основою для його введення може служити релятивістська теорія електрона Дірака. Побудувавши таку теорію, Дірак виявив, що при заданій енергії для електрона, що покоїться, завжди існують два незалежні рішення. Природне пояснення полягало в тому, що ці рішення відрізняються значенням деякої внутрішньої степені свободи. Ця внутрішня степінь свободи і є спін електрона. У квантовій фізиці спін, як і момент імпульсу, квантується і його проєкція на деяку виділену ось може приймати*

тільки кінцеве число значень. Число цих значень визначається величиною спіна s і дорівнює 2s + 1. Тому з теорії Дірака виходить існування 2 проєкцій і, отже, спін електрона рівний $s = \frac{1}{2}$ *. Величина спіна приведена в одиницях h. Іноді в популярних викладках можна зустріти пояснення спіна як момента обертання його навколо власної осі. Власне, з таким первинним розумінням спіна і пов'язана його назва "spin", в перекладі з англійської означає обертання, кружляння. Проте, таке розуміння приводить до швидкості обертання електрона, що перевищує швидкість світла, що фізично безглуздо. Тому спін – чисто квантова спостережувана, що не має класичного аналогу.*

Зрозуміло, спін не єдина спостережувана, що не має класичного аналогу.

7.2.5 Еволюція

Однією з найважливіших абстракцій фізики є уявлення про ізольовані системи. Саме ця абстракція дозволяє розібрати складні фізичні явища на простіші складові. Наскільки далеко таку абстракцію можна використати - складне і далеко не тривіальне питання. Виділення частини Всесвіту в якості ізольованої системи і припущення про те, що її еволюція визначається тільки своїми внутрішніми фізичними законами, не завжди сприймалися як природне явище. Проте обрізання усіх зв'язків з іншими частинами Всесвіту або облік їх в середньому - ефективний прийом, що часто використовується у фізиці. Квантова механіка також використовує цю абстракцію. Цікаво відмітити, що ізольована система при послідовному розгляді ізольована і від зовнішнього спостерігача. У цьому сенсі вона не допускає зовнішніх вимірювань або дій. Слід зазначити, що в деяких модифікаціях квантової теорії не достатньо навіть включити спостерігача в таку фізичну систему, треба ще включати і його свідомість в ній. Це цілком перекликається з висловлюваннями як Шредінгера, так і Вігнера. У цьому розділі спробуємо розібратися з описом еволюції квантових систем. Розпочнемо із стаціонарних станів, які в гільбертовому просторі можна "зображувати" як безліч взаємоортогональних променів. Нехай тепер початковий стан утворений суперпозицією таких станів. Наприклад:

$$|\psi\rangle = \alpha|\varphi_1\rangle + \beta|\varphi_2\rangle.$$

У відповідності з принципом суперпозиції еволюція таких станів відбувається незалежно і у будь-який момент часу є суперпозицією станів з тими ж коефіцієнтами. Так, еволюція станів

$$|\varphi_1\rangle \xrightarrow{t} |\varphi_{1t}\rangle,$$
$$|\varphi_2\rangle \xrightarrow{t} |\varphi_{2t}\rangle$$

приводить до еволюції їх суперпозиції згідно

$$|\varphi_t\rangle \xrightarrow{t} \alpha\,|\varphi_{1t}\rangle + \beta|\varphi_{2t}\rangle.$$

Це означає, що початковий вектор стану системи переходить з часом у вектор стану, який лінійним чином залежить від початкового стану квантової

системи. Іншими словами, еволюція станів з часом відбувається під впливом лінійного оператора

$$|\varphi_t\rangle = U(t)\,|\varphi_0\rangle. \qquad (7.17)$$

Ясно, що лінійний оператор U не залежить від початкового стану системи. Тепер згадаємо, що в якості представника стану променя в гільбертовому просторі ми домовилися використати одиничний вектор, що задає напрям променя. Тому з фізичної точки зору можна вважати, що довжина вектора $|\varphi_t\rangle$ зберігається в процесі еволюції. Це означає, що оператор U задовольняє умові:

$$U^{+}U = 1.$$

Такі лінійні оператори прийнято називати унітарними. Припустимо тепер безперервність еволюції стану $|\varphi_t\rangle$ з часом. Тоді:

$$\frac{d}{dt}|\varphi_t\rangle = \left(\lim_{t\to 0}\frac{U(t)-1}{t}\right)|\varphi_0\rangle.$$

Отримуваний в ліміті лінійний оператор чисто уявний, тому уведемо для нього позначення $H = ih\left(\lim_{t\to 0}\frac{U(t)-1}{t}\right)$ і після очевидних перетворень отримаємо рівняння:

$$ih\frac{d}{dt}|\varphi_t\rangle = H(\mathrm{t})\,|\varphi_t\rangle. \qquad (7.18)$$

Отримане таким чином рівняння встановлює закон, згідно з яким еволюціонує стан будь-якої квантової системи. У це рівняння входить поки невідомий нам оператор H (t). Цей оператор і визначає властивості квантової системи. Згадаємо, що еволюція відповідної класичної системи визначалася повною енергією системи або гамільтоніаном. Тому, якщо використати фізичні міркування про відповідність між квантовими і класичними системами, можна очікувати, що в рівняння входить оператор повної енергії системи. Зрозуміло, з фізичної точки зору повна енергія є квантовою спостережуваною, з якою пов'язаний оператор H (t). Для замкнутої системи цей оператор не залежить від часу. Рівняння (7.18) відповідає шредінгерівському виду рівняння еволюції квантових систем. У шредінгерівському підході вектор стану еволюціонує з часом згідно рівняння (7.18). При такій еволюції вектор стану заміщає в гільбертовому просторі деякий "конус".

Встановимо зв'язок між оператором еволюції U і оператором H. Для цього підставимо в рівняння (7.18) вид вектора станів (7.17):

$$ih\frac{dU}{dt}|\varphi_0\rangle = H\,U|\varphi_0\rangle.$$

Враховуючи, що початковий вектор довільний, з цього рівняння виходить операторне рівняння:

$$ih\frac{dU}{dt} = HU. \qquad (7.19)$$

Обговоримо тепер інший підхід до опису еволюції квантових систем, який базується на гейзенбергівському вигляді рівнянь руху. Повернемося до рівняння (7.17) і запишемо з нього

$$|\varphi_0\rangle = U(t)^{-1}\,|\varphi_t\rangle.$$

Дивлячись на цю формулу, легко помітити, що це перетворення переводить залежний від часу вектор стану в незалежний від часу нерухомий вектор $|\varphi_0\rangle$. Використовуючи це перетворення, можна перенести залежність від часу з векторів станів на оператори фізичних спостережуваних

$$A' = U^{-1}AU.$$

Це означає, що якщо раніше фізичній спостережуваній відповідав оператор, що не залежить від часу, то в підході Гейзенберга залежними від часу стають динамічні спостережувані або такі лінійні оператори, що відповідають їм. Тому в картині Гейзенберга рівнянням руху підкоряються оператори або спостережувані. Отримаємо ці рівняння.

Позначимо залежний від часу оператор індексом *t*

$$A_t = U^{-1}(t)AU(\mathrm{t})$$

або

$$U(\mathrm{t})\,A_t = AU(\mathrm{t}).$$

Продиференціюємо це представлення оператора за часом

$$\frac{dU(t)}{dt}\,A_t + U(\mathrm{t})\,\frac{dA_t}{dt} = A\,\frac{dU(t)}{dt}.$$

Використаємо тепер рівняння (7.19), перетворимо його до вигляду

$$HUA_t + ih\,U(t)\,\frac{dA_t}{dt} = ihA\,\frac{dU(t)}{dt}.$$

Подіємо на це рівняння оператором $U^{-1}(t)$ і після використання позначення $H_t = U^{-1}HU$, отримаємо

$$ih\,\frac{dA_t}{dt} = A_t\,H_t - H_tA_t\,.$$

Вводячи позначення для комутатора двох операторів $[A_t\,H_t] = A_t\,H_t - H_tA_t$, запишемо це рівняння в остаточному виді:

$$ih\,\frac{dA_t}{dt} = [A_t\,H_t]. \qquad (7.20)$$

Це гейзенбергівське рівняння руху для спостережуваних в квантовій механіці. Чудовою властивістю цього рівняння, поміченою Діраком, є виняткова близькість до рівняння руху класичних спостережуваних. Дійсно, якщо згадати запис рівнянь руху класичної механіки для спостережуваних, функцій координат і імпульсів з використанням дужок Пуассона $\{A, B\} = \frac{\partial A}{\partial q_i}\,\frac{\partial B}{\partial p_i} - \frac{\partial A}{\partial p_i}\,\frac{\partial B}{\partial q_i}$

$$\frac{df}{dt} = \{f, H\},$$

то цей чудовий зв'язок стає очевидним. У квантовому випадку дужка Пуассона переходить в комутатор, спостережувані функції у відповідні оператори. Ця аналогія між рівняннями руху класичної механіки і квантової механіки переконує нас в правильності вибору в якості H повної енергії системи. У квантовій механіці динаміку системи визначає оператор Гамільтона H. При цьому абсолютно не важливо, існує класичний варіант або ні. У разі, коли класичний аналог існує, зв'язок між ними досить тісний. У цих випадках часто припускають, що залежність від канонічних спостережуваних в квантовому гамільтоніані така ж, як і в класичному. Це дозволяє отримувати квантовий гамільтоніан по виду класичної функції Гамільтона. Строго кажучи, тут є присутніми ряд тонкощів, що пов'язаних з некомутативністю квантових спостережуваних, що входять в класичний аналог у вигляді добутку. При цьому стає неясним і нетривіальним порядок розставляння операторів. Виходячи з рівняння (7.20), можна визначити квантові інтеграли руху. Дійсно, спостережувана A_t зберігається з часом, якщо комутує з гамільтоніаном H. Природно, гамільтоніан H комутує сам з собою і є інтегралом руху. Для ізольованої квантової системи, на яку не діють зовнішні сили, окрім енергії, інваріантами є імпульс і момент кількості руху.

Давайте тепер розглянемо квантову систему з постійною енергією. Тоді рівняння (7.18) можна легко проінтегрувати, враховуючи, що усі величини, які входять в нього, комутують одна з іншою

$$U = e^{-i\frac{Ht}{h}}.$$

Тут врахована початкова умова $U|_{t=0} = 1$. У квантовій механіці кожному стаціонарному стану відповідає власне значення енергії E_i. Тоді еволюція власних векторів, за якими можна розкласти довільний стан, відбувається згідно з рівнянням

$$|\varphi_i\rangle \overset{t}{\rightarrow} e^{\frac{E_i t}{ih}}|\varphi_i\rangle .$$

Згадаємо тепер, що $E_i t$ за розмірністю співпадає з дією S. Тому еволюцію початкового стану $|\varphi_0\rangle$ можна описати і більш загальним виразом без апеляції до власних чисел енергії

$$|\varphi_0\rangle \overset{t}{\rightarrow} e^{-i\frac{S(t)}{h}} |\varphi_0\rangle.$$

Цей вираз являється джерелом фейманівського підходу до формулювання квантової механіки [5].

7.2.6 Вимірювання

Глибоким принципом квантової фізики є принципова зміна ролі спостерігача. Якщо в класичній фізиці спостереження за поведінкою класичної системи передбачало можливість спостереження при повній відсутності дії на неї ідеальним спостерігачем, то в квантовій системі це неможливо. Навіть ідеальний спостерігач не може не впливати на неї при спостереженні. Роль спостерігача змінюється роллю вимірювальника. При цьому єдиним джерелом

об'єктивної інформації про мікросвіт служать виключно вимірювання. Наші органи чуття дають інформацію про макроскопічний світ і розвивають макроскопічну інтуїцію. Тому доводиться замінити їх на прилади, що вивчають мікросвіт. В принципі, навіть наше мислення і мова, як елементи макроскопічної культури, можуть виявитися непристосованими для квантової фізики.

З цієї причини в квантовій механіці поняття вимірювання зазнає ряд степенів ідеалізації і стає абстрактним поняттям. Квантове вимірювання є в сильній степені абстрагованим від приладів, що здійснюють які–небудь вимірювання. Слід зауважити, що поняття квантового вимірювання, строго кажучи, визначене неоднозначно. Не вдаючись до деталей обгрунтування і тонкощів цього поняття, перейдемо до прагматичного трактування вимірювань. Слідуючи фон Нейману, опишемо вимірювання, що проводиться над квантовою системою, яка знаходиться в стані

$$|\psi\rangle = a_1|\varphi_1\rangle + a_2\,|\varphi_2\rangle.$$

Проводячи вимірювання над системою, можемо отримати один з двох результатів вимірювання, що відповідають стану $|\varphi_1\rangle$ з вірогідністю $|a_1|^2$ або стану $|\varphi_2\rangle$ з вірогідністю $|a_2|^2$. Згідно прийнятого в квантовій механіці постулату, в результаті вимірювання система перейде в стан $|\varphi_1\rangle$, якщо результат вимірювання відповідає цьому стану. Якщо ні, то система перейде в стан $|\varphi_2\rangle$. Цей миттєвий перехід називають редукцією або колапсом хвильової функції. Такий миттєвий перехід або квантовий стрибок, зрозуміло, не описується рівнянням квантової еволюції. На цей процес редукції квантових станів існує декілька точок зору. Важливо помітити, що причина, по якій цей постулат виглядає досить неясним, в квантовій механіці пов'язана з прийняттям принципу ізольованості замкнутої вимірюваної системи. Саме цей принцип і приводить до необхідності прийняття цього постулату (див., наприклад [5]).

Обговоримо тепер детальніше вимірювання деякої спостережуваної *A*. Для простоти припускатимемо, що ця спостережувана має кінцевий спектр значень або кінцеве число власних чисел. Це означає, що вектор, що визначає довільний стан системи безпосередньо перед вимірюванням, розкладається по кінцевому набору власних векторів спостережуваної *A*:

$$|\psi\rangle = \sum_{i=1}^{n} a_i\ |\varphi_i\rangle. \qquad (7.21)$$

Тут n – число власних векторів. При вимірюванні спостережуваної *A* отримаємо одне з власних чисел, наприклад, λ_i цієї спостережуваної. Вірогідність набуття цього конкретного значення співпадає з квадратом амплітуди $|a_i|^2$, що стоїть перед відповідним власним вектором. Відповідно до прийнятого постулату, вектор стану в результаті такого вимірювання *A* миттєво редукується у відповідний власний вектор спостережуваної *A*:

$$|\psi\rangle \rightarrow\ |\varphi_i\rangle.$$

Такий процес вимірювання приводить до двох найважливіших наслідків. Передусім, він приводить до принципового індетермінізму квантової фізики. Дійсно, заздалегідь не можна передбачити, яке значення спостережуваної буде

отримано при вимірюванні, тобто, не можна передбачити, в який стан перейде система після вимірювання. У кращому разі, повторюючи вимірювання над численними копіями початкової системи, вдасться з'ясувати вірогідність переходу у відповідні стани. Проте, навіть таке максимально досяжне знання не дозволяє встановити початковий стан фізичної системи (7.21). Причину цього неважко помітити. Знання вірогідності означає знання квадратів амплітуд $|a_i|^2$. Це дає знання тільки модуля комплексної амплітуди, але не її фази θ_i ($a_i = |a_i|e^{i\theta_i}$). Отже, не можна за неповними даними відновити початковий стан $|\psi\rangle$. Це означає неможливість упізнати минуле в квантовій фізиці навіть по ідеально повних квантових вимірюваннях сьогодення. Легко помітити вражаючу різницю з класичним детермінізмом. Як вважалося, завжди можна отримати шляхом вимірювання повні дані про стан класичної системи. Використовуючи їх як початкові умови і вирішуючи рівняння руху у зворотному напрямі часу, в класичній фізиці можна упізнати минуле. Тут ми стикаємося з необхідністю у відмові від класичного детермінізму. Таким чином, ідеальні дані вимірювання квантової фізики не мають тієї повноти, як ці вимірювання класичної фізики. У свою чергу, це означає відмову від принципу абсолютизації стану фізичної системи, як втім і від принципу абсолютизації фізичного процесу, як існуючого незалежно від вимірювань.

Розглянемо тепер можливість вимірювання двох спостережуваних A і B. Зрозуміло, нетривіальним є одночасне вимірювання цих спостережуваних, а не послідовне в часі. У останньому випадку це зведеться до вимірювання однієї спостережуваної над відповідними станами системи. В принципі, можна подумки представити такий прилад, який, роблячи одне вимірювання, вимірює дві спостережувані величини. Проте, повинна існувати принципова можливість такого вимірювання. Ясно, що для цього ці спостережувані повинні мати однакові набори власних векторів. З'ясуємо, якими властивостями для цього повинні володіти спостережувані A і B. Нехай вони мають однакові власні вектори. Це означає, що

$$A|\varphi_i\rangle = \lambda_i|\varphi_i\rangle,$$
$$B|\varphi_i\rangle = \mu_i|\varphi_i\rangle.$$

Подіємо на перше рівняння оператором B, а на друге оператором A.

$$BA|\varphi_i\rangle = B\lambda_i|\varphi_i\rangle = \lambda_i B|\varphi_i\rangle = \lambda_i \mu_i|\varphi_i\rangle,$$
$$AB|\varphi_i\rangle = A\mu_i|\varphi_i\rangle = \mu_i A|\varphi_i\rangle = \mu_i\lambda_i|\varphi_i\rangle = \lambda_i \mu_i|\varphi_i\rangle.$$

Порівнюючи праві частини рівності, легко довести, що однакові власні вектори мають спостережувані, відповідні лінійні оператори яких комутують один з одним, тобто,

$$AB - BA = [A,B] = 0.$$

Таким чином, в квантовій фізиці можливе одночасне вимірювання тих спостережуваних, що тільки комутують одна з іншою. Зрозуміло, не усі спостережувані фізичної системи комутують одна з іншою. Як йде справа з такими спостережуваними? Перш, ніж обговоримо це питання, нагадаємо статистичні властивості спостережуваних величин. Для будь-якої

спостережуваної можна визначити середнє значення. Для нормованих векторів станів $\langle\psi|\psi\rangle = 1$ середнє значення спостережуваної A визначається як

$$\overline{A} = \langle\psi|A|\psi\rangle.$$

Окрім середнього значення, для фізичних статистичних величин важливу роль відіграють дисперсія або середньоквадратичні відхилення від середнього значення

$$\overline{\Delta A} = \left(\left\langle\psi\left|\left(A - \overline{A}\right)^2\right|\psi\right\rangle\right)^{\frac{1}{2}}.$$

Ця величина характеризує, наскільки в середньому великі відхилення спостережуваної від середнього значення. Тепер розглянемо середнє від квадрата спеціально підібраного оператора

$$\left\langle\psi\left|\left(\left(A - \overline{A}\right) + ix\left(B - \overline{B}\right)\right)^2\right|\psi\right\rangle \geq 0.$$

Це середнє позитивно визначене при будь-якому значенні числового параметра x. Виконуючи усереднювання, отримаємо

$$\left\langle\psi\left|\left(A - \overline{A}\right)^2\right|\psi\right\rangle - (x)^2\left\langle\psi\left|\left(B - \overline{B}\right)^2\right|\psi\right\rangle -$$
$$-ix\left\langle\psi\left|\left(\left(A - \overline{A}\right)\left(B - \overline{B}\right) - \left(B - \overline{B}\right)\left(A - \overline{A}\right)\right)\right|\psi\right\rangle \geq 0$$

або, вводячи позначення $C = i\left[\left(A - \overline{A}\right)\left(B - \overline{B}\right)\right] = i\left(\left(A - \overline{A}\right)\left(B - \overline{B}\right) - \left(B - \overline{B}\right)\left(A - \overline{A}\right)\right)$ і враховуючи властивість, що легко доводиться, $\left[\left(A - \overline{A}\right)\left(B - \overline{B}\right)\right] = [A,B]$, запишемо попередню рівність в зручній формі

$$\left\langle\psi\left|\left(A - \overline{A}\right)^2\right|\psi\right\rangle - x^2\left\langle\psi\left|\left(B - \overline{B}\right)^2\right|\psi\right\rangle - x\langle\psi|C|\psi\rangle \geq 0.$$

Ліва частина цієї нерівності позитивна при будь-якому виборі числа x. Виберемо його з умови екстремальності лівої частини

$$-2x\left\langle\psi\left|\left(B - \overline{B}\right)^2\right|\psi\right\rangle - \langle\psi|C|\psi\rangle = 0.$$

Тоді

$$x = -\frac{\langle\psi|C|\psi\rangle}{2\left\langle\psi\left|\left(B - \overline{B}\right)^2\right|\psi\right\rangle}.$$

Після підстановки цього значення x в приведену вище нерівність і досить простих перетворень отримаємо

$$\overline{\Delta A} \cdot \overline{\Delta B} \geq \frac{1}{2}[\overline{A,B}]. \qquad (7.22)$$

Нерівність (7.22) має величезне значення в квантовій фізиці. Так, з цієї нерівності виходить, що вимірювання комутуючих спостережуваних в ідеалі може бути здійснено точно. Для таких спостережуваних дисперсії можуть

одночасно занулитися. Проте, якщо спостережувані не комутують, то точний вимір однієї з них з малою дисперсією робить розкидування другою спостережуваною величезним. Це означає неможливість одночасного вимірювання таких величин. Зокрема, так відбувається для пар спостережуваних [*A, B*] = const, що часто зустрічаються в квантовій фізиці.

Приклад 7.11. *Так, для канонічно зв'язаних величин координати* q_k *і відповідного імпульсу* p_k *виконується комутаційне співвідношення згаданого вище типу*

$$[q_k, p_k] = ih.$$

Тут по k *підсумовування відсутнє. Використовуючи цю комутаційну рівність, з нерівності (7.22) отримаємо*

$$\overline{\Delta q_k} \cdot \overline{\Delta p_k} \geq \frac{h}{2}.$$

Це співвідношення отримано Гейзенбергом. Цей індекс просто нумерує степені свободи. Ця нерівність зіграла видатну роль в побудові квантової фізики. З цього співвідношення невизначеності виходить, що якщо координата деякої степені свободи вимірюється точно ($\overline{\Delta q_i} = 0$)*, то вимірювання імпульсу цієї степені свободи можливо тільки з нескінченною дисперсією або, швидше правильно сказати, вимірювання імпульсу стає неможливим. Таке співвідношення невизначеності виконується для координат і імпульсу будь-якої степені свободи. Фізично отримати таку нерівність можна, виходячи з простих міркувань. Припустимо, ми хочемо виміряти положення і імпульс деякої квантової частки. В якості детектувального приладу використовуємо фотон. Ясно, що точність вимірювання координати частки визначається довжиною хвилі фотона. Нехай* $\Delta q \approx \lambda$. *Тоді для здійснення точнішого вимірювання положення частки треба використати фотони з малою довжиною хвилі. Чим менше* λ*, тим точніше буде виміряно положення частки. Проте, фотон володіє імпульсом* $p = hk = h2\pi / \lambda$*. Тому зменшення довжини хвилі* λ *приводить до збільшення імпульсу фотона і, отже, імпульсу, який передано частці, що вимірюється. Це означає збільшення невизначеності імпульсу частки при такому вимірюванні* $\Delta p \approx h2\pi / \lambda$*. Перемножуючи ці невизначеності, отримаємо*

$$\Delta q \cdot \Delta p \approx \lambda \frac{h2\pi}{\lambda} \approx h2\pi.$$

Це і є співвідношення невизначеності Гейзенберга. Воно показує неможливість в квантовій фізиці спостереження величин, яким в класичній фізиці приписуються одночасно певні значення.

Фактично це означає не тільки те, що одночасне вимірювання таких величин неможливе в квантовій фізиці, але навіть те, що не існує таких станів, при яких і спостережувані мають певні значення одночасно. Принцип невизначеності дозволяє по іншому осмислити корпускулярно – хвильовий дуалізм часток матерії. Дійсно, передусім слід замислитися над тим, що ми можемо називати часткою. Згідно макроскопічної інтуїції об'єкт, що

локалізований в просторі (тобто, те, що має визначені координати) і має певну швидкість, можна визнати в якості частки. Але, як випливає з принципу невизначеності, одночасно квантовий об'єкт і не може мати цих двох властивостей. Ці властивості в квантовій механіці є взаємно додатковими і може реалізуватися тільки або одна, або інша властивість. На цьому заснований глибокий принцип додатковості Бора. У цьому сенсі квантова частка і володіє корпускулярно – хвильовим дуалізмом.

Співвідношення невизначеності і їх узагальнення встановлені для багатьох спостережуваних квантової фізики. Зокрема, добре відома нерівність

$$\Delta\varepsilon\Delta\tau \geq \frac{h}{2},$$

що встановлює зв'язок між дисперсією або невизначеністю енергії $\Delta\varepsilon$ і похибкою у вимірюванні часу $\Delta\tau$. Це співвідношення невизначеності було отримане ще Бором. Воно встановлює, що точність у вимірюванні енергії об'єкту обмежується неточністю у вимірюванні часу події. Так, наприклад, енергія квантової системи, яка має кінцевий час життя τ, не може бути визначена точніше, ніж це витікає з попередньої нерівності.

Величезну роль для збереження суперпозиції станів грає можливість повної ізоляції системи від зовнішнього світу. Порушення ізольованості стимулює перехід системи в один з її стаціонарних станів. Такий перехід може служити прикладом декогеренції системи. Іншими словами, підвищення точності вимірювань неминуче веде до обліку незамкнутості квантових систем.

Спрощуючою обставиною служить побудова феноменологічних теорій вимірювання відкритих квантових систем в припущенні відсутності впливу на зовнішнє середовище. Наслідком цього є і поява в квантовій фізиці принципово не спостережуваних характеристик таких систем.

Приклад 7.12. *Гарною демонстрацією особливостей квантової механіки служить завдання військового приймання, запропоноване Елітцуром і Вайдманом. Привабливість цього завдання тісно пов'язана з його колоритним формулюванням. Припустимо, що на деякому військовому заводі йде виробництво атомних бомб з детонатором особливої чутливості. Цей тип детонатора спрацьовує навіть при попаданні на нього одного фотона видимого діапазону (див. Рис. 7. 7). Зрозуміло, як і на будь-якому заводі, певна частина продукції відноситься до браку. Брак викликаний порушенням механізму детонатора, при якому бомба не вибухає при попаданні фотона на відбивач детонатора. Такий дефект виникає на етапі складання готового виробу. Природно, механізм детонатора функціонує за класичними законами. Головне питання, чи можна відокремити якісну продукцію від бракованої? Іншими словами, вимагається знайти бомбу, що точно функціонує правильно. Ясно, що з класичної точки зору це завдання не здійснене. Причина в тому, що перевірка якості приведе до вибуху якісної продукції і тим самим зробить мету недосяжною. Абсолютно інакше це завдання виглядає у рамках квантової ідеології. Виявляється, що з точки зору квантової теорії це завдання має розв'язок. При цьому виділення якісної продукції технічно легко здійснене.*

Для того, щоб розібратися в цьому, розпочнемо з обговорення деяких простих елементів динаміки фотона. На рис. 7.8 показані два типи дзеркал, здатних відбивати фотони. У лівій частині зображено звичайне дзеркало, що відбиває фотони. Фотон, що падає, в стані $|F\rangle$ *після відображення переходить в стан* $i|R\rangle$. *Множник* i *пов'язаний із зрушенням фази при відображенні. Існує ще один тип дзеркала, який знадобиться в подальшому. Це напівпрозоре дзеркало. З класичної точки зору пучок фотонів, що падає на таке дзеркало, розділяється на два пучки проходження і відображення від нього фотонів. Зовсім інакше ситуація виглядає з точки зору квантової механіки. Дійсно, стан* $|F\rangle$ *одного фотона після взаємодії із дзеркалом еволюціонує в стан фотона* $|T\rangle + i|R\rangle$. *Іншими словами, навіть один фотон знаходиться в суперпозиції станів фотона, що проходить і відображається одночасно. Ні про будь-яке ділення фотона на половину не може бути і мови. Для того, щоб виявити результат такої суперпозиції станів після проходження напівпрозорого дзеркала фотоном, використовуючи уже звичайні дзеркала, зведемо разом стани, що пройшли і відобразилися. Як це виконати, показано на рис. 7.9. На рис. 7.10 наведена схема приймання якісної продукції.*

Джерело I випускає фотон в стані $|F\rangle$. *Взаємодіючи з напівпрозорим дзеркалом, стани* $i|R\rangle$ *і* $|T\rangle$ *відображаються від звичайних дзеркал, при цьому нехай відбувається наступна еволюція станів:*

$$|R\rangle \rightarrow i\,|R'\rangle\,,$$
$$|T\rangle \rightarrow i\big|R''\big\rangle.$$

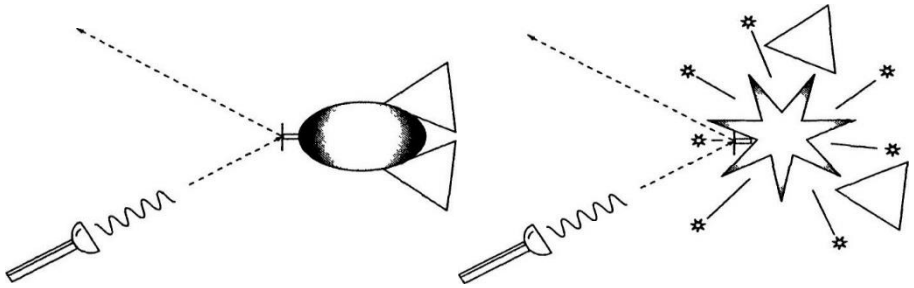


Рис. 7.7. У лівій частині зображена бракована бомба, яка не вибухає при попаданні фотона на відбивач детонатора. Справа – якісна продукція вибухнула при попаданні фотона

Таким чином, суперпозиція станів еволюціонує як:

$$|T\rangle + i|R\rangle \rightarrow i\big|R''\big\rangle + i\cdot i\,|R'\rangle = i\big|R''\big\rangle - |R'\rangle.$$

Тепер проаналізуємо, що станеться після взаємодії станів на останньому напівпрозорому дзеркалі. Так, стан $|R'\rangle$ *, що падає на нього, еволюціонує так:*

$$|R'\rangle \rightarrow |F\rangle + i|P\rangle,$$

а стан $\big|R''\big\rangle$ *еволюціонує згідно*

$$\left|R''\right\rangle \rightarrow |P\rangle + i|F\rangle.$$

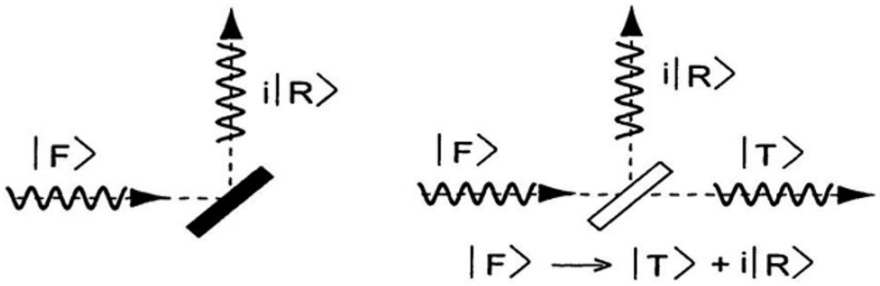


Рис. 7.8. Зліва відображення фотона від звичайного дзеркала.

Справа символічно показаний процес відображення фотона від напівпрозорого дзеркала. Стан, що відповідає двом альтернативам: відображенню і проходженню, які позначені $i|R\rangle$ і $|T\rangle$. Після взаємодії з таким дзеркалом фотон знаходиться в стані $|T\rangle + i|R\rangle$.

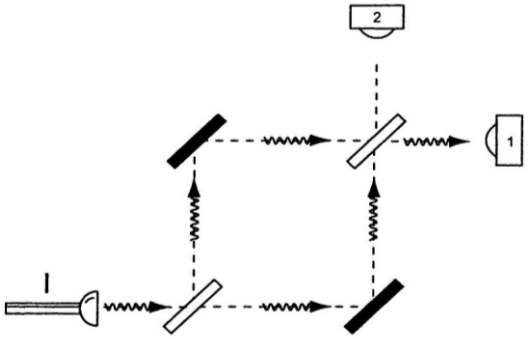


Рис. 7.9. Більш складна схема, що демонструє наслідок квантового характеру поведінки фотонів

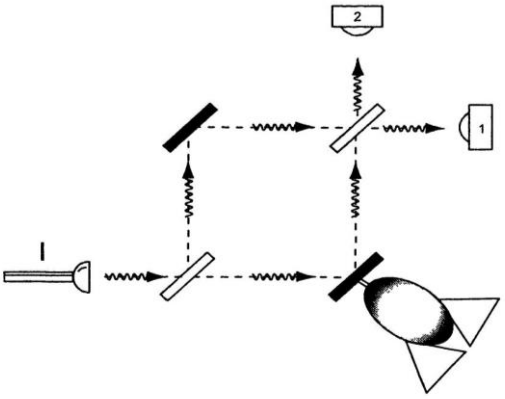


Рис. 7.10. Схема приймання якісної продукції

Залишається тепер встановити, у що перейде стан фотона $i\left|R''\right\rangle$ - $|R'\rangle$. *Враховуючи лінійність еволюції, отримаємо:*

$$i\left|R''\right\rangle\text{-}|R'\rangle \rightarrow i(|P\rangle + i|F\rangle) - (|F\rangle + i|P\rangle) = -2|F\rangle.$$

Звичайно, двійка не істотна. Це не міняє стану, да і умова нормування її знищить. Важливо, що фотони можуть бути зафіксовані тільки детектором 1. На детектор 2 фотони не будуть падати ніколи. Можна сказати, що це результат квантової інтерференції станів. Тепер можна пояснити принцип вибору якісної продукції на згаданому вище військовому заводі (див. Рис. 7.10). Розпочнемо з того, що в якості дзеркала в правому нижньому кутку використовуємо дзеркало детонатора бомби. Ясно, що в цьому випадку бомба має сенс приладу, що визначає стан фотона після взаємодії з першим напівпрозорим дзеркалом. Якщо бомба неякісна, то, згідно з попереднім розглядом, фотон буде зафіксований тільки детектором 1. Нехай тепер бомба якісна. Тоді можливі дві квантові ситуації, які перекладаються на класичний рівень: фотон падає на детонатор і фотон не падає на детонатор. У першому випадку бомба вибухає, а в другому - ні. Іншими словами, бомба грає роль приладу, який вимірює стан $|T\rangle + i|R\rangle$ *фотона. Цей прилад вибухає, якщо вимірює стан* $|T\rangle$ *і не вибухає, якщо вимірює стан* $|R\rangle$ *. Порівнюючи амплітуди цих станів, легко помітити, що ці дві події мають однакову вірогідність. Отже, в половині випадків якісна бомба вибухне, в половині - ні. У разі, коли якісна бомба не вибухає, вона вимірює стан фотона* $|T\rangle + i|R\rangle$ *і, відповідно, редукує його в стан* $i|R\rangle$*. При цьому, після відображення, стан фотона буде* $|R'\rangle$*. В результаті взаємодії з останнім напівпрозорим дзеркалом фотон придбає стан*

$$--|R'\rangle \rightarrow -|F\rangle - i|P\rangle.$$

Таким чином, ясно, що в половині випадків спрацьовуватиме детектор 2. Тоді, проводячи такі вимірювання над бомбами при спрацьовуванні детектора 2, можна строго стверджувати, що бомба, що піддається тестуванню, ЯКІСНА. Зрозуміло, така схема тестування з економічної точки зору має явні недоліки. Звичайно, її можна поліпшити і зменшити долю втрат при тестуванні. Проте, для нас важливо, що проблема, що не має рішення в класичному підході, легко вирішилася в квантовому варіанті.

Обговоримо тепер роль переплутаних станів в квантовій механіці. Вже на стадії створення квантової механіки такі стани були використані при обговоренні її складних проблем. Повернемося до "парадоксу" Ейнштейна, Подільського, Розена. Слово парадокс узято в лапки, тому що правильніше говорити про експеримент Ейнштейна, Подільського, Розена, який, зрозуміло, не є парадоксом. Головна причина привертання уваги до переплутаних станів полягала в спробі довести, що квантова механіка неповна і потрібна її перебудова у дусі класичної фізики. Проте, велика частина фізиків не прийняла цієї ідеології. Для цього були глибокі причини. Проте, для нас важливо обговорити ідеї, що лежать в основі цього експерименту. Автори вважали, що правильна теорія повинна задовольняти деяким природним вимогам. Припустимо, що в деякій точці простору з вакууму народилася електрон - позитронна пара. Спін цих двох часток може набувати тільки два протилежні значення із-за закону збереження спіна. Стан таких часток буде описуватися вектором

$$|\psi_+\rangle = \frac{1}{\sqrt{2}}|0,1\rangle + \frac{1}{\sqrt{2}}|1,0\rangle.$$

Нехай ці частки віддалилися одна від іншої на величезну відстань. Тоді, згідно класичної інтуїції, хочеться припустити, що ці частки мають визначені характеристики незалежно від їх вимірювань. Це припущення називається припущенням реалізму. З аналогічних міркувань використовуємо і припущення локальності. Це означає, що вимірювання характеристик, скажімо, електрона, визначається тільки вимірюваннями, що проводяться над ним, і не залежать від вимірювань, що проводяться або не проводяться над видаленим позитроном. Виконання цих двох припущень називають локальним реалізмом. Як ми побачимо нижче, ці припущення призводять до певних обмежень, які вже можуть бути перевірені експериментально. Ці обмеження формулюються у вигляді нерівностей, які прийнято називати нерівностями Белла. Саме Белл перший сформулював одну з таких нерівностей [7]. Наслідуючи його логіку, розглянемо ці нерівності. Повертаючись до електрона і позитрона, ми можемо вимірювати проєкцію спінів уздовж деяких трьох вісів. Припустимо, що існує вірогідність певних напрямів спінів уздовж цих вісів. Це базується на принципі реалізму. Ясно, що вимірювання спіна уздовж будь-якої осі може давати тільки напрям спіна уздовж або проти вибраної осі. Нехай ці вісі впорядковані і, наприклад, таке позначення p $(\uparrow,\downarrow,\downarrow)$ буде означати вірогідність того, що відносно першої осі проєкція спіна частки спрямована уздовж неї, відносно другої - проти, як і відносно третьої. У ситуації з електроном і позитроном напрям спіна, скажемо, електрона, можна завжди визначити відносно двох напрямів. Наприклад, якщо вимірювати напрям спіна відносно першого напряму у електрона і вимірювати напрям спіна відносно другої осі у позитрона. Враховуючи, що напрям спіна електрона відносно будь-якої осі протилежний направленню спіна позитрона відносно цієї осі, легко встановити таким чином напрям проєкції спіна електрона відносно двох напрямів. Таким чином, можуть бути виміряні $P_{1,2}\ (\uparrow,\downarrow)$ - вірогідність електрона мати спін, спрямований уздовж першого і проти другого напрямів, $P_{1,3}\ (\uparrow,\downarrow)$ – уздовж першого і проти третього і $P_{2,3}\ (\uparrow,\downarrow)$ - уздовж другого і проти третього. Цю вірогідність легко отримати з вірогідності відносно трьох вісів або напрямів:

$$P_{1,2}\ (\uparrow,\downarrow) = \mathrm{p}\ (\uparrow,\downarrow,\downarrow) + \mathrm{p}\ (\uparrow,\downarrow,\uparrow),$$

$$P_{2,3}\ (\uparrow,\downarrow) = \mathrm{p}\ (\downarrow,\uparrow,\downarrow) + \mathrm{p}\ (\uparrow,\uparrow,\downarrow),$$

$$P_{1,3}\ (\uparrow,\downarrow) = \mathrm{p}\ (\uparrow,\downarrow,\downarrow) + \mathrm{p}\ (\uparrow,\uparrow,\downarrow).$$

Додаючи два перших співвідношення і віднімаючи третє, отримаємо:

$$P_{1,2}\ (\uparrow,\downarrow) + P_{2,3}\ (\uparrow,\downarrow) = P_{1,3}\ (\uparrow,\downarrow) + \mathrm{p}\ (\uparrow,\downarrow,\uparrow) + \mathrm{p}\ (\downarrow,\uparrow,\downarrow)\ .$$

Враховуючи, що вірогідності p $(\uparrow,\downarrow,\uparrow)$ і p $(\downarrow,\uparrow,\downarrow)$ додатні, легко отримати нерівності Белла

$$P_{1,2}\ (\uparrow,\downarrow) + P_{2,3}\ (\uparrow,\downarrow) \geq P_{1,3}\ (\uparrow,\downarrow). \qquad (7.23)$$

Власне, ця нерівність типу нерівності трикутника. Сума будь-яких двох вірогідностей більша за третю. Отримання цих нерівностей не пов'язане з

квантовою фізикою, це відображення тільки локального реалізму. Тому слід тепер перевірити, чи виконуються ці нерівності в квантовій механіці.

Проте, перш ніж перейти до цього, нагадаємо деякі необхідні факти про таку квантову спостережувану як спін. Для частки із спіном $\frac{1}{2}$ власні значення оператора моменту приймають тільки два значення $\pm\frac{1}{2}$. Так, власні вектори оператора $\widehat{M}_z$, що відповідають цим значенням, будемо позначати $|\uparrow\rangle$ і $|\downarrow\rangle$ відповідно.

$$\widehat{M}_z|\uparrow\rangle = \frac{1}{2}|\uparrow\rangle,$$

$$\widehat{M}_z|\downarrow\rangle = -\frac{1}{2}|\downarrow\rangle.$$

Проте, оператори моментів імпульсу не комутують один з одним, а задовольняють наступним комутаційним співвідношенням:

$$[\widehat{M}_x, \widehat{M}_y] = ih\widehat{M}_z,$$

$$[\widehat{M}_y, \widehat{M}_z] = ih\widehat{M}_x,$$

$$[\widehat{M}_z, \widehat{M}_x] = ih\widehat{M}_y.$$

Це означає, що власні функції оператора $\widehat{M}_z$ не є власними функціями операторів $\widehat{M}_x$ і $\widehat{M}_y$. Дія цих операторів на ці вектори визначається рівностями:

$$\widehat{M}_x|\uparrow\rangle = \frac{1}{2}|\downarrow\rangle,$$

$$\widehat{M}_x|\downarrow\rangle = \frac{1}{2}|\uparrow\rangle,$$

$$\widehat{M}_y|\uparrow\rangle = \frac{i}{2}|\downarrow\rangle,$$

$$\widehat{M}_y|\downarrow\rangle = -\frac{i}{2}|\uparrow\rangle.$$

Матриці, що представляють компоненти операторів моменту імпульсу, називаються матрицями Паулі

$$\widehat{M}_k = \frac{1}{2}\,\sigma_k,$$

де індекс $k = x, y, z$, а явний вид матриць Паулі наступний:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Ці матриці мають ряд простих, таких, що легко перевіряються, властивостей, наприклад,

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I, \tag{7.24}$$

$$\sigma_x\sigma_y = -\sigma_y\sigma_x = i\sigma_z. \tag{7.25}$$

Вектор стану частки із спіном $\frac{1}{2}$ визначається суперпозицією станів $|\uparrow\rangle$ і $|\downarrow\rangle$. Більш послідовно слід вважати вектор стану спінором, який зазвичай записується у виді

$$|\psi\rangle = \begin{pmatrix} \psi_+ & (\vec{x}) \\ \psi_- & (\vec{x}) \end{pmatrix}.$$

Враховуючи, що оператор моменту імпульсу є оператором інфінітезимальних поворотів навколо відповідних вісів, можна представити оператор повороту навколо осі, спрямованої уздовж одиничного вектора, як

$$\hat{U} = e^{i\vec{n}\cdot\vec{M}\theta}.$$

Це оператор повороту на кут θ навколо осі, спрямованої уздовж $\vec{n}$.

Для оператора власного моменту імпульсу, тобто, спіна, враховуючи властивості матриць Паулі (7.24), це співвідношення можна спростити і записати у виді:

$$\hat{U} = e^{\frac{1}{2}\theta\vec{n}\cdot\vec{\sigma}} = cos\left(\frac{\theta}{2}\right) + i\, sin\left(\frac{\theta}{2}\right)\vec{n}\cdot\vec{\sigma}\,. \tag{7.26}$$

Для його виведення треба використати розкладання в ряд Тейлора і властивості матриць Паулі. Це співвідношення знадобиться нижче. Здійснимо поворот, наприклад, вектора стану $|\downarrow\rangle$ на кут θ навколо осі x

$$\hat{U}|\downarrow\rangle = \left(cos\left(\frac{\theta}{2}\right) + i\, sin\left(\frac{\theta}{2}\right)\sigma_x\right)|\downarrow\rangle = cos\left(\frac{\theta}{2}\right)|\downarrow\rangle + i\, sin\left(\frac{\theta}{2}\right)|\uparrow\rangle.$$

Таким чином, легко зрозуміти, що спостережувані значення проєкцій спіна на будь - яку ось співпадають з $c \pm \frac{1}{2}$. Тепер перейдемо до перевірки нерівностей Белла з точки зору квантової механіки (рис. 7.11). Нехай перший напрям співпадає з віссю z. Тоді вимірювання проєкції спіна електрона зі значенням $\frac{1}{2}$ переводить стан наступним чином:

$$\frac{|\uparrow,\downarrow\rangle + |\uparrow,\downarrow\rangle}{\sqrt{2}} \rightarrow |\uparrow\rangle|\downarrow\rangle.$$

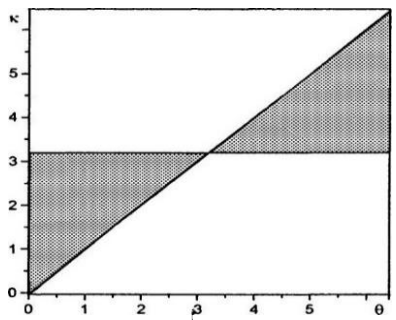


Рис. 7.11. На площині (θ, κ) показані сірим області порушення нерівностей Белла (7.23) для квантових значень вірогідності

Таким чином, для оцінки вірогідності виявити електрон в стані з напрямом уздовж осі z і проти напряму, поверненого, наприклад, на кут θ навколо осі x, можна отримати, вимірюючи стан видаленого позитрона.

Ця вірогідність співпадає з вірогідністю позитрона мати напрям спіна уздовж осі, поверненої на кут θ навколо осі x. Для того, щоб оцінити цю вірогідність, треба здійснити поворот вектора стану позитрона $|\downarrow\rangle$ на кут θ навколо осі x. Такий поворот ми здійснювали вище і тому легко виконати обчислення шуканої вірогідності. Вона співпадає з $sin\left(\frac{\theta}{2}\right)^2$, як модуль квадрата амплітуди при відповідному стані.

Таким чином, вірогідність рівна:

$$P_{12}(\uparrow,\downarrow) = \frac{1}{2}\, sin\left(\frac{\theta}{2}\right)^2.$$

Тут в якості першого напряму вибрана ось z , другий напрям співпадає з напрямом, поверненим на кут θ навколо осі x, а третій з напрямом, поверненим на кут λ також навколо осі x. Зрозуміло, такий вибір не принциповий, а тільки спрощує використовувані формули. Тоді аналогічний вид має вірогідність

$$P_{13}(\uparrow,\downarrow) = \frac{1}{2}\, sin\left(\frac{x}{2}\right)^2$$

і, нарешті, вірогідність

$$P_{23}(\uparrow,\downarrow) = \frac{1}{2}\, sin\left(\frac{\theta - x}{2}\right)^2.$$

Використовуючи ці вирази, можна переконатися, що нерівність Белла (7.23) для квантових вірогідностей порушується при певних значеннях кутів (див. Рис. 7. 11). Це означає, що локальний реалізм суперечить квантовій механіці. Для того, щоб переконатися, який вибір робить Природа, були зроблені ряд експериментів. Експерименти підтвердили співвідношення, що виходять з квантової теорії. Таким чином, Природа слідує квантовим законам.

7.3 Квант інформації - кубіт

Поняття класичного біта або "атома" інформації повинне істотно модифікуватися в квантовому випадку. У класиці біт може приймати тільки два стани: 0 або 1. Іншими словами, простір станів біта - кінцева множина, що складається з двох значень $\{0,1\}$. У квантовій механіці, зрозуміло, існують прості системи, які можуть служити зразком для введення кубіту. Це дворівневі системи. Наприклад, можна розглянути частку із спіном в магнітному полі. У базисних станах її спін може бути спрямований уздовж поля (вгору) $|\uparrow\rangle$ або проти нього (вниз) $|\downarrow\rangle$. Інакше кажучи, ця квантова система має тільки два базисні стани. Для подальших цілей зручно перепозначити ці базисні стани як $|1\rangle$ і $|0\rangle$. У квантовій фізиці ці стани інтерпретуються як базисні вектори в гільбертовому просторі. Тоді довільний стан такої частки описується вектором станів або хвильовою функцією $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Комплексні коефіцієнти α і β визначають

вірогідність знайти частку в стані $|1\rangle$ як $|\beta|^2$ і в стані $|0\rangle$ як $|\alpha|^2$. Природно тому вірогідність знайти частку в якому - небудь стані рівна:

$$|\alpha|^2 + |\beta|^2 = 1.$$

Таким чином, стан такої квантової частки визначається одиничним вектором $|\psi\rangle$ у двовимірному гільбертовому просторі C^2. Це і є потрібний квант інформації або кубіт. Природно, під кубітом можна розуміти абстрактну квантову систему з двома базисними векторами станів. Модель кубіта у вигляді частки із спіном в магнітному полі не єдина. В якості іншої моделі кубіта можна розглядати стани електрона в атомі, приписуючи двом квантовим орбітам два базові стани. Ще одним прикладом може служити фотон з можливими двома типами поляризації і багато інших квантових систем. Здатність кубіту знаходитися в суперпозиції базисних станів істотно відрізняється від звичайних класичних представлень. Для розуміння цього слід представити стан монети трохи орел і трохи реверс. Для кубіту реалізується навіть стан

$$|\psi\rangle = |+\rangle = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle,$$

який відповідає рівно вірогідності базисних станів. Відповідно до постулату квантової механіки ясно, що вимірювання таких станів приведуть в половині випадків до результату 0, а в половині - до результату 1. Для такого стану зарезервуємо на майбутнє позначення $|+\rangle$.

Тепер слід обговорити відмінності між класичним бітом і квантовим кубітом. Головна відмінність полягає в тому, що класичний біт має тільки два стани, тоді як стани кубіту заповнюють гільбертовий простір. Квантова еволюція кубіту відбувається по нескінченній безлічі його можливих станів. Ця важлива відмінність може бути використана для створення ефективніших, в порівнянні з класичними, квантових алгоритмів (див. Рис. 7.12). Проте, не слід думати, що в одному кубіті можна зберігати нескінченну кількість інформації. Як не парадоксально, в кубіті можна зберегти стільки інформації, скільки і в класичному біті. Причина цього в тому, що вимірювання стану кубіту дають або стан $|0\rangle$, або стан $|1\rangle$. Ніякої іншої інформації про стан кубіту отримати неможливо. Для наочних уявлень про стан кубіту і його еволюції іноді зручно використати геометричний образ його фазового простору. Нагадаємо, що в квантовій фізиці стани $|\psi\rangle$ визначені з точністю до множення на фазовий множник $e^{i\theta}$. Іншими словами, стани, що відрізняються на фазовий множник, не відрізняються один від одного і вважаються співпадаючими. Фаза стану – квантова властивість, що немає аналогу в класичній фізиці. Тому станам кубіту можна надати природну геометричну інтерпретацію, що враховує умову нормування $|\alpha|^2 + |\beta|^2 = 1$. Запишемо довільний вектор стану у виді:

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\alpha}\, sin\left(\frac{\theta}{2}\right)|1\rangle.$$

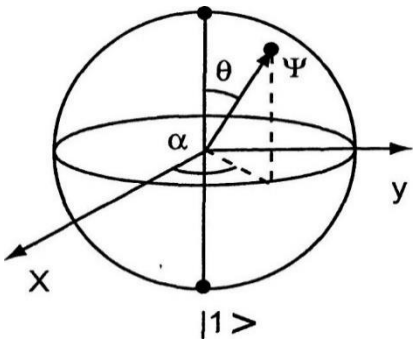


Рис. 7.12. Ілюстрація геометричного трактування стану кубіта як точки на одиничній сфері в тривимірному просторі. Полюси відповідають базисним станам. Ця сфера іноді називається сферою Блоха

Враховуючи попередні зауваження, можливість такого представлення легко зрозуміти. Величина θ змінюється в інтервалі $\pi \leq \theta \leq \pi$, величина α – в інтервалі $0 \leq \alpha \leq 2\pi$. Тепер можна помітити, що ці числа можна інтерпретувати як точку на одиничній сфері в тривимірному просторі (див. Рис. 7.12). Таким чином, хоча кубіт і не може зберігати більше інформації, ніж класичний біт, хочеться вірити, що Природа може використати усю суперпозицію станів в процесі обчислень, зрозуміло, до вимірювання. Іншими словами, вона може використати величезну приховану інформацію.

Перейдемо тепер до розкладання інформації по кубітам. У класичному випадку число станів N бітів було кінцевим і рівним 2^N . Квантовий варіант різноманітніший. Легко зрозуміти, що для N кубітів число базисних станів буде таким же. Іншими словами, стан цієї системи буде описуватися 2^N базисними векторами виду ($\underbrace{|0\rangle \cdots |0\rangle}_{N} = |0 \cdots 0\rangle, |1 \cdots 0\rangle$) і іншими, де в дужках стоять всі можливі набори 0 і 1.

Нагадаємо, що число, що стоїть на i –му місці, вказує базисний стан i-го кубіту у вибраному базисі $\{|0\rangle, |1\rangle\}$. Цікаво відмітити, що їх можна пронумерувати бітовим двійковим числом n і записувати у вигляді $|n\rangle$. Так, для двох кубітів отримаємо 4 базисні вектори $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, які записані в лексографічному порядку. Легко помітити, що ці вектори представляють числа 0, 1, 2, 3, які записані в двійковій системі числень. Для трьох кубітів число базисних векторів стану збільшується $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$ і дорівнює $2^3 = 8$. Можливий стан системи N кубітів визначається суперпозицією усіх базисних векторів станів:

$$|\psi\rangle = \sum_{i=1}^{2^N} a_i |i\rangle.$$

Тут $|i\rangle$ означає i-ий базисний вектор, що пронумерований в природному порядку. Комплексні амплітуди базисних станів a_i визначають вірогідність $|a_i|^2$

виявлення стану $|i\rangle$ при вимірюванні. Ці вірогідності задовольняють умові нормування

$$\sum_{i=1}^{2^N} |a_i|^2 = 1.$$

Таким чином, стани N кубітів належать 2^N – мірному гільбертовому простору $C^2 \otimes C^2 \otimes \ldots \otimes C^2 = C^{2^N}$. Розмірність цього простору росте експоненціально із зростанням числа кубітів. Це має як свої позитивні сторони, так і є джерелом багатьох проблем, пов'язаних з квантовими обчисленнями. Наочна геометрична інтерпретація фазового простору для системи кубітів невідома.

7.4 Елементарні квантові операції

Перейдемо тепер до обговорення елементарних операцій, за допомогою яких виконуються квантові обчислення. Далі побачимо, що як і в класичному випадку, ці операції повинні застосовуватися до невеликої кількості кубітів. Розпочнемо з операцій, які застосовуються до одного кубіта. Іншими словами, побудуємо квантовий гейт на одному кубіті. Єдиною нетривіальною однобітовою операцією в класичних обчисленнях був гейт ***NOT***. Нагадаємо дію цього гейта. Так, якщо на його вхід поступало значення 0, то на виході отримували 1, а якщо на вхід подавалося значення 1, то на виході з'являлося значення 0. Спробуємо побудувати його квантовий аналог. Нижче приведено порівняння класичного і квантового варіантів. На рис. 7.13 приведена дія квантового гейта $\widehat{\boldsymbol{NOT}}$ на довільний стан кубіту. Природною ідеєю є побудова квантового гейта, що діє на базисні стани кубіту аналогічно класичному аналогу.

Класичний варіант		Квантовий варіант
$0 \xrightarrow{NOT} 1$		$\|0\rangle \xrightarrow{\widehat{NOT}} \|1\rangle$
	$\Leftrightarrow$	
$1 \xrightarrow{NOT} 0$		$\|1\rangle \xrightarrow{\widehat{NOT}} \|0\rangle$

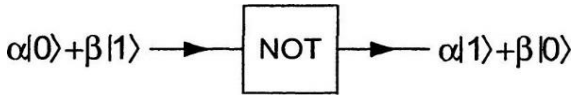


Рис. 7.13. Дія квантового гейта $\widehat{NOT}$ на довільний стан кубіту

Для реалізації цієї ідеї використовуємо природні квантові властивості. Стан кубіту визначається комплексним вектором станів, і його еволюція, згідно з квантовою фізикою, повинна відбуватися під дією унітарного оператора еволюції. Тому квантовий гейт повинен бути унітарним оператором. Вектори $|1\rangle$ і $|0\rangle$ можна і зручно представити у вигляді стовпців

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix},$$

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

Це нагадує позначення, які були використані раніше для опису частки із спіном спінорами. При такому представленні вектора станів унітарний оператор еволюції представляє собою унітарну матрицю 2 х 2. Неважко встановити вид матриці, при множенні на яку базисні стани еволюціонують згідно з класичним гейтом *NOT*:

$$\widehat{NOT}|0\rangle = |1\rangle,$$

$$\widehat{NOT}|1\rangle = |0\rangle.$$

Наступна матриця співпадає з уже знайомою нам матрицею Паулі

$$\widehat{NOT} = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} = \hat{\sigma}_x.$$

Унітарність матриць Паулі обговорювалася раніше. Дія квантового гейта на довільний стан кубіту описується співвідношенням (див. Рис. 7.13)

$$\widehat{NOT}\,(\alpha\,|0\rangle + \beta|1\rangle) = \alpha\,|1\rangle + \beta|0\rangle.$$

Аналогічно і дія інших квантових гейтів на стовпець вектора станів відповідає перемноженню на 2 х 2 унітарної матриці. Тому актуальним стає питання про вид довільної 2 х 2 унітарної матриці. Ці матриці і відповідають усім однокубітним квантовим гейтам. Розглядаючи довільну матрицю 2 х 2 з комплексними матричними елементами

$$\widehat{U} = \begin{pmatrix}\alpha & \beta\\\gamma & \delta\end{pmatrix},$$

отримаємо умови на матричні елементи із вимоги її унітарності $U\,U^{+} = I$. Це приводить до системи рівнянь виду

$$|\alpha|^2 + |\beta|^2 = 1, \tag{7.27}$$

$$|\gamma|^2 + |\delta|^2 = 1, \tag{7.28}$$

$$\alpha^*\,\gamma + \beta^*\,\delta = 0. \tag{7.29}$$

Замість комплексно зв'язаного рівняння до рівняння (7.29) використовуємо відому властивість унітарних матриць - рівність одиниці детермінанта матриці

$$\alpha\gamma - \beta\delta = 1. \tag{7.30}$$

З рівняння (7.29) легко помітити, що $\gamma = -\,m\,\beta^*$ і $\delta = m\,\alpha^*$, де m довільне число. Підстановка цих співвідношень в рівняння (7.28) дає:

$$|m|^2\,(|\alpha|^2 + |\beta|^2) = 1.$$

Порівнюючи отримане рівняння з рівнянням (7.27), легко встановити, що

$|m|^2 = 1$ і, отже, $m = e^{i\varphi}$.

Підставивши знайдені вище значення γ і δ в рівняння (7.30) і з умови рівності детермінанта одиниці, отримаємо:

$$e^{i\varphi}\,(|\alpha|^2 + |\beta|^2) = 1,$$

що означає $e^{i\varphi} = 1$. Таким чином, залишається задовольнити рівнянню (7.27), що достатньо просто:

$$\alpha = \cos(\theta)\, e^{i\varphi_1},$$

$$\beta = \sin(\theta)\, e^{i\varphi_2}.$$

Дійсно, із-за входження тільки модулів в перші два рівняння фази довільні, а модулі α і β повинні лежати на колі одиничного радіусу. Приведені розв'язки відповідають одиничному колу, заданому в параметричному виді. Зрозуміло, доповнені довільними фазовими множниками. Отже, загальний вигляд унітарних матриць визначається як

$$\hat{U} = \begin{pmatrix} \cos(\theta)\, e^{i\varphi_1} & \sin(\theta)\, e^{i\varphi_2} \\ -\sin(\theta)\, e^{-i\varphi_2} & \cos(\theta)\, e^{-i\varphi_1} \end{pmatrix}. \tag{7.31}$$

Після підрахунку числа параметрів легко виявити, що ці матриці тривимірні і утворюють групу унітарних унімодулярних матриць SU (2). Зазвичай обмежуються $\varphi_1 = \varphi_2 = 0$ і замінюють $\theta \to \theta/2$ [8]. Останнє зручно з точки зору вибору області зміни кута θ. Таким чином

$$\hat{U}_\theta = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{7.32}$$

Важливо відмітити, що з отриманого виду $\hat{U}_\theta$ стає ясним існування нескінченного числа навіть однокубітних квантових гейтів. Параметр θ має простий геометричний сенс. Розглянемо дію оператора U_π на базисні стани кубіту.

$$\hat{U}_\pi|0\rangle = -|1\rangle,$$

$$\hat{U}_\pi|1\rangle = |0\rangle.$$

Таким чином, на базисних станах цей гейт діє як класичний "NOT". Знак мінус в першій формулі не істотний, оскільки може бути прибраний на будь-якому етапі за рахунок фазового множника і, отже, стани $|1\rangle$ і $-|1\rangle$ еквівалентні з точки зору квантової механіки.

Розглянемо тепер гейт $\hat{U}_{-\pi/2}$. Цей гейт переводить стан $|0\rangle$ в однорідну суперпозицію обох станів.

$$\hat{U}_{-\pi/2}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

$$\hat{U}_{-\pi/2}|1\rangle = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle).$$

Класичний аналог цього гейта відсутній. Цей гейт часто називають гейтом Адамара і позначають латинською буквою *H* (див. Рис.7.14). Для різноманітності гейт Адамара будемо описувати унітарною матрицею

$$\widehat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

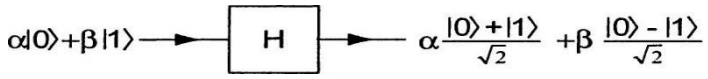


Рис. 7.14. Дія квантового гейта Адамара, що відповідає $\widehat{U}_{-\pi/2}$

Дія цього гейта відрізняється тільки фазовим множником від $\widehat{U}_{-\pi/2}$. Цей гейт може бути виражений і в термінах матриць Паулі як $\widehat{H} = \frac{1}{\sqrt{2}} (\hat{\sigma}_x + \hat{\sigma}_z)$. Наведемо приклад ще одного однокубітного гейта, позначимо його унітарною матрицею *K*

$$\widehat{K} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

Дія цього гейта на базисні стани наступна:

$$\widehat{K}\,|0\rangle = |0\rangle\,,$$

$$\widehat{K}\,|1\rangle = i|1\rangle$$

і зводиться до зміни фазового множника при дії на базисний стан $|1\rangle$. Нагадаємо, що для класичних схем був присутнім тільки один однобітовий гейт “NOT”. Однокубітових квантових гейтів значно більше – нескінченна кількість. У цьому сенсі стає складним питання про повний набір гейтів для квантових обчислень.

В підсумку відмітимо, що матриця довільного унітарного перетворення може бути представлена добутком трьох матриць обертання. Якщо позначити матрицю повороту на кут θ навколо осі, спрямованої уздовж одиничного вектору $\vec{e}$ як

$$\hat{R}_e(\theta) \text{ і } \vec{g} \text{ як } \hat{R}_g(\beta),$$

то

$$\widehat{U} = e^{i\varphi}\, \hat{R}_e(\alpha)\, \hat{R}_g(\beta)\, \hat{R}_e(\gamma),$$

де $\vec{e}$ і $\vec{g}$ непаралельні.

Зазвичай, за наявності вже вибраної декартової системи координат з'являється два однаково природних значень одиничних векторів.

Для цього можна використати напрями вісів координат. Тоді, використовуючи природні позначення, приведемо ці два представлення:

$$\widehat{U} = e^{i\varphi}\, \hat{R}_z(\alpha)\, \hat{R}_y(\beta)\, \hat{R}_z(\gamma),$$

$$\widehat{U} = e^{i\varphi}\, \hat{R}_x(\alpha)\, \hat{R}_y(\beta)\, \hat{R}_x(\gamma),$$

які називаються відповідно *zy* і *xy* розкладаннями. Залишається нагадати, що матриці обертання мають вигляд

$$\hat{R}_x(\theta)=\begin{pmatrix}\cos(\vartheta/2) & -i\sin(\theta/2)\\ i\sin(\theta/2) & \cos(\vartheta/2)\end{pmatrix},$$

$$\hat{R}_y(\theta)=\begin{pmatrix}\cos(\vartheta/2) & -\sin(\theta/2)\\ \sin(\theta/2) & \cos(\vartheta/2)\end{pmatrix},$$

$$\hat{R}_z(\theta)=\begin{pmatrix}\cos(\vartheta/2)+i\sin(\theta/2) & 0\\ 0 & \cos(\vartheta/2)-i\sin(\theta/2)\end{pmatrix}.$$

Правильність такого розкладання можна встановити безпосередньо, проводячи перемноження матриць і порівнюючи результат з матрицею (7.31). Із більш загальних міркувань цей результат легко зрозуміти, враховуючи відомий ізоморфізм групи обертань $SO(3) \simeq SU(2)$. Тут $SO(3)$ – тривимірна група обертань з детермінантом, рівним одиниці.

Для побудови повного набору гейтів нам потрібні і гейти, що діють на декілька кубітів. Ясно, що треба орієнтуватися на логічно оборотні гейти, оскільки усі квантові операції, що представлені унітарними операторами, оборотні. Це означає, що безповоротний двобітний гейт "**AND**" не має безпосереднього квантового аналога. Розглянемо декілька важливих двокубітних квантових гейтів. Зрозуміло, має сенс розглядати тільки двокубітні квантові гейти, які не зводяться до простої незалежної дії двох однокубітних гейтів.

У квантових схемах важливу роль відіграють гейти з керованим кубітом. Дія такого оператора визначається однокубітним гейтом $\hat{U}$ згідно простого правила. Визначимо оператор $\widehat{C(U)}$ з контролюючим першим кубітом наступним чином:

$$\widehat{C(U)}|0\rangle\,|x\rangle = |0\rangle\,|x\rangle,$$

$$\widehat{C(U)}|1\rangle\,|x\rangle = |1\rangle\hat{U}\,|x\rangle. \qquad (7.33)$$

Іншими словами, якщо контролюючий кубіт знаходиться в стані $|0\rangle$, то стан контрольованого кубіту не міняється. Якщо стан контролюючого кубіту $|1\rangle$, то стан контрольованого кубіту змінюється під впливом оператора $\widehat{U}$. Квантова схема, що відповідає таким гейтам, приведена на рис.7.15.

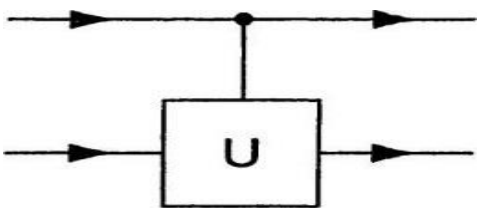


Рис. 7.15. Квантова схема двокубітного
гейта з контролюючим кубітом

Важливим оператором або гейтом такого типу є контрольоване "NOT", далі позначатимемо його як $\widehat{CNOT}$. Цей гейт діє на два кубіти і має два виходи. Причина в оборотності унітарних операторів. Базисні стани двох кубітів представимо у виді:

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

В цьому представленні квантовий гейт “CNOT” реалізується унітарною матрицею

$$\widehat{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}.$$

Унітарність цієї матриці легко перевірити. Перемноживши стовпці, що відповідають базисним станам, на цю матрицю, легко вивчити дію гейта "CNOT" на базисні стани

$$\widehat{CNOT}|00\rangle = |00\rangle,$$
$$\widehat{CNOT}|01\rangle = |01\rangle,$$
$$\widehat{CNOT}|10\rangle = |11\rangle,$$
$$\widehat{CNOT}|11\rangle = |10\rangle.$$

Для розуміння дії цього гейта зручно використати графічне зображення у виді класичних гейтів (рис. 7.16), де на виході стан $|x \oplus y\rangle$ приймає значення залежно від стану $|x\rangle$. Операція $\oplus$ означає додавання за модулем 2.

Порівнюючи зміну другого кубіту залежно від станів кубітів на вході, легко помітити зв'язок з дією класичного гейта "XOR". Тому іноді цей гейт називають і квантовим гейтом "XOR". Узагальнення гейта з контрольованим кубітом можна здійснювати у двох напрямках. По-перше, можна в якості оператора $\widehat{U}$ використати оператори, що діють на декілька кубітів, наприкад, N. При цьому визначення (7.33) переноситься на N +1 кубітні гейти. По-друге, можна узагальнити його на гейти з великим числом контрольованих кубітів. Далі важливим виявиться саме такий гейт CC (NOT) = C^2 (NOT) з двома контролюючими кубітами. Визначимо його таким чином:

$$C^2\,\widehat{NOT}\,|x_1\,x_2\rangle|y\rangle = \begin{cases}|x_1\,x_2\rangle\,|y\rangle, & \text{якшо } |x_1 \cdot x_2\rangle = 0,\\ |x_1\,x_2\rangle\widehat{NOT}\,|y\rangle, & \text{якщо } |x_1 \cdot x_2\rangle = 1.\end{cases} \quad (7.34)$$

Тут контролюючі кубіти – це перший і другий кубіти, третій кубіт контрольований. Схема цього квантового гейта показана на рис. 7.17.

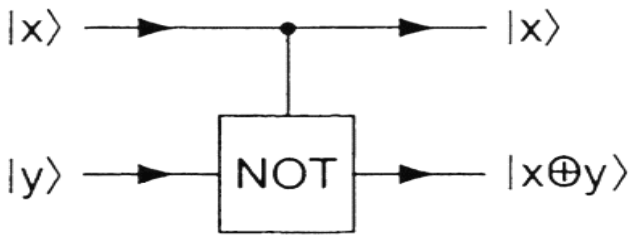


Рис. 7.16. Графічна діаграма дії унітарного оператора $\widehat{CNOT}$ у вигляді гейта з двома вхідними кубітами.

Стрілки вказують напрям часу. Результат дії - це нові стани цих двох кубітів на виході. Квантовий гейт $\widehat{CNOT}$ міняє стан одного кубіту

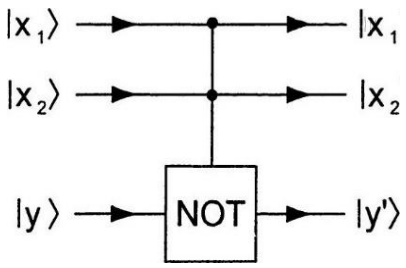


Рис. 7.17. Гейт $C^2\,\widehat{NOT}$

Дію цього гейта на базисні стани легко отримати:

$$C^2\,\widehat{NOT}\,|000\rangle = |000\rangle,$$
$$C^2\,\widehat{NOT}\,|001\rangle = |001\rangle,$$
$$C^2\,\widehat{NOT}\,|010\rangle = |010\rangle,$$
$$C^2\,\widehat{NOT}\,|011\rangle = |011\rangle,$$
$$C^2\,\widehat{NOT}\,|100\rangle = |100\rangle,$$
$$C^2\,\widehat{NOT}\,|101\rangle = |101\rangle,$$
$$C^2\,\widehat{NOT}\,|110\rangle = |111\rangle,$$
$$C^2\,\widehat{NOT}\,|111\rangle = |110\rangle\,.$$

Легко помітити близькість цього гейта до класичного гейта Тоффолі. Отже, використовуючи введені раніше квантові гейти, можна побудувати і квантовий гейт Тоффолі.

Приклад 7.13. *Обмін кубітами можна організувати, використовуючи двокубітну схему, побудовану із застосуванням елементарної операції* $\widehat{\boldsymbol{CNOT}}$**.** *Неважко перевірити, що потрібні властивості має схема, яка зображена на рис. 7.18. Унітарна матриця, що відповідає цій схемі, здійснює обмін між двома кубітами, діючи на стовпець, що визначає їх стан.*

$$U_{\leftrightarrow} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

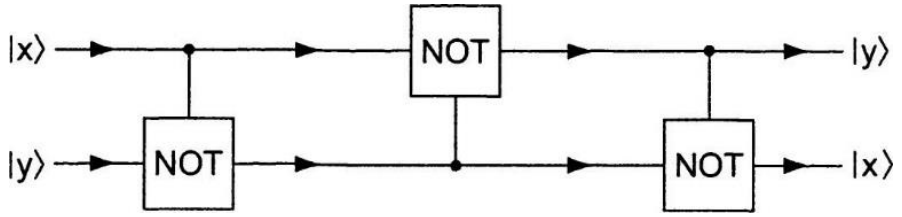


Рис. 7.18. Квантова схема обміну кубітами

Реалізацію квантового гейта Тоффолі можна здійснити і іншим способом. Для його побудови використаємо квантовий гейт "C (NOT) " і однокубітні гейти $\widehat{H}$; з'єднання їх в схему, як показано на рис. 7.19, формує квантовий аналог гейта Тоффолі. Використовуючи квантовий гейт Тоффолі, можна побудувати квантові схеми, що проводять квантові обчислення. Побудова таких схем аналогічна побудові класичних схем, відмінність тільки у використанні квантових елементів. Кожна така схема і проводить квантові обчислення над станами вхідних кубітів, а на виході вимірюється результат обчислень. Квантові схеми найчастіше використовуються при аналізі квантових алгоритмів. Проте, можна ввести і квантові машини Т'юринга. У цьому розділі основна увага була зосереджена на елементах близькості класичних і квантових схем. Проте, зрозуміло, що квантові схеми мають і абсолютно нові властивості, що і робить їх такими цікавими. Деякі принципово нові властивості квантових схем обговоримо в подальших розділах. Розпочнемо з обговорення існування базису квантових схем.

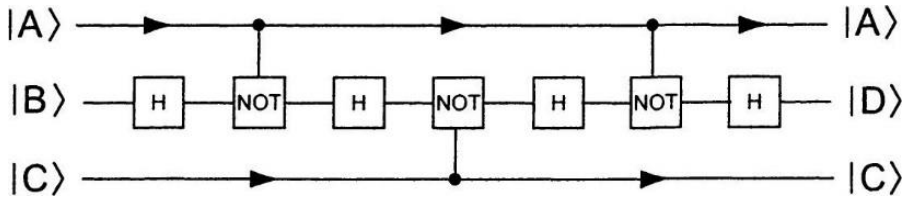


Рис. 7.19. Схема квантового гейта Тоффолі. Видно, що цей гейт побудований з трьох квантових гейтів *C (NOT)* і чотирьох однокубітних унітарних операторів Адамара H

7.5 Універсальні квантові схеми

Нагадаємо, що повний базис для класичних обчислень був кінцевий і складався, наприклад, з гейтів NOT, AND, 0R. Як йде справа в квантових схемах? Ясно, що якщо навіть однокубітних гейтів нескінченна кількість, то сподіватися на скінченність повного базису квантових гейтів не доводиться. Справа може

бути навіть значно гірше, якщо в такий базис треба включити нескінченну кількість двокубітних гейтів, трьохкубітних і так далі. На щастя, все не настільки погано. Можна довести, що нескінченний набір усіх двокубітних гейтів дозволяє отримати будь-який унітарний оператор і, отже, побудувати будь-яку квантову схему [8]. У таких схемах дія гейтів буде здійснюватися тільки на два кубіти в окремий момент часу. Ця властивість класичних схем зберігається і в квантових схемах. До цього зводиться ціна за точне представлення унітарного оператора квантовою схемою.

Приклад 7.14. *Для розуміння можливості такого базису квантових схем розглянемо простий приклад. Розкладемо довільну унітарну матрицю U на добуток унітарних матриць, які діють тільки на один кубіт. Нехай*

$$U = \begin{pmatrix} a & b & c \\ d & f & g \\ h & k & l \end{pmatrix}$$

-унітарна 3 х 3 матриця. Доведемо, що можна знайти такі унітарні матриці, що діють тільки на один кубіт, що

$$U_3 U_2 U_1 U = I. \tag{7.35}$$

З цього і виходить шукане представлення у вигляді добутку унітарних матриць, що діють тільки на один кубіт

$$U = U_1^+ U_2^+ U_3^+.$$

Для доведення рівності (7.35) виберемо у якості

$$U_1 = \begin{pmatrix} a^*/s & d^*/s & 0 \\ d/s & -a/s & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Тут $s = \sqrt{|a|^2 + |d|^2}$. *Ця унітарна матриця задовольняє необхідним властивостям. Перемножуючи матрицю* U_1 на *U , отримаємо:*

$$U_1 U = \begin{pmatrix} a' & d' & c' \\ 0 & f' & g' \\ h' & k' & l' \end{pmatrix}.$$

Залежність штрихованих матричних елементів від матричних елементів матриць U_1 і *U при перемноженні виходить автоматично. Проте, ця залежність поки що не важлива для подальших міркувань. Тепер перемножимо цей добуток на унітарну матрицю*

$$U_2 = \begin{pmatrix} a'^*/s' & 0 & h'^*/s' \\ 0 & 1 & 0 \\ h'/s' & 0 & -a'/s' \end{pmatrix}.$$

Тут $s' = \sqrt{|a'|^2 + |h'|^2}$. *Перемножуючи цю матрицю на матрицю* $U_1 U$, *отримаємо:*

$$U_2U_1U = \begin{pmatrix} 1 & b'' & c'' \\ 0 & f'' & g'' \\ 0 & k'' & l'' \end{pmatrix}.$$

Тепер згадаємо, що усі використовувані матриці унітарні, а значить унітарний і їх добуток. Це означає, що коефіцієнти $b'' = 0$ *і* $c'' = 0$, *а підматриця* $\begin{pmatrix} f'' & g'' \\ k'' & l'' \end{pmatrix}$ *також унітарна. Таким чином,*

$$U_2U_1U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & f'' & g'' \\ 0 & k'' & l'' \end{pmatrix}.$$

Матриця U_3 *має наступний вигляд*

$$U_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & f''^* & g''^* \\ 0 & k''^* & l''^* \end{pmatrix},$$

Перемножуючи матрицю U_3 *на добуток матриць* U_2U_1U *, отримаємо*

$$U_3U_2U_1U = I.$$

Звідси виходить необхідне представлення U

$$U = U_1^+U_2^+U_3^+$$

у вигляді добутку унітарних матриць спеціального виду. Приведена вище процедура може бути легко узагальнена на більш складні $2^N \times 2^N$ *матриці. При цьому можна довести, що будь-яка така унітарна матриця може бути представлена у вигляді добутку* 2^N $(2^N - 1)$ */ 2 унітарних матриць виду*

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & f & g & 0 & \cdots & 0 \\ 0 & \cdots & k & l & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix},$$

де матриця $\begin{pmatrix} f & g \\ k & l \end{pmatrix}$ *унітарна. Цим і завершимо пояснення повноти базису унітарних операторів, що діють на пари кубітів.*

Інша можливість полягає у використанні у якості повного базису квантових схем усіх однокубітних гейтів і двокубітного гейта $\widehat{\text{CNOT}}$. Приклад схеми, представленої з цього набору гейтів, наведений на рис. 7.19. Доказ цього досить простий і його можна знайти, наприклад, в [8].

Проте, при класичних і квантових обчисленнях часто достатньо обмежитися наближеними реалізаціями або апроксимаціями точних схем. В цьому випадку вже можна сподіватися на існування кінцевого базису для квантових схем. Проте, слова про апроксимацію або наближене представлення операторів вимагають певних уточнень. Передусім, слід уточнити поняття

близькості для унітарних операторів. Для цього треба увести відповідну норму для операторів або відстань в просторі операторів. Для цього природно використати звичайні властивості квантової теорії.

7.6 Паралельність квантових обчислень

Розглянемо тепер квантову схему, що обчислює деяку функцію f. Символічно квантова схема, що обчислює функцію f , зображена на рис.7.20. Стан перших N кубітів не змінюється, що дозволяє обчислювати функцію оборотним способом і, отже, для цього існує відповідний унітарний оператор. Дія схеми показана на базисних станах. Зрозуміло, якщо початковий стан нижніх кубітів нульовий, то $|x, 0\rangle \rightarrow |x, f(x)\rangle$, і схема обчислює значення функції $f(x)$. Тепер давайте подамо на верхні кубіти суперпозицію усіх базисних станів з відповідним коефіцієнтом нормування, тобто, $\frac{1}{\sqrt{2^N}} \sum|x\rangle$, а на нижні кубіти нульовий стан. В силу лінійності квантової механіки результуючий стан на виході схеми буде:

$$\sum |x\rangle\,|0\rangle \rightarrow \sum |x\rangle\ |f(x)\rangle.$$

Несуттєвий множник нормування в цих формулах упущений. Іншими словами, за одне проходження квантової схеми отримані усі значення функції. Це і є природна властивість паралельності квантових обчислень [2, 4]. Окрім цього, можна помітити, що стан на виході схеми – це переплутаний стан вхідних і вихідних станів кубітів. Незважаючи на таку, здавалося б, очевидну вигоду, все не настільки тривіально. Справа у тому, що використати цей квантовий паралелізм обчислень не зовсім просто. Більш проста причина полягає в останній стадії проведення квантових обчислень - вимірюванні результату. На цій стадії в результаті вимірювань можна отримати тільки одне значення функції. Отже, у такий спосіб використати паралельність обчислень не можна. Проте, це не означає, що його не можна використати іншими способами. Насправді таку паралельність обчислень можна ефективно використати при з'ясуванні глобальних властивостей функцій.

Приклади таких алгоритмів обговоримо в наступних розділах.

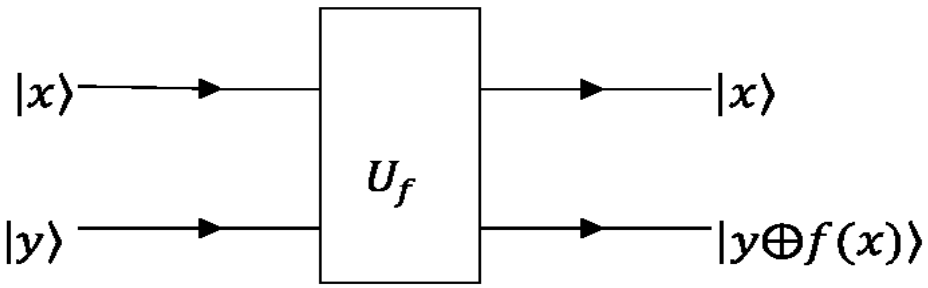


Рис. 7.20. Символічне зображення квантової схеми, що відповідає обчисленню функції f. U_f – унітарний оператор, що діє вказаним чином на верхні і нижні кубіти

На закінчення звернемо увагу, що класичні паралельні обчислення менш ефективні [4]. Квантова паралельність обчислень збільшує експоненціально об'єми обчислень при всього лише лінійному зростанні просторових об'ємів.

7.7 Неможливість клонування квантового стану

У класичних обчисленнях операція копіювання або класичний ксерокс звичайна нешкідлива операція. Абсолютно інакше виглядає її квантовий варіант. Виявляється, що невідомий квантовий стан неможливо скопіювати або клонувати. Виражаючись образно, неможливо клонувати незнайому людину. У цьому, зокрема, проявляється одна з важливих відмінностей класичних і квантових обчислень. Доведемо теорему про неможливість клонування.

Розпочнемо з вибору деякого стану, в якому знаходиться кубіт. Нехай цей стан $|a\rangle$. Клонування цього стану означає, що є присутнім ще один кубіr, який для визначеності знаходиться в стані $|0\rangle$, а після проведення клонування виявиться в стані $|a\rangle$. Тепер ясно, що для клонування спочатку треба мати два кубіти, що знаходяться в стані $|a,0\rangle$. Для здійснення клонування повинен існувати унітарний оператор $\widehat{U}_c$, який діє на початковий стан згідно з наступним рівнянням:

$$\widehat{U}_c|a,0\rangle = |a,a\rangle.$$

Ясно, що дія цього оператора на інший початковий стан, наприклад $|\beta,0\rangle$, абсолютно аналогічна і оператор $\widehat{U}_c$ не залежить від вибору стану $|a\rangle$. Іншими словами, цей оператор здійснює клонування будь-якого стану, наприклад,

$$\widehat{U}_c|\beta,0\rangle = |\beta,\beta\rangle.$$

Тепер візьмемо в якості стану, що клонується, стан рівноправної суперпозиції, який має вигляд:

$$|\gamma\rangle = \frac{1}{\sqrt{2}}(|a\rangle+|\beta\rangle).$$

Тоді початковий стан кубітів буде $|\gamma,0\rangle$ і знову подіємо на нього оператором клонування $\widehat{U}_c$

$$\widehat{U}_c|\gamma,0\rangle = \frac{1}{\sqrt{2}}(\widehat{U}_c|a,0\rangle + \widehat{U}_c|\beta,0\rangle) = \frac{1}{\sqrt{2}}(|a,a\rangle + |\beta,\beta\rangle) \neq |\gamma,\gamma\rangle.$$

Після виконання цих операцій приходимо до протиріччя, що означає відсутність оператора клонування або неможливість клонування довільного або, точніше, невідомого стану. Ця теорема, звичайно, не означає неможливість клонування взагалі. Просто не можна клонувати невідомий стан. Втім, навіть у разі невідомих станів, якщо відомо, що вони ортогональні, тобто,

$$\langle a|\beta\rangle = 0,$$

клонування стає можливим.

Приклад 7.15. *Звернемо увагу, що гейт контрольованого NOT можна розглядати як стан, що клонує* $|0\rangle$ *. Дійсно, якщо на його вхід подати стан* $|01\rangle$,

то на виході отримаємо $|00\rangle$. *Іншими словами, цей гейт здійснив клонування стану* $|0\rangle$.

Можливість клонування і "парадокс" Ейнштейна, Подільського, Розена призводили б до можливості передачі сигналів зі швидкістю, що перевищує швидкість світла.

Теорема про неможливість клонування має не лише фундаментальне значення, але і важлива для практичних застосувань. Одним з таких застосувань може служити неможливість підслуховування квантової передачі, якщо використовуються два неортогональні стани.

Для доказу припустимо, що передача з пункту А в пункт В здійснюється, використовуючи неортогональні стани $|a\rangle$ і $|b\rangle$, такі, що $\langle a|b\rangle \neq 0$. Нехай деякий зловмисник, використовуючи унітарне перетворення $\hat{U}_{spy}$, за допомогою приладу, що знаходиться в стані $|s\rangle$, намагається підслуховувати квантове повідомлення. Процес підслуховування можна описати наступними рівняннями:

$$\hat{U}_{spy}\,|sa\rangle = |s'a\rangle,$$

$$\hat{U}_{spy}\,|sb\rangle = |s''b\rangle.$$

Ці рівняння означають, що зловмисник діє таємно. Він міняє стан тільки свого приладу залежно від "букви", що передається, залишаючи незмінним повідомлення, що передається. Стан приладу $|s'\rangle$ означає, що підслухано $|a\rangle$, $|s''\rangle$ – що підслухано $|b\rangle$. При такій страте- гії зловмисника підслуховування неможливо виявити. Для аналізу можливості такої стратегії обчислимо скалярний добуток цих двох рівнянь згідно з правилами квантової механіки

$$\langle as|\,\hat{U}^{+}_{spy}\hat{U}_{spy}|sb\rangle = \langle as'|\hat{I}|s''b\rangle = \langle a|b\rangle\langle s'|s''\rangle.$$

Вираз ліворуч спрощується з урахуванням унітарності оператора $\hat{U}_{spy}$. Нагадаємо, що для унітарних операторів $\hat{U}^{+}_{spy}\hat{U}_{spy} = \hat{I}$. З урахуванням цього отримуємо

$$\langle a|b\rangle = \langle a|b\rangle\langle s'|s''\rangle\ .$$

Таким чином, якщо $\langle a|b\rangle \neq 0$, то це рівняння виконується тільки, якщо

$$\langle s'|s''\rangle = 1.$$

Це означає, що

$$|s'\rangle = |s''\rangle.$$

Тому зловмисник не може розрізнити букви неортогонального алфавіту і, отже, не може витягнути інформацію з повідомлення, що передається. Зрозуміло, якщо він хоче бути непоміченим. Отже, приємна неможливість підслуховування квантових передач може служити додатковим стимулом для розвитку квантових обчислень.

7.8 Щільне кодування

Припустимо, що ми хочемо передати з пункту А в пункт В двохбітове повідомлення. Це означає, що треба передати одне з чотирьох повідомлень 00, 01, 10 або 11. Зрозуміло, поміщаючи таке повідомлення в 2 кубіти і безпосередньо передаючи їх з пункту А в пункт В, це бажання легко здійснити. У класичному випадку ніякої іншої можливості і немає. Інакше йде справа у разі квантової інформації. Тут можна спробувати використати приховані ресурси беллівських станів. Проблема, яку слід вирішити, формулюється просто. Як передати двохбітне повідомлення, передаючи з пункту А в пункт В тільки один кубіт? Давайте спробуємо виконати це, використовуючи пару переплутаних кубітів, що знаходяться в стані

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$

при цьому перший кубіт знаходиться в пункті А, а другий в пункті В. Сам факт знаходження другого кубіту в пункті В не допоможе в класичному випадку. Наприклад, якщо він знаходиться в стані 1, то можна передати тільки повідомлення 01 і 11, передаючи перший біт у відповідному стані, але це рівно 1 біт, а не 2. У квантовому випадку, впливаючи тільки на кубіт, що знаходиться в пункті А, можна створити 4 взаємно ортогональних стани двох кубітів. Це вже два біти інформації. Дійсно, подіємо на перший кубіт оператором

$$(\hat{\sigma}_x \otimes \hat{I})\,|\Phi_+\rangle = \frac{1}{\sqrt{2}}(\hat{\sigma}_x|0\rangle)|0\rangle + \frac{1}{\sqrt{2}}(\hat{\sigma}_x|1\rangle)|1\rangle =$$

$$= \frac{1}{\sqrt{2}}\,|1\rangle|0\rangle + \frac{1}{\sqrt{2}}\,|0\rangle|1\rangle \equiv \frac{1}{\sqrt{2}}\,(|10\rangle + |01\rangle).$$

Позначення $(\hat{\sigma}_x \otimes \hat{I})$ просто вказує, що перший оператор діє на перший кубіт, а на другий кубіт діє одиничний оператор, що не міняє його стан. Аналогічно отримуємо результат дії інших операторів $\hat{\sigma}_y$, $\hat{\sigma}_z$ і $\hat{I}$ на перший кубіт

$$(\hat{\sigma}_y \otimes \hat{I})\,|\Phi_+\rangle = \frac{1}{i\sqrt{2}}\,(|10\rangle - |01\rangle),$$

$$(\hat{\sigma}_z \otimes \hat{I})\,|\Phi_+\rangle = -\frac{1}{\sqrt{2}}\,(|00\rangle - |11\rangle),$$

$$(\hat{I} \otimes \hat{I})\,|\Phi_+\rangle = |\psi\rangle = \frac{1}{\sqrt{2}}\,(|00\rangle + |11\rangle).$$

Після дії на кубіт потрібним унітарним оператором передаємо кубіт з пункта А в пункт В. Одержувачу залишається з'ясувати, дії якого оператора піддавався присланий кубіт і отримати 2-х бітове повідомлення. Для цього в пункті В застосуємо до цих кубітів операцію $\widehat{\text{CNOT}}$ і після цього проведемо вимірювання кубіту, що знаходився в пункті В (див. Рис. 7. 21).

Якщо вимірювання визначає стан $|0\rangle$, то стан кубітів належить одному з пари станів $\frac{1}{\sqrt{2}}\,(|00\rangle \pm |11\rangle)$, якщо прилад реєструє стан $|1\rangle$, то стан кубітів один із двох станів $\frac{1}{\sqrt{2}}\,(|10\rangle \pm |01\rangle)$. Дійсно, розглянемо дію $\widehat{\text{CNOT}}$ на беллівські базові стани

$$\widehat{CNOT}\left(\frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right)\right) = \frac{1}{\sqrt{2}}\left(|00\rangle \pm |10\rangle\right),$$
$$\widehat{CNOT}\left(\frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right)\right) = \frac{1}{\sqrt{2}}\left(|01\rangle \pm |11\rangle\right).$$

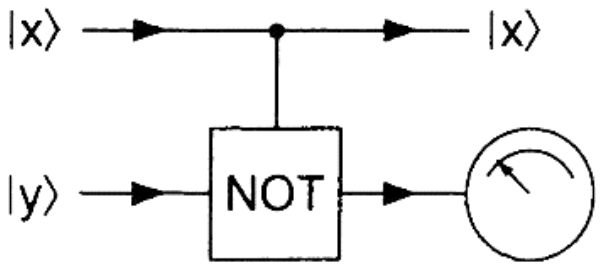


Рис. 7. 21. Схема, що дозволяє по вимірюванню другого кубіту визначити, чи належить їх стан парі станів $\frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right)$ або парі $\frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right)$. Процес вимірювання кубіту символічно позначений приладом з шкалою і стрілкою

З результату стає ясним попереднє твердження. Тепер на кубіт, що ще не підданий вимірюванню, подіємо оператором Адамара і проведемо вимірювання (див. Рис.7. 22).

Стан кубіту, що прибув, після вимірювання кубіту, що знаходився в пункті *B*, залежить від знаку суперпозиції і буде в одному з двох станів $\frac{1}{\sqrt{2}}\left(|0\rangle \pm |1\rangle\right)$.

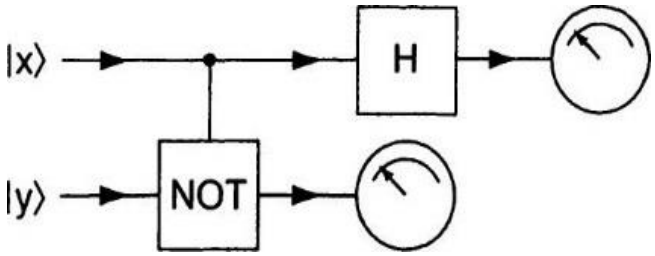


Рис. 7.22. Визначення станів кубітів

Дію оператора Адамара на ці стани легко проаналізувати:

$$\widehat{H}\ \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = \frac{1}{2}\left(|0\rangle + |1\rangle + |0\rangle - |1\rangle\right) = |0\rangle,$$
$$\widehat{H}\ \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \frac{1}{2}\left(|0\rangle + |1\rangle - |0\rangle + |1\rangle\right) = |1\rangle.$$

Таким чином, вимірювання стану цього кубіту дозволяє встановити знак суперпозиції і остаточно встановити, який з чотирьох операторів застосовувався в пункті *A*. Тим самим отримати двобітне повідомлення при передачі тільки одного кубіту. Цей метод дістав назву щільного кодування. Сенс такої назви ясний. Для передачі 2 бітів досить передати 1 кубіт. Цікаво відмітити ще одну важливу властивість щільного кодування. Це підвищений захист інформації, що передається. Дійсно, для того, щоб прочитати інформацію, що передається,

недостатньо перехопити кубіт, що пересилається, треба ще отримати доступ до другого кубіту, що не пересилається.

7.9 Квантові машини Т'юринга

Наслідуючи Шора, обговоримо простий варіант квантової машини Т'юринга. Роль комірок стрічки виконують тепер кубіти. Над ними розташовується голівка квантової машини Т'юринга [9]. Робота машини розпочинається з установки кубітів в певний початковий стан. Звичайно цей стан $|0, \dots, 0\rangle$ кінцевого числа кубітів. Голівка машини переміщається і діє згідно з класичною програмою. Один такт роботи складається з наступних дій (див. Рис. 7.23):

1. Згідно класичної програми машина бере один або два кубіти.
2. Діє на них однокубітним унітарним оператором $\hat{U}_\theta$ або двокубітним $\hat{U}_{XOR}$ відповідно.
3. Повертає кубіти на попередні місця в новому стані.

Повторюючи багаторазово такі цикли, голівка машини виконує приписи програми. Після зупинки (якщо вона відбувається) проводиться вимірювання кінцевого стану кубітів, що і є результатом квантових обчислень. Особливо слід підкреслити, що усі здійснювані нею дії не можуть грунтуватися на знанні стану якого-небудь кубіту. Для цього знання треба провести вимірювання стану кубіту в процесі роботи машини. Така дія привела б до безповоротної зміни стану кубіту і порушенню процесу обчислень. Тому багато умовних циклів безпосередньо не можуть виконуватися на квантовій машині Т'юринга. Важливе питання, яке слід обговорити, це: чи дають квантові обчислення економію ресурсів в порівнянні з класичними обчисленнями?

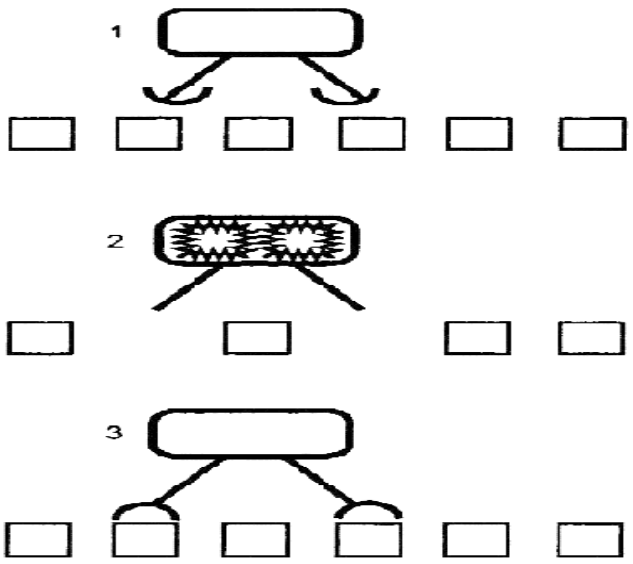


Рис. 7.23. Показані стадії виконання квантовою машиною Т'юринга одного такту з двома кубітами. На першій стадії вибирається пара кубітів. На другій стадії здійснюється їх взаємодія і стани міняються під дією унітарного оператора. На третій стадії кубіти повертаються на свої місця

7.10 Квантовий алгоритм пошуку в несортованій базі даних

Одним з добре відомих класичних алгоритмів є алгоритм пошуку деякого обраного елемента з великого набору *N* елементів. Цей алгоритм входить в якості підпрограми у велетенське число різноманітних програм. З фізичної точки зору це відповідає пошуку білої кулі серед *N* - 1 чорних куль, що лежать в урні. Зрозуміло, не можна підглядати, а можна тільки виймати кулі по черзі і тоді розглядати їх, визначаючи, вийняли потрібну кулю або ні. Біла куля буде знайдена класичним алгоритмом з вірогідністю 1/2 після *N*/2 спроб. Чи існує алгоритм пошуку, що дозволяє знайти потрібний елемент за менше число спроб? Несподівано виявилось, що такий алгоритм існує. Саме такий, але вже квантовий алгоритм пошуку був запропонований Гровером. Важливість існування такого алгоритму в тому, що він демонструє переваги квантових обчислень над класичними. Тому обговоримо цей алгоритм детальніше.

Нехай серед $N = 2^n$ елементів треба вибрати один. Розпочнемо з того, що кожному елементу x_i зіставимо певний стан *n* кубітів виду $|\dots,0,0,\dots,1,0,1,\dots,1\rangle$. Таких станів рівно $N = 2^n$. Тому одному елементу можна зіставити рівно один стан. Тоді завдання зводиться до пошуку одного стану, що відповідає шуканому елементу серед 2^n станів. Для того, щоб можна було виділити шуканий елемент від інших елементів x_i , $i \neq 0$, він повинен відрізнятися від них. Іншими словами, шуканий елемент повинен мати властивість, що відрізняє його від інших елементів. У фізичному прикладі це був колір кулі. З нормальної точки зору існування такої властивості означає існування функції *C* (*x*), такої, що C (x_0) = 1, а C (x_i) = 0, якщо $i \neq 0$. Тоді можна припустити, що можна побудувати квантову схему, яка в стані розрізняти шуканий стан кубітів від інших станів. Якщо не вдаватися до внутрішнього устрою такої схеми, то її дію можна зображувати, як на рис. 7.24. Важливо відмітити, що дія цієї схеми оборотна і, отже, такий унітарний оператор може бути побудований. Часто така схема називається оракулом.

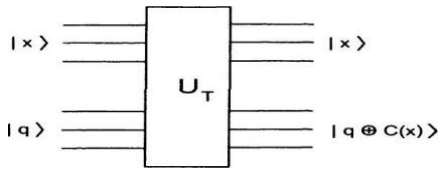


Рис. 7.24. Квантова схема, яка здатна відрізнити
шуканий стан від інших станів

Таким чином, дія цієї квантової схеми на базисні стани описується як

$$\widehat{U}_T|x\rangle|q\rangle = |x\rangle|q \oplus C(x)\rangle,$$

де $|x\rangle$ – стан "верхніх" кубітів, а $|q\rangle$ – стан кубітів схеми, що перевіряє. Знак $\oplus$, як і раніше, означає додавання за модулем 2. Легко помітити, що якщо подіяти схемою $\widehat{U}_T$ на стан $|x_0\rangle|0\rangle$, то схема переведе кубіт оракула в стан $|1\rangle$. Проте дію оракула можна привести до простішої і зручнішої форми. Для цього розглянемо

дію цієї схеми, якщо кубіт оракула знаходиться в стані суперпозиції виду $\frac{1}{\sqrt{2}}$ ($|0\rangle$— $|1\rangle$). Дія $\widehat{U}_T$ на такий стан кубітів зводиться до наступного виразу:

$$\widehat{U}_T|x\rangle \left(\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)\right) = (-1)^{C(x)}|x\rangle \left(\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)\right).$$

Найважливіше спостереження з цього рівняння, що стан кубітів оракула не міняється при функціонуванні цієї схеми. Це означає, що можна не стежити за станами кубітів оракула і звести дію цієї схеми до наступної:

$$\widehat{U}_T|x\rangle = (-1)^{C(x)}|x\rangle. \quad (7.36)$$

Слід підкреслити, що ця схема не здійснює пошук як такий. Її роль зводиться тільки до розпізнавання властивостей станів кубітів, що пред'являються їй. Якщо повернутися до класичного прикладу з кулями, то її роль зводиться до можливості відрізнити чорний колір кулі від білого.

Тепер приведемо блок-схему алгоритму Гровера, що здійснює пошук потрібного елемента в неструктурованій базі даних (див. Рис. 7.25).

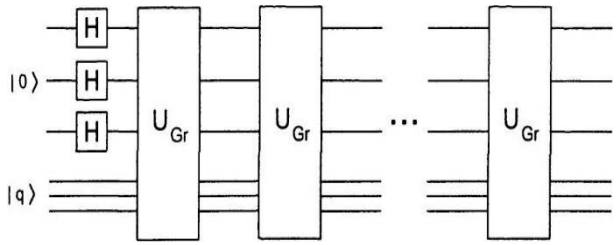


Рис. 7.25. Блок- схема квантового алгоритму пошуку.

На вході усі верхні регістри знаходяться в стані $|0\rangle$. Під дією операторів Адамара їх стан переводиться в стан однорідної суперпозиції. Далі застосовується приблизно $\sqrt{N}$ разів унітарний оператор Гровера $\widehat{U}_{gr}$. Нижні регістри відіграють допоміжну роль як робочий простір оракула.

На вхід цієї схеми подаються кубіти в нульових станах $|0\rangle$. Подальша дія на них операторів Адамара приводить їх стан в стан однорідної суперпозиції:

$$|\varphi\rangle = \frac{1}{\sqrt{N}} \sum_0^{N-1}|x\rangle.$$

Залишається розглянути облаштування квантової схеми, що відповідає дії унітарного оператора $\widehat{U}_{gr}$. Ця квантова схема приведена на рис.7.26.

Дія оракула, що входить в цю схему, описана вище. Тому залишається описати дію унітарного оператора $\widehat{U}_R$, що входить в цю схему. Його дія зводиться до зсуву фази відповідно до правил:

$$\widehat{U}_R|0\rangle = |0\rangle,$$

$$\widehat{U}_R|x\rangle = -|x\rangle, \text{ якщо } x \neq 0.$$

Цей оператор легко записати в термінах проєкційних операторів як

$$\widehat{U}_R = 2|0\rangle\langle 0| - I,$$

де I - одиничний оператор, що не міняє стан. Таким чином, схема оператора Гровера влаштована досить просто. Легко зрозуміти, що її можна також записати, використовуючи формалізм проєкційних операторів у виді:

$$\widehat{U}_{gr} = (2|\varphi\rangle\langle\varphi| - I) \tag{7.37}$$

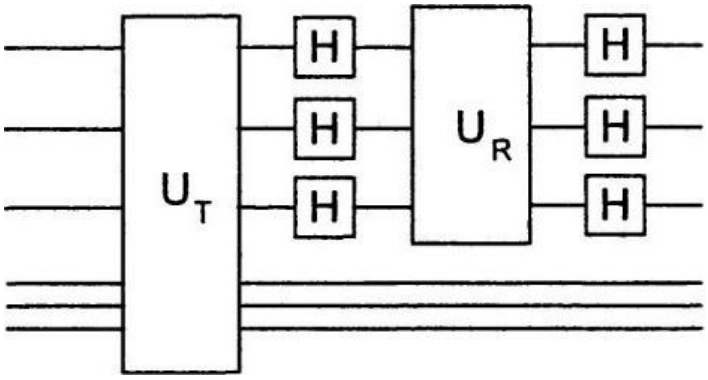


Рис. 7. 26. Квантова схема, що пояснює
"мікроскопічний" пристрій оператора Гровера $\widehat{U}_{gr}$

Тепер розглянемо процес здійснення пошуку квантовим алгоритмом. Припустимо для простоти, що шуканий стан тільки один – $|x_0\rangle$. Для розуміння дії алгоритму зручно перейти до простих геометричних представлень. Для цього представимо стан $|\varphi\rangle$ у вигляді суми двох одиничних векторів. Вектора

$$|\alpha\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle,$$

де підсумовування виконується по усіх станах, за винятком шуканого стану при пошуку. Другий вектор, в нашому випадку, це шуканий вектор

$$|\beta\rangle = |x_0\rangle.$$

У більш загальному випадку, якщо шукається декілька векторів, то вектор β вибирається у вигляді однорідної суперпозиції шуканих станів. Таким чином,

$$|\varphi\rangle = \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle + \frac{1}{\sqrt{N}} |\beta\rangle. \tag{7.38}$$

У такому представленні вектор $|\varphi\rangle$ розташований в площині, “натягнутій” на вектори $|\alpha\rangle$ і $|\beta\rangle$. Амплітуди при цих векторах можна записати в термінах кута між вектором $|\varphi\rangle$ і напрямом, наприклад, вектора $|\alpha\rangle$ (див. Рис. 7.27).

$$|\varphi\rangle = \cos\left(\frac{\theta}{2}\right) |\alpha\rangle + \sin\left(\frac{\theta}{2}\right) |\beta\rangle . \tag{7.39}$$

Для аналізу дії оператора $\widehat{U}_{qr}$, згідно з рівнянням (7.37), розглянемо спочатку дію оракула (7.36) на стан (7.38). Легко зрозуміти, що

$$\widehat{U}_T|\varphi\rangle = \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle - \frac{1}{\sqrt{N}} |\beta\rangle.$$

У такій дії оператора легко узнати відображення вектора $|\varphi\rangle$ відносно осі $|\alpha\rangle$. В термінах кута результат дії $\widehat{U}_T$ зводиться до заміни $\theta \rightarrow -\theta$.

$$\widehat{U}_T|\varphi\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle - \sin\left(\frac{\theta}{2}\right)|\beta\rangle$$

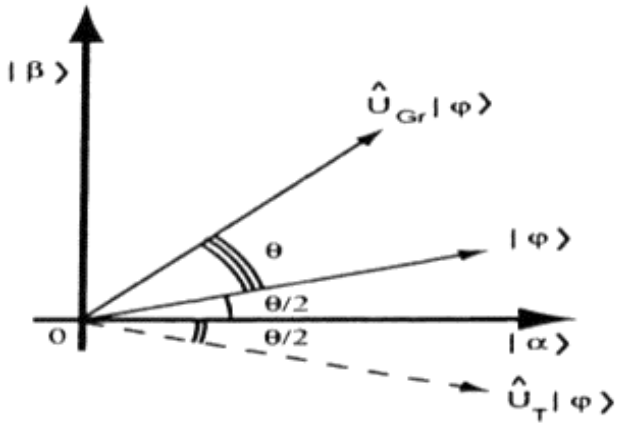


Рис. 7.27. Геометричні побудови, які ілюструють дію операторів $\widehat{U}_T$ і $\widehat{U}_{qr}$ на початковий вектор $|\varphi\rangle$ в площині, "натягнутій" на вектори $|\alpha\rangle$ і $|\beta\rangle$

Тепер подіємо на отриманий стан кубітів оператором $\widehat{U}_{gr} = (2|\varphi\rangle\langle\varphi| - I)$. Дія цього оператора на довільний вектор зводиться до його відображення відносно осі, що задається вектором станів $|\varphi\rangle$ [10]. З рис. 7.27 легко помітити, що таке відображення можна замінити поворотом початкового вектора на кут θ.

$$(2|\varphi\rangle\langle\varphi| - I)\left(\cos\left(\frac{\theta}{2}\right)|\alpha\rangle - \sin\left(\frac{\theta}{2}\right)|\beta\rangle\right) =$$

$$= \left(\cos\left(\frac{3\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{3\theta}{2}\right)|\beta\rangle\right).$$

Це означає, що (див. співвідношення (7.37))

$$\widehat{U}_{gr}|\varphi\rangle = \left(\cos\left(\frac{3\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{3\theta}{2}\right)|\beta\rangle\right). \qquad (7.40)$$

Таким чином, в площині, "натягнутій" на вектори $|\alpha\rangle$ і $|\beta\rangle$, дія оператора Гровера на вектор однорідної суперпозиції $|\varphi\rangle$ зводиться до повороту його на кут θ у напрямі осі $|\beta\rangle$ [10]. Це означає, що амплітуди, що раніше були на усіх базисних станах однакові, в результаті застосування оператора $\widehat{U}_{gr}$ втратили таку рівноправність. В силу повороту у напрямі осі $|\beta\rangle$ амлітуда стану трохи збільшилася (див. Рис.7. 28).

Ясно, що якщо повторити дію оператора $\widehat{U}_{gr}$ m разів, то результат такої дії наступний:

$$\widehat{U}_{gr}^{m}|\varphi\rangle = \left(\cos\left(\frac{2m+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2m+1}{2}\theta\right)|\beta\rangle\right). \qquad (7.41)$$

З'ясуємо тепер, за скільки дій оператора Гровера вектор повернеться у напрямі осі $|\beta\rangle$. Для цього оцінимо кут θ. Повертаючись до співвідношення (7.39) і порівнюючи його із співвідношенням (7.38), легко отримати

$$\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}.$$

Враховуючи, що в задачі пошуку інтерес представляє ситуація, коли $N \gg 1$ і тому вектор спрямований практично уздовж осі $|\alpha\rangle$. Це означає малу величину кута $\theta \ll 1$ і для оцінки використовуємо

$$\sin\left(\frac{\theta}{2}\right) \approx \frac{\theta}{2} = \frac{1}{\sqrt{N}}.$$

Отже, можна знайти значення m, при якому кут $\frac{2m+1}{2}\,\theta$ стане порядку $\frac{\pi}{2}$ і результуючий вектор стану буде спрямований уздовж осі $|\beta\rangle$. Прирівнюючи

$$\frac{2m+1}{2}\,\theta \approx \frac{\pi}{2}$$

і підставивши значення θ, отримаємо

$$\frac{2m+1}{2}\,2 \approx \frac{\pi\sqrt{N}}{2}.$$

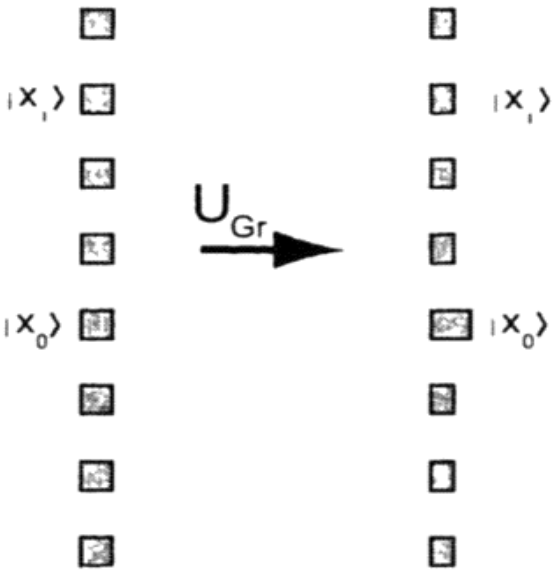


Рис. 7.28. Символічно показані зміни амплітуд базисних станів, що входять у вигляді рівноправної суперпозиції, під впливом оператора $\widehat{U}_{gr}$. Видно, що збільшилася амплітуда шуканого стану $|x_0\rangle$ і зменшилися амплітуди інших станів $|x_i\rangle$, $i \neq 0$. Величина амплітуди відповідає ширині прямокутника

Після проведення елементарних перетворень і нехтуючи одиницею в порівнянні з $2m$, знайдемо

$$m \approx \frac{\pi\sqrt{N}}{4}.$$

Це означає, що після приблизно $\frac{\pi\sqrt{N}}{4}$ застосувань оператора Гровера амплітуда стану $|\beta\rangle = |x_0\rangle$ стане близькою до одиниці, а амплітуди при інших базисних станах стануть дуже малі (див. Рис. 7.29).

Тепер для того, щоб знайти шуканий стан, досить просто провести вимірювання на виході квантової схеми алгоритму пошуку (див. Рис. 7.25). Оскільки вірогідність виявити відповідний базисний елемент дорівнює квадрату модуля амплітуди при ньому, то стан $|x_0\rangle$ буде виявлений з вірогідністю, близькою до одиниці. Вірогідність при вимірюванні отримати інший стан близька до нуля.

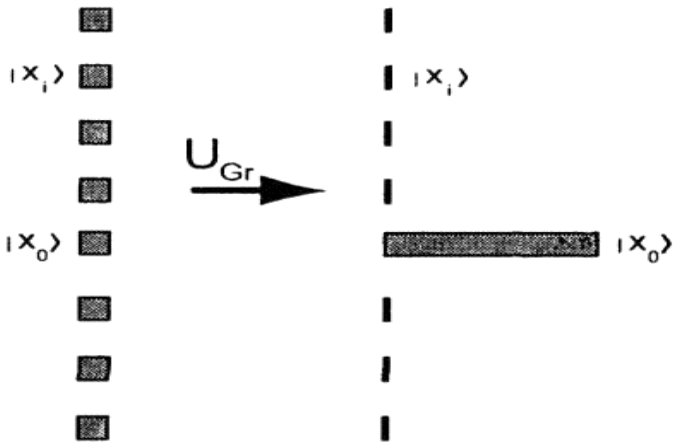


Рис. 7. 29. Символічно показані зміни амлітуд базисних станів при $\frac{\pi\sqrt{N}}{4}$ - кратному застосуванні оператора Гровера. Ясно, що знайти стан $|x_0\rangle$ при такій відмінності в амплітудах стає простою задачою

Після обговорення функціонування квантової схеми пошуку самим неясним питанням залишається: як можна було придумати такий алгоритм? Для того, щоб зрозуміти, як, виходячи з простих фізичних міркувань, можна прийти до цього алгоритму, звернемося до квантової задачі переходу з одного стану в інший деякої квантової системи. Основне питання в такій постановці зводиться до пошуку гамільтоніану системи, еволюція якої, розпочинаючись з вектора стану $|\varphi\rangle$, виявлялася в деякий момент часу в стані $|x_0\rangle$. Ясно, що бажано побудувати оператор Гамільтона, використовуючи тільки ці стани і не залучаючи зайвих сутностей. Давайте в якості відповідного гамільтоніана виберемо

$$\hat{H} = |x_0\rangle\langle x_0| + |\varphi\rangle\langle\varphi|. \qquad (7.42)$$

Оператор еволюції, згідно квантової механіки, є унітарним оператором виду (див. підрозділ 7.2.5):

$$\hat{U} = e^{-i\hat{H}t}.$$

Тепер питання зводиться до існування такого часу *t*, при якому

$$|x_0\rangle = e^{-i\hat{H}t}\,|\varphi\rangle.$$

Розглянемо, що відбувається при малих значеннях часу *t*. Використовуючи розкладання експоненти в ряд Тейлора, отримаємо наближений еволюційний оператор:

$$e^{-i\hat{H}t}\,|\varphi\rangle \approx (\hat{I} - i\hat{H}t)\,|\varphi\rangle.$$

Спробуємо зрозуміти, як еволюціонує вектор стану $|\varphi\rangle$ на малих значеннях часу. Для цього перетворимо праву частину попереднього рівняння

$$(\hat{I} - i\hat{H}t)\,|\varphi\rangle = (1 - it)|\varphi\rangle - it\,\langle x_0|\varphi|x_0\rangle. \qquad (7.43)$$

Звідси можна зробити висновок, що на малих значеннях часу *t* вектор $|\varphi\rangle$ злегка повертається у бік вектору станів $|x_0\rangle$. В цьому легко переконатися, оцінюючи зміну амплітуд базисних станів. Для аналізу унітарного оператора еволюції на довільних значеннях часу розглянемо його детальніше. Розпочнемо з уведення ортогонального базису, по якому можна розкласти стан $|\varphi\rangle$. Нехай:

$$|\varphi\rangle = a|x_0\rangle + b\,|\phi\rangle,$$

де амплітуди *a* і *b* для простоти вважатимемо дійсними і задовольняючими умові нормування $a^2 + b^2 = 1$. В силу ортогональності $\langle x_0|\phi\rangle = \langle\phi|x_0\rangle = 0$. Переходячи до представлення цих базисних векторів у вигляді стовпців

$$|\phi\rangle = \begin{pmatrix}0\\1\end{pmatrix},$$

$$|x_0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$$

і вивчаючи дію оператора $\hat{H}$ на ці базисні вектори, легко отримати матричне представлення цього оператора виду

$$\hat{H} = \begin{pmatrix}1 + a^2 & ab\\ ab & 1 - a^2\end{pmatrix} \equiv I + a(a\hat{\sigma}_z + b\hat{\sigma}_x).$$

Тут $\hat{\sigma}_x\,\hat{\sigma}_z$ - матриці Паулі (див. розділ 7.2.6). Таким чином, оператор еволюції можна записати як

$$e^{-i\hat{H}t} = e^{-i(I + a(a\hat{\sigma}_z + b\,\hat{\sigma}_x))t} = e^{-iIt}\,e^{-ia(a\hat{\sigma}_z + b\hat{\sigma}_x)t}.$$

При отриманні цього виразу враховано, що одиничний оператор комутує з усіма операторами, що входять в гамільтоніан. Ясно, що загальну зміну фази квантового стану враховувати не потрібно. Тому запишемо цей оператор у виді:

$$e^{-i\hat{H}t} = \cos\big(at(a\hat{\sigma}_z + b\hat{\sigma}_x)\big) - i\sin\big(at\,(a\hat{\sigma}_z + b\hat{\sigma}_x)\big).$$

Для розуміння подальшого нагадаємо розкладання в ряд Тейлора синуса і косинуса:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

Важливо підкреслити, що розкладання косинуса ведеться тільки по парних степенях, а синуса тільки по непарних. Розглянемо тепер просту степінь оператора $at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)$:

$$\left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right)^2 = \left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right)\left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right) =$$
$$= (at)^2(a^2\hat{\sigma}_z\hat{\sigma}_z + ab\hat{\sigma}_z\hat{\sigma}_x + ab\hat{\sigma}_x\hat{\sigma}_z + b^2\hat{\sigma}_x\hat{\sigma}_x).$$

Підкреслимо, що матриці Паулі $\hat{\sigma}_x$ і $\hat{\sigma}_z$ не комутують і тому міняти їх порядок у вище приведеному добутку не можна.

Врахуємо властивості матриць Паулі, що легко перевіряються, $\hat{\sigma}_x^2 = \hat{\sigma}_z^2 = \hat{I}$ і $\hat{\sigma}_x\hat{\sigma}_z + \hat{\sigma}_z\hat{\sigma}_x = 0$ (див. в розділі 7.2.6 співвідношення (7.24) і (7.25)) і запишемо остаточний вираз:

$$\left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right)^2 = (at)^2(a^2 + b^2)\hat{I} = (at)^2\hat{I}\,.$$

Ясно, що усі парні степені цього оператора пропорційні одиничному оператору і, отже,

$$\cos\left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right) = \cos(at)\hat{I}\,.$$

З цих же міркувань ясно, що усі непарні степені, що входять в розкладання синуса, пропорційні першій степені оператора $(a\hat{\sigma}_z + b\hat{\sigma}_x)$ і, отже,

$$\sin\left(at\ (a\hat{\sigma}_z + b\hat{\sigma}_x)\right) = \sin(at)(a\hat{\sigma}_z + b\hat{\sigma}_x).$$

Остаточно отримаємо:

$$e^{-i\hat{H}t} = \cos(at)\hat{I} - i\sin(at)(a\hat{\sigma}_z + b\hat{\sigma}_x).$$

Тепер розглянемо еволюцію стану $|\varphi\rangle$ під дією цього оператора

$$e^{-i\hat{H}t}|\varphi\rangle = \cos(at)|\varphi\rangle - i\sin(at)(a\hat{\sigma}_z + b\hat{\sigma}_x)\,|\varphi\rangle\,.$$

Використовуючи розкладання $|\varphi\rangle$ по ортогональному базису, легко довести, що $(a\hat{\sigma}_z + b\hat{\sigma}_x)\,|\varphi\rangle = |x_0\rangle$ і тому

$$e^{-i\hat{H}t}|\varphi\rangle = \cos(at)|\varphi\rangle - i\sin(at)|x_0\rangle.$$

Ясно, що на протязі часу, при якому $at \approx \pi/2$, амплітуда стану $|x_0\rangle$ стане близькою до одиниці, а амплітуди інших станів зменшаться до нуля. Оцінку швидкості кута повороту a здійснимо за станом однорідної суперпозиції станів:

$$|\varphi\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle\,.$$

Тоді швидкість кута повороту $a \approx \langle x_0|\varphi\rangle = \frac{1}{\sqrt{N}}$ (див. рівняння (7.43)). Отже, якщо провести вимірювання системи вчасно $t = \frac{\pi\sqrt{N}}{2}$, то з вірогідністю 1 виявимо шуканий стан $|x_0\rangle$. Залишається зрозуміти, як по відомому оператору еволюції або гамільтоніану побудувати квантову схему.

Таким чином, використовуючи чудовий квантовий алгоритм Гровера, можна знайти елемент $|x_0\rangle$ з набору *N* елементів за $\sqrt{N}$ кроків. В порівнянні з класичним алгоритмом пошуку, приблизно за *N/2* кроків, перевага досить помітна.

Нині алгоритм Гровера один з небагатьох квантових алгоритмів, що демонструють переваги квантових алгоритмів в порівнянні з класичними. Цікаво обдумати, чи можна, знаючи цей алгоритм, витягнути білу кулю з урни за таке число проб? Відомий також інший чудовий приклад квантового алгоритму, отриманий Шором, який в експоненціальне число разів прискорює класичну задачу факторизації числа.

7.11 Квантовий алгоритм факторизації чисел Шора

Розглянутий раніше квантовий алгоритм пошуку призводить усього лише до незначної економії ресурсів. Значно більшого успіху можна досягти в квантовому алгоритмі факторизації чисел, запропонованому Шором. З точки зору класичних алгоритмів задача розкладання числа на прості множники є складним завданням, що вимагає експоненціальне число ресурсів. Зараз вважається, що для факторизації *n*-бітового цілого числа вимагається $\boldsymbol{exp\ (n^{\frac{1}{3}} log^{\frac{2}{3}} n)}$ операцій. При виконанні квантового алгоритму Шора на це завдання потрібно всього $\boldsymbol{O\ (n^2 \log n \log\log n)}$ операцій. Такий квантовий алгоритм призводить до експоненціальної економії ресурсів. Ця демонстрація переваг квантового алгоритму в порівнянні з класичними змушує до детальнішого його обговорення [26].

Розпочнемо з обговорення задачі факторизації числа *N* на прості множники. Виберемо тепер ціле число *x*. Припустимо, що *N* і *x* взаємно прості числа. Використовуючи алгоритм Евкліда, це досить легко перевірити. Якщо це не так, то можна повторити вибір. Розглянемо тепер, як поводиться послідовність чисел $\boldsymbol{x}$ **mod (***N***),** $\boldsymbol{x^2}$ **mod (***N***),** $\boldsymbol{x^3}$ **mod (***N***), … .** Ці співвідношення цілком аналогічні використовуваним раніше рівностям по **mod 2.** У більш загальному випадку вираз $\boldsymbol{x}$ **mod (***N***)** дорівнює залишку від ділення *x* на число *N*. Ясно, що якщо такий залишок дорівнює нулю, то *x* ділиться на *N*. Монотонно наростаюча послідовність

$$1, x, x^2, x^3, \dots, x^n, \dots$$

після виконання операції модуля mod *N* перетворюється на періодичну послідовність

$$\underbrace{1, x, \dots}_{k}, \underbrace{1, x, \dots}_{k}, \underbrace{1, x, \dots}_{k}, \dots$$

Число *k* називається порядком за модулем *N*. Легко помітити, що період цієї послідовності дорівнює мінімальній степені, при якій $\boldsymbol{x^k}$**mod (***N***) = 1.** Існування такого *k* в теорії чисел відомо з теореми Ейлера. Припустимо, що, використовуючи деякий алгоритм, знайдемо період *k*. Нехай він буде парним

числом. Якщо це не так, то можна вибрати інше x і повторити пошук періоду. Якщо період k парний, то рівняння

$$x^k \bmod (N) = 1$$

можна записати як

$$\left(x^{\frac{k}{2}} - 1\right)\left(x^{\frac{k}{2}} + 1\right) = 0 \bmod (N).$$

Це означає, що один з двох співмножників лівої частини кратний числу N. Таким чином, залишається знайти найбільший спільний дільник (НСД) чисел $\left(x^{\frac{k}{2}} - 1, N\right)$ і $\left(x^{\frac{k}{2}} + 1, N\right)$. Серед цих чисел і буде один або два множники, на які розкладається початкове число N. Цей класичний алгоритм по суті зводить проблему факторизації числа до пошуку періоду функції $t \rightarrow x^t \bmod (N)$.

Приклад 7.16. *Розглянемо в якості N = 119 і x = 8. Спробуємо розв'язати проблему факторизації N = 119 приведеним вище алгоритмом. Нижче приведемо таблицю, в якій в першому рядку виписаний номер члена послідовності або аргумент функції* $t \rightarrow x^t$ *mod (N), а в нижньому рядку відповідне значення цієї функції* x^t *mod (N).*

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
x^t mod (N)	1	8	64	36	50	43	10	615	1	8	64	36	50	43

З приведеної таблиці легко визначити порядок 8 за модулем 119 або період цієї послідовності. Ясно, що $\boldsymbol{k = i}$*, відповідно, парне. Тому треба знайти найбільший спільний дільник (НСД) чисел (*$8^4 - 1, 119) = (4095, 119)$ *і (*$8^4 + 1, 119) = (4097, 119)$*. Виконуючи це завдання, знаходимо, що НСД (4О95, 119) = 7, а НСД (4097, 119) = 17. Таким чином, знайдені навіть два множники, на які розкладається число 119 – це числа 7 і 17. Виконуючи ділення на ці множники* $\frac{119}{7} = 17$*, легко переконатися в цьому. Числа 7 і 17 прості і, отже, 119 = 7 х 17. На цьому і закінчується факторизація числа 119.*

Таким чином, якщо ми хочемо запропонувати ефективний квантовий алгоритм факторизації чисел, то треба запропонувати квантовий алгоритм пошуку періоду заданої послідовності чисел. В принципі для вирішення такої проблеми можна використати Фур'є – перетворення. Ясно, що в спектрі розкладання періодичної послідовності в ряд Фур'є будуть сплески на частотах, кратних основній частоті. Наявність яскраво виражених сплесків істотно спрощує завдання пошуку періоду. Треба тільки переконатися, чи буде відповідний квантовий алгоритм дійсно ефективним в порівнянні з класичним. Такий погляд на проблему факторизації корисний ще і тим, що попутно буде обговорене квантове перетворення Фур'є. Роль цього перетворення у фізиці і математиці важко переоцінити. Передусім слід зрозуміти, що ми розумітимемо під квантовим перетворенням Фур'є. Зрозуміло, далі говоритимемо про дискретне перетворення Фур'є. Таке перетворення переводить вектор

початкових даних $(x_0, x_1, x_2, \dots, x_j, \dots \dots, x_{N-1})$ у вектор перетворених даних $(y_1, y_2, \dots, y_k, \dots, y_{N-1})$ згідно з наступним правилом:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{N}} x_j.$$

Тут i - уявна одиниця, а число $k = 0, 1, 2, \dots, N - 1$. Слід нагадати, що числа x_j і y_k можуть бути комплексними. Далі зручно вважати, що число $N = 2^n$. Наслідуючи загальну ідею узагальнення класичних обчислень, визначимо квантове дискретне Фур'є-перетворення як унітарний оператор, що діє на базисних станах аналогічно класичному визначенню. Тоді визначимо дію унітарного оператора $\widehat{U}_{FF}$ на базисний стан n кубітів $|j\rangle$ $(0 \leq j \leq 2^n - 1)$ як:

$$\widehat{U}_{FF}|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle. \qquad (7.44)$$

Для того, щоб побачити близькість цих визначень, розглянемо дію цього оператора на суперпозицію станів:

$$\widehat{U}_{FF} \sum_{j=0}^{N-1} x_j |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{N}} x_j \right) |k\rangle = \sum_{k=0}^{N-1} y_k |k\rangle.$$

Легко помітити повну відповідність з класичним дискретним перетворенням Фур'є. Дійсно, амплітуди y_k співпадають з дискретним перетворенням Фур'є величин x_j. Квантове перетворення Фур'є (7.44) можна представити в мультиплікативній формі, яка виявляється зручнішою для подальшого викладу. Для виведення цієї форми запишемо рівняння (7.44) як

$$\widehat{U}_{FF}|j\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{2^n}} |k\rangle =$$

Тепер використовуємо для номерації базисних векторів двійкове представлення чисел $k = k_1 2^{n-1} + k_2 2^{n-2} + \dots + k_n 2^0$, яке записують у бінарній формі як $k = k_1\, k_2 \dots k_n$. Далі нам знадобиться і представлення у вигляді двійкового дробу $k = \frac{k_l}{2} + \frac{k_{l+1}}{2^2} + \dots + \frac{k_m}{2^{m-l+1}}$ із записом у бінарному представленні у вигляді $0.\, k_l k_{l+1} \dots \dots k_m$. Коефіцієнти в цих розкладаннях можуть набувати тільки два значення $k_l = 0{,}1$. Використовуючи ці форми запису, перетворимо попередній вираз до вигляду

$$= \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^{1} \dots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{l=1}^{n} k_l 2^{-l}\right)} |k_1 \dots k_n\rangle =$$

Суму в експоненті запишемо як добуток

$$= \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^{1} \dots \sum_{k_n=0}^{1} \prod_{l=1}^{n} e^{2\pi i j\, k_l 2^{-l}} |k_l\rangle =$$

Тут також використано $|k_1 \dots k_n\rangle = |k_1\rangle \dots |k_n\rangle$. Використовуючи прості алгебраїчні властивості, перетворимо далі

$$= \frac{1}{2^{\frac{n}{2}}} \prod_{l=1}^{n} \left(\sum_{k_l=0}^{1} e^{2\pi i j k_l\, 2^{-l}} |k_l\rangle \right) = \frac{1}{2^{\frac{n}{2}}} \prod_{l=1}^{n} \left(|0\rangle + e^{2\pi i j\, 2^{-l}} |1\rangle \right).$$

Остаточно отримуємо перетворення Фур'є у вигляді добутку

$$\widehat{U}_{FF}|j_1 \dots j_n\rangle = \frac{1}{2^{n/2}} \left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right) \cdot \left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right) \dots$$
$$\dots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle\right). \quad (7.45)$$

Це представлення дискретного перетворення Фур'є дозволило досить просто встановити квантову схему для його обчислення, яка приведена на рис. 7.30 [11, 12]. На цій схемі однокубітний гейт R_m визначається унітарною матрицею

$$R_m = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^m}} \end{pmatrix}.$$

Безпосередня перевірка дії цієї схеми переконує, що вона здійснює квантове дискретне перетворення Фур'є після здійснення операцій обміну між кубітами, які не вказані на рис. 7.30. Нагадаємо, що схема обміну між двома кубітами містить 3 гейти $\widehat{CNOT}$ (див. підрозділ 7.4 і рис. 7.18). Використовуючи вид цієї схеми, можна оцінити ефективність здійснення перетворення Фур'є квантовим алгоритмом. Для цього оцінимо число гейтів, що входять в квантову схему. Число гейтів, що діють на перший кубіт *n*, треба скласти з числом гейтів, що діють на другий кубіт *n* - 1, і додавши інші гейти, отримаємо суму *n* + (*n* - 1) + (*n* - 2) +…+ 1 = (*n* + 1) *n* /2. Число гейтів для здійснення обміну пропорційно *n* і не змінює порядок числа гейтів n^2 квантового перетворення Фур'є. Для порівняння нагадаємо, що класичні алгоритми дискретного перетворення Фур'є вимагають значно більшого числа операцій порядка $n2^n$. У цьому сенсі квантове перетворення Фур'є ефективніше, ніж класичне – можна сказати експоненціально ефективніше. Зрозуміло, використання цієї ефективності являється не зовсім простою проблемою. Покажемо, що це можливо для пошуку періоду заданої функції.

Таким чином, перейдемо до обговорення алгоритму пошуку періоду деякої періодичної функції.

$$f(x+l) = f(x).$$

Ця рівність виконується для усіх *x* і *l* періоду цієї функції.

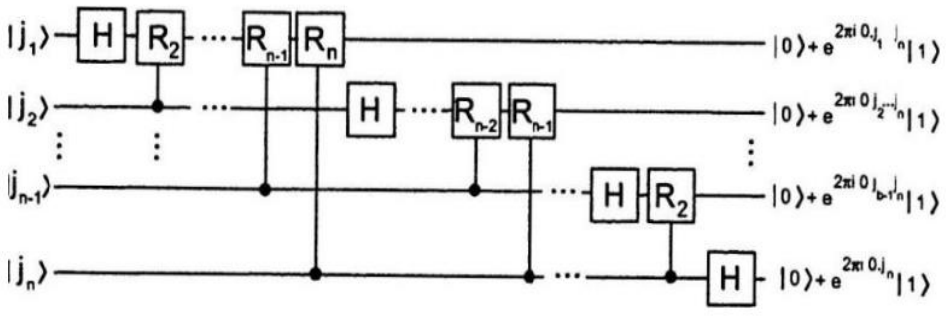


Рис. 7.30. Схема, що ефективно обчислює квантове перетворення Фур'є. На виході цієї схеми для скорочення упущені гейти, що здійснюють обмін між кубітами для обертання слідування кубітів

Наше завдання побудувати ефективний алгоритм пошуку періоду *l*. Для побудови такого алгоритму використаємо ідею квантового паралелізму і приведемо кубіти в стан однорідної суперпозиції виду (див. розділ 7.6).

$$|f\rangle = \frac{1}{N}\sum_{x=0}^{N-1}|x\rangle\ |f(x)\rangle.$$

Цей стан містить інформацію про багатьох властивостях функції $f(x)$ і про її період також. Головне питання: як витягнути дані про період? Тут ми стикаємося із звичайною проблемою використання квантового паралелізму. Дійсно, вимірювання другого регістру дасть деяке число y_m. Тоді стан однорідної суперпозиції сколапсує до станів першого регістру, що призводять до цих значень $y_m = \mathrm{f}\,(x_0 + ml)$. Тут m - довільне ціле число (див. Рис. 7.31). Такі стани відрізняються на числа, кратні періоду. Іншими словами, вони призводять до станів

$$|\varphi\rangle = \frac{1}{M}\sum_{m=0}^{M-1}|x_0 + ml\rangle\,.$$

Число $0 \le x_0 < l$ можна розглядати як випадково вибране на періоді функції. Такий випадковий вибір був здійснений при вимірюванні інших регістрів. Тому вимірювання перших регістрів вже не дасть ніякої інформації про період функції, а тільки повідомить нас про величину випадкового зрушення x_0.

Для того, щоб обійти цю складність, застосуємо до стану $|\varphi\rangle$ алгоритм квантового перетворення Фур'є. Після такого перетворення картина зміниться і відповідні зміни показані на рис. 7.32. Найважливіше спостереження полягає в зникненні зрушення. Тепер зробимо вимірювання. В результаті буде виміряно значення $g = p\frac{M}{l}$, кратне *l*.

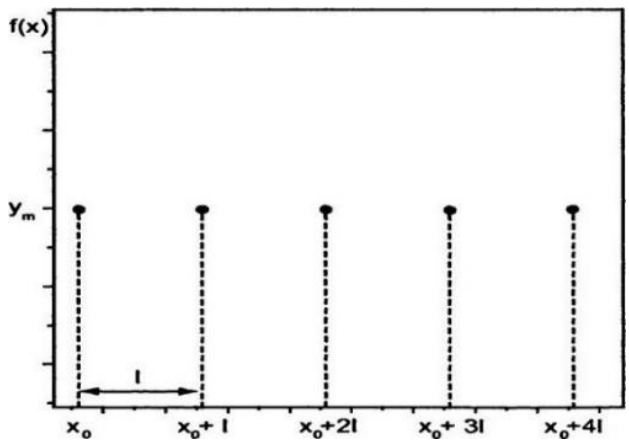


Рис. 7.31. Показано виміряне значення функції. Також зображено значення $\boldsymbol{x_0}$, що виникло в результаті вимірювання і зрушенні на *l* однакових значень функції

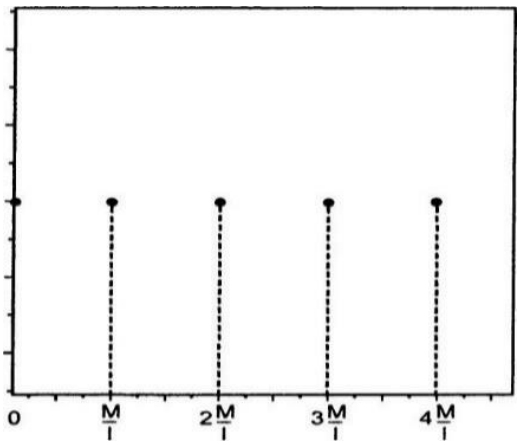


Рис. 7.32. Легко помітити зникнення зрушення після Фур'є – перетворення і періодичність з новим періодом *M / l*

Зрозуміло, числа g і M відомі, а p і l не відомі. Запишемо це співвідношення у виді $\frac{g}{M} = \frac{p}{l}$. Припустимо, що ціле число p, яке випадково з'явилося, взаємно просте з періодом l. Тоді період l можна знайти після проведення можливих скорочень в дробі $\frac{g}{M}$. Вірогідність того, що p взаємно просте з l , можна оцінити. Оцінки цієї вірогідності обгрунтовані на деяких фактах теорії чисел. Для нас важливо тільки, що вона не перевищує $\frac{1}{log_2 l}$. Таким чином, при повторенні цього алгоритму близько O $(log_2 M)$ період l може бути знайдений з будь-якою точністю, що цікавить нас. Використання цього алгоритму і завершує алгоритм Шора. Слід зазначити, що квантовий формалізм перетворення Фур'є є найважливішою складовою більшості ефективних квантових алгоритмів, відомих на сьогодні.

Детально квантовий алгоритм Гровера, квантовий алгоритм факторизації чисел Шора та квантове дискретне перетворення Фур'є розглянуті авторами в наступних розділах.

7. 12 Квантовий комп'ютер

У цьому підрозділі обговоримо принципову схему квантового комп'ютера. Незалежно від фізичної реалізації елементів цієї схеми загальні риси квантового комп'ютера в принципі досить просто описати. Інша справа, як втілити цей загальний опис в реальних фізичних системах. Із загальної точки зору квантовий комп'ютер - це фізичний пристрій, здатний виконувати логічні операції над квантовими станами. Логічним операціям відповідають унітарні перетворення, що не порушують квантову суперпозицію в процесі обчислень. Розглянемо принципи побудови квантових комп'ютерів більш детально.

1. Ясно, що передусім в просторі квантового комп'ютера повинен існувати великий набір кубітів. Роль кубітів можуть відіграти квантові дворівневі системи. Важливо, щоб такі кубіти були добре ізольовані один від одного і від оточуючого середовища.

Приклад 7.17. *Існує багато пропозицій по фізичній реалізації кубітів. Це окремі атоми або іони, молекули з декількома різними ядерними спінами і множина інших фізичних об'єктів. Не маючи будь-якої можливості детально зупинитися хоч би на частині з них, обмежимося обговоренням тільки одного. Зупинимо вибір на оптичному фотоні. Фотони - квантові нейтральні "частки". До переваг такого об'єкту можна віднести слабку взаємодію фотонів один з одним і з довкіллям, можливість поширення на великі відстані. Як представити собі кубіт на основі фотонів? Для кубіту на основі поодинокого фотона в якості двобазисних станів годяться два стани фотона з протилежною поляризацією. Існує і інша можливість представлення кубіту, що називається подвійним представленням. Давайте розглянемо пару резонаторів, що мають двократно вироджену фотонну моду з частотою ω. Це означає, що базисні стани такого кубіту відповідають стану* $|01\rangle$ *- фотон знаходиться в першому резонаторі і* $|10\rangle$ *– фотон знаходиться в другому резонаторі. Загальний стан кубіту - це суперпозиція базисних станів* $\alpha|01\rangle + \beta|10\rangle$.

2. В квантовому комп'ютері повинна передбачатися можливість ініціалізації n кубітів вхідного регістра. Під ініціалізацією мається на увазі переведення цих кубітів у базисний стан $\underbrace{|0,0,\dots,0,0\rangle}_{n}$. Це означає, що квантовий комп'ютер має бути вбудований в класичне оточення, що дозволяє управляти станами і еволюцією квантових кубітів і здійснювати вимірювання їх станів.

Приклад 7.18. *Для вирішення цієї проблеми необхідно з'ясувати, як генерувати поодинокі фотони експериментально? Це необхідно для створення великого набору кубітів. В принципі, найбільш простим способом генерації поодиноких фотонів є послаблення лазерного випромінювання. Іншим способом, можливо, більш ефективним, являється використання нелінійних оптичних середовищ, в яких можливі процеси параметричного розпаду. Зокрема, до таких середовищ відноситься* KH_2PO_4. *Такий процес розпаду супроводжується збереженням енергії і імпульсу. Так, падаюче випромінювання частоти ω на таке середовище приводить до генерації фотонних пар з частотами* $\omega_1 + \omega_2 = \omega$ *і хвильовими векторами* $\vec{k}_1 + \vec{k}_2 = \vec{k}$. *Таким чином, якщо детектор зареєструє фотон частоти* ω_1, *то в системі є одиничний фотон частоти* ω_2. *Тому, якщо встановити на виході селектор, що відкривається при реєстрації одного фотона частоти* ω_1, *отримаємо джерело одиничних фотонів (див. Рис.7.33). Нанотехнології поставляють і інші типи джерел одиничних фотонів. Так, цікавим джерелом поодиноких фотонів може виступати квантова точка з InP, що знаходиться на вершині конуса. Конус з діаметром вершини декілька мікрон витравлюють з підкладки GaAs. Після цього за допомогою молекулярно-променевої епітаксії вирощують шари GaInP і тонкий шар InP, який і збирається в єдину квантову точку, потім цю точку покривають захисним шаром з GaInP. Було виявлено, що така структура надійно випромінює якраз поодинокі фотони.*



Рис. 7.33. Символічно зображено облаштування
джерела поодиноких фотонів

3. Кубіти повинні бути керованими, тобто, повинна бути передбачена процедура дій на окремий кубіт або на пари кубітів. Таку процедуру дій треба робити для виконання квантових алгоритмів. В принципі, число таких керованих кубітів повинно перевищувати 10^3 для більш-менш повноцінного квантового комп'ютера.

Приклад 7.19. *Обговоримо фізичні можливості в управлінні станами фотонів і їх еволюцією. Виявляється, що довільні квантові гейти можуть бути сконструйовані з фазообертача, світлодільника і дзеркала. Розпочнемо з опису облаштування фазообертача. Він є прозорою пластиною, коефіцієнт заломлення якої n відрізняється від коефіцієнта заломлення вакууму* n_0. *При проходженні фотона через таку пластинку завтовшки l його фаза міняється на* e^{ikl}. *Враховуючи, що хвильовий вектор k = nw/c (c - швидкість світла у вакуумі), отримаємо зміну фази* $e^{\frac{inl\omega}{c}}$. *Таким чином, зрушення фази фотона, що пройшов через пластинку, відносно того, що рухається у вакуумі, буде* $e^{\frac{i(n-n_0)\,l\,w}{c}}$.

Світлодільник влаштований дещо складніше (див. Рис. 7.34). Зазвичай він виготовляється з двох призм з частково посрібленим або металевим шаром між ними. Цей шар має коефіцієнт відображення R і коефіцієнт проходження T = 1 - R. Фазообертачів і світлодільників достатньо для створення довільного унітарного оператора, що діє на поодинокий кубіт. Проте, потрібні гейти, що діють і на декілька кубітів. З останнім елементом, середовищем Керра, і пов'язаний, мабуть, головний недолік представлення кубітів оптичними фотонами.

З нелінійної оптики відомо, що існують середовища, коефіцієнт заломлення n (I) яких залежить від інтенсивності I

$$n\,(I) = n_1 + n_2 \cdot I.$$

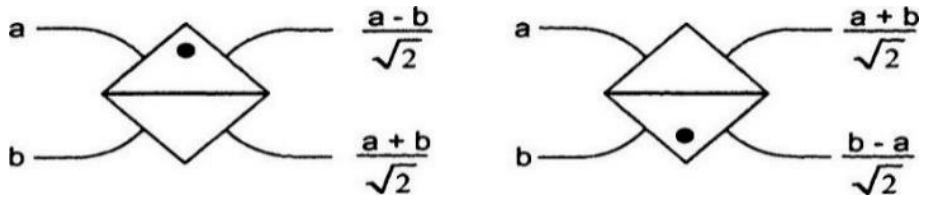


Рис. 7.34. Оптичний світлодільник. Шар металу, що розділяє призми, показаний жирною лінією по середині. Вхідні і вихідні канали відповідають R = 0.5. Зображені світлодільники є взаємооборотними

Наявність такої залежності називається ефектом Керра. Роль цього середовища полягає в забезпеченні взаємодії між фотонами. Саме це потрібно для створення квантового гейта $\widehat{\text{CNOT}}$. На жаль, для звичайних середовищ n_2 занадто мало. Так, для легованого скла $n_2 = 10^{-7} \div 10^{-14}$ (см^2/ Вт), для напівпровідників $n_2 = 10^2 \div 10^{-10}$ (см^2/Вт). Враховуючи обговорюваний раніше базис квантових гейтів, ясно, що фазообертачів, світлодільників і середовища Керра достатньо для їх побудови і, отже, для створення, в крайньому випадку теоретично, будь-яких квантових схем.

4. У квантовому комп'ютері має бути передбачена боротьба з перешкодами або шумами. Іншими словами, необхідно забезпечити гарний захист від довкілля. Помилка при виконанні однієї квантової операції не повинна перевищувати 10^{-4}. Така помилка вважається прийнятною для квантового комп'ютера з використанням кодів, що коригують помилки. При спробі задовольнити цим вимогам стикаємося з проблемою декогеренції. Ця проблема полягає в розпаді суперпозиційних квантових станів і перетворенні їх в просту суміш. Власне, з цим процесом і пов'язано пояснення парадоксу кота Шредінгера. Ясно, що час обчислень на квантовому комп'ютері має бути багато менше часу декогеренції. Тому треба або ізолювати квантовий комп'ютер від довкілля або знайти можливість штучного збільшення часу збереження когерентності квантових станів.

Приклад 7.20. *У певному значенні фотони і, отже, кубіти на їх основі добре ізольовані один від одного і довкілля. Як відзначалося вище, це і приводить до головної проблеми при здійсненні квантових обчислень на оптичному квантовому комп'ютері.*

5. Нарешті, треба забезпечити можливість вимірювання квантових станів вихідних кубітів. Врешті-решт проблема вимірювання квантових станів є незмінним атрибутом квантових комп'ютерів.

Приклад 7.21. *Експериментальна техніка реєстрації окремих фотонів добре розвинена і продовжує інтенсивно розвиватися. У оптичному діапазоні зазвичай використовуються детектори, які грунтуються на фотоелектричному ефекті. Такі детектори повинні володіти рядом важливих властивостей. Так, недосконалості детекторів, що зменшують вірогідність детектування окремого фотона, мають бути малими. Кількісною характеристикою цього являється квантова ефективність - вірогідність народження в детекторі поодиноким фотоном фотопари носіїв заряду, що*

дають вклад у фотострум. Величина квантової ефективності апріорі задовольняє нерівності $0 \leq \eta \leq 1$ *і характеризує якість детектора. Окрім цього, детектори характеризуються смугою пропускання, рівнем шуму і "темновим" струмом. "Темновий" струм пов'язаний з генерацією фотопар навіть за відсутності фотонів. Для детектування поодиноких фотонів використовуються фотоперемножувачі, лавинні фотодіоди, надпровідні детектори. Останні базуються на використанні наявності критичного струму, використовують також ефект Джозефсона, андріївське відображення і інші ефекти. Так, існують надпровідникові однофотонні детектори для реєстрації одиничних фотонів інфрачервоного (ІЧ) діапазону, які мають ряд переваг перед детекторами аналогічного діапазону, такими як ФЕП та ін.*

Таким чином, якщо реалізація оптичного квантового комп'ютера і не виглядає багатообіцяючою, то його обговорення дозволило встановити загальну архітектуру квантових комп'ютерів. Інші квантові комп'ютери, які базуються на інших фізичних елементах, структурно являються цілком схожі на розглянутий вище зразок. Необхідно зазначити, що ідеї, що лежать в основі оптичного квантового комп'ютера, у будь-якому випадку виявляться виключно корисними в проблемах квантової інформатики, квантової криптографії і інших галузей науки і техніки.

8 КВАНТОВА ТЕЛЕПОРТАЦІЯ

Тема цього розділу послужила джерелом натхнення для багатьох релігійних і фантастичних художніх творів. Телепортація – здатність "миттєвого" переміщення в просторі настільки здавалася неможливою, що давно привертала пильну увагу і живий інтерес. Досить легко представити, як радикально її реалізація повинна вплинути на світ, що оточує нас. Звичайний транспорт втратить своє значення і стане непотрібним, війни або змінять якісно свою тактику або стануть і зовсім безглуздими, навіть космічні подорожі стануть набагато доступніші, не кажучи вже про екзотичні подорожі у відпускний період. У арсенал багатьох фокусників входить "телепортація" монет, кроликів і навіть слонів. Зрозуміло, спостереження за такими фокусами ще не переконує в існуванні телепортації. Один з перших романів, присвячених телепортації, був написаний Артуром Конан Дойлем під назвою "Дезінтеграційна машина". Сюжет цього роману грунтований на зустрічі професора Челленджера з винахідником машини, здатної розкласти людину на атоми, а потім зібрати її у іншому місці. Жахнувшись наслідками цього винаходу для людства, професор Челленджер дезінтегрує винахідника, "забуваючи" його зібрати. Інший сучасний приклад звернення до теми телепортації дає кінематограф. Досить згадати популярний фільм "Муха", не кажучи вже про серіал "Зоряний шлях". Популярним сюжетом багатьох фантастичних романів є збільшення числа об'єктів, що телепортуються, або суб'єктів. Як ми обговоримо нижче, з теореми про неможливість клонування невідомих станів слідує обнадійливий висновок про сумнівність такої події. Потішно, що тема телепортації нещодавно перекочувала з художньої в наукову літературу. Досягненням дослідників в цій області і присвячений цей розділ.

Одна із можливих схем квантової телепортації розглянута авторами вище (див. підрозділ 5.3).

Розглянемо наступні види телепортації.

8.1 Телепортація квантового стану по класичному каналу

Нечасто в науковій літературі з'являється можливість супроводжувати виклад наукового факту цікавою вигаданою історією. З цієї нагоди розпочнемо з проблеми, що виникла у деякого шахрая і його подруги. Припустимо, що квантовий ключ від сейфа з капіталами шахрая переданий на зберігання його подрузі. Квантовий ключ - це деякий спеціальний квантовий стан кубіту. Після деякого часу діяльність цього шахрая була викрита. Він вимушений ховатися від караючих органів. Зрозуміло, такий процес супроводжується великими фінансовими витратами і шахраю потрібний ключ від сейфа, де гроші лежать. Ясно, що безпосередній зв'язок з подругою неможливий із-за ретельного спостереження за нею з боку, скажімо, поліції. Єдиний спосіб зв'язатися з подругою по телефону. Чи зможе вона передати квантовий ключ по телефону?

Іншими словами, передати квантовий стан по класичному каналу зв'язку. Ясно, що ключ – це стан кубіту, наприклад,

$$|\varphi\rangle = \alpha|0\rangle + \beta\,|1\rangle,$$

де амплітуди α і β невідомі подрузі. Цей розумний захід обережності заздалегідь був вжитий шахраєм. Зрозуміло, дізнатися їх подруга не може. Будь-яка спроба упізнати ключ означає проведення вимірювання. Проте, при будь-якому вимірюванні можна виявити стан $|0\rangle$ або стан $|1\rangle$ і при цьому відбудеться редукція або колапс вектору станів і повторне вимірювання стане безглуздим. Більше того, квантовий ключ буде безповоротно загублений. Тому упізнати амплітуди α і β не можна. Спроба створити багато копій квантового ключа і провести вимірювання над копіями також приречена на провал. Клонування невідомого квантового стану, згідно раніше обговорюваній теоремі (див. розділ 7.7), теж не можливо. Як бачимо, виникає серйозна проблема. Все-таки персонажі можуть виконати таку передачу завдяки передбачливості шахрая. Задовго до виникнення неприємностей він приготував пару кубітів, що знаходяться в переплутаному стані, скажемо,

$$|\varphi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Після цього один кубіт цієї пари був переданий його подрузі, а другий незмінно знаходився з ним. Після такої підготовчої роботи проблема стає вирішуваною. Опишемо дії подруги для виконання такого відповідального завдання (див. Рис. 8.1). Класичний канал зв’язку позначений подвійними лініями. Два верхніх кубіти знаходяться у розпорядженні подруги, нижній - у шахрая. Результати вимірювання n і m можуть бути 0 або 1. Залежно від цих значень $\hat{\sigma}^0 = \hat{I}$ або $\hat{\sigma}^1 = \hat{\sigma}$ будуть застосовуватися чи ні відповідні матриці Паулі.

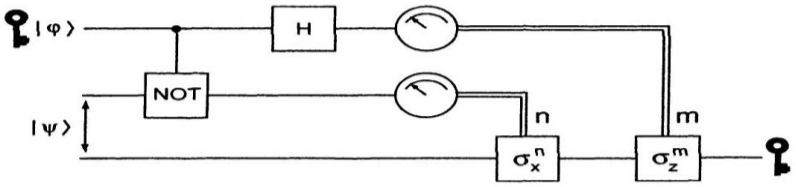


Рис. 8.1. Квантова схема здійснення телепортації

Передусім в її розпорядженні знаходяться два кубіти - ключ і представник переплутаної пари. До вимірювання стани кубітів були

$$|\varphi\rangle\,|\psi\rangle = (\alpha|0\rangle + \beta\,|1\rangle)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) =$$

$$= \frac{1}{\sqrt{2}}\alpha|0\rangle|00\rangle + \frac{1}{\sqrt{2}}\alpha|0\rangle|11\rangle + \beta|1\rangle|00\rangle + \beta|1\rangle|11\rangle.$$

Дія гейта $\widehat{\mathrm{CNOT}}$ на перших два кубіти переводить цей стан в

$$\widehat{\mathrm{CNOT}}\,(12)|\varphi\rangle\,|\psi\rangle =$$

$$= \frac{1}{\sqrt{2}}\,|00\rangle\alpha|0\rangle + \frac{1}{\sqrt{2}}\,|01\rangle\alpha|1\rangle + \frac{1}{\sqrt{2}}\,|11\rangle\beta|0\rangle + \frac{1}{\sqrt{2}}\,|10\rangle\beta|1\rangle.$$

Далі на стан першого кубіту діє оператор Адамара, в результаті стан переходить в

$$\widehat{H}(1)\widehat{CNOT}\,(12)\,|\varphi\rangle\,|\psi\rangle = \frac{1}{\sqrt{2}}(\,\widehat{H}|0\rangle)|0\rangle\alpha|0\rangle + \frac{1}{\sqrt{2}}\,(\widehat{H}|0\rangle)|1\rangle\alpha|1\rangle +$$

$$+\frac{1}{\sqrt{2}}\big(\,\widehat{H}|1\rangle\big)|1\rangle\beta|0\rangle\; +\frac{1}{\sqrt{2}}\big(\,\widehat{H}|1\rangle\big)|0\rangle\beta|1\rangle =$$

$$=\frac{1}{2}\,(|0\rangle+|1\rangle)|0\rangle\alpha|0\rangle +\frac{1}{2}\,(|0\rangle+|1\rangle)|1\rangle\alpha|1\rangle +$$

$$+\frac{1}{2}\,(|0\rangle-|1\rangle)|1\rangle\beta|0\rangle +\frac{1}{2}\,(|0\rangle-|1\rangle)|0\rangle\beta|1\rangle =$$

Після виконання елементарних перетворень отримаємо

$$=\frac{1}{2}|00\rangle(\alpha|0\rangle+\;\beta|1\rangle)+\frac{1}{2}\;|10\rangle(\alpha|0\rangle-\;\beta|1\rangle)+$$

$$+\frac{1}{2}|01\rangle(\beta|0\rangle+\;\alpha|1\rangle)+\frac{1}{2}|11\rangle(-\beta|0\rangle+\;\alpha|1\rangle)=$$

Тепер подруга проводить вимірювання станів перших двох кубітів. Дивлячись на попередній вираз, легко побачити чотири можливих результатів вимірювань. Це може бути 00, 01, 10 або 11. В результаті вимірювань стан третього кубіту $|\phi\rangle$, що знаходиться у шахрая, колапсує в один з чотирьох станів. Який саме виникає стан, залежить від результатів вимірювань подруги. Так,

якщо виміряно 00, то стан редукує в $|\phi\rangle = \alpha|0\rangle + \;\beta|1\rangle$,

якщо виміряно 01, то стан редукує в $|\phi\rangle = \beta|0\rangle + \;\alpha|1\rangle$,

якщо виміряно 10, то стан редукує в $|\phi\rangle = \alpha|0\rangle - \;\beta|1\rangle$,

якщо виміряно 11, то стан редукує в $|\phi\rangle = \alpha|1\rangle - \beta|0\rangle$.

При телефонному дзвінку шахрай отримує інформацію, який саме результат вимірювань був отриманий. В залежності від інформації, що поступила, він діє на свій кубіт наступними квантовими гейтами (див. Рис. 8.1)

якщо 00, то застосовується $\hat{I}\,|\phi\rangle = \alpha|0\rangle + \;\beta|1\rangle = |\varphi\rangle$,

якщо 01, то $\hat{\sigma}_x|\phi\rangle = \hat{\sigma}_x(\beta|0\rangle + \;\alpha|1\rangle) = \;\alpha|0\rangle + \;\beta|1\rangle = |\varphi\rangle$,

якщо 10, то $\hat{\sigma}_z|\phi\rangle = \alpha\hat{\sigma}_z|0\rangle - \;\beta\hat{\sigma}_z|1\rangle = \alpha|0\rangle + \;\beta|1\rangle = |\varphi\rangle$,

якщо 11, то $\hat{\sigma}_z\hat{\sigma}_x|\phi\rangle = \alpha\hat{\sigma}_z\hat{\sigma}_x|1\rangle - \;\beta\hat{\sigma}_z\hat{\sigma}_x|0\rangle \;= |\varphi\rangle$.

Легко бачити, що за даними вимірювання, переданими по класичному каналу зв’язку, шахраєві вдається телепортувати невідомий квантовий стан з міста, де знаходиться його подруга, в місце, де він ховається. Точніше, телепортацію здійснює його подруга. Відповідно до правил жанру, цей стан зник в одному місці і з'явився в іншому, дуже віддаленому пункті. Така схема телепортації була запропонована в роботах C.H Bennetta [13]. Декілька важливих додаткових спостережень можна вилучити з цього процесу. Стан $|\varphi\rangle$, що телепортується, при вимірюванні в початковому пункті зникає і, відповідно, не виникає протиріч з теоремою про неможливість клонування невідомого квантового стану. Наступне спостереження у відсутності принципового обмеження на відстань телепортації квантового стану. Це дуже цікаво. Важливо тільки зберегти переплутаність пари станів при рознесенні їх на велику відстань

один від одного. Ясно, що перевищити швидкість поширення світла при телепортації не вдасться із-за необхідності в передачі інформації по класичному каналу. Тому телепортація зі швидкістю, що перевищує швидкість світла, тема тільки фантастичних творів, як і поява дублів при телепортації.

Легко бачити, що телепортується не кубіт, а квантовий стан. Так, якщо кубіти зчепленої пари мають іншу фізичну природу, чим кубіт, що містить “ключ», то стан переноситься на інший об'єкт. Наприклад, якщо перший кубіт – це фотон, а зчеплені спінові стани – стани ядер, то стан фотона переноситься на спіновий стан ядра.

Окрім цього, експериментальна перевірка телепортації породжує ще один тест, який може спростувати квантову механіку. Нагадаємо, що будь – яку науку можна спростувати одним контрприкладом або експериментом. На превеликий жаль, навіть велике кінцеве число експериментів не може довести її правильність. У цьому ховається джерело надій і розчарувань для багатьох дослідників.

8.2 Експериментальна телепортація

Важливість квантової фізики стимулювала ряд експериментальних перевірок можливості телепортації по класичних каналах зв'язку. Одним з перших експериментів була телепортація квантового стану фотона. Схема експерименту приведена на рис. 8.2 [14]. Основним елементом цієї експериментальної установки є джерело (N) фотонних пар в зчепленому стані. Діє це джерело таким чином. Ультрафіолетовим лазером збуджується фемтосекундний імпульс, який падає на нелінійний кристал (N). У нелінійному кристалі з квадратичною нелінійністю можливий процес спонтанного параметричного розпаду фотона. Тому ультрафіолетовий фотон розпадається на два фотони. Енергії цих "червоних" фотонів приблизно однакові. Закон збереження енергії і імпульсу для фотонів дає:

$$\overline{h}\,\omega_p = \overline{h}\,\omega_1 + \overline{h}\,\omega_2\ ,$$

$$\overline{h}\ \vec{k}_p = \overline{h}\ \vec{k}_1 + \overline{h}\ \vec{k}_2 .$$

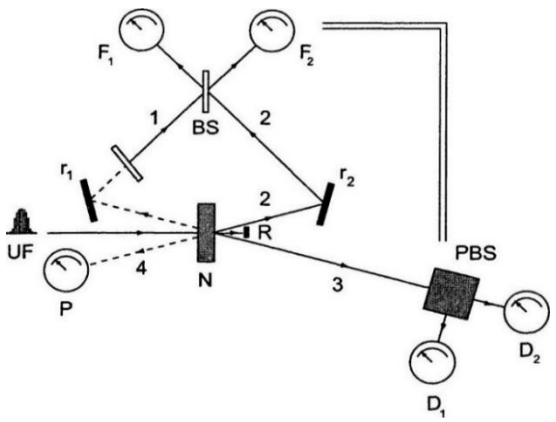


Рис. 8.2. Умовна схема експерименту по телепортації.
Класичний канал зв'язку показаний подвійною лінією

Слід зауважити, що окрім квадратичної нелінійності, кристал повинен мати і іншу важливу властивість. У його середовищі повинен виконуватися спеціальний синхронізм типу II. Такі властивості мають кристали, наприклад, β – борату барія. У такому кристалі, завдяки двопроменезаломленню, фотони, що виникли при розпаді, утворюють "зв'язані" пари фотонів. Один з цих фотонів звичайно поляризований, а другий – незвично поляризований фотон. Фотони таких типів створюють веселку колірних конусів (див. Рис.8.3), які мають неспівпадаючі вісі. Синхронізм фотонів в кристалах, що призводить до таких явищ, відноситься до другого типу II. При синхронізмі першого типу I розсіяні фотони лінійно поляризовані і їх конуси соосні. Таким чином, нелінійний кристал може служити джерелом переплутаних станів. Дійсно, в напрямах перетину конусів (див. Рис. 8.3) поляризація фотонів не визначена. Єдине, що можна стверджувати, що вона протилежна. Ці фотони і утворюють переплутані стани. Значить, падаючи на нелінійний кристал (N), лазерний імпульс при розсіянні (див. Рис. 8.2) утворює пару (2, 3) беллівських або переплутаних станів. Відбитий промінь від дзеркала R при зворотному проходженні формує і іншу пару (1,4). Фотон 1 є носієм того стану, який і буде телепортований. Фотон 4 далі буде використаний тільки для контролю за появою фотона 1. За допомогою дзеркал r_1 і r_2 фотон 1 і фотон 2 зводяться на світлодільнику (BS). Після інтерференції фотонів на світлодільнику проводяться вимірювання детекторами F_1 і F_2 . По класичному каналу передається інформація про співпадіння відліків детекторів F_1 і F_2 . Пояснимо, про що говорить таке співпадіння. Неважко перевірити, що спрацьовування цих детекторів означає, що фотони 1 і 2 знаходяться в антисиметричному стані

$$|\Psi_-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) .$$

Тому спрацьовування детекторів F_1 і F_2 означає, що фотони 1 і 2 знаходяться в стані $|\Psi_-\rangle$. Фізична причина цього пов'язана з симетрією беллівських або переплутаних станів. Дійсно, інші переплутані стани симетричні відносно перестановки кубітів і тільки стан $|\Psi_-\rangle$ антисиметричний. Проте, слід пам'ятати, що фотони є бозонами (див. Підрозділ 7.2.3). Тому вектор станів $|\Psi_-\rangle$ повинен бути антисиметричним відносно просторової перестановки координат кубітів для збереження його симетричності. Інші три заплутані стани симетричні відносно просторової перестановки. Тепер слід зрозуміти, який результат дасть інтерференція фотонів на дільнику (BS) (див. Рис.8.4). З аналізу інтерференції виходить, що два детектори спрацюють тільки на просторово антисиметричному векторі станів.

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}|0,0\rangle + \frac{1}{\sqrt{2}}|1,1\rangle ,$$
$$|\Phi_-\rangle = \frac{1}{\sqrt{2}}|0,0\rangle - \frac{1}{\sqrt{2}}|1,1\rangle ,$$
$$|\Psi_+\rangle = \frac{1}{\sqrt{2}}|0,1\rangle + \frac{1}{\sqrt{2}}|1,0\rangle .$$

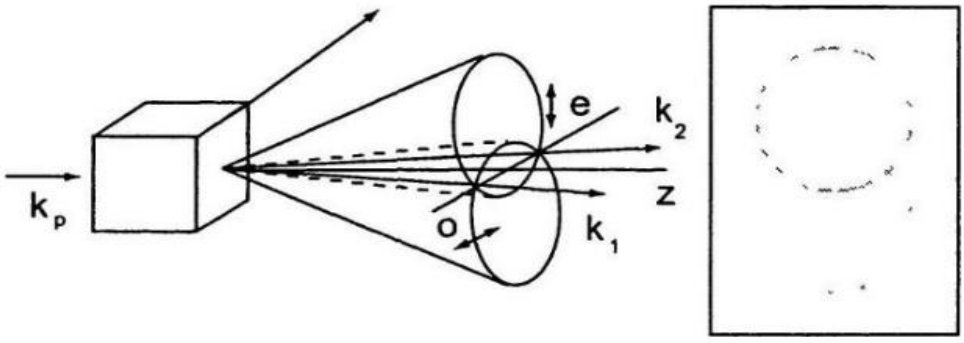


Рис. 8.3. Нелінійний кристал типу II. Справа показана картина перерізу колірних конусів, пропущених через вузькосмуговий фільтр. В точках перетину звичайного і незвичайного конусів поляризація фотонів не визначена. Саме у цих напрямах поширюються поляризаційно скоректовані пари фотонів

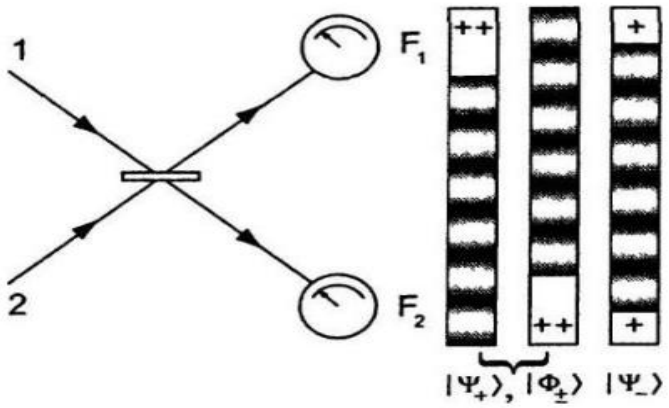


Рис. 8.4. Ілюстрація спрацьовування детекторів F_1 і F_2 на різних переплутаних станах. Інтерференція станів символічно показана справа, де і відмічені детектори, що спрацьовують

Таким чином, в експерименті телепортувався фотон 1 з 45^0 поляризацією. Телепортація відбувалася, коли пара фотонів 1 і 2 знаходилася в стані $|\Psi_-\rangle$. При вимірюванні детекторами F_1 і F_2 відбувається колапс вектору станів і фотон 3 повинен знаходитися в стані з 45^0 поляризацією. Для визначення цієї події використовувався поляризаційний дільник (PBS), що розділяє поляризації під + 45^0 і -45^0.

Ясно, що телепортація реєструється, якщо при одночасному спрацьовуванні детекторів F_1 і F_2 спрацьовує детектор D_2, що реєструє поляризацію + 45^0 і не спрацьовує детектор D_1, що реєструє поляризацію -45^0.

Результати експерименту (див. Рис.8.5) переконливо продемонстрували, що квантова телепортація реалізовується. Ліворуч теоретичні передбачення співпадінь потрійних відліків детекторами F_1 і F_2 і зверху D_1, а знизу D_2. Телепортація стану з + 45^0 поляризацією ідентифікується по провалу детектування D_1 фотонів з поляризацією – 45^0 і постійному рівню детектування D_2 стану з + 45^0 . Справа експериментальні дані по вимірюванню співпадінь

відліків (F_1 , F_2 і D_1) зверху і (F_1 , F_2 і D_2) знизу. Вертикальна ось - число співпадінь при тривалості спостережень 4000 сек. Горизонтальні вісі - час затримки між прибуттям фотона 1 і 2 на світлодільник.

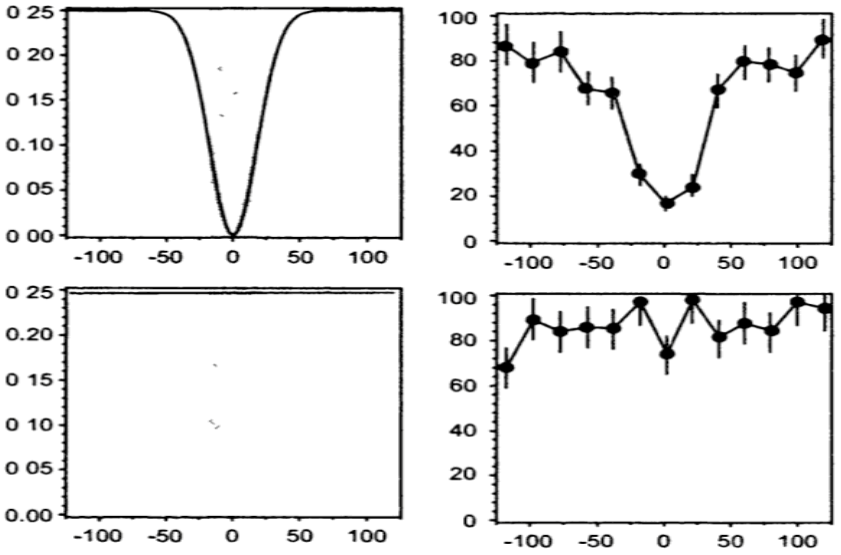


Рис. 8.5. Результати експерименту квантової телепортації

Зрозуміло, що в цьому розділі розглянута тільки частина найпростіших експериментальних результатів. Експерименти по телепортації фотонів і навіть атомів тривають. У 2006 році був досягнутий великий прогрес в цьому напрямі. Фізикам вдалося телепортувати не стан мікроскопічного об'єкту, а стан макроскопічного. Їм вдалося переплутати стани газу атомів цезію з фотонами і потім телепортувати, використовуючи фотони, стан початкового газу атомів цезію, що віддалені на відстань 0.5 метра від пункту відправки. Ці експерименти викликають великий інтерес не лише з прикладної точки зору, як можливість передачі станів на великі відстані, що є важливим елементом самих квантових комп'ютерів. Окрім цього, телепортація, що реалізовується, дає можливість по новому подивитися на фундаментальні проблеми передачі інформації у біологічних і молекулярних системах.

8.3 Квантова однокубітова телепортація

Розглянута вище телепортація використала три кубіти. Це кубіт, стан якого треба телепортувати, і пару кубітів, що знаходяться в переплутаному стані. Таку схему телепортації можна спростити, використовуючи для телепортації тільки два кубіти [15,16]. Ясно, що стан одного з них і буде телепортуватися, а другий носить допоміжну роль, утворюючи переплутаний стан з першим кубітом. Зазвичай, цей допоміжний кубіт називають ancilla або служниця в перекладі з англійської мови. Далі його називатимемо службовим кубітом. Таким чином, початково в системі присутні всього два кубіти. Перший кубіт (1), стан якого буде телепортуватися, і другий кубіт (2) **ancilla** знаходяться спочатку в незалежних станах

$$|\psi\rangle = |\varphi\rangle\,|0\rangle.$$

Вектор стану першого кубіту

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

як і раніше, невідомий відправнику. Службовий кубіт знаходиться в стані $|0\rangle$. Стани цих кубітів переплутаємо, діючи на них гейтом $\widehat{CNOT}$. Контрольованим кубітом являється службовий кубіт (див. Рис.8.6). Виникаючий стан і утворює переплутаний стан

$$\widehat{CNOT}\,(1,2)\,|\psi\rangle = \alpha|00\rangle + \beta|11\rangle\,.$$

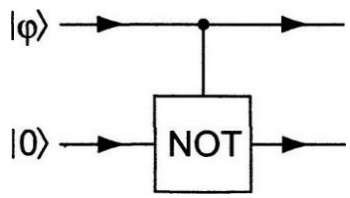


Рис. 8.6. Схема, що формує переплутаний стан двох кубітів

Відразу після цього службовий кубіт передамо у віддалений пункт передбачуваної телепортації. Потім на кубіт, стан якого передбачалося телепортувати, подіємо гейтом Адамара. В результаті стан кубітів змінюється згідно

$$\hat{H}\,(1)\,\widehat{CNOT}\,(1,2)\,|\psi\rangle = \alpha(\hat{H}|0\rangle)|0\rangle + \beta(\hat{H}|1\rangle)\,|1\rangle =$$

$$= \alpha\,\frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle\,)\,|0\rangle + \frac{1}{\sqrt{2}}\,\beta\,(|0\rangle - |1\rangle\,)\,|1\rangle =$$

перетворимо запис цього стану в зручному для подальших спостережень виді

$$=\frac{1}{\sqrt{2}}\,[|0\rangle(\alpha|0\rangle - \beta|1\rangle) + |1\rangle(\alpha|0\rangle + \beta|1\rangle)]=$$

після цього проведемо вимірювання над підготовленим для телепортації першим кубітом в стандартному базисі векторів станів $|0\rangle$ і $|1\rangle$. Повертаючись до попереднього запису стану кубітів, легко зрозуміти, що можливі тільки два результати вимірювань. Наш прилад може зафіксувати стан $|0\rangle$ або стан $|1\rangle$ першого кубіту. При цьому в результаті вимірювання відбувається редукція вектору станів,

якщо спостерігається $|0\rangle$, то стан колапсує в $\alpha|0\rangle + \beta|1\rangle$,

якщо спостерігається $|1\rangle$, то стан колапсує в $\alpha|0\rangle - \beta|1\rangle$.

Значить, службовий кубіт після вимірювання може знаходитися в одному з двох приведених вище станів. З'ясуємо тепер, яким унітарним оператором відновлюється стан $|\varphi\rangle$ у службового кубіта, що телепортується. Легко помітити, що

якщо вимірюється $|0\rangle$, то $\hat{I}\,(\alpha|0\rangle + \beta|1\rangle) = |\varphi\rangle$,

якщо вимірюється $|1\rangle$, то $\hat{\sigma}_z\,(\alpha|0\rangle - \beta|1\rangle) = |\varphi\rangle$.

Нагадаємо, що дія гейта $\hat{\sigma}_z$ на базисні стани визначається співвідношеннями $\hat{\sigma}_z|0\rangle = |0\rangle$ і $\hat{\sigma}_z|1\rangle = -|1\rangle$. Отже, для здійснення телепортації після проведення описаного вище вимірювання над кубітом, що знаходився раніше в стані, який передбачалося телепортувати у віддалений пункт, передається результат вимірювання.

У цьому віддаленому пункті на службовий кубіт діють одиничним оператором (n = 0), якщо повідомлений результат вимірювання 0 або оператором $\hat{\sigma}_z$ (n = 1), якщо повідомляється результат вимірювання 1 (див. Рис. 8.7).

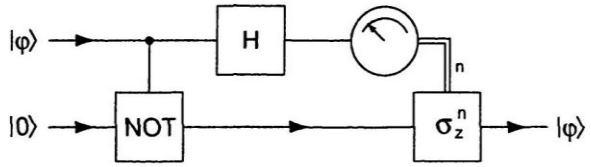


Рис. 8.7. Квантова схема однобітової телепортації. Класичний канал управління показаний подвійною лінією. Якщо вимірювання визначає стан $|0\rangle$, то n = 0, якщо $|1\rangle$, то n = 1

У будь-якому випадку службовий кубіт переходить в стан $|\varphi\rangle$. В результаті здійснилася однобітова телепортація. Звернемо увагу, що однобітову телепортацію можна здійснити і за іншою квантовою схемою, приведеною на рис. 8.8. Дійсно, до вимірювання стан кубітів

$$\widehat{CNOT}\,(2,1)\,\hat{H}\,(2)|\psi\rangle = \frac{1}{\sqrt{2}}\,[|0\rangle\,(\alpha|0\rangle + \beta|1\rangle) + |1\rangle\,(\alpha|1\rangle + \beta|0\rangle)].$$

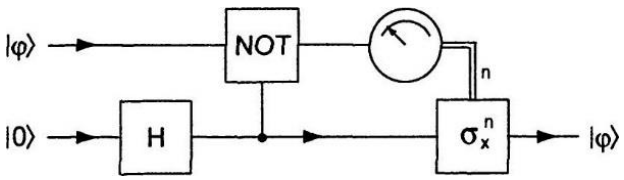


Рис. 8.8. Ще одна квантова схема однобітової телепортації. Якщо вимірювання визначає стан $|0\rangle$, то n = 0, якщо $|1\rangle$, то n = 1

Знову при результаті вимірювання 0 першого кубіту стан другого буде $\alpha|0\rangle + \beta|1\rangle$, а при результаті 1 стан буде $\alpha|1\rangle + \beta|0\rangle$. Таким чином, при результаті 0 для телепортації вже не треба застосовувати ніяких операторів, n = 0. У випадку 1 слід застосувати $\hat{\sigma}_x$ (n = 1) до другого кубіту і перевести його в стан $\alpha|0\rangle + \beta|1\rangle$. Це завершує процес телепортації.

8.4 Квантова телепортація переплутаності

Розглянемо тепер можливість телепортації переплутаності станів кубітів. Нагадаємо, що, як правило, переплутаність станів раніше виникала в результаті взаємодії кубітів. Нижче обговоримо можливість телепортації переплутаності

станів на кубіти, що ніколи не взаємодіють один з одним. Для цього розглянемо дві пари кубітів. Нехай початково кубіти 1 і 2 знаходяться в переплутаному стані

$$|\varphi_{12}\rangle = \alpha|00\rangle + \beta|11\rangle$$

і незалежна пара кубітів 3 і 4 знаходиться в переплутаному стані

$$|\varphi_{34}\rangle = \gamma|00\rangle + \delta|11\rangle .$$

Значить, початковий стан цих пар кубітів визначається вектором станів

$$|\varphi\rangle = (\alpha|00\rangle + \beta|11\rangle)(\gamma|00\rangle + \delta|11\rangle) =$$

який, згідно прийнятих нами угод, можна записати як

$$= \alpha\,\gamma|0000\rangle + \alpha\delta\,|0011\rangle + \beta\gamma|1100\rangle + \beta\delta|1111\rangle =$$

тепер проведемо вимірювання 2 і 3 кубітів у базисі $|\Phi_{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$ і $|\Psi_{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$. Для того, щоб зрозуміти результат таких вимірювань, запишемо стани 2 і 3 кубітів в цьому базисі

$$= \alpha\,\gamma|0\rangle|00\rangle|0\rangle + \alpha\delta\,|0\rangle|01\rangle|1\rangle + \beta\gamma|1\rangle|10\rangle|0\rangle + \beta\delta|1\rangle|11\rangle|1\rangle =$$

$$= \frac{1}{\sqrt{2}}\,\alpha\gamma|0\rangle(|\Phi_+\rangle + |\Phi_-\rangle)|0\rangle +$$

$$+ \frac{1}{\sqrt{2}}\,\alpha\delta|0\rangle(|\Psi_+\rangle + |\Psi_-\rangle)|1\rangle +$$

$$+ \frac{1}{\sqrt{2}}\,\beta\gamma|1\rangle(|\Psi_+\rangle - |\Psi_-\rangle)|0\rangle +$$

$$+ \frac{1}{\sqrt{2}}\,\beta\delta|1\rangle(|\Phi_+\rangle - |\Phi_-\rangle)|1\rangle =$$

Після елементарних перетворювань отримаємо:

$$= \frac{1}{\sqrt{2}}\,\alpha\gamma|0\rangle|\Phi_+\rangle\,|0\rangle + \frac{1}{\sqrt{2}}\,\beta\delta|1\rangle|\Phi_+\rangle\,|1\rangle +$$

$$+ \frac{1}{\sqrt{2}}\,\alpha\gamma|0\rangle|\Phi_-\rangle\,|0\rangle - \frac{1}{\sqrt{2}}\,\beta\delta|1\rangle|\Phi_-\rangle\,|1\rangle +$$

$$+ \frac{1}{\sqrt{2}}\,\alpha\delta|0\rangle|\Psi_+\rangle\,|1\rangle + \frac{1}{\sqrt{2}}\,\beta\gamma|1\rangle|\Psi_+\rangle\,|0\rangle -$$

$$- \frac{1}{\sqrt{2}}\,\alpha\delta|0\rangle|\Psi_-\rangle\,|1\rangle - \frac{1}{\sqrt{2}}\,\beta\gamma|1\rangle|\Psi_-\rangle\,|0\rangle.$$

Тепер легко зрозуміти, що можливі чотири результати вимірювань станів 2 і 3 кубітів. Це Φ_+, Φ_- , Ψ_+ і Ψ_-. Так, якщо результат вимірювання дає Φ_+, то вектор стану 1 і 4 кубітів колапсує у вектор стану

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha\gamma|00\rangle + \frac{1}{\sqrt{2}}\,\beta\delta|11\rangle.$$

Якщо результат вимірювання Φ_-, вектор стану редукує в

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha\gamma|00\rangle - \frac{1}{\sqrt{2}}\,\beta\delta|11\rangle.$$

Вірогідність кожного з цих результатів вимірювання $|\alpha\gamma|^2 + + |\beta\delta|^2$. Якщо Ψ_+, то

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha\delta|01\rangle - \frac{1}{\sqrt{2}}\,\beta\gamma|10\rangle.$$

Нарешті, при Ψ_-

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha\delta|01\rangle + \frac{1}{\sqrt{2}}\,\beta\gamma|10\rangle.$$

Вірогідність спостереження одного з цих двох станів $|\alpha\delta|^2 + |\beta\gamma|^2$. Звернемо увагу, що раніше незалежні стани першого і четвертого кубітів тепер стали переплутаними. Особливо цікаво, що ці кубіти не взаємодіяли в процесі такої перебудови станів. Для простоти припустимо, що спочатку кубіти 3 і 4 були максимально переплутані, тобто, $\gamma = \delta = \frac{1}{\sqrt{2}}$. Далі, дотримуючись схеми рис. 8.9, по класичному каналу зв'язку можна повідомити один з чотирьох результатів вимірювань. Таким чином, якщо повідомляється, що в результаті вимірювань було виявлено стан Φ_+, то перший і четвертий кубіти знаходяться в стані

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha|00\rangle + \frac{1}{\sqrt{2}}\,\beta|11\rangle$$

і для переведення його в стан, що телепортується, вже нічого робити не потрібно.

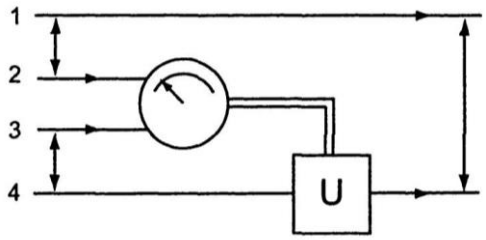


Рис. 8.9. Квантова схема телепортації переплутування.
Початково переплутані стани 1 і 2 кубітів і 3 і 4 кубітів.
На виході схеми переплутані стани 1-го і 4-го кубітів

В цьому випадку унітарний оператор $\hat{U} = \hat{I}$. Розглянемо випадок, коли повідомляється результат вимірів Φ_-, тоді

$$|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha|00\rangle - \frac{1}{\sqrt{2}}\,\beta|11\rangle,$$

тепер при отриманні цього результату на четвертий кубіт подіємо однокубітним гейтом $\hat{U} = \hat{\sigma}_z$ і переведемо переплутаність в ту, що телепортується, $|\varphi_{14}\rangle = \frac{1}{\sqrt{2}}\,\alpha|00\rangle + \frac{1}{\sqrt{2}}\,\beta|11\rangle$. При отриманні Ψ_+ і Ψ_- треба для телепортації подіяти оператором $\hat{U} = \hat{\sigma}_x$ і $\hat{U} = -i\hat{\sigma}_y$ відповідно. Таким чином, використовуючи класичний канал зв'язку, можна телепортувати квантову переплутаність станів.

8.5 “Класична” телепортація

У 2007 році була запропонована інша схема телепортації, що не використовує переплутаності станів [17]. Така назва, як використана вище, не

означає неквантовість телепортації. У класичній фізиці телепортація неможлива. Просто, на думку авторів, вона ближче за духом до телепортації, яка описана у багатьох фантастичних романах і може бути тому названа "класичною". Ідея полягала в тому, щоб узяти усю інформацію про пучок часток і перенести її на фотони, які можна передати по оптоволоконному кабелю на велику відстань. У віддаленому пункті прийому за цією інформацією відтворити початковий пучок часток. Все виглядає так, як ніби в одному місці пучок часток зникає, а в іншому з'являється. Цікаво помітити, що для реалізації цієї ідеї потрібно було використати Бозе-Ейнштейнівський конденсат. Хоча такий стан речовини був, як випливає з назви, відкрито Шатьєдронатом Бозе і Ейнштейном ще в 1925 році, але тільки в 1995 році його вдалося отримати в лабораторії експериментально. Охолоджуючи газ бозе часток до шалено низьких температур ($10^{-6} - 10^{-9}$ градуса K^0), можна добитися появи Бозе конденсату. Телепортація за цією схемою починається з Бозе конденсату атомів рубідія.

Зрозуміло, вибір атомів рубідія не принциповий, просто саме вони використовувалися в експерименті. На цей конденсат падає пучок атомів рубідія, який прагне перейти в стан Бозе конденсату (див. Рис. 8.10). Ліворуч відправник ψ_1^{in} у формі Бозе конденсату, що поглинає пучок атомів ψ_2^{in} і випускає світло, що несе усю квантову інформацію про атоми. Справа одержувач ψ_1^{out}, що поглинає світло і поновлює пучок атомів по квантовій інформації. Поля Ω^{in} і Ω^{out}, що управляються, потужні поля, а E^{in} і E^{out} - слабкі пробні поля. При цьому зменшення енергії пучка супроводжується її випромінюванням у формі фотонів. Світло, що з'явилося, передається по оптоволоконному кабелю до іншого віддаленого Бозе конденсату атомів рубідія. Потрапляючи в конденсат, фотони породжують пучок атомів рубідія в точності в тому ж стані, що і використаний для появи світла. Найважливішим є те, що фотони містили усю квантову інформацію про початковий пучок атомів рубідія.

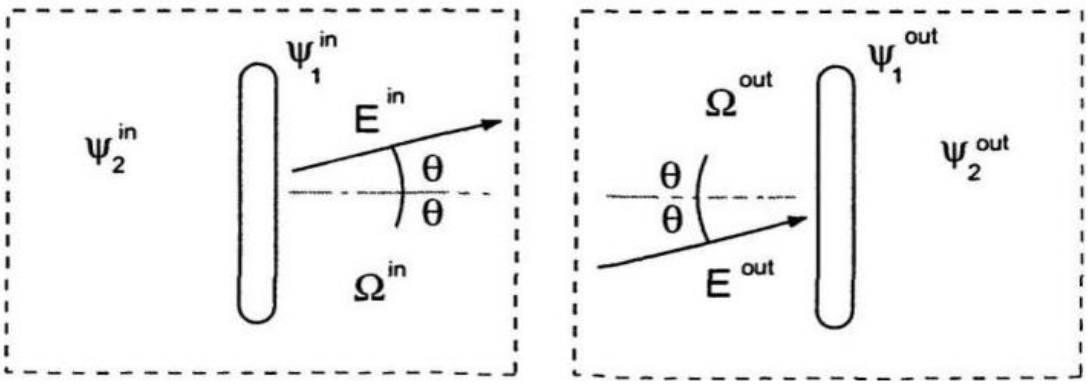


Рис. 8.10. Експериментальна **с**хема “класичної” телепортації

Таким чином, згідно з таким протоколом телепортації, на станції, що телепортує (див. Рис.8.10 ліворуч), роль якої грає Бозе конденсат, падає пучок атомів рубідія ψ_2^{in}, класичне поле Ω^{in} , що управляється, і слабке оптичне поле E^{in}. Схематичне розташування енергетичних рівнів як станції, що посилає, так і

станції, що приймає, показано на рис. 8.11. На цьому малюнку стан $|1\rangle$ відповідає Бозе конденсату, стан $|2\rangle$ - вхідному пучку атомів, а стан $|3\rangle$ −стану збуджених атомів. Частота ω_0 - це частота Рабі переходу між станами $|2\rangle \leftrightarrow |3\rangle$. При конденсації пучка часток уся квантова інформація про нього переходить на поле E^{in} (Рис. 8.10). Конфігурація рівнів енергії дозволяє стимулювати переходи між захопленим і незахопленим полями. Ці переходи стимулюють інтенсивне контрольоване поле, яке позначене Ω^{in}, і поле захоплених бозе часток ψ_1^{in}. Оптичне поле E^{in} долає відстань між конденсатами і використовується для вилучення атомного пучка з приймальної станції. У основі такого процесу лежить механізм комбінаційного розсіяння світла, який в закордонній літературі прийнято називати ефектом Рамана. Звернемо увагу, що приготування посилаючої і приймаючої станції і керуючих полів потребує тільки класичної інформації і зовсім не потребує знання квантового стану початкового пучка. Стадія передачі інформації пробним полем за певних умов відбувається автоматично. Трансформація оптичного поля в пучок відбувається за механізмом, описаним Хайне і Хоупом [18].

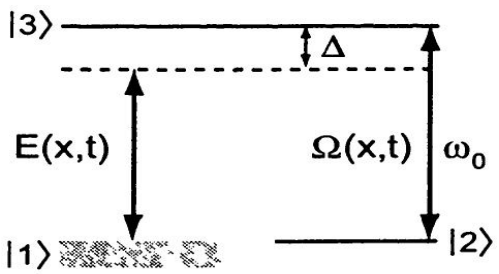


Рис. 8.11. Енергетичні рівні об'єктів, що беруть участь в телепортації. Щільно заселений рівень показаний широкою сірою смугою

9 КВАНТОВІ КЛЮЧІ І ЇХ ПЕРЕДАЧА

Обговорюючи квантову передачу ключів, згадаємо раніше розглянуту неможливість підслуховувати квантове повідомлення, що передається в неортогональному "алфавіті" (див. розділ 7.7). Для передачі ключів між Алісою і Бобом повинні існувати два канали зв'язку.

Один з них відкритий, класичний канал зв'язку. Повідомлення, що передаються по ньому, доступні будь-кому, хто бажає. Другий канал зв'язку - це квантовий канал. Надійність або захищеність передачі по квантовому каналу пов'язана з фізичним порушенням сигналу при підслуховуванні (див. розділ 7.7). Для простоти далі припускатимемо відсутність шумів в квантовому каналі.

Квантову передачу секретного ключа можна організувати за наступною простою схемою. Спочатку Аліса і Боб домовляються про квантові базиси станів, які вони будуть використовувати. Наприклад, будуть використовуватися два базиси

$$|\varphi_0\rangle = |0\rangle, |\varphi_1\rangle = |1\rangle \text{ і } |\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

які далі будемо позначати як 1 і 2 відповідно. При цьому стани $|0\rangle$ і $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ означатимуть 0, а стани $|1\rangle$ і $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ відповідатимуть 1. Підкреслимо, що ці базиси взаємно не ортогональні.

Нехай Аліса вибирає випадковий рядок бітів довжини $4n$. Тепер цей рядок слід закодувати станами $4n$ кубітів. Вибираючи випадковим чином базис для запису відповідного значення у кубіт, вона отримує $4n$ кубітів, що кодують початкову послідовність, яка приведена в табл. 9.1.

Таблиця 9.1.

Початковий рядок	0	1	1	0	1	. . .	1	0
Базис Аліси	1	2	2	2	1	. . .	1	1
Стан кубіту	$\|\varphi_0\rangle$	$\|\psi_1\rangle$	$\|\psi_1\rangle$	$\|\psi_0\rangle$	$\|\varphi_1\rangle$	. . .	$\|\varphi_1\rangle$	$\|\varphi_0\rangle$

Отриманий рядок кубітів Аліса пересилає по квантовому каналу зв'язку Бобу. Боб, отримуючи цю послідовність, вибирає випадковим чином базис, проводить вимірювання цих кубітів. Випадковий вибір базису робиться кожного разу при вимірюванні кубіту (табл.9.2).

Після цього Боб зв'язується по відкритому, класичному каналу зв'язку з Алісою і повідомляє, який базис він використав при вимірюванні, а Аліса повідомляє його про свій базис. Іншими словами, вони з'ясовують, в яких випадках вони використали однаковий базис. Зрозуміло, вони ні слова не говорять про результати вимірювання.

Таблиця 9.2.

Стан кубіту	$\|\varphi_0\rangle$	$\|\psi_1\rangle$	$\|\psi_1\rangle$	$\|\psi_0\rangle$	$\|\varphi_1\rangle$	. . .	$\|\varphi_1\rangle$	$\|\varphi_0\rangle$
Базис Боба	2	1	2	2	2	. . .	1	2
Рядок Боба	1	1	1	0	0	. . .	1	1

Залишаючи тільки значення бітів, при кодуванні і вимірюванні яких використовувався однаковий базис, у Аліси і Боба з'являється однаковий рядок бітів. Довжина цього рядка, враховуючи незалежність вибору базисів, приблизно **2*n*** (табл. 9.3).

Таблиця 9.3.

Базис Аліси	1	2	2	2	1	. . .	1	1	1
Базис Боба	2	1	2	2	2	. . .	1	2	1
Рядок, що залишився			1	0		. . .	1		1

В нашому прикладі це

10 . . . 11

Тепер залишається перевірити, чи здійснювалося підслуховування квантової передачі, чи ні. Для цього Аліса вибирає випадковим чином *n* бітів цього рядка і повідомляє їх значення Бобу по відкритому каналу зв'язку. Боб порівнює їх зі значеннями цих кубітів, отриманих ним при вимірюванні. Якщо число тих, що співпали, невелике, то це означає, що відбувалося підслуховування, яке привело до зміни станів кубітів. Тому процес обміну ключем треба почати знову. Якщо значення співпадають, то у Аліси і Боба залишається загальний рядок з *n* бітів, який і може використовуватися як секретний ключ.

Зрозуміло, запропоновані і інші квантові схеми обміну ключем [19]. Зупинимося на схемі, що використовує як ресурс переплутані стани кубітів. Припустимо, що між Алісою і Бобом колись були розподілені пари кубітів, що знаходяться в переплутаному стані

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}\,(|00\rangle + |11\rangle).$$

В принципі такі пари можуть розсилатися Алісі і Бобу з деякого нейтрального джерела. Проводячи вимірювання у випадковому базисі, аналогічно як і у попередньому випадку станів своїх кубітів, Аліса і Боб отримують послідовності 0 і 1. Встановлюючи по класичному каналу випадки співпадіння базисів вимірювання, вони виділяють секретний ключ. Зрозуміло, частину цієї послідовності можна використати для перевірки секретності або збереження зачепленості пар при пересилці. Основна відмінність полягає в тому, що ключ в такій схемі випадковий з фізичних причин, пов'язаних з вимірюванням станів кубітів. У попередній схемі таку випадковість забезпечувала Аліса. У цьому сенсі вона і генерувала секретний ключ. При використанні зачеплених станів ні Аліса, ні Боб його не генерують. У цьому сенсі

його випадковість об'єктивніша. До недоліків цієї схеми можливо слід віднести те, що її реалізація складніша, ніж попередньої схеми.

Не дивно, що експериментально вдалося реалізувати схему, що не використовує зачепленості квантових станів. Експериментальна передача секретного ключа по квантовому каналу зв'язку уперше була здійснена ще в 1992 році на відстань приблизно 40 см, використовуючи імпульси зеленого світла. Значно пізніше така передача була здійснена на відстань 23 км, що має вже практичний інтерес.

9.1 Класичні коди, що виправляють помилки

Коди кодування і декодування відіграють важливу роль не лише для збереження секретності. Існує ще одна важлива проблема, рішення якої тісно пов'язане з цими об'єктами. Про цю проблему вже згадувалося в розділі 7.12. Це необхідність боротьби з шумами і перешкодами, які перешкоджають обчисленням як класичним, так і квантовим. Основна ідея досить проста і вона використовується нами досить часто навіть на побутовому рівні. Так, якщо ми спілкуємося з кимось в шумній обстановці, як правило, ми часто дублюємо або повторюємо потрібну пропозицію, щоб бути упевненим в тому, що вона правильно зрозуміла співрозмовнику. Аналогічно відбувається при спілкуванні з іноземцем. Для того, щоб бути правильно зрозумілим, одна і та ж пропозиція, можливо з варіаціями, повторюється кілька разів. Іншими словами, ідея полягає у використанні надмірної інформації для передачі повідомлення. Тоді навіть при втраті частини інформації можна відновити повідомлення. На цьому принципі і базується боротьба з перешкодами і шумами при проведенні обчислень або при передачі інформації по каналах зв'язку.

Обговоримо це детальніше. Давайте спробуємо передати біт інформації. Для цього здійснимо кодування 0, переводячи його в стан трьох бітів 000, а 1 закодуємо як 111. Для простоти припускатимемо, що кодування і декодування повідомлень здійснюється абсолютно точно. Шум і помилки з'являються на стадії передачі інформації. Це означає, що шум або перешкоди при передачі біта міняють його з вірогідністю p і зберігають з вірогідністю 1 - p (див. Рис. 9.1). Вірогідність p є характеристикою каналу зв'язку. Припустимо, що все-таки $p < ½$.

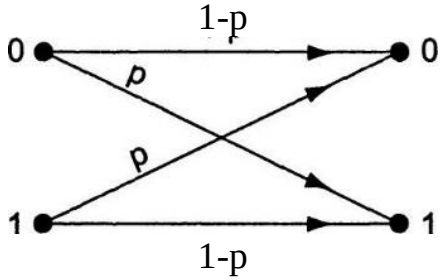


Рис. 9.1. Діаграма передачі інформації з вірогідністю помилки p.
Це схема симетричного бінарного каналу передачі інформації.
Така ж схема відповідає зберіганню інформації,
тоді стрілки символізують зміну часу

Ясно, що при передачі закодованого повідомлення, наприклад 000, може статися помилка і отримане повідомлення буде 100. Якщо припустити, що канал зв’язку не зовсім поганий, то можна чекати появи помилки тільки в одному біті. У нашому прикладі помилка в першому біті. Виходячи з цього, можна відновити повідомлення, виправляючи його за ознакою більшості значень. Тоді 100 → 000 і, отже, значення біта, що передається, було 0. Таким чином, при декодуванні виправляється помилка, викликана шумом в каналі зв'язку. Використовуваний вище код називається кодом з повторенням. Число значень, що кодують початкове значення, може бути будь-яким і визначає кількість доданої надмірної інформації. Зрозуміло, це у свою чергу визначає надійність повідомлення. У кодах з повторенням рядка 000 і 111 називаються, відповідно, логічним нулем і логічною одиницею. Обчислимо тепер вірогідність помилки при такому способі декодування. Ясно, що виправлення помилки в більшості випадків не спрацює, якщо помилка з'явиться в двох або трьох бітах. Вірогідність цього легко виявити. Для цього треба скласти вірогідність трьох подій, коли не міняється тільки один біт з трьох, з вірогідністю зміни усіх трьох бітів. В результаті отримаємо $p_1 = +p^3 = 3p^2(1-p)\ 3p^2 - p^3$. Неважко перевірити, що при p < 1/2 вірогідність $p_1 < p$ і, отже, передача повідомлення з використанням коду з повторенням надійніша (див. Рис. 9.2). Зрозуміло, при використанні таких кодів знижується швидкість передачі інформації.

Потрібно відмітити, що теорія класичних кодів, що виправляють помилки, добре розвинена. Головна причина цього в існуванні безлічі технічних застосувань цієї теорії. Перш ніж перейти до обговорення квантових кодів, що виправляють помилки, обговоримо деякі елементи класичної теорії. Природно, багато рис класичних кодів, що виправляють помилки, можна перенести на квантові коди.

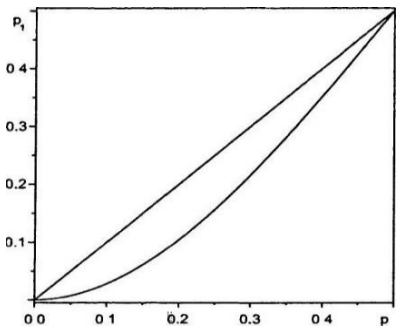


Рис. 9.2. Показана вірогідність p_1 як функція p.
Добре видно, що p_1< p в області значень p < 1/2.
Пряма лінія відповідає p, а крива – p_1 (p)

Розпочнемо з теорії класичних лінійних кодів. Лінійний код, що кодує m бітів інформації в $n > m$ бітовий простір коду, визначається матрицею G розміру n x m з елементами 0, 1. Кодування здійснюється просто множенням m бітового повідомлення x на матрицю G. При цьому закодоване повідомлення буде Gx. Усі арифметичні операції додавання, множення і інші здійснюються за модулем 2.

Приклад 9.1. *Неважко помітити, що раніше розглянутий код з повторенням відноситься до лінійних кодів. Дійсно, якщо вибрати в якості G матрицю,*

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \qquad (9.1)$$

то перемноживши початкове повідомлення 0 на G, отримаємо (0,0,0), а перемноживши 1 на G, отримаємо (1, 1, 1). Цим і закінчується порівняння з кодом з повторенням.

Лінійні коди, що кодують *m* бітів інформації в *n* бітах, прийнято розділяти на типи, позначаючи їх як коди типу [*n,m*]. Так, код, що приведений в прикладі вище, відноситься до коду типу [3, 1]. Представлення лінійних кодів з використанням матриці *G*, що являється виробляючою, зручно для опису кодування, але не для опису коригування помилок. Тому перейдемо до іншого еквівалентного представлення - в термінах перевірочної матриці *H*. Для коду типу [*n,m*] ця матриця розміром (*n* - *m*) х *n* має наступну властивість

$$Hy = 0,$$

де *y* - всі вектори з *n* координатами, що належать [0, 1]. Власне, ці вектори і складають код типу [*n, m*]. Елементи перевірочної матриці також 0 або 1. Для доказу еквівалентності треба вказати конструктивний алгоритм побудови матриці *H* при заданій *G* і навпаки. Для переходу від *H* до *G* виберемо *m* лінійно незалежних векторів $z_1, \ldots, z_m$, для яких ядро *H* є лінійною оболонкою. Матрицю *G* побудуємо із стовпців $z_1, \ldots, z_m$. Для переходу від матриці *G* до *H* візьмемо *n* - *m* лінійно незалежних векторів, ортогональних стовпцям матриці *G*, і використовуємо їх як рядки матриці *H*.

Приклад 9.2. *Знову повертаючись до коду типу [3, 1], побудуємо 3 - 1 = 2 лінійно незалежних вектори, які ортогональні стовпцю матриці (9.1). Це відповідає умові, що сума координат цих векторів дорівнює нулю за модулем 2. Тоді в якості таких векторів можна вибрати (0,1,1) і (1,1,0). Тоді перевірочна матриця, згідно приведеному вище алгоритму, рівна:*

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Перевіримо тепер, на яких векторах Hx = 0. Легко довести, що таких векторів тільки два: (0,0,0) і (1,1,1).

Тепер звернемося до процедури виявлення помилки. Припустимо, що перед пересилкою повідомлення *x* кодується лінійним кодом в $y = Gx$. Після проходження по зашумленому каналу зв'язку виникає помилка ε і повідомлення стає $y' = y + \varepsilon = Gx + \varepsilon$. Подіємо на отримане повідомлення перевірочною матрицею, отримаємо

$$Hy' = Hy + H\varepsilon,$$

враховуючи, що $Hy = 0$, отримаємо

$$Hy' = H\varepsilon.$$

Величина Hy' називається синдромом помилки. Синдром містить інформацію про помилку, яка дозволяє сподіватися на відновлення повідомлення. Припустимо, що помилка сталася на одному біті.

Тоді Hy' дорівнює нулю у відсутності помилки і $H\varepsilon_i$, якщо помилковий i-й біт. Вектор ε_i з відмінною від нуля i-ю координатою (рівною 1). Тоді, порівнюючи Hy' з можливими значеннями $H\varepsilon_i$ при $H\varepsilon_i \neq H\varepsilon_j$, встановлюємо положення помилки. Таким чином, синдром помилки сигналізує про її появу і дає шлях до її виправлення.

Для пояснення виправлення помилки використовуємо ще одне важливе поняття - відстань між двома бінарними послідовностями. Важливою для подальшого буде відстань Хеммінга d (x,y), яка дорівнює кількості цифр, в яких ці дві послідовності відрізняються між собою. Інтуїтивно, роль цієї відстані визначається тим, що чим більша відстань d (x, y) між послідовностями, тим легше їх розрізняти навіть за наявності шуму. Кодові слова $y_1, y_2, \ldots, y_k$ представляють собою послідовності 0, 1 довжиною n. Після проходження через канал зв'язку $y \to y'$. Як відзначалося вище, припустимо, що жертвою помилки стає тільки один біт. Тоді незмінене кодове слово y належить околиці розміром $d(y, y') \leq 1$ (див. Рис. 9.3).

Зрозуміло, так буде, якщо відстані між кодовими словами досить великі і задовольняють нерівностям $d(y_i, y_j) \geq 2l + 1$. Зокрема, для випадку кодування з повторенням $l = 1$. Дійсно, за визначенням ясно, що d (000, 111) = 3. Якщо відстань $d(y_i, y_j) \geq 2l + 1$ (при $i \neq j$), то виправляються до l помилок, що появилися при передачі по каналу з шумом.

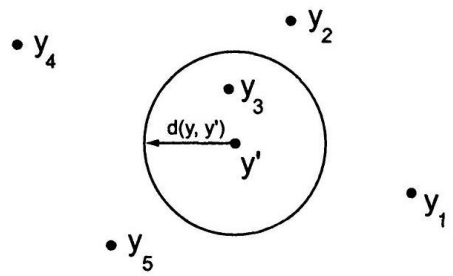


Рис. 9.3. Показана ідея вибору незіпсованого кодового слова y за відстанню Хеммінга від отриманого повідомлення y'.
У зображеному прикладі найближче кодове слово до отриманого повідомлення y' - це y_3. Його вибір і усуває помилку

9.2 Квантові коди, що виправляють помилки

Обнадійливі зразки класичних кодів, що коригують помилки, викликають бажання перенесення їх на квантові обчислення. Проте, все не так просто і, отже, цікавіше. У квантовій області потрібні нові ідеї. Причина цього може бути легко усвідомлена. Передусім, навіть аналог кодування біта в трьохбітовому

повідомленні в квантовій області викликає необхідність роздумів. Очевидна причина в неможливості клонування квантових станів (див. Підрозділ 7.7). Ще одна трудність підстерігає при спробі узагальнити на квантовий випадок вимірювання станів кубітів для ухвалення рішення про наявність помилки і подальшого її виправлення і декодування. Причина в руйнуванні квантових станів при вимірюванні.

Окрім цього, квантові помилки якісно відрізняються від класичних. Так, під дією шуму стан кубіту може змінитися безперервно нескінченним числом способів. Це означає, що на перший погляд потрібна величезна точність для ідентифікації помилки і, отже, величезні ресурси для її упізнання.

Ці причини деякий час стримували розвиток методів виправлення помилок в квантовій області. Проте, можливості просування в цій області все-таки були знайдені ще в 1995 році. Нижче постараємося пояснити механізми подолання цих труднощів. Розпочнемо з обговорення кодування стану кубіту

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

станами трьох кубітів. Узагальнюючи класичне кодування біта в трьох бітах, необхідно добитися

$$|0\rangle \rightarrow |0_L\rangle = |000\rangle \text{ і } |1\rangle \rightarrow |1_L\rangle = |111\rangle.$$

Стани $|0_L\rangle$ і $|1_L\rangle$ часто називають логічним нулем і одиницею відповідно.

Цієї мети можна досягти, використовуючи квантові гейти контрольованого "NOT".

Квантова схема, що здійснює таке кодування, приведена на рис. 9.4. В результаті на виході цієї квантової схеми стани кубітів будуть $\alpha|000\rangle + \beta|111\rangle$. Це цілком аналогічно класичному кодуванню з повторенням.

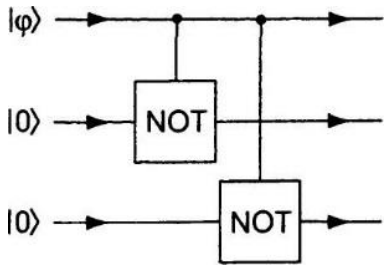


Рис. 9.4. Квантова схема, що здійснює трьохкубітне кодування

Тепер розглянемо виникнення помилки при передачі повідомлення по квантовому каналу зв'язку. Розпочнемо з типу помилки найбільш близького до класичного. Нехай в результаті передачі стан кубіту не міняється з вірогідністю 1 - p і з вірогідністю p міняється як $|\varphi\rangle \rightarrow \widehat{NOT}|\varphi\rangle$. Наочно це можна представляти, наприклад, як переворот спіна. Така зміна дуже нагадує виникнення помилки під дією шуму в класичному каналі зв'язку. Тому канал з таким механізмом появи помилки називається каналом з класичною помилкою. Як і раніше, припускаємо, що кодування здійснюється ідеальним чином і помилка може з’явитися тільки на стадії передачі.

При передачі трьох кубітів по каналу з класичною помилкою при значеннях $p < 1/2$ можна чекати появи помилки тільки в одному з кубітів. Тоді ясно, що існує тільки чотири можливості зміни станів кубітів. Це $\alpha|000\rangle + \beta|111\rangle, \alpha|001\rangle + \beta|110\rangle$, $\alpha|010\rangle + \beta|101\rangle$ і $\alpha|100\rangle + \beta|011\rangle$. Після цього виникає проблема встановлення і виправлення помилки. Як і в класичному випадку, в квантовому треба визначити синдром помилки. В якості синдрому помилки для каналу з класичною помилкою використовуємо 4 проєкційні оператори

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111|.$$

Якщо помилка відсутня, то синдром $\langle\psi|P_0|\psi\rangle = 1$, тут $|\psi\rangle$ – стани кубітів після проходження каналу зв'язку. Інші синдроми нульові $\langle\psi|P_1|\psi\rangle = 0$ при $i = 1, 2, 3$.

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011|.$$

Якщо помилка виникла в першому кубіті, то буде відмінний від нуля тільки синдром $\langle\psi|P_1|\psi\rangle = 1$.

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$$

Аналогічно з помилками у другому і третьому кубітах.

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$$

Звернемо увагу, що визначення синдрому не міняє квантовий стан при його вимірюванні. Власне, при вимірюванні синдрому ми не будемо мати інформації про квантовий стан, тобто, про α і β, тому таке вимірювання і не міняє квантовий стан.

Після визначення синдрому помилки її треба виправити і відновити початковий стан. При знанні, який синдром відмінний від нуля і, отже, в якому кубіті сталася помилка, залишається "перевернути" стан саме цього кубіту. Це можна зробити, подіявши на відповідний кубіт оператором $\widehat{NOT}$. Таким чином, виправляються помилки каналу зв'язку з класичною помилкою або амплітудні помилки. Проте, як обговорювалося вище, помилки в квантовому випадку більш різноманітніші класичних. Окрім амплітудних помилок, в квантових каналах зв'язку можуть виникати фазові помилки або помилки у фазі квантових станів. Такі помилки вже не мають класичного аналога. Розглянемо в якості простого випадку фазових помилок таку зміну стану кубіту, коли він з вірогідністю p міняє стан $\alpha|0\rangle + \beta|1\rangle$ на стан $\alpha|0\rangle - \beta|1\rangle$. Слід зазначити, що ця квантова помилка не має класичного аналога. Таку помилку можна описати дією оператора $\hat{\sigma}_z$ на стан кубіту з вірогідністю p. Власне, так і вводяться моделі помилок для детального аналізу їх дії і наслідків. Наприклад, розглянуту вище амплітудну помилку можна моделювати дією оператора $\hat{\sigma}_x$ з визначеною вірогідністю. Популярною моделлю множини помилок є набір операторів Паулі ($\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z, \hat{I}$), що діють з певною вірогідністю на стан кубіту. Останній одиничний оператор означає відсутність помилки. Для того, щоб зрозуміти, як виправляти таку фазову помилку, досить зробити корисне спостереження. Давайте в якості базисних

станів використаємо стани $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ і $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Дію оператора $\hat{\sigma}_z$ на ці стани легко встановити. Так, $\hat{\sigma}_z|+\rangle = |-\rangle$, а $\hat{\sigma}_z|-\rangle = |+\rangle$. Таким чином, в цьому базисі оператор фазової помилки діє на базисні стани як оператор амплітудної помилки.

Тоді ясно, що працюючи в цьому базисі, можна виправити фазову помилку в точності так, як виправлялася раніше амплітудна помилка.

Для цього в якості закодованого стану виберемо логічні нуль і одиницю

$$|0_L\rangle = |+++\rangle,$$

$$|1_L\rangle = |---\rangle.$$

Після цього усі дії з виправлення помилки абсолютно аналогічні розглянутим раніше при корекції амплітудної помилки. Можливо, тільки схема кодування заслуговує більше детального вивчення і приведена на рис. 9.5. Таким чином ясно, що можна виправити амплітудну і фазову помилки, що виникають в одному кубіті.

Виявляється, що цих ідей вистачає для створення коду, що виправляє будь-яку помилку, що виникає в одному кубіті. Кодуюча схема для дев'яти кубітів була запропонована Шором. Відповідна квантова схема приведена на рис. 9.6. Неважко помітити, що ця схема являється об'єднанням трьох кубітних кодів, що спочатку виправляє фазову помилку. Потім схема трьох кубітів кодується кодом, що виправляє амплітудні помилки. Цей спосіб об'єднання кодів досить часто використовується і в складніших випадках. Таке об'єднання кодів називається каскадним. Інший варіант коду, що виправляє помилки, був запропонований Стеном [20]. Цей семикубітовий код відноситься до широкого класу симплектичних кодів [21]. Цей код також дозволяє виправляти помилку в одному кубіті. Він, як і код Шора, не є оптимальним, але він має корисні властивості, які будуть необхідні надалі. Найоптимальніший код – це п'ятикубітовий код, що дозволяє виправити будь-яку помилку в одному кубіті [21]. Перевірочна матриця або матриця перевірки парності цього коду має вигляд:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

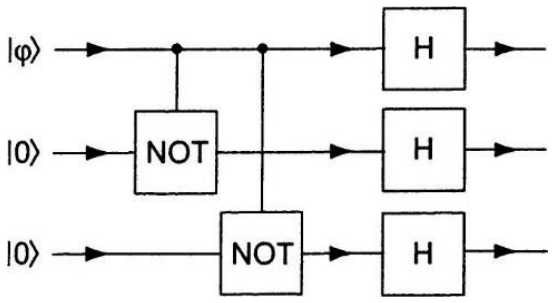


Рис. 9.5. Квантова схема кодування для виправлення помилки у фазі

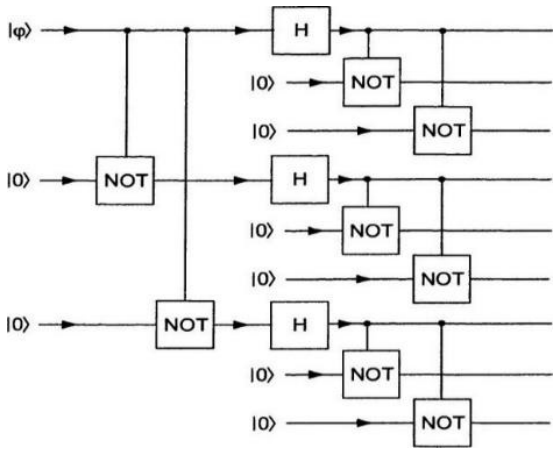


Рис. 9.6. Квантова схема виправлення помилки в одному кубіті

Кодові вектори $|x\rangle$ задовольняють рівнянню $Hx = 0$ або відповідній системі лінійних рівнянь. Розв'язуючи цю систему рівнянь, знаходимо усі кодові слова. Після цього стани, що відповідають логічному нулю і одиниці, вибираються у вигляді суперпозиції цих кодових векторів. Так, логічному нулю відповідає суперпозиція векторів, кодові слова яких містять парне число одиниць, а логічній одиниці – з непарним числом одиниць.

$$|0_L\rangle = \frac{1}{\sqrt{8}}\{|0000000\rangle + |1010101\rangle + |0110011\rangle + \\ +|1100110\rangle + |0001111\rangle + |1011010\rangle + \\ +|0111100\rangle + |1101001\rangle\},$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}\{|1111111\rangle + |0101010\rangle + |1001100\rangle + \\ +|0011001\rangle + |1110000\rangle + |0100101\rangle + \\ +|1000011\rangle + |0010110\rangle\}.$$

Тепер обговоримо визначення синдрому помилки. В принципі, можливість цього можна зрозуміти, повертаючись до перевірочної матриці і помічаючи, що усі стовпці цієї матриці різні. Тому при виникненні помилки в одному кубіті можна завжди виявити її положення. Процедуру визначення синдрому помилки виконаємо ще одним, часто використовуваним способом. Для цього доповнимо початкові кубіти додатковими. Таких додаткових кубітів додамо рівно стільки, скільки синдромів для визначення положення помилки (див. Рис. 9.7).

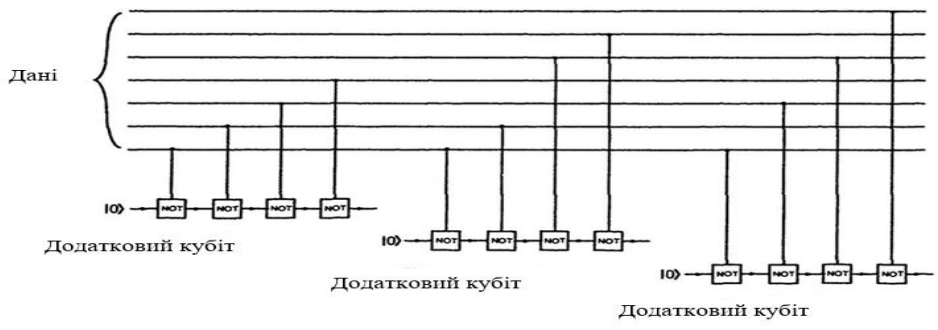


Рис. 9.7. Схема визначення помилки по вимірюванню синдрому помилки в схемі кодування Стена

Тепер, якщо на вході додаткових кубітів встановлений стан $|0\rangle$, а дані - це кодовий вектор $|x_L\rangle$, то на виході стан додаткових кубітів стане $|Hx_L\rangle$. Якщо сталася помилка в одному з кубітів, то вимірювання станів додаткових кубітів редукує вихідні дані робочих кубітів в правильний стан або в стан з перевернутим значенням одного кубіту. При цьому ці вимірювання дозволяють визначити положення кубіту з помилкою. Виправлення його легко здійснити, застосовуючи оператор $\widehat{NOT}$ або $\hat{\sigma}_x$ до цього кубіту. Виправлення фазової помилки також можна здійснити, злегка модифікуючи схему рис. 9.7, заздалегідь пропускаючи дані через гейти Адамара і роблячи аналогічно на виході схеми. Зрозуміло, в цьому розділі обговорювалися тільки найпростіші ідеї, що відносяться до квантових кодів виправлення помилок. Існує велика кількість спеціальної літератури, яка присвячена цій області, що бурхливо розвивається.

9.3 Квантовий ефект Зенона і пригнічення помилок

Існують і інші ідеї пригнічення помилок в квантових системах. Одна з них пов'язана з квантовим парадоксом Зенона або ефектом Зенона. Цей ефект полягає в тому, що вимірювання квантової системи, що повторюються, перешкоджає її переходу в інший стан. Зі збільшенням частоти вимірювань можна перейти до узагальнення вимірювань на безперервне вимірювання стану квантової системи. Різні сценарії експериментальної перевірки цього ефекту пропонувалися різними авторами. Наприклад, в [22,23] пропонувалося використати трирівневу систему, роль якої виконував іон в магнітній пастці. Експериментальна перевірка за близьким сценарієм була здійснена вже в 1990 році.

Обговоримо цей ефект детальніше. Нехай деяка дворівнева квантова система в початковий момент часу $t = 0$ знаходиться в стані $|\varphi_0\rangle = |0\rangle$. Другий ортогональний базисний стан, природно, позначатимемо $|\varphi_1\rangle = |1\rangle$. У моменти часу $t_1, t_2, \ldots, t_N$ система піддається вимірюванню. Зручно припускати, що вимірювання здійснюються через рівні проміжки часу $\Delta t = t_i - t_{i+1} = T / N$. Тут T загальний час, за який здійснено N вимірювань над системою. Безперервне вимірювання відповідає граничному переходу $\Delta t \to 0$ при $N \to \infty$. До першого вимірювання, як втім і між будь-якими вимірюваннями, квантова система еволюціонує згідно з рівнянням Шредінгера. У загальному випадку цю еволюцію можна описати суперпозицією базисних станів, амплітуди яких α і β міняються з часом.

$$|\varphi_t\rangle = \alpha(t)|0\rangle + \beta(t)|1\rangle,$$

Початкові значення цих амплітуд $\alpha(t)|_{t=0} = 1$, а $\beta(t)|_{t=0} = 0$. Ясно, що у момент часу $t_1 = \Delta t$ з вірогідністю $|\alpha(\Delta t)|^2 = 1 - P$ вимірювання виявить стан $|0\rangle$, а з вірогідністю $|\beta(\Delta t)|^2 = P$ – стан $|1\rangle$. При цьому P можна розглядати як вірогідність переходу із стану $|0\rangle$ в стан $|1\rangle$ за час Δt. Легко зрозуміти, що якби при $t = 0$ система знаходилася в стані $|1\rangle$, то вірогідність переходу в стан $|0\rangle$ в точності співпадала би з P. Тепер легко передбачити результат першого вимірювання. Так, початковий стан φ_{i_0} $(i_0 = 0,1)$ при вимірюванні у момент

часу Δt з вірогідністю P поміняється на стан φ_{i_1} $(i_0 \neq i_1)$ і з вірогідністю $Q = 1 – P$ залишиться в попередньому стані. Ясно, що друге і будь-яке подальше вимірювання не відрізняється від першого і тому легко передбачити результати усіх подальших вимірювань $i_0, i_1, i_2, \ldots, i_N$. Вірогідність, що вимірювання дадуть $i_0, i_1, i_2, \ldots, i_N$, наступна:

$$P(i_0, i_1, i_2, \ldots, i_N) = P_1 P_2 \ldots P_N,$$

де, якщо $i_k \neq i_{k-1}$, то $P_k = P$ і якщо $i_k = i_{k-1}$, то $P_k = Q = 1 - P$. Таким чином, ця вірогідність дорівнює

$$P(i_0, i_1, i_2, \ldots, i_N) = P^n Q^{N-n}.$$

Число n дорівнює числу змінювань стану при N вимірюваннях над системою. Підрахуємо вірогідність P_s того, що кінцевий стан співпадатиме з початковим $i_N = i_0$. Ясно, що якщо зміна стану системи сталася парне число разів, то система залишається в початковому стані. Тому вірогідність потрібної події визначається виразом:

$$P_s = \sum_{n=\text{парні}} \binom{N}{n} P^n Q^{N-n}.$$

Тут $\binom{N}{n} = \frac{N!}{n!\,(N-n)!}$ – біномні коефіцієнти. Обчислюючи цю суму, отримаємо вірогідність виживання початкового стану при N вимірюваннях:

$$P_s = \frac{1}{2}(1+(Q-P)^N) = \frac{1}{2}(1+(1-2P)^N).$$

Тепер залишається оцінити вірогідність переходу P при малих $\Delta t = T/N$. Припустимо, що вірогідність такого переходу пропорційна квадрату Δt, тобто,

$$P = \gamma^2 \Delta t^2 = \left(\frac{\gamma T}{N}\right)^2.$$

Тоді вірогідність виживання:

$$P_s = \frac{1}{2}\left(1 + \left(1 - 2\frac{\gamma^2 T^2}{N^2}\right)^N\right)$$

і при великих N, з урахуванням чудового ліміту, перетворимо його до виду:

$$P_s = \frac{1}{2}\left(1 + e^{-2\frac{\gamma^2 T^2}{N}}\right).$$

З цього виразу легко помітити, що при $N \to \infty$

$$P_s \to 1.$$

Це означає, що при безперервному вимірюванні квантової системи стає неможливою зміна її стану. Це і є квантовий ефект Зенона. Помітимо, що при окремому вимірюванні вірогідність зміни стану квантової системи має кінцеве значення. Слід зауважити, що при поясненні ефекту Зенона використовувалося $P = \gamma^2 \Delta t^2$, що виконується, зрозуміло, не для кожної квантової системи. Наприклад, вірогідність розпаду нестабільного стану $P = 1 – e^{\gamma t}$. Легко помітити, що при малих значеннях часу ця вірогідність пропорційна значенням часу. В

цьому випадку питання про ефект Зенона залишається відкритим і вимагає детальнішого аналізу. Ефекту Зенона присвячена значна література, до якої для з'ясування багатьох його аспектів і повинен звернутися зацікавлений читач. Для нас важливе загальне уявлення про нього, що дозволяє уяснити можливість його використання для уникнення помилок. Засновані на ньому сценарії були запропоновані у багатьох роботах. Основна ідея полягає в придушенні декогерентизації прямим вимірюванням системи. До переваг таких методів можна віднести відсутність стадії корекції помилок, що істотно зменшує ресурси, необхідні для здійснення обчислень. Разом з цим є присутніми і очевидні недоліки. Наприклад, неясною є можливість проведення обчислень на тлі "безперервних" вимірювань стану системи.

Да і релізації вимірювань часто вносять додаткові канали декогерентизації. Швидше можна чекати корисності ефекту Зенона для тривалого зберігання інформації в кубітах квантового комп'ютера.

9.4 Стійкі квантові гейти

У попередньому розділі обговорювався тільки один аспект проблеми появи помилок. Вважалося, що квантові гейти не вносять помилок, іншими словами, є ідеальними, а помилки виникають при передачі по зашумленим каналам зв'язку або при тривалому зберіганні (див. Рис. 9.8).

Проте, така картина далека від реальності. Природно, усі квантові операції, включаючи ті, що навіть здійснюють визначення синдрому помилки, вимірювання і ініціалізацію, самі по собі вносять шум і, отже, служать джерелами помилок. Тому дуже важливо, щоб способи виправлення помилок вносили менше помилок, ніж виправляють. Тільки у такому разі можна сподіватися на виконання досить тривалих обчислень. Для цього, передусім, слід потурбуватися про створення таких квантових гейтів, які не призводили б до поширення і збільшення числа помилок. Звернемо увагу, що такий важливий гейт, як $\widehat{CNOT}$, не має потрібних якостей.

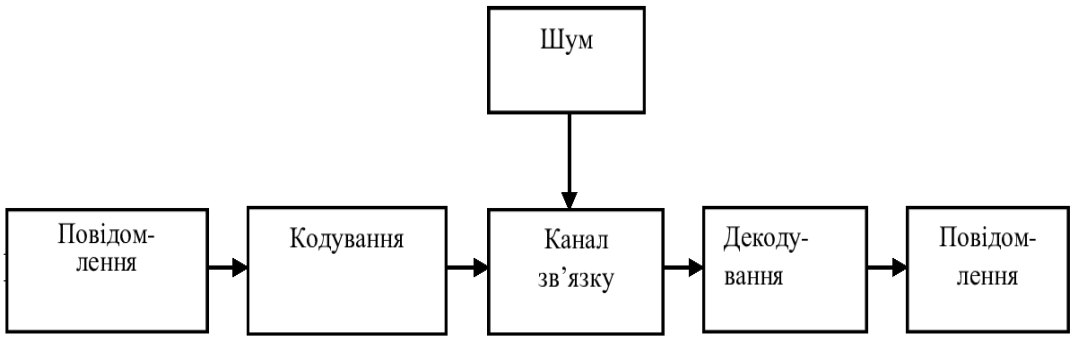


Рис. 9.8 Блок - схема виникнення і боротьби з помилками при передачі по каналах зв'язку з шумом

Дійсно, якщо розглянути обробку помилки цим гейтом, то неважко переконатися в її розмноженні (див. Рис. 9.9). Тому реалістичні квантові

обчислення потребують розробки завадостійких квантових гейтів і вимірювань. Такі гейти і вимірювання повинні перешкоджати поширенню помилки і збільшенню їх числа не більше, ніж до такого рівня, який може бути виправлений кодами, що коригують помилки. Одна із загальних ідей для створення таких гейтів полягає в здійсненні, що відповідає операції безпосередньо над закодованими станами. В цьому випадку декодування стає непотрібним. Згідно цієї ідеології кожен кубіт замінюється блоком кубітів.

При цьому використовується код, що виправляє помилки. Після цього кожен квантовий гейт замінюється на схему, що виконує над блоком кубітів дію, що відповідає дії цього гейта над кубітом.

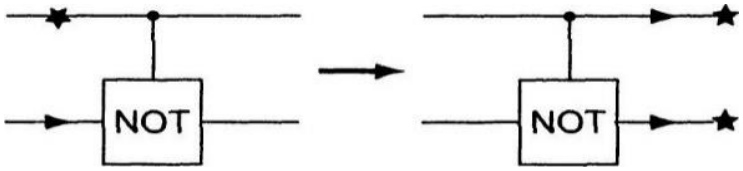


Рис. 9.9. Показаний квантовий гейт $\widehat{CNOT}$, який при надходженні помилки на контролюючий кубіт передає її і на контрольований кубіт. Символічно наявність помилки помічена зірочкою $\widehat{CNOT}$

Такі схеми називаються закодованими логічними елементами. Після застосування закодованого елементу знову використовується код виправлення помилок, що запобігає накопиченню помилок. Зрозуміло, важливо, щоб використовувані схеми не призводили до поширення і розмноження помилок. Такі схеми, які замінюють гейти, називаються елементами, стійкими до помилок. Відмітимо, що навіть така передбачливість не вирішує усіх проблем. Залишається ще потурбуватися, щоб коди, що виправляють помилки, самі не вносили їх в закодовані кубіти, або, точніше, не вносили надто багато помилок. Принаймні вони повинні виправляти більше помилок, ніж вносити. Неважко помітити, що така стратегія вимагає збільшення ресурсів і числа операцій.

При використанні одноступінчатого кодування число кубітів збільшується на порядок, а при n-ступінчатому у 10^n разів. Тому центральне питання полягає в можливості досягнення стабілізації обчислень, використовуючи прийнятні ресурси. Зараз вважається, що квантові обчислення можуть виконуватися необмежено довго, якщо вірогідність помилки на одній будь-якій елементарній операції не перевищує критичного рівня p_c. Значення цього порогового рівня за деякими оцінками складає $p_c \leq 10^{-5} \div 10^{-4}$. Продемонструємо створення завадостійких гейтів, починаючи з гейта Адамара. Для зручності використовуємо семикубітний код Стена. Кодування станів при такому способі кодування здійснюється згідно (див. розділ 9.3)

$$|0_L\rangle = \frac{1}{\sqrt{8}}\{|0000000\rangle + |1010101\rangle + |0110011\rangle +$$
$$+|1100110\rangle + |0001111\rangle + |1011010\rangle +$$
$$+|0111100\rangle + |1101001\rangle\},$$

$$|1_L\rangle = \frac{1}{\sqrt{8}}\{|1111111\rangle + |0101010\rangle + |1001100\rangle +$$
$$+|0011001\rangle + |1110000\rangle + |0100101\rangle +$$
$$+|1000011\rangle + |0010110\rangle\}.$$

Легко безпосередньо перевірити, що схема, яка приведена на рис. 9.10, діє на ці стани цілком аналогічно гейту Адамара. Залишається переконатися, що він завадостійкий. Під завадостійким гейтом можна розуміти таку схему, в якій збій, що стався в одній з її компонент, приводить не більше, ніж до однієї помилки у вихідному блоці кубітів. Під компонентою розуміється будь-яка елементарна операція, гейти, вимірювання і тому подібне. Дивлячись на приведену схему (рис. 9.11), можна помітити повну відсутність взаємодії кубітів. Це означає, що поява помилки в одному із вхідних кубітів не призводить до її розмноження. Помилковим на виході буде вміст тільки цього кубіту. В принципі, ця схема демонструє загальний принцип побудови завадостійких гейтів. Якщо вдається побудувати гейт, що діє побітно, то така схема дає завадостійкий гейт. Ясно, що такі схеми не збільшують число помилок і, відповідно, вони можуть виправлятися кодами, що коригують помилки.

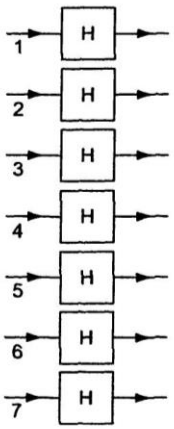


Рис. 9.10. Квантова схема, яка діє як гейт Адамара
при використанні семикубітного коду Стена

$$U|0_L\rangle = \frac{1}{\sqrt{2}}(|0_L\rangle + |1_L\rangle),$$
$$U|1_L\rangle = \frac{1}{\sqrt{2}}(|0_L\rangle - |1_L\rangle).$$

Іншим з цікавих ресурсів для створення завадостійких гейтів і вимірювань служить квантова телепортація. Такі квантові гейти з використанням однобітової телепортації були запропоновані в багатьох роботах, наприклад [24, 25]. Ці гейти дозволяють здійснювати операції над віддаленими один від одного кубітами. На закінчення розділу відмітимо, що обговорювані в ньому питання ще не досить усвідомлені і дослідження в цьому напрямі інтенсивно розвиваються.

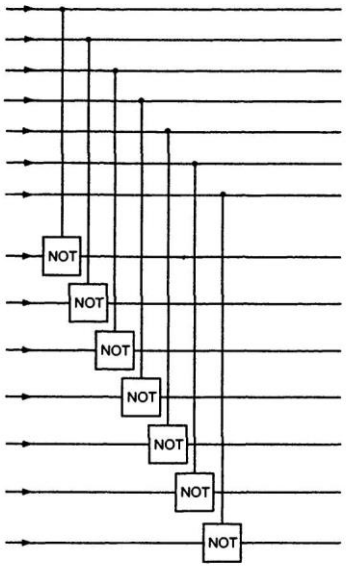


Рис. 9.11. Схема, що відповідає завадостійкому CNOT, що контролюється

10 ВСТУП В ТЕОРІЮ ТА ПРАКТИКУ КВАНТОВИХ ОБЧИСЛЕНЬ

Розділ присвячений розгляду найсучаснішого напрямку науки – квантової інформатики та її найбільш важливих додатків, квантової криптографії та квантового криптоаналізу.

Ще в 1965 році майбутній співзасновник компанії INTEL Гордон Мур прогнозував щорічне подвоєння щільності транзисторів в напівпровідникових мікросхемах, тобто, експонентне зменшення розмірів елементів. Закон Мура справджується вже протягом 56 років і на сьогодні розмір елементів транзистора (і, відповідно, розмір області, в якій зберігається одиниця інформації - біт) становить долі мікрона. У зв'язку з цим виникає питання про принципові обмеження на розміри, швидкодію та теплообмін між елементами комп'ютерних схем. На думку фізика Р. Фейнмана "закони фізики не заперечують зменшення розмірів комп'ютера до тих пір, поки біти не досягнуть розмірів окремих атомів і закони квантової механіки не стануть домінуючими". В той час стало питання, чи можливо створити комп'ютер, в якому поведінка системи бітів буде визначатись законами квантової механіки? Процес обчислень завжди вважався виключно класичним процесом. Робота окремих елементів квантового комп'ютера описується законами квантової фізики для звичайних обчислень стану окремого біта та передбаченістю результату операції над ним. Причому визначення стану біта (фізичне вимірювання) не впливає на його стан, різні фізичні реалізації логічної операції призводять до однакової зміни стану біта, і це робить можливим розглядати процес обчислень на класичному комп'ютері незалежно від його апаратного втілення.

В основі ж квантової механіки лежить, як ми відмічали раніше, *принцип невизначеності Гейзенберга* та ймовірністний характер станів окремих елементів, а також знищення стану квантового об'єкта (біта) при визначенні стану (вимірюванні), що, здається, знаходиться в повному протиріччі із вимогою повної визначеності, однозначності та зворотності комп'ютерних операцій.

Однак, на початку 80-х Полю Беньоффу (Paul Benioff), Річарду Фейнману та Давіду Дейчу (David Deutsch) вдалось поєднати дві дисципліни, що досі вважались взаємовиключними, квантову фізику та інформатику. Вони показали, що квантова теорія не тільки не обмежує обчислювальних можливостей, але дозволяє суттєво їх розширити. Беньофф висловив ідею універсального квантового комп'ютера машини, яка здійснює логічні операції, спираючись на квантові алгоритми, що не мають аналогів в класичній фізиці. Фейнман показав, що квантовий комп'ютер повинен бути більш потужним, ніж класичний, а Д. Дейч запровадив ідею квантової паралельності, що описує здатність квантового комп'ютера працювати із квантовою суперпозицією чисел. Тим самим ці вчені заклали фундамент нової сучасної галузі досліджень - квантових інформаційних технологій або квантової інформатики.

На початку 90-х квантова інформатика набула прикладного змісту завдяки роботам Екерта (A. K. Ekert), Беннетта (C. H. Bennett) та групи з Женевського університету. Виявилось, що, такі "недоліки" квантового об'єкту, як можливість

одночасного перебування у сукупності станів та суттєвий вплив процесу вимірювання на стан об'єкта, відкривають нові можливості в криптографії.

Досягнення фізики останніх років (бозе-ейнштейнівська конденсація атомів газу, квантовий ефект Хола, штучні періодичні структури – квантові точки, колодязі тощо), а також розвиток лазерних та оптоволоконних технологій зробили можливим реалізацію найближчим часом квантового комп'ютера. На сьогодні квантова інформатика вбирає в себе досягнення різних дисциплін - фізики, теорії інформації, обчислювальних методів, телекомунікацій, матеріалознавства тощо і переживає період бурхливого розвитку, який можна порівняти тільки із розвитком ідей квантової фізики в кінці 19-го - початку 20-го сторіччя.

Даний розділ має на меті ознайомлення із основними проблемами та напрямками розвитку квантової інформатики.

Деякі питання квантової механіки

Квантова інформатика в повній мірі використовує властивості квантових об'єктів, такі, як:

- здатність однієї частки перебувати в кількох станах одночасно (суперпозиція станів об'єкта);
- здатність системи із кількох часток перебувати в корельованиих (переплутаних) станах і пов'язана з цим нелокальність;
- суттєвий вплив процесу вимірювання на стан об'єкта;
- неможливість клонування квантових станів.

Розглянемо ці властивості докладніше.

Принципова відмінність квантової фізики від класичної полягає в тому, що:

1) Закономірності квантової фізики носять статистичний (ймовірні-стний), а класичної фізики - динамічний (детерміністичний) характер;

2) Процес вимірювання стану мікрооб'єкта істотно впливає на його стан і тому в квантовій фізиці, на відміну від класичної, неможливо абстрагуватися від фізичної природи процесу вимірювання. Внаслідок цього, стан квантового об'єкта описується через так звані базисні стани, що задаються можливостями вимірювального приладу.

Пояснимо це на прикладі. Так, стан фотона (кванта світла) заданої частоти характеризується його поляризацією. Поляризаційний стан фотона визначається за допомогою поляризатора-аналізатора, вісь якого ми можемо розташувати довільним чином. Припустимо, вісь поляризатора розташована вертикально. Якщо досліджуваний фотон був поляризований вертикально, то він пройде через поляризатор, якщо горизонтально, то не пройде. Таким чином, можливі стани поляризатора утворюють *базис;* вертикальній поляризації відповідає вектор стану $|1\rangle$, горизонтальній поляризації – вектор стану $|0\rangle$ (Використані позначення для так званих векторів стану були вперше запроваджені Діраком).

Стан фотона описують за допомогою амплітуди ймовірності або хвильової функції, яка в зазначеному вище базисі має вигляд:

$$|\psi\rangle = a|0\rangle + \beta|1\rangle\ . \qquad (10.1)$$

Коефіцієнти α та β можуть бути комплексними. Відповідно, безпосередньо хвильова функція та коефіцієнти не є *спостережуваними* (їх неможливо виміряти). Із результатів вимірювання можна лише визначити квадрати модулів $|a|^2$ та $|\beta|^2$, що характеризують ймовірність знайти систему в стані $|0\rangle$ або $|1\rangle$ відповідно.

Так, для вертикально поляризованого фотона $\alpha = 0, \beta = 1$, для горизонтально поляризованого, навпаки, $\alpha = 1, \beta = 0$. Якщо фотон був поляризований під кутом 45° до вертикальної осі, то $\alpha = \beta = 1/\sqrt{2}$, тобто, фотон з однаковою ймовірністю ½ пройде чи не пройде через поляризатор. Зазначимо також, що якщо обидві величини, *a* та β, відмінні від нуля, то кажуть, що фотон знаходиться в *суперпозиційному* стані, а якщо одна з них дорівнює нулю (тобто, в результаті експерименту ми визначаємо стан фотона з ймовірністю 1), то в *чистому* стані. Зауважимо, що введені поняття пов'язані із вибором базису вимірювального приладу. Процес вимірювання за допомогою поляризатора переводить (проєктує) фотон із стану $|\psi\rangle$ в стан $|1\rangle$ або знищує його. Повторне вимірювання не дає нової інформації про $|\psi\rangle$, оскільки стан фотона змінюється вже після першого вимірювання.

Квантові властивості притаманні і системам, що складаються з кількох квантових об'єктів. В таких системах можна спостерігати ефекти кореляції, які носять нелокальний характер. Знову звернемося до прикладу – процесу параметричної генерації. Він полягає в тому, що кристал з нелінійними оптичними властивостями (скажімо, KO_3) опромінюють лазерним випромінюванням оптичного діапазону (довжина хвилі 655 нм). На виході кристала утворюються два пучки інфрачервоного діапазону (довжина хвилі 1310 нм) із двічі меншою частотою. Цей процес можна представити як утворення двох "інфрачервоних" фотонів із одного "оптичного". В залежності від взаємної орієнтації кристала та поляризації вхідного випромінювання можливі дві принципово різні ситуації:

1) інфрачервоні фотони ведуть себе незалежним чином, як класичні частки, а вся система знаходиться в так званому *чистому* стані;

2) між інфрачервоними фотонами існує кореляція і стани фотонів *переплутані.*

Розглянемо ці ситуації докладніше.

а) Незалежність фотонів означає, що вони мають різні, але визначені (за орієнтацією кристала) поляризації, наприклад, перший фотон знаходиться в стані $|1\rangle$, а інший - в стані $|0\rangle$. При вимірюванні стану одного фотона, скажімо, першого, ми не отримуємо інформації про стан іншого фотона, але при цьому і не змінюємо його. В цьому випадку як система, так і кожна з часток перебувають в чистому стані.

Хвильова функція такої системи є добутком хвильових функцій її елементів, наприклад,

$$|\psi\rangle = |1\rangle_1 \, |0\rangle_2 \,, \qquad (10.2)$$

де нижні індекси позначають перший та другий фотон.

б) В разі корельованих фотонів обидві частки мають різні поляризації, але неможливо *апріорі* встановити, яка з них в якому стані. Кажуть, що фотони знаходяться в *переплутаних* (*entangled*) станах, хоча система в цілому може знаходитися в чистому стані. Головною особливістю переплутаних станів є те, що, вимірюючи стан одного фотона, ми одразу ж змінюємо стан іншого і втрачаємо інформацію про стан системи в цілому. З іншого боку, для визначення стану системи необхідно одночасно зробити вимірювання за обома фотонами. Описана кореляція між фотонами зберігається навіть при розповсюдженні фотонів в різних напрямках (навіть при відстані між фотонами більше, ніж 10 км). Поняття переплутаних станів може бути легко узагальнене на системи, що складаються із більшої кількості часток.

Хвильова функція системи в переплутаних станах не може бути представлена у вигляді добутку хвильових функцій окремих елементів в будь-якому базисі. Типовий приклад переплутаних станів системи з двох фотонів задається чотирма так званими *станами Белла* (див. підрозд. 4.5)*:*

$$|\psi_+\rangle = \left(1/\sqrt{2}\right)\left\{|1\rangle_1|0\rangle_2 + |0\rangle_1|1\rangle_2\right\}, \tag{10.3}$$

$$|\psi_-\rangle = \left(1/\sqrt{2}\right)\left\{|1\rangle_1|0\rangle_2 - |0\rangle_1|1\rangle_2\right\}, \tag{10.4}$$

$$|\phi_+\rangle = \left(1/\sqrt{2}\right)\left\{|1\rangle_1|1\rangle_2 + |0\rangle_1|0\rangle_2\right\}, \tag{10.5}$$

$$|\phi_-\rangle = \left(1/\sqrt{2}\right)\left\{|1\rangle_1|1\rangle_2 - |0\rangle_1|0\rangle_2\right\}. \tag{10.6}$$

Підкреслимо, що переплутаність станів і ефекти кореляції спостерігаються лише в системах, в яких частки слабо взаємодіють одна з однією і з навколишнім середовищем. З ростом кількості часток взаємодія між частками та часток з навколишнім середовищем призводить до швидкого знищення когерентності.

Врешті-решт, відзначимо ще дві особливості квантових станів:

1) Стан квантової системи неможливо клонувати. Інакше кажучи, якщо ми маємо фотон у довільному і невідомому нам стані, ми не можемо створити ще один фотон в точнісінько такому ж стані, не знищивши перший фотон. Якби клонування було можливим, то створюючи копії квантового об'єкта, ми б могли вимірювати одночасно такі параметри стану, які відповідають некомутуючим операторам, а це заборонено принципом невизначеності Гейзенберга.

2) Стан квантової системи із часом змінюється завдяки впливу зовнішніх тіл (процес декогеренції). Будь-яка ізольованість квантової системи носить відносний характер.

Призначення квантового комп'ютера

Основне призначення квантового комп'ютера полягає не тільки в прискоренні обчислень за допомогою квантових ефектів, а у використанні нових квантових алгоритмів, принципово неможливих в класичних обчислювальних машинах.

До 1994 року квантовий комп'ютер вважався прекрасним, але суто теоретичним здобутком, оскільки не було розроблено квантових алгоритмів, які

б були здатними розв'язувати якісь цікаві та нагальні проблеми, а також із-за відсутності належних фізичних систем для апаратної реалізації такого комп'ютера. Але в 1994 році Пітер Шор запропонував ефективні квантові алгоритми розв'язку двох проблем асиметричної криптографії - проблеми факторизації цілих чисел, а також обчислення логарифма від дискретних чисел. Після вказаного Цірак та Золлер розробили апаратне забезпечення квантового комп'ютера, користуючись відомою технікою маніпулювання іонами в пастках. Досягнення сучасної мікроелектроніки та матеріалознавства роблять можливим реалізацію квантового комп'ютера на напівпровідникових кристалах з так званими квантовими точками. Нижче наведено загальні відомості про принцип дії та апаратне втілення квантового комп'ютера.

Особливості кодування інформації на квантових об'єктах

В класичному комп'ютері за одиницю інформації вважають один біт, який реалізується як класична дворівнева система, що може з певністю перебувати в одному із двох станів - "0" або "1". Стан такої системи описується одним числом, отже, за допомогою N бітів можна закодувати *одне* з 2^N чисел.

На відміну від класичних, квантові системи відкривають великі можливості для збільшення щільності кодування та паралельної обробки інформації. Так, в квантовому комп'ютері елементарною одиницею інформації вважають кубіт (qubit), тобто, квантовий біт (термін був введений в 1995 році Шумахером). Кубіт реалізується як квантова дворівнева система, що може перебувати не тільки в чистих станах $|0\rangle$ та $|1\rangle$, але й в суперпозиції цих станів $\alpha\,|0\rangle + \beta\,|1\rangle$.

Таким чином, стан кубіта описується двома числами, тому один кубіт містить вдвічі більше інформації, ніж класичний біт, а для системи з N кубітів зростання щільності кодування інформації у порівнянні з класичним буде експоненційно великим. Щоб пояснити істотне зростання щільності кодування, необхідно звернутися до схеми телепортації (див. розд. 8). Відомо, що за допомогою класичної дворівневої системи можна переслати один біт інформації. У випадку, якщо ж для передачі інформації скористатись квантовим об'єктом, наприклад, фотоном, що знаходиться в переплутаному з іншим (базовим) фотоном стані, то за допомогою фотона можна передати якнайменше $\log_2 3 = 1{,}58$ бітів. Дійсно, система з двох переплутаних фотонів може знаходитись в одному з 4-х станів Белла (10.3-10.6), які можна розрізнити експериментально. Як вже зазначалось, для того, щоб перевести пару фотонів з одного стану Белла в інший, достатньо змінити стан одного фотона. Отже, приймаючи один із станів (10.3-10.6) за основний (несучий) і маніпулюючи одним із пари фотонів, можна закодувати одне з трьох повідомлень (станів) і в такий спосіб передати повідомлення в 1,58 біт. Використання системи із N корельованих часток дозволяє підвищити щільність кодування до $\log_2(2^N\text{-}1) \approx N$ біт, порівняно із $\log_2 N$ біт для класичної системи.

Також важливою властивістю квантових об'єктів є можливість здійснювати паралельні операції. Для цього розглянемо систему із N кубітів, що перебуває в переплутаному стані:

$$|\psi\rangle = 2^{-N/2}\{|00000\ldots.0\rangle|00000\ldots.01\rangle + \cdots.. + |00001011\ldots.10\rangle + \cdots\}. \quad (10.7)$$

Ефективно така система кодує відразу 2^N чисел. Операція над нею, завдяки когерентності станів різних кубітів, впливає на всі доданки в сумі (10.7) і це дозволяє обробляти відразу всі 2^N чисел.

Алгоритми квантових обчислень

Для реалізації будь-якого комп'ютера поряд із носієм інформації необхідна можливість виконувати логічні операції над ним. Бажано, щоб такі операції могли бути зворотними. Під зворотністю розуміють можливість відтворити вихідні дані за кінцевими, тобто, результатом обчислень. В класичному комп'ютері зворотність реалізують введенням додаткових регістрів. Так, при додаванні двох чисел, a і b, результат, $c = a \oplus b$ розміщують в регістрі b, а вміст регістра a залишається незмінним, тобто, схематично: $|a, b\rangle \Rightarrow |a, a \oplus b\rangle$ (Символ $\oplus$означає додавання за модулем 2). Така процедура дозволяє за сумою $a \oplus b$ та значенням a знайти b. На щастя, зворотність квантово-механічних рівнянь дозволяє виконувати аналогічні зворотні операції і над кубітами. Логічні операції за типом операції додавання іноді ще називають логічними вентилями. Квантові логічні вентилі реалізуються, як унітарні перетворення над кубітами, які можна здійснювати за допомогою імпульсів з правильно підібраною тривалістю та несучою частотою. Можливість реалізації логічних вентилів було продемонстровано тими ж самими розробниками ідеї квантового комп'ютера– Дейчем, Екертом та ін. Таким чином, квантовий комп'ютер може відтворювати ті ж самі операції, що і класичний, і в цьому процес програмування не залежить від природи і типу фізичного носія інформації.

Більш вражаючою є можливість застосування суто квантових алгоритмів. На сьогодні їх відомо лише кілька (це, як правило, алгоритми, що стосуються розв'язання задач, які вважаються експоненційно складними для класичних комп'ютерів: факторизація великих чисел, пошук за базою даних, задача "комівояжера"), і саме тут відкривається широкий простір для винахідників.

Взагалі, квантові алгоритми дуже різні для різних задач і сильно залежать від їх втілення на конкретних об'єктах, але всі вони призводять до зменшення складності задачі (збільшення ефективної швидкості) від експоненційної до поліноміальної, і все це - завдяки інтерференційним властивостям квантових об'єктів. Не зупиняючись на деталях алгоритмів, описаних в спеціальній літературі, проілюструємо різницю між класичним та квантовим алгоритмом на простій аналогії. Припустимо, у нас є екран з двома щілинами і необхідно визначити, чи обидві щілини відкриті, чи одна з них (будь-яка) закрита. Класичний алгоритм будується на перевірці по черзі кожної з щілин, для отримання відповіді необхідно загалом зробити два виміри. Аналог квантового алгоритму полягає в тому, щоб освітити одночасно обидві щілини, і якщо за екраном буде спостерігатись інтерференція, то обидві щілини відкриті, якщо ні, то, принаймі, одна з них закрита. Для отримання відповіді в цьому випадку достатньо зробити один вимір.

Це пов’язано із квантовим характером явища - квантова частка одночасно випробує всі альтернативи (тобто, можливість пройти через одну та другу

щілину) і реалізує всі можливості, що мають однакову ймовірність. Очевидне збільшення ефективності квантового алгоритму тим більше, чим більша кількість альтернатив («щілин»), що їх перебирає система за один крок. З іншого боку, з квантовою природою кубітів пов’язані і певні ускладнення. Дійсно, завдяки суперпозиції та інтерференції, операції над квантовими об’єктами значно прискорюють процес обробки квантової інформації, але для її зчитування необхідно перевести квантову інформацію в класичну, тобто, провести вимірювання стану. Однак, чим з більшої кількості часток складається система, тим більше можливих станів вона має і тим більше вимірювань необхідно виконати, щоб встановити достеменну відповідь. Вибір ефективного алгоритму, таким чином, у великій мірі спирається на визначення найбільш ефективного способу вимірювання кінцевого результату.

Квантові алгоритми Шора і Гровера розглянуті детально авторами нижче.

Фізична реалізація квантового комп'ютера

Основна проблема у побудові квантового комп'ютера полягає в тому, що необхідно створити систему, яка б задовольняла майже несумісним таким вимогам:

- кубіти (елементи квантового комп'ютера) повинні бути максимально ізольовані один від одного та від навколишнього середовища;
- можливість корельованого впливу на пару кубітів, тобто, потрібно не тільки вміти змінювати стан одного кубіту, але й ще вмикати та вимикати взаємодію між парою сусідніх кубітів;
- система кубітів повинна бути досить стабільною, щоб зберігати корельованість станів, але одночасно й легко відновлюваною для нового циклу обчислень;
- реалізовувати таку сукупність зворотних перетворень над системою кубітів, які б дозволили виконати будь-яку потрібну логічну операцію;
- під час обчислень система повинна зберігати квантові властивості, але наприкінці треба зробити вимірювання, яке б однозначно визначило стан системи і у такий спосіб перевело б квантову інформацію в класичну.

На сьогодні існує кілька підходів до проблеми реалізації квантового комп'ютера, а саме, твердотільні моделі з використанням так званих (т.з.) квантових точок (мікроскопічних областей в багатошарових плівках, в яких може бути локалізований окремий електрон), системи надпровідних гранул, в яких основним елементом (кубітом) служить куперовська пара електронів тощо. Найбільш розроблена ідея комп'ютера на окремих атомах (або іонах). В її основі лежать зроблені в останні 5 років відкриття, що стосуються методів лазерного охолодження атомів та іонів (М. А. Андерсон, 1995). Завдяки цим методам іони можна зупинити до температури (середньої кінетичної енергії) майже 100 нК і утримувати в такому стані за допомогою магнітного поля протягом 1 сек!

Отже, квантовий комп'ютер - це вже реальність. Звісно, в указаній схемі виникає безліч технічних проблем. Так, на сьогодні кількість іонів (і, відповідно, кубітів), якими вдалось маніпулювати, досить обмежена (всього ЗО іонів), але кілька років тому могли маніпулювати лише 4-5. Є певні проблеми із зчитуванням спектра флюоресценції – це дуже слабкі сигнали, мають місце

ефекти насичення і таке інше, але статистична обробка результатів вимірів з використанням звичайного комп'ютера дозволяє отримати відповідь (зчитати інформацію) з великою точністю. Навіть незважаючи на вакуум, іони швидко "гріються" і стан системи розпадається, але за час релаксації встигають зробити до мільйона операцій. Крім того, використання спеціальних процедур корекції помилок, які виникають завдяки хаотизації системи, дозволяють значно збільшити час роботи квантового комп'ютера і підвищити надійність результатів. Докладніше про це - в наступному підрозділі.

Декогеренція і квантова корекція помилок

Як було зазначено вище, в основі квантових обчислень лежить їх властивість знаходитись в переплутаних станах. Однак, для ефективного використання цих властивостей необхідно, щоб квантова система не взаємодіяла із навколишнім середовищем, оскільки останнє знищує когерентність станів. Такий процес називають процесом *декогеренції.* Оскільки повністю виключити взаємодію із середовищем неможливо, то намагаються, по-перше, використовувати такі об'єкти, стани яких "розпадаються" не надто швидко (наприклад, збуджені стани іонів Ca або Be "живуть" секунду, за яку можливо виконати мільярд операцій). Інша можливість боротьби з декогеренцією спирається на процедуру корекції помилок.

Процедура корекції помилок широко застосовується в класичних комп'ютерах, оскільки і там, завдяки різним збоям, виникають помилки при передачі та обробці даних. В «класиці» проблему корекції розв'язують, вводячи додатковий «контрольний» біт, тобто, якимось чином дублюючи інформацію. Безпосереднє застосування такої процедури в квантовому комп'ютері неможливо, оскільки в процесі обробки інформації система повинна визначити значення "контрольного" біта, отже, зробити вимір і в такий спосіб знищити стан, що вимірюється. Отже, квантова корекція помилок повинна спиратись на процедуру, яка б дозволила, з одного боку, встановити, чи змінилось самовільно значення кубіту і якщо так, то змінити його на правильне, а з іншого боку, не вимірювати безпосередньо стану кубіту, щоб не знищити інформацію.

Принцип квантової корекції помилок був запропонований Шором та Стеаном (A. M. Stean). Для цього первісну інформацію кодують надлишково, наприклад, замість одного кубіта використовують три: стан $\alpha|0\rangle + \beta|1\rangle$ кодують векторами $\alpha|000\rangle + \beta|111\rangle$. Обробка інформації побудована таким чином, щоб під дією операторів перетворень всі три інформаційні кубіти змінювались однаково. Декогеренція може призвести до зміни стану одного з кубітів, наприклад, $|000\rangle \Rightarrow |010\rangle$.

Для визначення помилки з'ясовують, чи знаходяться всі три частинки (кубіти) в однаковому стані, чи ні. Таке вимірювання можна зробити, наприклад, використовуючи логічний вентиль "НІ з контролем". Як вже згадувалось, таку операцію легко здійснити за допомогою лазерних імпульсів, вона зворотна і тому не знищує інформаційний стан.

Якщо вимірювання показує наявність помилки, корекція легко здійснюється за допомогою зміни стану одного із кубітів на альтернативний.

Квантова корекція помилок під час процесу обчислення, хоча і потребує значного дублювання інформації і, як наслідок, збільшення оперативного об'єму системи, дозволяє значно збільшити тривалість процесу обчислень (кількість операцій до розпаду системи) і зробити роботу квантового комп'ютера майже незалежною від впливу зовнішнього середовища.

10.1 Основні поняття і сутність квантової криптографії

Криптографія (тайнопис) являє собою один із напрямків криптології – науки, що займається проблемами захисту інформації шляхом її перетворення. Основні напрямки застосування криптографічних методів – передача конфіденційної та достовірної інформації по каналах зв'язку (наприклад, через електронну мережу), встановлення достовірності та автентичності (достеменності) повідомлення, що передається, зберігання інформації (документів, баз даних) на носіях в зашифрованому вигляді тощо. Серед проблем, які вирішує криптографія, найбільш вразливою завжди була проблема розподілу ключів. Ключ – це інформація (набір кодів), за допомогою якої здійснюється процедура шифрування даних при обміні між різними суб'єктами (сторонами).

Як правило, зашифрована інформація передається по відкритих каналах, а ось обмін ключами між суб'єктами повинен відбуватись суворо конфіденційно, тобто, особисто або по закритих каналах.

В поширеному протоколі, що обумовлюється так званим стандартом БЕЗ, довірені сторони повинні мати однаковий ключ для зашифрування і розшифрування інформації. Для забезпечення стійкості криптосистеми відносно можливих атак (тобто, для виключення розшифрування інформації сторонніми особами) ключ повинен бути випадковим та абсолютно секретним. Найскладнішою є вимога забезпечення абсолютної секретності ключа - навіть при передачі ключа по закритим каналам не виключена можливість непомітного для довірених сторін несанкціонованого втручання в канал сторонніх осіб (порушників).

Квантова криптографія значно полегшує задачу секретної передачі інформації. Вона вирішує проблему розподілу ключів, використовуючи властивість квантових систем змінювати свій стан в процесі вимірювання та неможливість клонування квантових станів. Процес "підслуховування" передбачає можливість копіювання (відтворення) стану носія інформації, що в разі передачі інформації за допомогою суто квантових об'єктів неможливо. Отже, будь-яка спроба несанкціонованого втручання в канал передачі інформації буде виявлена.

Основні фундаментальні властивості квантових систем, які використовуються в квантовій криптографії, наступні:

1) Вимірювання фізичних характеристик квантових систем (спостережуваних).

У результаті процесу вимірювання деякої фізичної величини стан квантової системи змінюється. Це обумовлено впливом на квантовий об'єкт

вимірювального приладу, який принципово неможливо зробити як завгодно слабким. Чим точніше вимірювання, тим сильніший вплив, що воно здійснює, і лише при вимірюваннях дуже малої точності вплив на об'єкт вимірювання може бути досить слабким.

Крім того, збурювання, яке вноситься взаємодією квантового об'єкта з вимірювальним приладом, може бути передбачено тільки статистично й тому не може бути виключено. Цей факт перебуває в різкому протиріччі із класичною теорією вимірювань, яка базується на припущенні, що взаємодія між об'єктом і приладом якщо й не може бути зроблена як завгодно малою, то принаймні може бути точно врахованою й, отже, у принципі її можна виключити.

На рис.10.1 приведена класифікація квантових технологій.

2) Неможливість точного клонування невідомих квантових станів (теорема про заборону клонування).

Внаслідок лінійності й унітарності квантової механіки неможливо створити точну копію невідомого квантового стану. Таким чином, зловмисник не може виготовити точну копію кубітів або кудитів, що передаються по комунікаційному каналу, щоб провести вимірювання над копією, а оригінал переслати законному користувачеві каналу, не проводячи над ним вимірювання. Цей факт лежить в основі більшості протоколів квантової криптографії, тому що змушує зловмисника вимірювати передавані кудити або переплутувати їх зі своїми допоміжними квантовими системами, що призводить до зміни станів цих кудитів. Ці зміни станів, що передаються, можуть виявити законні користувачі, виконуючи квантові вимірювання й обмінюючись результатами цих вимірювань по звичайному відкритому каналу зв'язку. Відзначимо, що ймовірність правильно клонувати довільний стан кубіту, створивши одну його копію, дорівнює 5/6.

Якщо потрібно створити n копій невідомого стану кубіту, то ймовірність правильного клонування зменшується та при $n \to \infty$ прямує до 2/3 .

3) Неортогональні квантові стани неможливо розрізнити.

Квантова система із двома станами кубітів може перебувати не тільки в базисних станах $|0\rangle$ та $|1\rangle$ (які відповідають, наприклад, вертикальній та горизонтальній поляризації окремого фотона), але й у стані лінійної суперпозиції

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad (10.8)$$

де α і β – комплексні числа, що задовольняють умові $|\alpha|^2 + |\beta|^2 = 1$.

Вимірюючи стан кубіту, ми знайдемо, що кубіт з імовірністю $|\alpha|^2$ несе значення "0", а з імовірністю $|\beta|^2$ – значення "1".

Внаслідок законів квантової механіки неможливо виконати вимірювання, що дозволило б розрізнити стани $|\Psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ та $|\Psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, крім випадку, коли скалярний добуток $\langle\Psi_1|\Psi_2\rangle = 0$, тобто, стани $|\Psi_1\rangle$ і $|\Psi_2\rangle$ ортогональні.

Аналогічно, вимірюючи довільний стан кудиту

$$|\Phi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle, \tag{10.9}$$

ми отримаємо "0" з імовірністю $|\alpha|^2$, "1" з імовірністю $|\beta|^2$ та "2" з імовірністю $|\gamma|^2$. Відзначимо, що суперпозиція квантових станів не має аналога в класичній фізиці.

4) Переплутування (квантові кореляції).

Дві або більше квантових системи можуть бути переплутані. Так, пара фотонів у синглетному поляризаційному стані наступна:

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle_1|1\rangle_2 - |0\rangle_2|1\rangle_1\right), \tag{10.10}$$

де індекси позначають номери фотонів – це приклад максимально переплутаного стану двох кубітів. Такий стан називають парою Ейнштейна– Подольського– Розена (ЕПР-парою).

Якщо вимірювання виконується над одним із двох переплутаних кубітів в стані $|\Psi^-\rangle$ (див. 10.9), наприклад, в базисі $\{|0\rangle, |1\rangle\}$, який називається обчислювальним базисом, то результат буде "0" або "1" з однаковою ймовірністю 1/2. Стан другого кубіту антикорельований з першим, тобто, якщо перший кубіт в результаті вимірювання перейшов у стан $|0\rangle$, то другий перейде в стан $|1\rangle$ і навпаки. Без проведення вимірювання, однак, жодний із цих двох кубітів не перебуває в певному стані. Відзначимо, що переплутування, як і суперпозиція станів – винятково квантові ефекти, що не мають аналога для об'єктів класичної фізики.

Квантові протоколи розподілення секретних ключів пропонують інший підхід до вирішення цієї проблеми. Теоретично, квантова криптографія може забезпечити захищене від перехоплення розподілення ключа, оскільки, на відміну від класичної криптографії, вона заснована на законах фізики, а не на тому факті, що для успішного перехоплення потрібні були б величезні обчислювальні потужності. Внаслідок вищезазначених властивостей квантових систем зловмисник вносить у передану окремими фотонами інформацію деяку кількість помилок, які можуть бути виявлені легітимними користувачами. Відзначимо, що закони квантової механіки дозволяють не тільки виявити збурювання станів, але й зв'язати рівень помилок при вимірюваннях у легітимних користувачів з кількістю інформації, що міг отримати зловмисник. Це дозволяє провести процедуру підсилення секретності, при якій довжина переданого ключа зменшується на деяке число бітів, що залежить від рівня помилок при передачі. У результаті кількість інформації про ключ, що може мати зловмисник після цієї процедури, обмежена зверху як завгодно малою величиною з імовірністю як завгодно близькою до одиниці. Таким чином, протоколи квантового розподілення ключів, на відміну від більшості класичних

схем, мають теоретико-інформаційну стійкість, що не залежить від обчислювальних та інших технічних можливостей зловмисника.

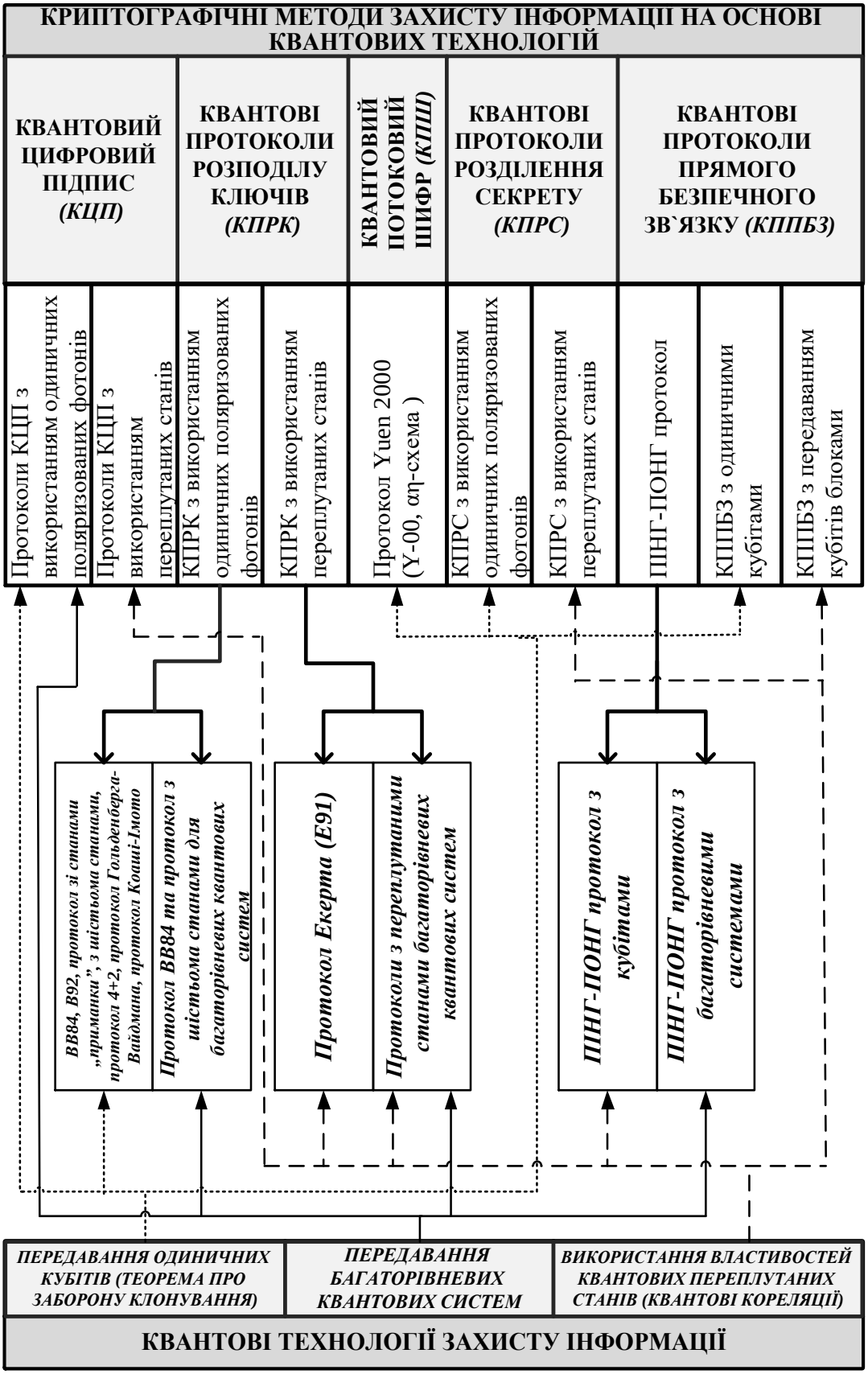


Рис. 10.1. Класифікація квантових технологій

10.2 Протоколи квантового поширення ключа

10.2.1 Протокол BB84

Ідея використати квантові об'єкти для захисту інформації від підробки й несанкціонованого доступу вперше була висловлена Стефаном Вейснером (Stephen Weisner) в 1970 р. Через 10 років Беннет і Брассард, які були знайомі з роботою Вейснера, запропонували використати квантові об'єкти для передачі секретного ключа. В 1984 р. вони опублікували статтю, у якій описувався протокол квантового поширення ключа BB84.

Носіями інформації в протоколі BB84 є фотони, які поляризовані під кутами 0, 45, 90, 135 градусів. Відповідно до законів квантової фізики, за допомогою вимірювання можна розрізнити лише наступне: якщо для двох ортогональних станів відомо, що фотон поляризований або вертикально, або горизонтально, то шляхом вимірювання можна встановити, як саме; теж саме можна стверджувати щодо поляризації під кутами 45 і 135 градусів. Однак з вірогідністю відрізнити вертикально поляризований фотон від фотона, поляризованого під кутом 45 градусів, неможливо.

Ці особливості поводження квантових об'єктів лягли в основу протоколу квантового поширення ключа. Відправник кодує дані, що відправляють, задаючи певні квантові стани. Одержувач реєструє ці стани. Потім одержувач і відправник спільно обговорюють результати спостережень. В остаточному підсумку з як завгодно високою вірогідністю можна бути впевненим, що передані й прийняті кодові послідовності тотожні. Обговорення результатів стосується помилок, внесених шумами або зловмисником, і ні в найменшій мірі не розкриває вмісту переданого повідомлення. Може обговорюватися парність повідомлення, але не окремі біти. Відкритий канал зв'язку не обов’язково повинен бути конфіденційним, необхідно, щоб тільки забезпечував автентифікацію.

Щоб обмінятися ключем, Аліса і Боб використовують наступні дії:

1. Аліса посилає Бобу біт A_i, задаючи певні квантові стани – поляризацію в 0, 45, 90, 135 градусів. Відлік кутів можна вести від напрямку "вертикально нагору" по годинниковій стрілці.

2. Боб розпізнає двома аналізаторами: один розпізнає вертикально-горизонтальну поляризацію, інший - діагональну. Для кожного фотона Боб випадково вибирає один з аналізаторів і записує тип аналізатора і результат вимірювань. Отриманий так званий «сирий» ключ $B_i = A_i$ з імовірністю $P = 75\%$. Тобто, він містить 25 % помилок.

3. По загальнодоступному каналу зв'язку Боб повідомляє Алісі, які аналізатори використалися, але не повідомляє, які результати були отримані.

4. Аліса по загальнодоступному каналу зв'язку повідомляє Бобу, які аналізатори він вибрав правильно. Ті фотони, для яких Боб невірно вибрав аналізатор, відкидаються.

5. Для виявлення перехоплення Аліса й Боб вибирають випадкову ділянку ключа й порівнюють його по загальнодоступному каналу зв'язку. Якщо відсоток помилок великий, то він може бути віднесений на рахунок Єви і процедура повторюється спочатку.

Як джерело світла може використатися діод або лазер. Як провідник використовують або простір, або оптичні кабелі.

10.2.2 Приклад шифрування протоколом BB84

Умовні позначення:

Ці правила із легкістю можуть бути замінені на протилежні (аби тільки Аліса і Боб домовилися між собою), однак у таблицях 10.1 і 10.2 прийняті саме ці позначення.

Якби Єва робила перехоплення інформації за допомогою устаткування, подібного до устаткування Боба, то приблизно в 50 відсотках випадків вона вибрала б невірний аналізатор, не змогла б визначити стан отриманого нею фотона і відправила б фотон Бобу в стані, обраному навмання.

Таблиця 10.1

Позначення аналізатора	**Поляризація фотонів**
+	Прямокутний
X	Діагональний

При цьому в половині випадків вона вибере невірну поляризацію й, таким чином, приблизно в 25 відсотках випадків результати вимірювань Боба можуть відрізнятися від результатів Аліси. Це досить помітно й швидко виявиться. Однак, якщо Єва перехоплює тільки 10% інформації, тоді рівень помилок буде 2.5%, що менш помітно.

Таблиця 10.2

Послідовність фотонів Аліси	|	/	/	—	\	|	|	—	—
Послідовність аналізаторів Боба	+	**X**	+	+	**X**	**X**	**X**	+	**X**
Результати вимірювань Боба	0	**0**	**1**	**1**	1	0	1	1	**0**
Аналізатори обрані вірно	так	так		так	так			так	
Ключ	0	**0**		**1**	1			1	

10.2.3 Зниження рівня помилок і збільшення таємності ключа

Внесені помилки можуть бути виявлені й усунуті за допомогою підрахунку парності так, що вирівнюється парність у блоках з декількох бітів, при цьому після перевірки один біт з кожного блоку відкидається. Беннет в 1991 році запропонував наступний протокол.

1. Відправник і одержувач домовляються про довільну перестановку бітів у рядках, щоб зробити положення помилок випадковими.

2. Рядки діляться на блоки розміру k (k вибирається так, щоб імовірність помилки в блоці була мала).

3. Для кожного блоку відправник й одержувач обчислюють і відкрито сповіщають один одного про отримані результати. Останній біт кожного блоку видаляється.

4. Для кожного блоку, де парність виявилася різною, одержувач і відправник роблять ітераційний пошук і виправлення невірних бітів.

5. Щоб виключити кратні помилки, які можуть бути не замічені, операції пунктів 1-4 повторюються для більшого значення k.

6. Для того, щоб визначити, залишилися чи ні невиявлені помилки, одержувач і відправник повторюють псевдовипадкові перевірки:

• Одержувач і відправник відкрито повідомляють про випадкове перемішування позицій половини бітів у їхніх рядках.

• Одержувач і відправник відкрито порівнюють парності. Якщо рядки відрізняються, парності повинні не збігатися з імовірністю 1/2.

• Якщо має місце відмінність, одержувач і відправник використають двійковий пошук і видалення невірних бітів.

7. Якщо відмінностей немає, після m ітерацій одержувач і відправник отримують ідентичні рядки з імовірністю помилки 2^{-m}.

Збільшення таємності ключа також може бути зроблене без додаткового обміну даними по відкритому каналу. Наприклад, при наявності в Аліси послідовності відправлених бітів A_i і послідовності прийнятих бітів B_i у Боба, обоє можуть зробити наступні перетворення:

$$A_i^{'} = A_{2i} xor A_{2i+1},$$
$$B_i^{'} = B_{2i} xor B_{2i+1},$$

що забезпечує таємність ключа від Єви, навіть якщо вона змогла перехопити і скопіювати кожен другий фотон, що пересилається.

10.2.4 Протокол Екерта

Якщо Аліса і Боб не збираються використати отриманий ними ключ відразу, то перед ними виникає нова проблема – як зберегти ключ у секреті? В 1991 р. Артур Екерт (Artur Ekert) запропонував протокол, що дозволяє вирішити обидві ці проблеми - поширення і зберігання ключа. Протокол Екерта заснований на ефекті EPR (Einstein-Podolsky-Rosen). Ефект EPR виникає, коли сферично симетричний атом випромінює два фотони в протилежних напрямках у сторону двох спостерігачів. Фотони випромінюються з невизначеною поляризацією, але в силу їхньої симетрії поляризації завжди протилежні. Такі стани двох фотонів називаються зчепленими. На основі ефекту EPR Екерт запропонував криптосхему, що гарантує безпеку пересилання і зберігання ключа. Відправник генерує деяку кількість EPR фотонних пар. Один фотон з кожної пари він залишає для себе, другий посилає своєму партнерові. При цьому, якщо ефективність реєстрації близька до одиниці, при одержанні відправником значення поляризації 1 його партнер зареєструє значення 0 і навпаки. Ясно, що в такий спосіб партнери кожний раз, коли потрібно, можуть

одержати ідентичні псевдовипадкові кодові послідовності. Практично реалізація даної схеми проблематична через низьку ефективність реєстрації і вимірювання поляризації одиночного фотона.

Неефективність реєстрації є платою за таємність. Варто враховувати, що при роботі в однофотонному режимі виникають чисто квантові ефекти. При горизонтальній поляризації і використанні вертикального поляризатора результат очевидний - фотон не буде зареєстрований. При 45° поляризації фотона і вертикальному поляризаторі ймовірність реєстрації 50%. Труднощі також полягають у тому, що в цей час не всі зчеплені стани, що відбивають, піддаються вимірюванню, не говорячи вже про створення ідеальних ємностей для зберігання фотонів.

10.2.5 Практична реалізація протоколу BB84

Схема реалізації односпрямованого каналу із квантовим шифруванням показана на рис. 10.2, 10.3. Передавальна сторона перебуває ліворуч, а приймаюча - праворуч. Осередки Покеля служать для імпульсної варіації поляризації потоку квантів передавачем і для аналізу імпульсів поляризації приймачем. Передавач може формувати один із чотирьох станів поляризації (0, 45, 90 й 135 градусів). Передані дані надходять у вигляді керуючих сигналів на ці осередки. Як канал передачі даних може використовуватися оптичне волокно. Як первинне джерело світла можна використати й лазер. На приймаючій стороні після осередку Покеля ставиться кальцитна призма, що розщеплює пучок на два фотодетектори (ФД), які вимірюють дві ортогональні складові поляризації. При формуванні переданих імпульсів квантів доводиться вирішувати проблему їхньої інтенсивності.

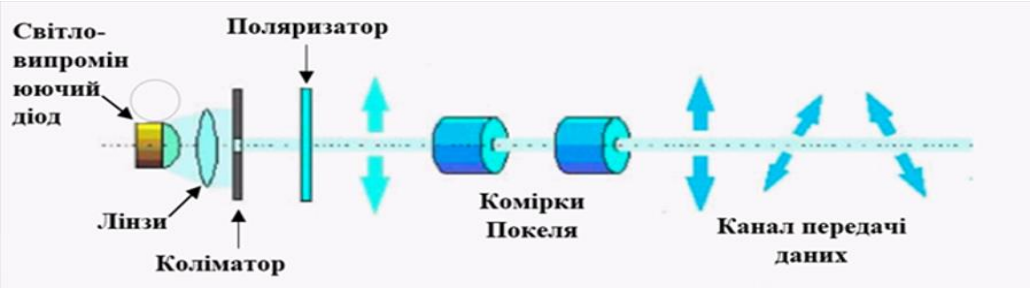


Рис.10.2. Схема сторони, яка передає

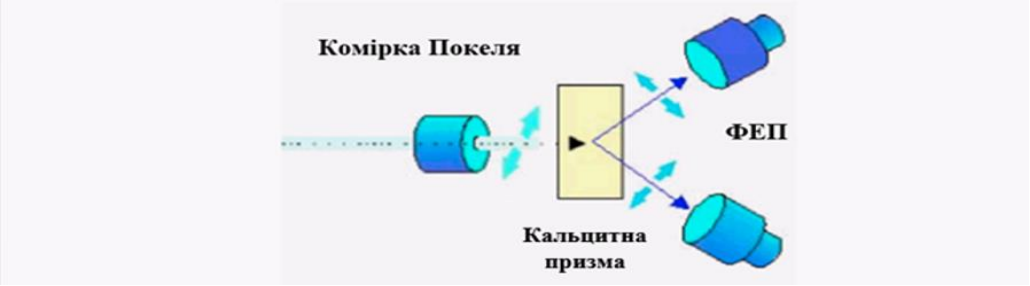


Рис.10.3. Схема реалізації односпрямованого каналу із квантовим шифруванням

Якщо квантів в імпульсі 1000, є ймовірність того, що 100 квантів по шляху буде відведено зловмисником на свій приймач. Аналізуючи пізніше відкриті переговори між передавальною і приймаючою сторонами, він може одержати потрібну йому інформацію. В ідеалі число квантів в імпульсі повинне бути біля одного. Тут будь-яка спроба відводу частини квантів зловмисником приведе до істотного росту числа помилок у приймаючої сторони. У цьому випадку прийняті дані повинні бути відкинуті й спроба передачі повторена. Але, роблячи канал більш стійким до перехоплення, ми в цьому випадку зіштовхуємося із проблемою "темнового" шуму (видача сигналу у відсутності фотонів на вході) приймача (адже ми змушені підвищувати його чутливість). Для того, щоб забезпечити надійне транспортування даних, логічному нулю і одиниці можуть відповідати певні послідовності станів, що допускають корекцію одинарних і навіть кратних помилок.

10.2.6 Протокол B92

В протоколі B92 також використовуються як носії інформації фотони, однак, поляризовані тільки у двох станах. Такі стани поляризації більш зручні для передачі даних на більші відстані по оптичних кабелях.

В алгоритмі B92 приймач і передавач створюють систему, що базується на інтерферометрах Маха-Цендера. Відправник визначає кути фазового зрушення, що відповідають логічному нулю і одиниці ($F_A = p/2$), а приймач задає свої фазові зрушення для логічного нуля ($F_B = 3p / 2$) і одиниці ($F_B = p$). У даному контексті зміна фази $2p$ відповідає зміні довжини шляху на одну довжину хвилі використовуваного випромінювання. Хоча фотони поводяться при детектуванні як частки, вони поширюються як хвилі. Імовірність того, що фотон, відправлений відправником, буде детектовано одержувачем, дорівнює:

$$P_d = \cos^2\left(\frac{F_A - F_B}{2}\right)$$

і характеризує інтерференцію амплітуд хвиль, що поширюються по верхньому і нижньому шляхах (див. Рис. 10.4). Імовірність реєстрації буде варіюватися від 1 (при нульовій різниці фаз) до нуля. Тут передбачається, що відправник і одержувач використовують фазові зрушення (F_A F_B) = ($0,3p / 2$) для нульових бітів і (F_A F_B) = ($p/2$, p) для одиночних бітів (для алгоритму BB84 використовуються інші припущення).

Для реєстрації одиночних фотонів, крім ФД, можуть використовуватися твердотільні лавинні фотодіоди (германієві й InGaAs). Для зниження рівня шуму їх варто прохолоджувати. Ефективність реєстрації одиночних фотонів лежить у діапазоні (10-40) %. При цьому варто враховувати також досить високе поглинання світла оптичним волокном (~0,3-3 ДБ/км).

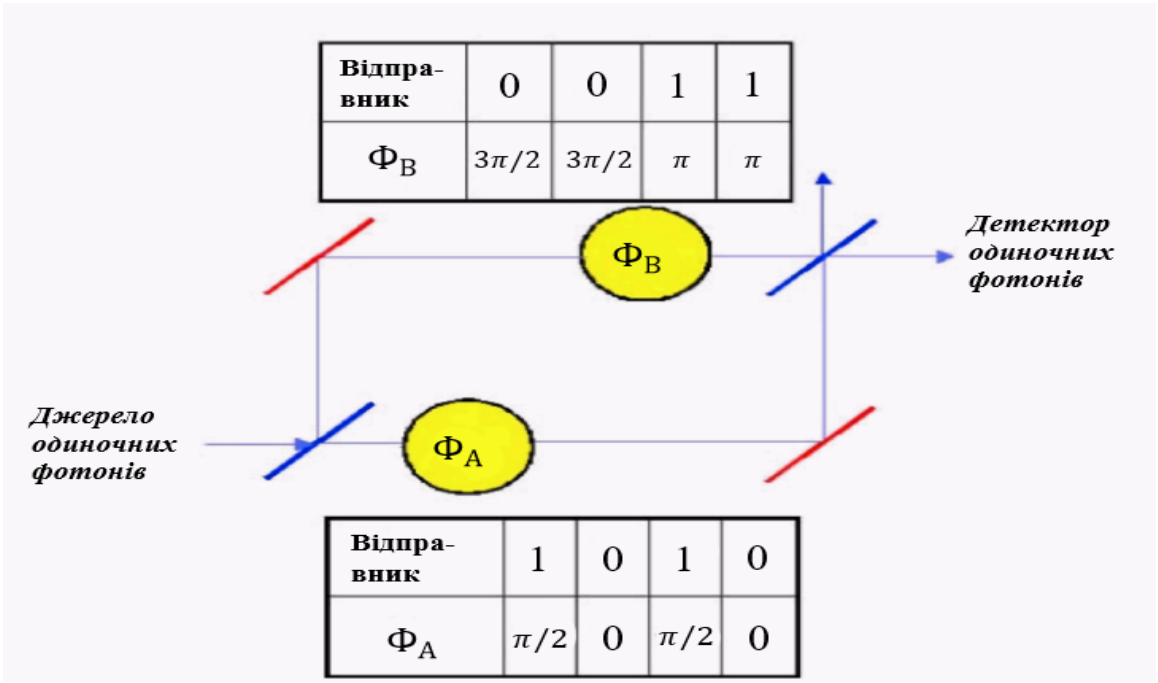


Рис. 10.4. Реалізація алгоритму B92

Схема інтерферометра із двома волокнами досить нестабільна через різні властивості транспортних волокон і може успішно працювати тільки при малих відстанях. Кращих характеристик можна досягти, мультиплексіруючи обидва шляхи фотонів в одне волокно (див. Рис. 10.5). У цьому варіанті відправник і одержувач мають ідентичні нерівноплічні інтерферометри Маха - Цендера (червоними кольорами відзначені дзеркала).

Різниця фаз довгого й короткого шляхів БВ багато більше часу когерентності світлового джерела. Із цієї причини інтерференції в межах малих інтерферометрів не відбувається (Б). Але на виході інтерферометра отримувача вона можлива (В). Імовірність того, що фотонні амплітуди складуться (центральний пік вихідного сигналу інтерферометра В), дорівнює:

$$P = \frac{1}{8}\left(1 + \cos\left(\Phi_A - \Phi_B\right)\right).$$

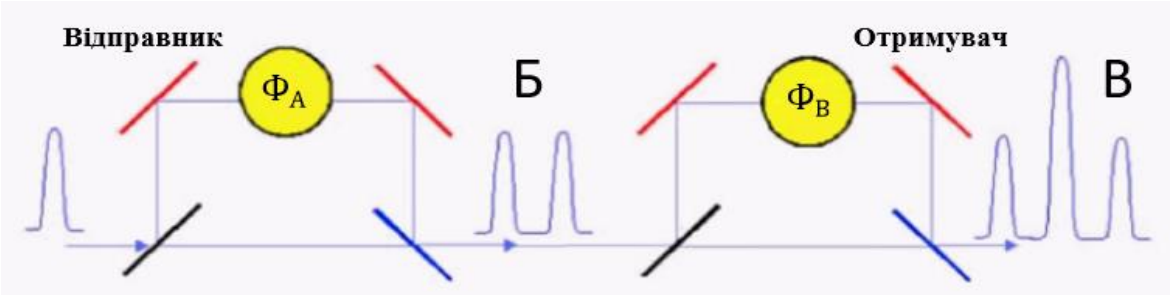


Рис. 10.5. Інтерферометр із одним транспортним волокном

Розсіювачі пучка (напівпрозорі дзеркала) можуть бути замінені на оптоволоконні об'єднувачі (coupler). Практичні вимірювання для транспортного

кабелю довжиною 14 км показали ефективність генерації біта ключа на рівні $2,2 * 10^{-3}$ при частоті помилок близько 1,2%.

10.2.7 Практичні аспекти квантової криптографії

При створенні практичних криптосистем, заснованих на квантовому поширенні ключа, доводиться зіштовхуватися з наступними проблемами:

- Низька швидкість передачі.
- Невеликі відстані.
- Неможливість створення квантових повторювачів.
- Інтенсивність імпульсів квантів.
- Атаки зловмисників на квантовий канал.

При формуванні переданих імпульсів квантів доводиться вирішувати проблему їхньої інтенсивності (див. п. 10.2.5).

Атаки на квантову передачу класифікуються в такий спосіб:

- Некогерентні (індивідуальні).
- Когерентні (масові).
- Спільні.
- Колективні.

10.3 Квантові протоколи узгодження ключів

Якщо дослідникам та конструкторам вдасться реалізувати універсальний квантовий комп'ютер, який зможе ефективно реалізувати алгоритми Шора та Гровера, то цілі розділи класичної криптографії опиняться під загрозою зникнення. Дійсно, швидкий алгоритм факторизації зробить неефективною асиметричну криптографію, причому це стосується не лише RSA, оскільки з не меншою ефективністю можна буде розв'язати й задачі дискретного логарифмування та знаходження точок кратності еліптичних кривих. В той же час алгоритм швидкого пошуку у невпорядкованому масиві змушує збільшувати довжину ключа симетричних криптоалгоритмів для компенсації зменшення їх криптостійкості, як мінімум у 2 рази.

Таким чином, необхідно шукати альтернативу класичним криптоалгоритмам. Сьогодні одним з новітніх підходів до рішення цієї проблеми є квантова криптографія.

Квантова криптографія - це розділ квантової інформатики, який вивчає методи захисту інформації з використанням квантових носіїв.

Можливість такого захисту визначається, перш за все, теоремою про неможливість клонування невідомого квантового стану. Існування цієї теореми привело до розробки протоколу квантової телепортації, вона ж забезпечує захист інформації від прослуховування в квантових каналах зв'язку.

Здається, тоді достатньо встановити такий канал зв'язку між учасниками інформаційного обміну і можна обмінюватися відкритими повідомленнями без використання криптографічного захисту. Однак, сьогодні це занадто дорого й повільно.

З другого боку, квантовий канал зв’язку можна використати для узгодження криптографічних ключів симетричних алгоритмів, оскільки популярні сьогодні протоколи узгодження ключа на основі асиметричних алгоритмів знаходяться під найбільшою загрозою з боку алгоритму Шора.

Найреальніший варіант, який лежить на поверхні, це потокові шифри, наприклад, шифр Вернама, ключ до якого узгоджується за допомогою квантових протоколів.

Саме так сьогодні і виглядає найпопулярніша схема квантової криптографії.

10.3.1 Квантовий протокол узгодження криптографічного ключа BB84

Цей протокол був розроблений Чарльзом Беннеттом та Жілем Брассаром у 1984 році, звідки й походить його назва. Потім він зазнав кількох модифікацій, однак ми розглянемо його першу інтерпретацію на чотирьох квантових станах.

Нехай ми маємо канал зв’язку, чистий настільки, що він може пропускати окремі фотони без значних втрат. Нехай у нас є також обладнання, яке вміє випромінювати фотони поодинці. Ці фотони мають довільну, але строго визначену поляризацію.

Назвемо поляризацією електромагнітних хвиль напрямлені коливання векторів електричного та магнітного полів. Будемо вважати, що такою ж поляризацією володіють й окремі фотони, хоча насправді це не зовсім так.

Припустимо, що фотони можуть мати або вертикально-горизонтальну поляризацію, яку будемо називати ортогональною, або діагональну, як це показано на рисунках 10.6 і 10.7. Цього можна досягти, використовуючи на виході нашого устаткування поляризаційний фільтр, який може пропускати фотони лише певної поляризації. Наприклад, якщо фільтр може пропускати лише фотони вертикальної поляризації і затримувати усі горизонтальної, то після нього всі фотони будуть мати тільки вертикальну поляризацію (див. Рис. 10.6, 10.7).

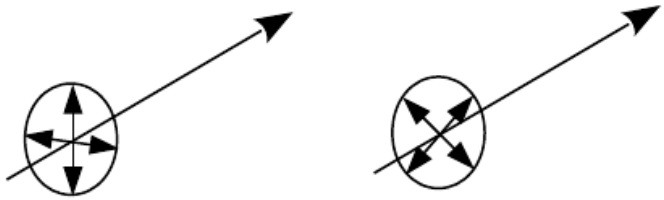


Рис. 10.6. Типи поляризації фотонів: ліворуч — ортогональна поляризація; праворуч — діагональна

Це приведе до зменшення кількості фотонів після фільтру, тобто, до зменшення інтенсивності потоку. Так ми платимо за те, що потік фотонів буде чітко поляризований, наприклад, вертикально.

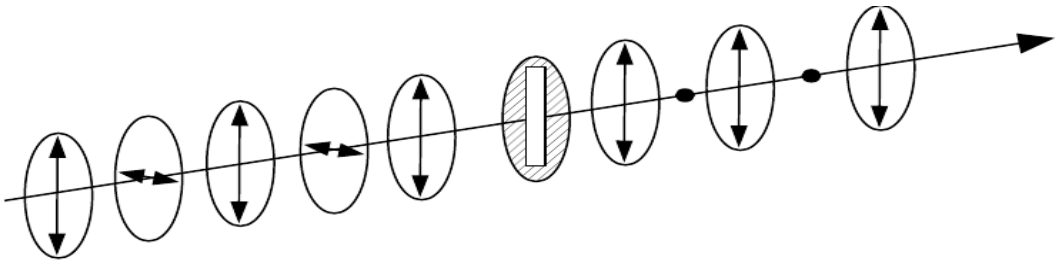


Рис. 10.7. Схема фільтрування потоку фотонів поляризаційним фільтром

Якщо ж лазер випромінює фотони діагональної поляризації, частину їх також буде затримано, однак тут ми можемо лише говорити про те, з якою ймовірністю той чи інший фотон буде затриманий. Отже, у цьому випадку працює звична для нас квантова схема. Можливі, наприклад, такі варіанти, які показано на рис. 10.8.

Одним словом, ми не знаємо, яким чином піде процес, а лише можемо говорити про певну ймовірність того чи іншого стану системи. Яким же чином цю ситуацію можна використати для узгодження криптографічних ключів?

Припустимо, що абонент А має таку систему, вдосконалену тим, що фільтр може довільним чином змінювати напрям поляризації, вибираючи з чотирьох позицій, які показано на рис. 10.9.

Вибір позиції щоразу випадковий. Будемо приймати за “1” вертикальну поляризацію фотона, а за “0” - горизонтальну. Одиниця та нуль діагональної поляризації показані на рис. 10.9.

Нехай абонент А передає потік одиничних фотонів, весь час змінюючи положення поляризаційного фільтра випадковим чином.

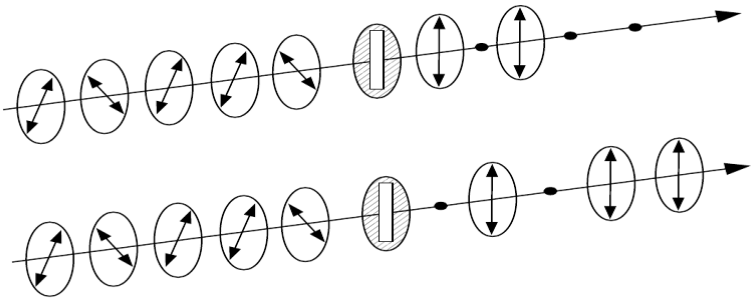


Рис. 10.8. Різні варіанти пропускання фільтром фотонів діагональної поляризації

На виході ми отримаємо потік фотонів кожної з чотирьох поляризацій. На приймальному боці у абонента Б є приймальна апаратура, яка використовує такий самий фільтр, також орієнтуючи його випадковим чином. Отже, якщо фільтр зорієнтовано горизонтально, а на нього потрапляє вертикально зорієнтований фотон, то його буде затримано і на виході приймальної апаратури не буде нічого. Якщо ж фільтр зорієнтовано діагонально, а фотон поляризований вертикально, то він з певною ймовірністю може бути пропущений. І ми не можемо стверджувати, що це станеться напевне.

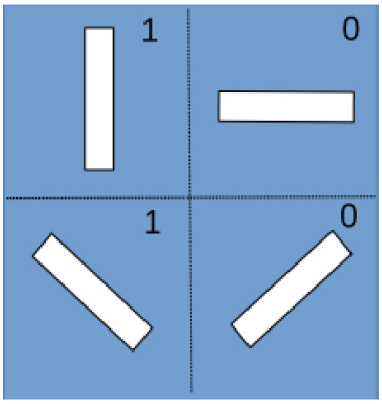


Рис.10.9. Схема поляризаційного фільтра,
використаного у протоколі узгодження ключа

Розглянемо, як ми можемо узгодити криптографічний ключ за допомогою такої схеми.

Припустимо, що лазер абонента А згенерував послідовність фотонів з такою поляризацією:



Тоді ми отримали такий бітовий рядок: 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1. Ці та подальші дії, а також отримані результати подано в табл. 10.3.

Таблиця 10.3

1	0	1	1	0	1	1	0	0	1	0	1	1	0	0	1
2	X	+	X	+	+	+	+	+	X	X	+	X	X	X	+
3	0	-	1	-	1	1	0	0	-	0	1	1	-	0	1
4	+	X	X	+	+	X	X	+	X	+	X	X	X	X	+
5	1		1		1	0	0	0		1	1	1		0	1
6	+		X		+	X	X	+		+	X	X		X	+
7			√		√			√				√		√	√
8			1		1			0				1		0	1
9					1									0	
10					√									√	
11			1					0				1			1

Записані в таблиці 10.3 дії слід читати в такий спосіб:

Перші п'ять дій виконуються квантовим каналом зв'язку.

1. В цьому рядочку подана бітова послідовність, яку згенерував лазер абонента А.
2. Орієнтація (базиси) поляризаційного фільтра, які були обрані абонентом А:
   X — діагональний; + - ортогональний.
3. Фотони (бітовий рядок), відправлений абонентом А в канал зв'язку.

4. Випадкові базиси, обрані абонентом Б під час прийому сигналу.
5. Результуючий бітовий рядок, отриманий абонентом Б.

Решта дій виконуються відкритим каналом зв'язку.

6. Абонент Б повідомляє А обрані ним базиси.
7. А відзначає, які базиси співпадають.
8. Узгоджені біти ключа, які можна використовувати, якщо не було порушень під час узгодження ключа.

Якщо є підозра, що канал прослуховується, виконуються наступні операції:

9. Абонент Б вказує деякі, вибрані випадково, біти ключа.
10. Абонент А підтверджує ці біти.

Результат

11. Секретні біти, які можна використовувати як узгоджений криптографічний ключ.

Коли цей одноразовий ключ буде повністю використано, протокол узгодження нової порції ключа повторюється спочатку.

Цей протокол досить малоефективний, оскільки, як ми бачимо, з переданих 15 бітів в цьому прикладі узгодили лише 4.

Протокол узгодження ключа B92

Протокол BB84 використовує чотири напрями поляризації фотонів. У 1992 році Чарльз Беннетт показав, що аналогічний протокол можна побудувати за допомогою лише двох напрямків поляризації. Протокол було названо B92 за аналогією з BB84. Він використовує фотони, поляризовані у двох різних напрямках: за “1” приймається поляризація під кутом +45°, а за нуль - під кутом 0°. Процедуру узгодження ключа за цим протоколом схематично подано на рис. 10.10.

Апаратура абонента А генерує фотони, поляризовані в напрямках 0 та +45 градусів. Відповідно, вони представляють нулі та одиниці, причому поляризація кожного фотона випадкова. Детектувальна апаратура на приймальному боці приймає фотони крізь фільтри, орієнтовані під кутами 90 та -45 градусів. При цьому, якщо фотон аналізується крізь фільтр, орієнтований під кутом 90° відносно його поляризації, то фотон буде затримано. Якщо ж цей кут становить 45°, то фільтр пропустить фотон з ймовірністю 50%.

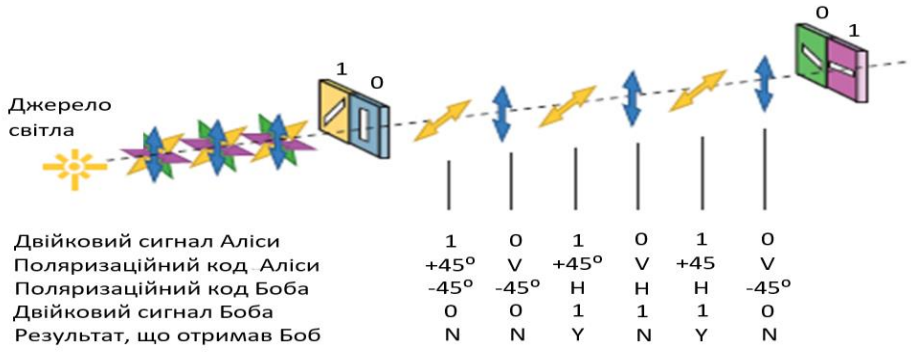


Рис. 10.10. Схема узгодження ключа за протоколом B92

Розглянемо приклад узгодження бітів ключа за цим протоколом. Припустимо, що абонент А згенерував таку послідовність фотонів:



Це означає бітовий рядок 1, 0, 1, 0, 1, 0. На приймальному боці абонент Б застосував до цих фотонів такі фільтри:



Ми бачимо, що в такому разі тільки для 3-го та 4-го фотонів напрям приймальних фільтрів був перпендикулярний до напряму поляризації фотонів. Відповідно, ці фотони точно були затримані апаратурою. Отже, ми отримаємо таку послідовність з прийнятих та затриманих фотонів: +, +, -, -, +, +, де "+" - це прийнятий фотон, а "-" – затриманий. Однак, тут не враховано той факт, що в разі кута між поляризацією фільтра і фотона в 45 градусів, крізь фільтр проходять лише 50% фотонів. Нехай і в нашому випадку з чотирьох можливих система пропустила лише два. Тоді послідовність пропущених та затриманих фотонів буде виглядати приблизно так: -, +, -, -, +, -. Залишилося з'ясувати, які саме двійкові значення отримав абонент Б. Напрям поляризації 2-го фотона на вході був 0° (це "0" за нашими домовленостями). Якщо він пройшов крізь приймальний фільтр, то його поляризація стала -45°, що також відповідає нулю, як це показано на мал. 10.10.

Отже значення, яке "приніс" другий фотон - нуль. Аналогічно, передостанній фотон мав напрям поляризації $+45^o$, що відповідає одиниці. Якщо ж його пропустила приймальна апаратура, то напрям поляризації став 90°, що відповідає одиниці. Таким чином, ми узгодили два біти ключа: 0, 1.

Для цього протоколу використовується така ж процедура просіювання ключа, як і для BB84.

Протокол B92 ще більш витратний, ніж BB84. Дійсно, ймовірність того, що абонент Б застосує для чергового фотона фільтр з поляризацією в 45°, в середньому буде дорівнювати 50%. В свою чергу, ймовірність того, що цей фільтр пропустить фотон, також дорівнює 50%. Отже, на приймальному боці зможуть виявити лише 25% відправлених абонентом А фотонів. Тим не менше, в цьому є й позитивний момент: дії зловмисника, який буде прослуховувати канал, буде легше виявити, оскільки спотворення інформації будуть більшими.

Атаки на протокол квантового узгодження ключа

Розглянемо, які дії повинен виконати зловмисник для того, щоб перехопити цей ключ, і чи зможе він це зробити.

Можливі варіанти такі:

1. Скопіювати ключ під час передавання неможливо, оскільки ми знаємо, що існує теорема про неможливість клонування невідомого квантового стану. Це саме такий випадок, оскільки зловмисник не знає, який стан того чи іншого

фотона. Саме це він і прагне з'ясувати. Копіювання змінить стан фотонів і перевірка абонентів А та Б це виявить. Отже, цей варіант не проходить.

2. Скористатися протоколом квантової телепортації зловмисник також не може, оскільки при цьому руйнується стан оригінального фотона.

Тим не менше, припустимо, що зловмисникові якимось невідомим для нас чином вдалося виміряти поляризацію кожного фотона, який мандрував каналом зв'язку. Хай навіть так! Однак, він не в змозі дізнатися, яка випадкова послідовність положень фільтра була на приймальному боці у абонента Б. Отже, зловмисник спробує вгадувати ці стани, тобто, застосує свою послідовність. Звичайна теорія ймовірностей говорить про те, що в такому разі він може вгадати не більше 50% положень фільтра, що значно менше, ніж потрібно. Однак, це ще не всі його проблеми. Якщо він при цьому сформує свій фотон для того, щоб відправити його абоненту Б, то його поляризація буде неправильною внаслідок редукції хвильової функції. Вимірюючи цей “неправильний” фотон в своєму базисі, абонент Б неодмінно помітить помилку і зрозуміє, що канал прослуховують. Значення похибки на приймальному боці можна оцінити з таких міркувань. Припустимо, зловмисник атакував не кожний фотон, а якусь їх частину з ймовірністю *p*. Тоді 1 - *p* сигналів приходить абоненту Б без змін. Для сигналів, що були атаковані, існує два варіанти розвитку подій: 1) зловмисник правильно вгадав базис, а, отже, правильно передав сигнал абоненту Б, не вносячи жодних спотворень; 2) базис був вгаданий неправильно, ймовірність чого дорівнює 0,5. Отже, з цією ймовірністю результат вимірювань був неправильним, і абсолютно точно він передав “неправильний” фотон абоненту Б. Це призводить до помилки на його боці з такою самою ймовірністю 0,5.

Ймовірність кожного сценарію дорівнює 0,5, отже, легко бачити, що на приймальному боці накопичується 25% неправильних результатів, що занадто багато для якісного каналу. Більше того, теоретичні оцінки для каналів зв'язку, згідно з критерієм Шеннона, зменшують цю ймовірність до 11%. Таким чином, прийнято вважати, що 11% помилок - це межа, до якої можливе секретне узгодження ключа.

Отже, будь-які дії зловмисника призведуть до того, що абонент Б, проходячи по отриманій послідовності, обов'язково помітить, що у деяких ситуаціях він застосував правильну схему, а чомусь результату (ні “0”, ні “1”) не отримав. Це буде свідчити, що канал зв'язку знаходиться під контролем сторонньої особи. Цей факт буде сигналом для переходу на інший канал і повторення усієї процедури узгодження ключа з самого початку.

Сьогодні на ринку немає однофотонних випромінювачів, тому в апаратурі для квантового узгодження ключа використовують слабкі когерентні імпульси, які випромінюються високоякісними лазерними діодами. Це дає змогу реалізувати так звану атаку розділення фотонів.

Якщо зловмисник помічає в імпульсі більше одного фотона, він відділяє один з них, дозволяючи решті безперешкодно відправитися до абонента Б. Далі зловмисник підмішує цей фотон до свого, утворюючи переплутану пару, і чекає оголошення базисів. Виконуючи потім часткові вимірювання свого фотона в

парі, він може точно визначити стан підмішаного фотона, не вносячи жодних змін в ключ.

Якщо ж імпульс складається з одного фотона, то стратегія зловмисника може бути різною. Наприклад, він може просто пропускати одиничні фотони, не втручаючись в хід подій, або заставити їх провзаємодіяти з пробним фотоном і відправити абоненту Б. Проби так само зберігаються до оголошення результатів просіювання ключа.

Таким чином, ми бачимо, що існують деякі типи атак на протокол квантового узгодження ключів, які, скоріше, можна назвати теоретичними. Однак, сьогодні існують і атаки на реалізацію. Одну з таких атак продемонстрував Вадим Макаров зі співробітниками.

Атака виконується на особливості апаратної частини устаткування.

Оскільки імпульси, якими обмінюються сторони, дуже слабкі, на приймальному боці використовують фотоелектричні помножувачі – лавинні фотодіоди. Ці фотодіоди можуть працювати в двох режимах. Коли на вхід поступає слабкий сигнал, фотодіод підсилює його, відправляючи на вихід електричний сигнал. Коли ж інтенсивність падаючого світла зростає, прилад перестає реагувати на окремі фотони, і працює лише з потужними імпульсами.

Цю особливість помножувачів використала для зламу протоколу і група В. Макарова. Вони повністю перехоплюють потік фотонів від абонента А і генерують потужні імпульси тієї самої поляризації, що й перехоплені фотони, які відправляють на попередньо “засліплений” фотодіод, який вже не здатен реагувати на окремі фотони.

Таку атаку було реалізовано на діючій системі квантового узгодження ключа в Сингапурському університеті, коли за 5 хвилин було перехоплено більше 8 млн. фотонів, залишившись повністю непомітними для учасників інформаційного обміну.

Тим не менше, від такої атаки існує дуже простий захист. Він полягає в тому, що перед приймачем розміщують додаткове джерело одиночних фотонів, яке відправляє їх у випадкові моменти часу. Якщо фотодіод на них не реагує, значить канал прослуховується.

Реалізації протоколу квантового узгодження ключа

Практичні роботи у галузі квантової криптографії виконують такі компанії як IBM, Mitsubishi, Toshiba, лабораторія GAP-Optique, Національна лабораторія в Лос-Аламосі, Каліфорнійський технологічний інститут, MagiQ, холдінг QinetiQ.

Досягнуті успіхи:

- Ніколя Гісін (GAP-Optique) – реалізовано протокол BB84 на відстані 67 км;
- компанія Mitsubishi - 87 км зі швидкістю 7,2 біти за секунду;
- Toshiba Research Europe - 100 км.

Створено також комерційну квантову криптосистему id 3000 Clavis Quantum Key Distribution System, яка підтримує:

- безпечне узгодження ключа на відстані до 100 км;
- підтримку протоколу BB84;

- вбудований протокол просіювання ключа;
- протокол шифрування та передавання файлів;
- бітрейт - 1500 бітів за секунду.

Розглянуті протоколи квантового узгодження криптографічного ключа вважаються основними. Існують, тим не менше, багато різних модифікацій, які не принципово відрізняються від оригіналів.

Протокол E91

Тим не менше, існує ще один дуже цікавий протокол узгодження криптографічного ключа, який ґрунтується на зовсім іншій ідеї. Цей протокол був запропонований Артуром Екертом у 1991 році і називається E91 (див. підрозділ 10.2.4). Інша назва цього протоколу - ЕПР-протокол, оскільки він використовує ЕПР-пари фотонів. Такі пари фотонів породжуються симетричним атомом. Поляризація фотонів невизначена, однак, оскільки вони переплутані, то їх напрями поляризації строго протилежні.

$$|S_0\rangle = 1/\sqrt{2}\,(|0\rangle|3\pi/6\rangle + |3\pi/6\rangle|0\rangle);$$

$$|S_1\rangle = 1/\sqrt{2}\,(|\pi/6\rangle|4\pi/6\rangle + |4\pi/6\rangle|\pi/6\rangle);$$

$$|S_2\rangle = 1/\sqrt{2}\,(|2\pi/6\rangle|5\pi/6\rangle + |5\pi/6\rangle|2\pi/6\rangle).$$

Якщо у абонента А є подібне обладнання, він зможе реалізувати такий протокол узгодження ключа.

Один фотон із згенерованої пари абонент залишає у себе, а другий – відправляє на приймальний бік абоненту Б. Обидва абоненти не знають, яка поляризація їхніх фотонів. Більше того, вони зможуть це з'ясувати лише після вимірювання. Отже, згенерувавши певну кількість переплутаних фотонів, учасники інформаційного обміну можуть починати вимірювання. Для отримання ідентичних послідовностей абонент А записує істинні значення, а абонент Б - їх доповнення до одиниці.

Така ідея цього протоколу. Звичайно, тут також існують процедури просіювання ключа.

Реалізація такого протоколу значно складніша за розглянуті раніше, оскільки сьогодні не існує джерел переплутаних пар фотонів з високим степенем кореляції та великим часом життя.

Квантові гроші Стівена Віснера

Як приклад того захисту, що його можуть надати користувачам квантові технології при впровадженні їх у життя, розглянемо ідею квантового захисту грошей, яка була запропонована Стівеном Віснером і ґрунтується на законах квантової фізики.

У кінці 60-х р. XX сторіччя аспірант Колумбійського університету Стівен Віснер (син директора MIT, радника багатьох президентів США з науки Джерома Віснера) запропонував своєму керівникові ідею квантового захисту грошей. Науковий керівник не звернув на цю ідею жодної уваги. С. Віснер відправив статтю в журнал.

Редакція відхилила статтю, визнавши її антинауковою.

Лише 1983 року стаття «Conjugate Coding» (журнал «ACM SIGACT News») вийшла в друці та була визнана однією з основних у галузі квантового захисту інформації.

Основна перевага ідеї полягає в тому, що захист побудовано не на складності підробки, а на принциповій неможливості цього.

Ідея полягала в наступному. Припустимо, ми маємо такі квантові пастки, в яких можемо утримувати окремі фотони необмежений час.

Тоді ми можемо реалізувати такий захист грошей.

На кожній купюрі розміщено деяку кількість таких пасток, які заповнюються поляризованими фотонами під час випуску купюр.

Банк зберігає дані про поляризацію кожного фотона на кожній купюрі у своїй базі даних.

Схематично С. Віснер уявляв це собі таким чином (див. рис. 10.11):

Детальніше підхід, запропонований С. Віснером, пояснено на рисунку 10.12.

Система захисту банкноти складається із 100 пасток, де знаходяться квантові частинки, кожна з яких має багато квантових атрибутів, які можна вимірювати у трьох напрямах.

Серійний номер банкноти пов’язаний з квантовою конфігурацією цих частинок і зберігається у базі даних емітента валюти. Для перевірки істинності банкноти банк шукає в базі даних номер банкноти і отримує квантову конфігурацію кожної із ста частинок, що знаходяться у пастках цієї банкноти. Після цього банк випадковим чином вибирає один з напрямів, в якому виконує часткові вимірювання квантових характеристик кожної частинки, не зачіпаючи решти напрямів. Якщо результати вимірів збігаються з даними, записаними в базі даних банку, він вважає, що банкнота справжня. Зловмиснику для копіювання банкноти необхідно виконати вимірювання в усіх трьох напрямах, що заборонено принципами квантової механіки.

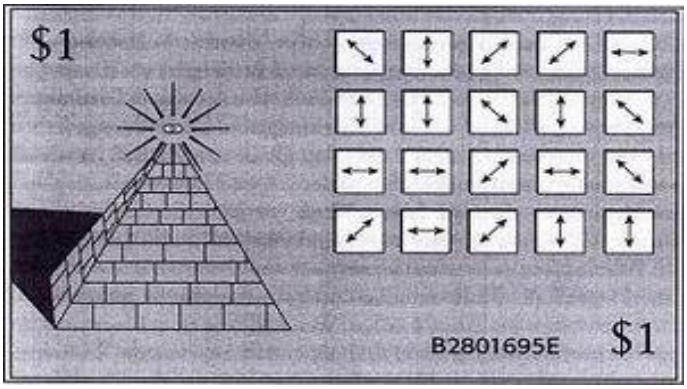


Рис. 10.11. Схематичне зображення поляризації фотонів

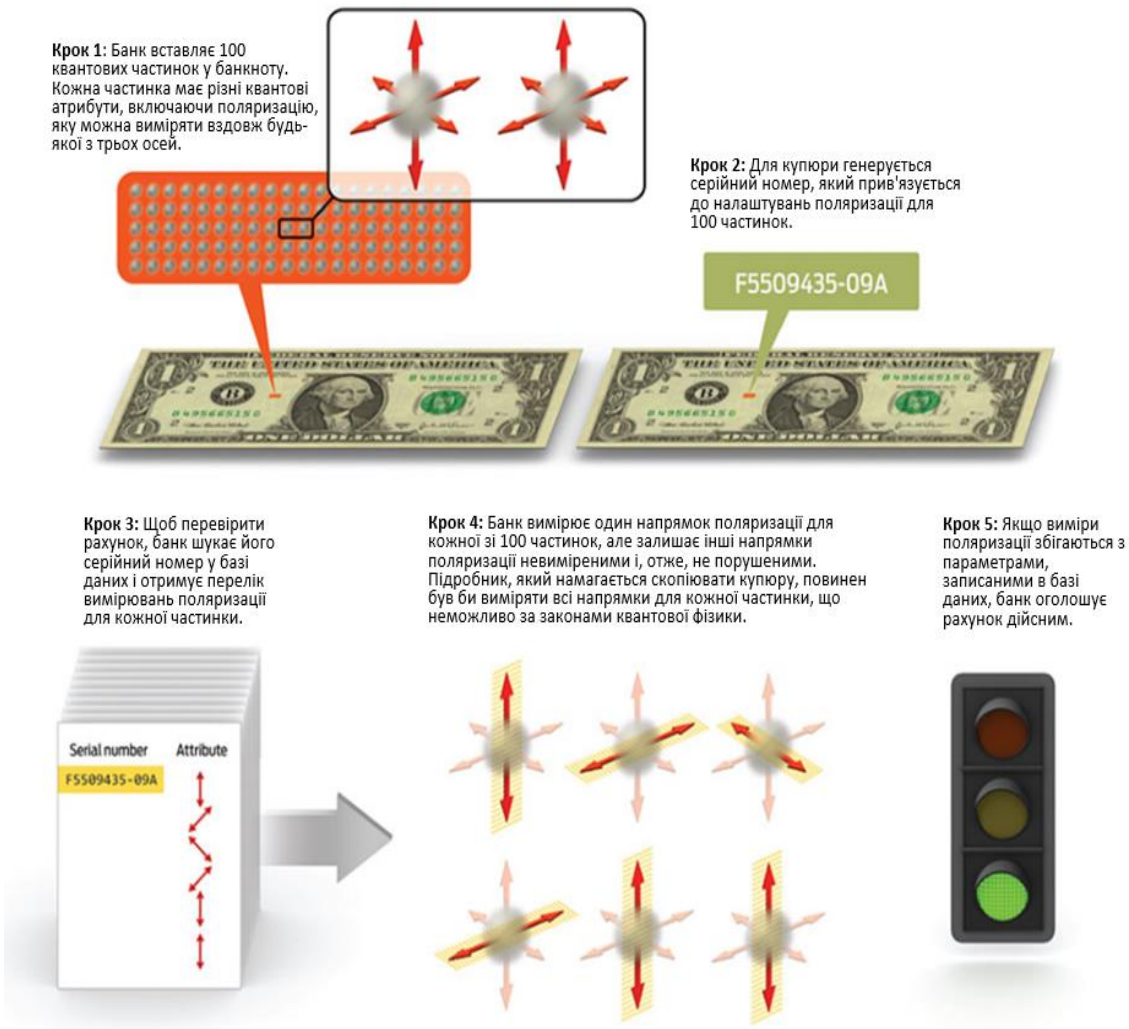


Рис. 10.12. Схематичне зображення захисту банкноти

Звичайно, недоліком такої схеми є ніким не контрольовані права банку на необмежений випуск банкнот.

Співробітники Масачусетського технологічного інституту запропонували модифікацію ідеї С. Віснера: банк генерує кілька станів квантових частинок та змішує їх при заповненні пасток. У цьому випадку банк не знає, який отримується результат. При цьому банк публікує «публічний ключ» – алгоритм, за допомогою якого кожне відділення може перевірити справжність купюри, але не зможе відтворити цей стан квантових частинок. Стверджується, що такий підхід позбавляє банк-емітент монополізму на випуск купюр.

Довгий час ідея С. Віснера була лише теоретичною іграшкою, оскільки не існувало квантових пасток, які могли б утримувати частинки хоча би макроскопічний час, не кажучи вже про необмежений.

Однак у 2012 році Серж Арош (Франція) та Девід Вайнленд (США) отримали Нобелівську премію з фізики за розроблені методики маніпулювання квантовими частинками, які могли утримуватися в пастках протягом однієї секунди, а це вже макроскопічний час. Проте, це відбувається за дуже низьких температур.

Інший варіант пасток пропонується науковцями Центру квантової фізики Гарвардського університету, які можуть утримати квантовий стан азоту при кімнатній температурі в матриці надчистих кристалів кремнію.

Таким чином, ідеї Стівена Віснера сформували фундамент нового напряму науки – **квантової інформаційної безпеки**.

Суть цього напряму – використання фундаментальних квантових властивостей мікросвіту для захисту інформації.

Основна перевага – теоретична стійкість квантових протоколів та способів захисту.

Основний недолік – складність практичної реалізації.

Повернемося тепер до документу NISTIR 8105 DRAFT Report on Post-Quantum Cryptography і таблиці 10.3. Як видно з усього наведеного матеріалу, ми стоїмо перед надзвичайно відповідальним етапом розвитку криптографічної науки, який так і називають у Національному інституті стандартів і технологій США, постквантовою криптографією. Згаданий документ було видано у квітні 2016 року, а пізніше, в грудні того ж року НІСТ оголосив конкурс криптоалгоритмів, стійких до атак за допомогою квантових комп'ютерів.

Сьогодні конкурс ще триває, проходить "півфінал", в якому кількість учасників скорочено до 26 алгоритмів. Ці алгоритми, на думку математиків НІСТ та спеціалістів з криптографії, вважаються найсильнішими кандидатами для розробки стандартів постквантової криптографії. Як повідомляють на сайті НІСТ (https://www.nist.gov/news-events/news/2019/01/nist-reveals-26-algorithms-advancing-post-quantum- crypto-semifinals), в подальшому увагу експертів буде зосереджено на продуктивності алгоритмів, які мають бути ефективними як для настільних комп'ютерів, так і для обчислювачів з обмеженими ресурсами (смартфони, кредитні картки, вбудована техніка, "розумний будинок") тощо.

Оскільки ніхто зараз не знає, якими мають бути параметри повномасштабного програмованого квантового комп'ютера, для участі в конкурсі відібрано криптоалгоритми, що використовують різні розділи математики, які, як передбачається, є стійкими до квантових атак.

Ці алгоритми представляють цілий спектр математичних ідей, зокрема, можна зупинитися на наступних:

1. Решіткові системи, побудовані з використанням геометричних структур та подані з використанням математичних масивів, відомих як матриці. Класичними прикладами є схеми шифрування Ring-Lear-ning with Errors чи старіші NTRU або GGH.

2. Криптокодові системи на основі кодів з виправленням помилок. Яскравими представниками є схеми шифрування МакЕліса та Нідеррайтера.

3. Криптографія на основі функцій хешування, зокрема, підпис Р.Ч. Меркля з публічним ключем на основі хеш-дерева.

4. Однією з найцікавіших схем є підпис з публічним ключем Ж. Патарена (схема HFE), запропонована ще 1996 року як узагальнення пропозицій Мацумото та Імаї.

По завершенні другого раунду конкурсу, можливо, буде третій, перш ніж НІСТ остаточно оголосить переможців - алгоритми, які будуть доповнювати або замінювати три стандарти, що вважаються найменш стійкими до квантових атак: FIPS 186-4 (стандарт цифрових підписів), NIST SP 800-56A та NIST SP 800-56B (розподіл ключів за допомогою асиметричних криптосистем). На прийняття цього рішення визначальним чином буде впливати сучасний стан розвитку квантового комп'ютингу. Поки що, як зазначалося нами, до реалізації масштабованого квантового комп'ютера ще далеко, але над проблемою працює велика кількість наукових колективів, що не може не впливати на прискорення вирішення технологічних викликів, які стоять на шляху цих колективів.

Практичні роботи

Практичні роботи в області квантової криптографії ведуть такі компанії як IBM, Mitsubishi, Toshiba, лабораторії GAP-Optique, Національна лабораторія в Лос-Аламосі, Каліфорнійський технологічний інститут (Caltech), Magi, холдинг Qineti (табл. 10.4).

Таблиця 10.4

Організація	**Дослідники**	**Досягнуті результати**
IBM	Чарльз Беннетт, Жіль Брассард	
Gap-Optique	Ніколас Гисин	67 км
Mitsubishi		87 км, 7.2 біт/с
Toshiba Research Europe		100 км

10.3.2 Стан робіт у галузі квантової криптографії та протоколів

Активні дослідження в області квантової криптографії ведуть IBM, GAP-Optique, Mitsubishi, Toshiba, Національна лабораторія в Лос-Аламосі, Каліфорнійський технологічний інститут, молода компанія Magi і холдинг Qineti, підтримуваний британським міністерством оборони.

Квантова криптографія, як сегмент ринку, тільки починає формуватися і тут поки що на рівні можуть грати і світові комп'ютерні корпорації, і невеликі починаючі компанії. В IBM тривають фундаментальні дослідження в області квантових обчислень, початі групою Чарльза Беннетта. Ними займається корпорація лабораторії Almaden Research Center. Про практичні досягнення IBM у квантовій криптографії відомо не багато - ці роботи мало рекламуються. Дослідникам з Лос-Аламоса вдалося передати фотонний ключ по оптоволокну на відстань 48 км зі швидкістю в кілька десятків кілобітів на секунду. Цього досить, щоб з'єднати між собою відділення банку або урядові заклади.

Створена при участі Женевського університету компанія GAP Optique під керівництвом Н. Гисина [27] об'єднує теоретичні дослідження із практичною

діяльністю. Фахівцям цієї фірми вдалося передати ключ на відстань 67 км із Женеви в Лозанну за допомогою майже промислового зразка апаратури. Цей рекорд був побитий корпорацією Mitsubishi Electric, що передала квантовий ключ на відстань 87 км, правда, на швидкості в 1 байт/с. Дослідження в області квантової криптографії ведуться й у європейському дослідницькому центрі Toshiba Research Europe Limited (TREL), розташованому в Кембриджі (Великобританія). Він спонсорується англійським урядом; у них беруть участь співробітники Кембриджського університету і Імперіал-коледжу в Лондоні. Зараз вони можуть передавати фотони на відстань до 100 км. Таким чином, технологія може бути використана тільки в межах одного міста. Є надія, що незабаром будуть випущені комерційні продукти.

Два роки тому доктор Ендрю Шилдс і його колеги з TREL і Кембриджського університету створили діод, здатний випускати одиничні фотони. В основі нового променевого діода лежить “квантова точка”- мініатюрний шматочок напівпровідникового матеріалу діаметром 15 нм і товщиною 5 нм, що може при подачі на нього струму захоплювати лише по одній парі електронів і дірок. Рекомбінація одного електрона з однією діркою приводить до випущення фотона. При цьому струм, що подається на “квантову точку”, підбирається так, щоб у рекомбінації брала участь тільки одна пара електрон - дірка. Але навіть якщо новий променевий діод випустить два фотони, які будуть характеризуватися різною довжиною хвилі, це дозволяє відітнути зайву частку за допомогою фільтра. Звичайні променеві діоди й лазери випускають фотони групами, що теоретично дає можливість доступу до визначення характеристик окремих фотонів, інші фотони продовжать свій шлях у незмінному виді. Щоб обійти труднощі, пов'язані із створенням джерел окремих фотонів, Фредерик Гроссан з Інституту оптики в Орсе (Франція) розробив методику, що дозволяє шифрувати повідомлення за допомогою імпульсів, що складаються з кількох сотень фотонів. На її безпеку не впливає навіть ослаблення сигналу на більших відстанях. Гроссан відмовився від окремих квантів світла й запропонував усереднювати значення амплітуди й фази електричного поля групи фотонів. Як і поляризація окремого фотона, ці змінні зв'язані одна з іншою принципом невизначеності. Однак, на відміну від поляризації фотона, що приймає одне із двох значень уздовж ортогонального напрямку, ці змінні можуть приймати безперервний ряд значень. Подібні дослідження у квантовій криптографії ведуться одночасно декількома групами. Але тільки групі Гроссана вдалося продемонструвати практичні перспективи, а також створити апаратуру і ПЗ для роботи із квантовим ключем. При вимірюванні безперервного ряду значень уже не обов'язково реєструвати кожний фотон. У ході експериментальної демонстрації вдалося передати зашифровані дані зі швидкістю 75 кбіт/с при тому, що більше половини фотонів губилося. Така схема потенційно має набагато більшу швидкодію, чим схеми з рахунком одиничних фотонів. Це робить її, на думку розроблювачів, досить привабливою для швидкої передачі секретних даних на відстані менше 15 км. Перспективи її використання на більших дистанціях вимагають додаткового вивчення.

У дослідження високошвидкісної квантової криптографії поглибилася і корпорація NEC в особі свого інституту NEC Research Institute. Над прототипами комерційних систем квантової криптографії, що діють по оптоволоконних лініях зв'язку, працює підрозділ телекомунікаційного гіганта Verizon Communications - BBN Technologies. Команда Північно-Західного університету (США) співробітничає з Telcordia Technologies і BBN Technologies, намагаючись довести технологію до комерційного застосування. Їм удалося передати зашифровані дані по оптоволокну зі швидкістю 250 Мбіт/с. Тепер необхідно довести, що схема дозволяє сигналам проходити крізь оптичні підсилювачі. У цьому випадку метод можна буде використати не тільки в спеціальних оптоволоконних лініях зв'язку між двома точками, а й у більш широких мережах. Ще ця команда працює над тим, щоб досягти швидкостей порядку 2,5 Гбіт/с. Дослідження Північно-Західного університету в області квантової криптографії фінансуються DARPA-оборонним відомством США. Міністерством оборони Великобританії підтримується дослідницька корпорація Qineti, що активно вдосконалює технологію квантового шифрування. Ця компанія з'явилася на світ в результаті ділення британського агентства DERA (Defence Evaluation and Research Agency) в 2001 р., увібравши в собі всі неядерні оборонні дослідження. Про свої досягнення вона широкій публіці поки не повідомляє.

До досліджень приєдналося й кілька молодих компаній, у тому числі швейцарська Id Quantique (www.idquantique. com), що представила комерційну систему квантової криптографії, і Magi Technologies (www.magiqtech.com) з Нью-Йорка, що випустила прототип комерційної квантової криптографічної технології власної розробки. Magi Technologies була створена в 1999 р. на засоби великих фінансових інститутів. Крім власних співробітників, з нею взаємодіють науковці із цілого ряду університетів США, Канади, Великобританії й Німеччини. Віце-президентом Magi є Олексій Трифонов, який в 2000 р. захистив докторську дисертацію в Петербурзькому університеті. Рік назад Magi одержала 7 млн. дол. від декількох інвесторів, включаючи засновника Amazon.com Джеффа Безоса. У продукті Magi засіб для розподілу ключів (quantum key distribution, QKD) названий Navajo - по імені індіанців Навахо, мову яких у Другій світовій війні американці використали для передачі секретних повідомлень, оскільки за межами США її ніхто не знав. Navajo здатний у реальному часі генерувати і поширювати ключі засобами квантових технологій і призначений для забезпечення захисту від внутрішніх і зовнішніх зловмисників. Продукт Navajo перебуває в стані бета-тестування і стане комерційно доступним наприкінці року. Кілька комунікаційних компаній тестують Navajo у своїх мережах. Інтерес до квантової криптографії з боку комерційних і військових організацій росте, тому що ця технологія гарантує абсолютний захист. Творці технологій квантової криптографії впритул наблизилися до того, щоб випустити їх з лабораторій на ринок. Залишилося небагато почекати, і вже дуже незабаром квантова криптографія забезпечить надвисокий рівень безпеки для тих, у кого параноя є звичним станом психіки - банкірів і співробітників спецслужб.

Немає жодного сумніву в тому, що криптографічний захист інформації є одним з наріжних каменів інформаційної безпеки. Особливо активно він

використовується на всіх ділянках захисту фінансової інформації: для організації доступу, захисту конфіденційності та цілісності банківської інформації; для підтримки криптовалют та інших цілей інформаційної безпеки. Отже, адекватне розгортання та підтримка криптографічного захисту інформації є надзвичайно важливим завданням, розрахованим на середньо- та довгострокову перспективу.

Для криптографічного захисту інформації розроблено багато криптосистем та криптографічних алгоритмів: симетричні та асиметричні системи шифрування для захисту конфіденційності, криптографічні хеш-функції для процесів аутентифікації та захисту цілісності інформації, системи електронного цифрового підпису для електронного документообігу, методи аутентифікації повідомлень на основі симетричних та асиметричних криптосистем та інші криптографічні процедури та протоколи. Ці криптографічні перетворення інформації довгий час безвідмовно служили людям і успішно використовувалися в усіх галузях людської діяльності, пов'язаних із інформаційними технологіями.

Тим не менше, не так давно, у світі з'явилися загрози, які поставили під сумнів цілісну картину криптографічних технологій, що успішно використовувалися до цього часу. Мова йде про квантові комп'ютери та сукупність алгоритмів для них, практичне впровадження яких сприятиме повному перегляду сучасних концепцій криптографічного захисту інформації.

Рух по цьому шляху вже розпочато. Так, у квітні 2016 року Національний Інститут стандартів і технологій США (NIST) випустив документ NISTIR 8105 DRAFT Report on Post-Quantum Cryptography, в якому оцінив розвиток квантового комп'ютингу та його вплив на стан криптографічного захисту інформації в недалекому майбутньому.

Висновки, до яких прийшли автори документу, скорочено подані у Таблиці 10.5.

Таблица 10.5.

Impact of Quantum Computing on Common Cryptographic Algorithms

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES-256	Symmetric key	Encryption	Larger key sizes needed
SHA-256, SHA-3		Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

Як видно з таблиці, квантовий комп'ютинг (звичайно, при впровадженні масштабованих квантових комп'ютерів) завдає відчутного удару по всіх напрямах криптографії. Зокрема, НІСТ вважає, що для збереження потрібного рівня захисту симетричні криптоалгоритми на зразок AES-256 потребують збільшення довжини криптографічного ключа, функції хешування SHA-256 або SHA-3 будуть стійкими до зламу за допомогою квантових алгоритмів при збільшенні довжини хеш-образу. Слід зауважити, що вплив квантового комп'ютингу не залежить (або слабо залежить) від архітектури функції хешування, оскільки SHA-256 та SHA-3 ґрунтуються на різних підходах: структурі Меркля-Дамгарда та "криптографічної губки" відповідно.

Що стосується асиметричної криптографії, зокрема, криптосистем RSA, а також електронного цифрового підпису як на основі кінцевих полів, так і на еліптичних кривих, НІСТ вважає їх використання небезпечним.

Звичайно, це не означає, що ми повинні вже зараз ламати встановлені системи захисту, вони ще довгі роки будуть стійкими для практичного використання. Прикладом такого підходу можна вважати відношення до криптографічних хеш-функцій MD-5, SHA-1, SHA-2, використання перших двох з яких сьогодні вважається небезпечним і строго не рекомендується, хоча остання залишається практично стійкою, не дивлячись на те, що архітектура її подібна до SHA-1.

Виходячи з наведеного, можна зробити висновок про те, що ми сьогодні є свідками одного з найбільших (якщо не найбільшого) перевороту у криптографічному світі, який можна порівняти, хіба що, з винаходом асиметричної криптографії на рубежі 70-80 років XX сторіччя.

Які ж досягнення квантового комп'ютингу призвели до такої "революції"? Чому квантові комп'ютери можуть зробити те, що вважається технічно неможливим у класичному комп'ютингу? Які квантові алгоритми дозволяють це зробити? І взагалі, які особливості квантових обчислень сприяють такому зростанню обчислювальних потужностей?

Відповіді на ці та інші питання ми спробуємо отримати в наступних розділах, хоча їх складно отримати, не розібравшись детально, що ж таке квантові обчислення. Тому спочатку ми розглянемо основні поняття квантових обчислень, які дозволять нам сформулювати потрібний термінологічний та алгоритмічний апарат.

11 ФІЗИЧНІ ОСНОВИ КВАНТОВОЇ КРИПТОГРАФІЇ

Вступ

В електроніці відомий *закон Мура* (див. розд. 10), згідно якого кількість транзисторів на кристалі інтегральних схем подвоюється кожні 24 місяці (Рис. 11.1). Відповідно, зменшуються розміри цих транзисторів, вдосконалюється технологічний процес. Сьогодні налагоджене масове виробництво мікропроцесорів на основі 20-22-нанометрових технологічних процесів, будуються заводи для 10-нанометрових виробництв, розробляються основи 7-нанометрового процесу. Чи значить це, що обчислювальні потужності, які доступні людству, будуть зростати необмежено? На жаль, відповідь на це питання негативна. Справа в тому, що сучасна технологія вже дуже близько підібралалася до розміру атомів речовини. Дійсно, зараз йдеться про 10-нанометровий технологічний процес, розмір елемента в якому (10 нм) усього в 100 разів більший за розмір атома (0.1 нм). Це означає, що кожен елемент транзистора, наприклад, затвор, буде складатися з ~100 атомів. А що далі? Відомо, що побудувати транзистор (та й будь-який інший елемент мікросхем) з одного атома неможливо.

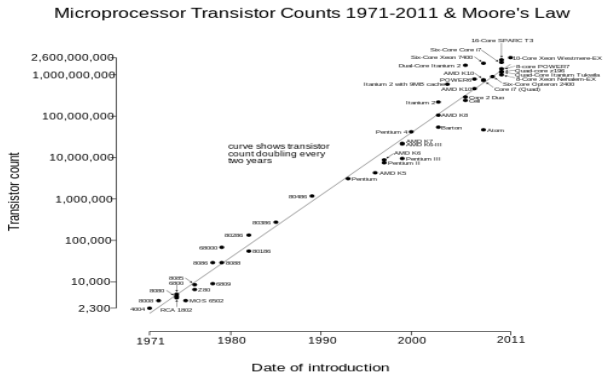


Рис. 11.1. Закон Мура, який демонструє подвоєння числа транзисторів у процесорах кожні 18 місяців

Однак, це ще не все. Існуючі комп'ютери використовують класичні алгоритми та способи збереження інформації. А ці методи для ряду дуже важливих задач вимагають астрономічних розмірів пам'яті і таких самих запасів часу. Наприклад, якщо ми хочемо з максимальною точністю змоделювати еволюцію молекули ДНК, що складається з кількох сотень мільярдів елементарних частинок, то ми повинні будемо керувати одночасно такою кількістю елементів пам'яті, що перевищує кількість елементарних частинок у Всесвіті. Так що навіть якщо би ми навчилися створювати транзистори з одного атома, ряд принципових задач однаково будуть недоступними для наших комп'ютерів. Ще один приклад. Вже зараз сучасні суперкомп'ютери, які моделю-

ють ядерні вибухи, урагани та інші погодні явища, використовують 1,5-2 тисячі процесорів та терабайти оперативної пам’яті, споживаючи енергію, якою могли б користуватися жителі невеликого міста. Отже, необхідно змінювати принципи виконання обчислень.

Які тут можливі варіанти? Перший, який зараз досить успішно працює, це розпаралелювання обчислень [4]. Сьогодні ми використовуємо багатоядерні процесори, розподілені обчислювальні системи тощо. Однак, і тут існують принципові обмеження. Наприклад, відомий *закон Амдала* стверджує, що час виконання обчислень паралельною системою не може бути меншим за час виконання найдовшого фрагменту коду. Більше того, якщо подивитися на рисунок 11.2, який ілюструє закон Амдала, то можна побачити, що навіть якби ми розпаралелили 95% усього коду, ми не досягнемо більше як 20-разового прискорення обчислень.

Ще одним, правда тимчасовим фактором, який сповільнює прогрес швидкості обчислень, є те, що розробники сучасного програмного забезпечення не зовсім дбають про готовність своїх програм до паралельних обчислень. Свідченням цього можуть служити синтетичні тести процесорів, які час від часу виконують комп’ютерні журнали на кшталт iXBT. На рисунку 11.3 показано результати тестування процесорів 8-ядерного Core i7-5960X та 12-ядерного Xeon E5-4650 v3. З малюнку видно, що процесор з меншою кількістю ядер, але на вищій тактовій частоті, виграє у багатоядерного Xeon.

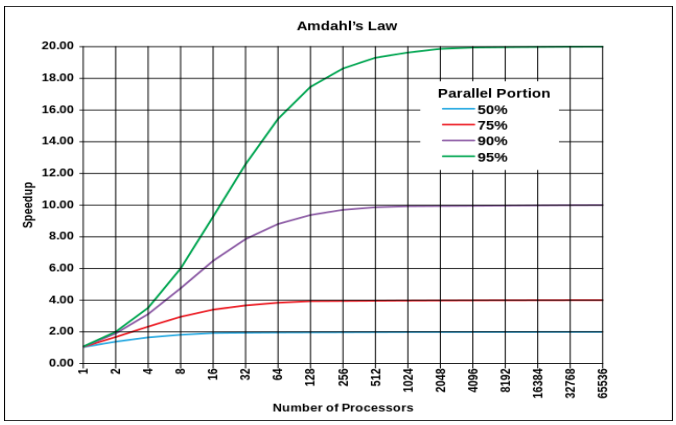


Рис.11.2. Ілюстрація закону Амдала

І знов ми повертаємось до часу виконання послідовних команд. Іншим, досить перспективним, варіантом виходу з такого становища могло б бути використання принципів квантових обчислень, які, як ми побачимо далі, мають дивовижні властивості квантового прискорення.

Однак, не все так просто. По-перше, поки квантова система виконує обчислення, доступ до результатів обмежено. Ми не можемо зупинити систему, подивитись проміжні результати і продовжити обчислення. Таке принципово неможливо. По-друге, надзвичайні проблеми виникають сьогодні при практичній реалізації квантових комп’ютерів. І хоча тут досягнуто деяких важливих результатів, говорити про повний успіх в побудові квантових комп’ютерів зарано. Сьогодні існують лише окремі зразки частково-квантових

обчислювальних систем *Orion*, які випускає канадська фірма D-Wave, але вони призначені лише для виконання окремих, чітко визначених задач. Тим не менше, пригадаємо, що перші комп'ютери, ENIAC, EDVAC та інші, також використовувалися лише для однієї задачі.

І усе ж таки значні переваги квантових комп'ютерів та обчислювальних алгоритмів на основі квантової парадигми змушують дослідників серйозно займатися цими новітніми технологіями, які можуть спричинити чергову революцію в галузі інформаційних технологій.

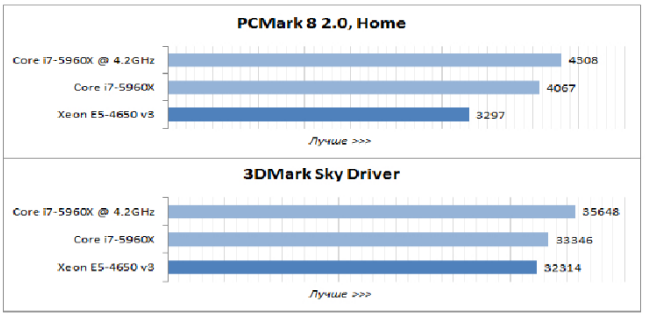


Рис. 11.3. Результати тестування швидкодії процесорів Core i7 порівняно з Xeon E5

Трохи історії квантових обчислень

У 1969 році аспірант Колумбійського університету Стівен Віснер подав до журналу IEEE Information Theory статтю, але вона не була прийнята до публікації, оскільки редакція вважала ідеї, висунуті автором, антинауковими. Крім того, технологічний рівень тих років не дозволяв навіть мріяти про такий рівень захисту. І тільки 1983 року робота Віснера "Conjugate coding" була опублікована в журналі SIGACT News та отримала високу оцінку в наукових колах (Більш детально робота С. Віснера розглянута у підпункті 10.3.1 – Квантові гроші Стівена Віснера).

Суть ідеї С. Віснера полягає в тому, що кожна грошова банкнота повинна мати 20 "світлових пасток", в кожній з яких розміщено по одному фотону з строго визначеним станом поляризації. Така банкнота маркувалася б спеціальним серійним номером, який би містив деяку інформацію про напрями поляризації кожного фотона. Оскільки фальшивомонетник не знає, яку комбінацію фільтрів було застосовано для запису в пастки, він не зможе правильно зчитати інформацію в пастках, і тим більше, повторити таку комбінацію на фальшивій банкноті. Банк, навпаки, зберігає цю послідовність у сховищі, і в будь-який час може перевірити коректність квантового запису на банкноті. Концепція С. Віснера довгий час була (та й зараз лишається) чисто теоретичною, оскільки створити пастки для тривалого зберігання квантового стану не вдається. Сьогодні можуть зберігати квантові частинки у стані суперпозиції на протязі секунд, що надзвичайно мало для такої задачі. Тим не менше, робота С. Віснера була однією з перших, яка мала відношення до захисту інформації з використанням квантових уявлень.

У 1981 році лауреат Нобелівської премії 1965 року, визначний фізик-теоретик Річард Фейнман на Першому конгресі по обчисленнях, який проходив у Массачусетському технологічному інституті, вперше запропонував використовувати способи обчислень на основі уявлень квантової механіки для моделювання квантових систем. У тому ж році професор Массачусетського технологічного інституту Томмазо Тоффолі запропонував квантовий вентиль Тоффолі, який використовується і сьогодні для побудови квантової логіки. У 1982 році В. Вуттерсом, В. Зуреком та Д. Діксом було висунуто так звану "теорему про заборону клонування" у квантових обчисленнях, яка доводить, що невідомий квантовий стан частинки неможливо скопіювати, не зруйнувавши самий стан.

Приблизно в ті ж роки дослідники почали розробляти принципи квантового зв'язку та квантові протоколи передавання даних. Звичайно, до їхньої практичної реалізації було ще далеко, але основи закладалися саме тоді. Наприклад, у 1984 році було запропоновано перший квантовий протокол узгодження криптографічного ключа, який отримав назву BB84 за першими літерами розробників, Чарльза Беннетта з дослідницької лабораторії IBM та Жіля Брассара з університету Монреаля. Сьогодні цей та інші квантові протоколи узгодження криптографічного ключа разом з потоковими шифрами складають основу квантової криптографії. Ч. Беннетт та Ж. Брассар у 1992 році запропонували також принципи квантового надщільного кодування, один з перших квантових протоколів, який надає можливість вибору однієї з чотирьох альтернатив за допомогою передавання одного кубіта. У цьому ж році Чарльз Беннет розробив модифікований протокол квантового узгодження ключа BB84, який отримав назву B92. Цей рік був дуже плодотворним для квантової інформатики. Девід Дойч також запропонував алгоритм, який вперше продемонстрував квантове прискорення обчислень. Оскільки існує квантова теорема про заборону клонування, дослідникам необхідно було розробити альтернативний метод переносу невідомого квантового стану, оскільки без таких операцій створення обчислювальної техніки неможливе. У 1993 році Ч. Беннетт та Ж. Брассар запропонували зовсім нову концепцію, яка отримала назву квантової телепортації. Розроблений протокол дозволяє перенести невідомий квантовий стан на новий квантовий об'єкт, тоді як квантовий стан оригіналу, тобто, об'єкта, стан якого переносився, повністю руйнувався. Цей процес не має нічого спільного з телепортацією реальних об'єктів у творах фантастів, оскільки йдеться не про фізичний об'єкт, а про його квантовий стан. Квантову телепортацію сьогодні успішно реалізовано на відміну від "фантастичної" телепортації.

Настала черга складніших алгоритмів, які безпосередньо впливають на стан сучасної криптографічної науки. Зокрема, алгоритм, розроблений американським математиком Пітером Шором у 1994 році, дозволяє факторизувати великі цілі числа майже за той самий час, який потрібен для їх множення. Зрозуміло, що після розробки цього алгоритму усі існуючі асиметричні криптоалгоритми незалежно від того, на якій математичній проблемі ґрунтується їх криптостійкість, можна вважати скомпрометованими.

За рік, у 1996 році, американський математик індійського походження Лов Гровер запропонував алгоритм швидкого пошуку у невпорядкованому масиві, який легко можна застосувати для пошуку криптографічного ключа будь-якого симетричного криптоалгоритму. Тут важливий саме невпорядкований масив, оскільки у впорядкованому ефективний пошук організувати легше класичними алгоритмами, наприклад, методом дихотомії. За оцінками фахівців, існування цього алгоритму змушує збільшити довжину ключа симетричних алгоритмів у 2-3 рази.

Коротко подамо досягнення у технічній реалізації квантових алгоритмів та протоколів:

- У 1997 році групами фізиків під керівництвом Антона Цайлінгера (Інсбрук, Австрія) та Франческо де Мартіні (Рим, Італія) було реалізовано протокол квантової телепортації стану одного квантового біта на відстань 1 метра. У 2012 році відстань, на яку телепортували квантовий стан, досягла вже 143 км;
- У 2001 році спеціалістами IBM реалізовано алгоритм П. Шора на прототипі квантового комп'ютера. Було розкладено число 15 на прості множники 3 та 5.
- 2005 рік – в Японії продемонстровано програмований квантовий комп'ютер на основі квантового оперативного запам'ятовуючого прис трою з двох квантових бітів.
- 2009 рік – такий же за параметрами квантовий комп'ютер продемонструвало Агентство національної безпеки США.
- 2012 рік– спільнота з чотирьох американських університетів, Південно-Каліфорнійського, Каліфорнійського в Санта-Барбарі, технологічного університету Дельфта та університету штату Айова реалізувала квантовий комп'ютер з двобітовим оперативним запам'ятовуючим пристроєм, який працює при кімнатних температурах. Пристрій складався з квантових комірок пам'яті – кристалів алмазу з домішками азоту. Повідомляють, що був реалізований на такому комп'ютері алгоритм Гровера.

Окремим рядком в історії квантових обчислень стоїть канадська компанія D-Wave. У лютому 2007 року керівництвом цієї компанії було оголошено про реалізацію квантового комп'ютера з шістнадцятибітовим процесором (назва комп'ютера — *Orion*). В листопаді 2007 року, буквально через півроку після *Orion*'u, було продемонстровано роботу прототипу 28-бітового квантового комп'ютера *Leda* на конференції, присвяченій суперкомп'ютерам, а вже у грудні 2009 року науковий співробітник Google Хартмут Невен продемонстрував на комп'ютері D-Wave роботу програми розпізнавання образів. У травні 2011 року компанія випустила квантовий комп'ютер *D-Wave One* з 128-кубітовим процесором. Планується створення комп'ютера з 1024 кубітовим процесором.

Зараз компанія D-Wave успішно випускає квантові комп'ютери "під замовлення". Сьогодні це обчислювальні системи, орієнтовані під конкретну задачу, які працюють під глибоким охолодженням. Вартість таких систем становить приблизно 10 млн. доларів США, отже, дозволити собі такий

комп'ютер можуть лише окремі транснаціональні корпорації. Серед них - Google, який експериментує з розпізнаванням образів за допомогою квантового комп'ютера; Martin-Lockheed, яка використовує квантовий комп'ютер для аналізу критичного програмного забезпечення свого винищувача-бомбардувальника F-35.

Квантові комп'ютери можуть застосовуватись для моделювання ураганів та інших кліматичних явищ, аналізу ДНК та складних ліків, аналізу складних криптоалгоритмів та інших задач, які не під силу класичним комп'ютерним системам, навіть розподіленим.

Тим не менше, компанія дуже довго знаходилася під вогнем критики з боку експертів у галузі квантових обчислень. Справа в тому, що інженери D-Wave ніколи не демонстрували свої комп'ютери світовій спільноті. Компанія продавала лише машинний час, відповідаючи відмовою на вимоги науковців дозволити дослідити роботу цих комп'ютерів. Відповідно, у науковців виникли підозри про те, чи насправді компанія D-Wave розробила комп'ютер, що задовольняє усі вимоги до квантових обчислювальних систем. Основною вимогою є те, що під час обчислень основні вузли комп'ютера повинні знаходитися у стані квантової суперпозиції.

І лише в травні 2013 року компанія запросила канадського професора Кетрін МакГ'ю для порівняльних тестів швидкодії квантового комп'ютера *D-Wave One* на процесорі *Vesuvius* з тодішнім флагманом фірми Intel, *Core i*7. Оскільки *D-Wave One* був спеціально спроектований для розв'язку задач дискретної оптимізації, тести проводили саме на них. В цих задачах *D-Wave One* виконав тести у 3600 разів швидше за свого конкурента. Прискорення обчислень на інших задачах було незначним. І нарешті, в січні 2014 року інженери D-Wave експериментально довели існування квантової суперпозиції в процесорі під час обчислень, для чого використали кубітову тунельну спектроскопію, що було підтверджено інженерами Google, які виявили, що у 1000-кубітовому комп'ютері D-Wave, який вони використовували, кубіти дійсно організовані у кластери по вісім кубітів. Це дозволило досягти зростання швидкодії до 100 млн. разів в одному з алгоритмів порівняно зі звичайним комп'ютером.

У вересні 2016 року на конференції у Сан-Франциско компанія показала свій новий комп'ютер *D-Wave 2X,* який містить 2000 кубітів. Як і раніше, він оптимізований для виконання задач дискретної оптимізації.

Таким чином, можна стверджувати, що наприкінці XX сторіччя на основі розвитку квантової механіки, теорії алгоритмів та теорії інформації було створено **новий розділ науки: квантову інформатику**, яка вивчає загальні принципи та закони, що керують динамікою складних квантових систем. Моделлю таких систем є **квантовий комп'ютер**. Сьогодні концептуальні математична та алгоритмічна бази квантової інформатики створені й успішно розвиваються. Що стосується повноцінного універсального квантового комп'ютера, то вважається, що його ще не створено. Існують лише квантові комп'ютери, розроблені та оптимізовані для окремих задач. Таку ситуацію людство вже проходило зі звичайними класичними комп'ютерами у 40-50-х роках XX сторіччя, коли вони виконували лише одну задачу, для якої їх було

розроблено. Можна очікувати створення універсального квантового програмованого комп'ютера найближчим часом.

Звісно, квантова криптографія, з одного боку, може вважатися самостійним науковим напрямом, а з другого - є частиною ширшої галузі сучасної науки - квантової інформатики. В обох випадках, для того, щоб зрозуміти досягнення цих галузей, необхідно починати з фізичних основ квантової інформатики.

Дослід з "кулеметом"

Згадані алгоритми та протоколи квантових обчислень ми розглянемо пізніше. Зараз же опишемо кілька простих прикладів, що дозволять нам частково зрозуміти дивну поведінку квантових об'єктів, яку абсолютно неможливо пояснити класичним чином. Явище квантової суперпозиції, яке нам дуже потрібне надалі, містить саму суть квантової механіки. Однак, ми досі не можемо його однозначно пояснити, тобто, не можемо відповісти на питання: "Чому це так?". Просто розповімо про те, як воно працює.

Спочатку розглянемо пристрій, схожий на кулемет. Це поганий кулемет. "Кулі", що він випускає, дуже сильно розсіюються (рис. 11.4).

Перед кулеметом розташована броньова плита, а в ній є два отвори, 1 і 2, крізь які кулі проходять вільно. За плитою знаходиться земляний вал, куди влучають кулі, що пройшли крізь отвори. Перед земляним валом рухається "детектор", наприклад, ящик з піском. Будь-яка куля, що влучає в "детектор", лишається в ньому. Коли дослід закінчено, ящик відкривають та рахують кулі, що влучили у нього. "Детектор" рухається в обох напрямках вісі x і дозволяє відповісти на питання: "Яка ймовірність того, що куля, що пройшла крізь плиту, влучить у земляний вал на відстані x від центру?" Відзначимо, що ми говоримо лише про ймовірність, тому що неможливо знати наперед, куди влучить куля. Цю ймовірність можна обчислити, підрахувавши, скільки куль влучило в "детектор" за певний проміжок часу, а потім розділити це число на загальну кількість куль, що влучали у земляний вал за цей же проміжок часу. Якщо швидкість стрільби на протязі експерименту не змінювалася, можна вважати ймовірність пропорційною кількості куль, що влучили у "детектор" за певний час.

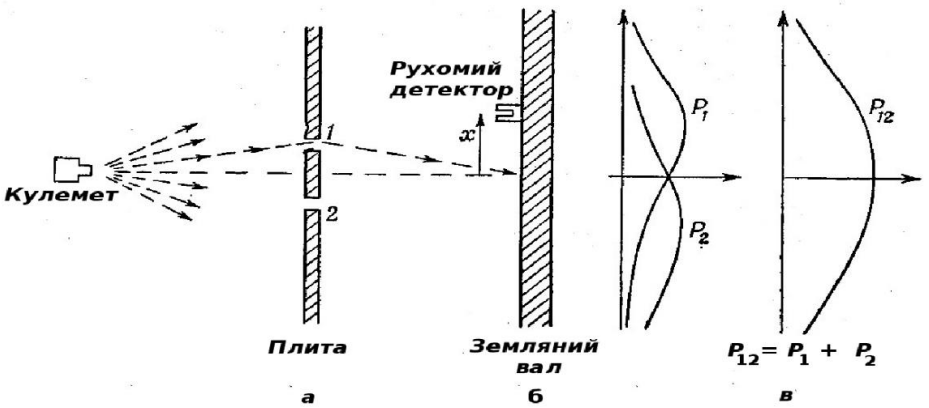


Рис.11.4 Ілюстрація до досліду з «кулеметом»: а – схема «установки»;
б – ймовірність «влучання» від кожної щілини;
в – ймовірність «влучання від обох щілин»

Ми будемо вважати, що кулі влучають лише цілими. Не буває половинок куль або четвертих частинок. Це означає, що кулі “приходять” лише дискретними однаковими порціями, по одній штуці. Навіть якщо швидкість стрільби стає дуже малою, в детекторі завжди знаходиться ціла кількість куль. Розмір порції не залежить від швидкості стрільби.

Результати такого уявного експерименту подані на рис. 11.4 в. Ми позначили ймовірність влучання кулі у детектор через P_{12}, щоб підкреслити, що вони можуть проходити і крізь перший отвір, і крізь другий. Очевидно, що ймовірність у центрі графіку буде більшою, а по його краях – меншою. Тим не менше, не так очевидно, що максимум ймовірності повинен бути при $x = 0$, тобто, у самому центрі графіку.

Це можна легко зрозуміти, якщо повторити дослід, закривши отвори 1 або 2. У першому випадку ми отримаємо криву P_2, а в другому – P_1 (рис. 11.4 б). Максимуми цих кривих, як і слід було чекати, розташовані просто навпроти відкритого отвору. Якщо порівняти рисунки 11.4 б та 11.4 в, то стає зрозумілим, що у випадку двох відкритих отворів ймовірності P_1 та P_2 просто додаються, утворюючи максимум рівно в початку координат $x = 0$, тобто: $P_{12}=P_1+P_2$. Такий результат ми назвемо “відсутністю інтерференції” з причин, про які ви дізнаєтеся пізніше. На цьому ми закінчимо експеримент з “кулеметом”, сформулювавши такі висновки:

- Кулі прибувають лише цілими, «квантуються»; не буває 1,5 або 0,25 кулі.
- Закриваючи отвори, ми отримали відповідні ймовірності $P1$ або $P2$.
- Якщо обидва отвори відкрито, то загальна ймовірність $P12 = P1 + P2$.
- Інтерференція відсутня.

Такі висновки абсолютно традиційні для макроскопічних об’єктів, що й не дивно, оскільки кулі і є макроскопічними об’єктами.

Дослід з хвилями

Розглянемо аналогічний дослід з хвилями на воді, схема якого подана на рисунку 11.5.

Устаткування являє собою мілку посудину, заповнену водою. Предмет, позначений як “Джерело хвиль”, коливаючись за допомогою маленького моторчика вгору-вниз, викликає кругові хвилі. Перегородка затримує хвилі, однак має два отвори, які, згідно з принципом Френеля, є джерелами вторинних хвиль, що досягають екрану та взаємодіють там, викликаючи картину, зображену на рисунку 11.5 *б* та 11.5 *в*. Перед екраном, як і в першому досліді, розміщено рухомий детектор. Якщо його “шкалу” відкалібрувати пропорційно квадрату висоти хвилі h, то відлік буде показувати інтенсивність хвилі. Таким чином, детектор буде вимірювати енергію, що її переносить хвиля, або, що точніше, ту частину енергії, яку передано саме детектору.

Перше, що можна стверджувати за допомогою такого хвилевого апарата, це те, що інтенсивність хвиль може бути довільною. Коли джерело рухається поволі, то детектор демонструє ледве помітний рух. Якщо інтенсивність руху джерела зростає, детектор показує більшу інтенсивність. Оскільки інтенсивність може бути будь-якою, ми вже не можемо стверджувати, що вона має хоча б якусь

дискретність, квантується. Тепер змусимо джерело хвиль працювати стабільно та будемо вимірювати інтенсивність хвиль для різних значень координати *x*. Ми отримаємо криву, подану на рисунку 11.5 *в*.

Як бачимо, хвилі взаємодіють між собою і, як це відомо зі шкільного курсу фізики, утворюють так звану інтерференційну картину. Саме таку картину у вигляді райдужних кілець ми бачимо на поверхні калюж, куди розлито трохи бензину або мастила. Зрозуміло, що первинна хвиля від джерела дифрагує на отворах 1 та 2, в результаті чого ми маємо дві когерентні вторинні хвилі, які й утворюють інтерференційну картину. Це типовий результат для хвильових процесів чи то електромагнітних, чи механічних хвиль.

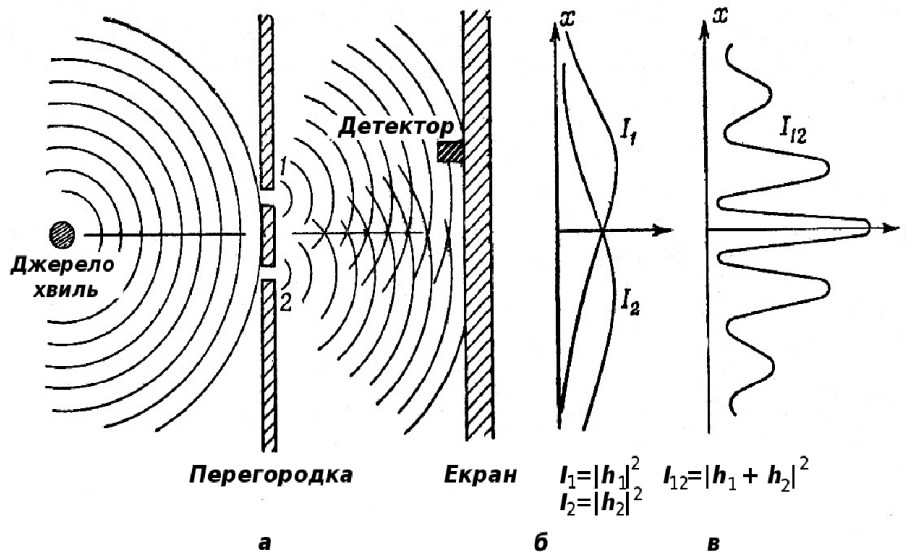


Рис.11.5. Ілюстрація досліду з хвилями на воді: а – схема установки; б – інтенсивність від кожної щілини; в – інтерференція хвиль від двох щілин

Тепер, якщо ми будемо закривати отвори по черзі, то отримаємо криві інтенсивності I_1 та I_2, зображені на рисунку 11.5 *б*. Очевидно, що сумарна інтенсивність I_{12} не дорівнює сумі інтенсивностей від кожної щілини I_1+I_2. В цьому й полягає суть явища інтерференції. На деяких ділянках (де спостерігаються максимуми графіка I_{12}) хвиля від одного отвору підсилює хвилю від другого. Утворюється так звана конструктивна інтерференція, забезпечуючи більшу інтенсивність. Це буває тоді, коли окремі хвилі приходять до деякої точки "у фазі", тобто, відстань від цієї точки до однієї з щілин більше (або менше) відстані до другої на ціле число довжин хвилі. Якщо ж різниця відстаней певної точки екрану до кожної зі щілин кратна не цілому числу довжин хвилі, виникає "деструктивна інтерференція", тобто, загальна інтенсивність зменшується. У цих точках можна побачити мінімуми кривої I_{12}.

Якщо згадати шкільну фізику, ми зможемо записати, що наш хвильовий процес можна описати такими формулами:

$$H_1 = h_1 e^{i\omega t}; H_2 = h_2 e^{i\omega t},$$

де h_1, h_2 — миттєві висоти хвилі, t — час, ω — частота коливань. Якщо відкриті обидві щілини, тоді будемо мати:

$$H_{1,2} = (h_1 + h_2)e^{i\omega t},$$

і це символізує той факт, що загальна висота хвилі є сумою висот хвиль від окремих щілин. Відомо, що інтенсивність пропорційна квадрату висоти хвилі, тобто:

$$I_1 = |h_1|^2; I_2 = |h_2|^2; I_{12} = |h_1 + h_2|^2.$$

Як бачимо, нічого спільного з тим, що було з кулями, не відбувається. Якщо ми розкриємо модуль $|h_1 + h_2|^2$, отримаємо:

$$I_{12} = |h_1|^2 + |h_2|^2 + 2|h_1||h_2|cos(\delta),$$

де δ – різниця фаз між h_1 та h_2. Уводячи інтенсивності, можемо отримати:

$$I_{12} = I_1 + I_2 + 2\sqrt{I_1 I_2}cos(\delta).$$

Як бачимо, останній вираз суттєво відрізняється від випадку “кулемета”. Останній доданок цієї формули - так званий “інтерференційний доданок”, якого немає у випадку класичних об’єктів. На цьому ми закінчимо приклад з хвилями, зробивши такі висновки:

- інтенсивність хвиль може бути довільною - “квантування” у цьому випадку не існує;
- закриваючи отвори почергово, ми отримали інтенсивності хвиль I_1 та I_2;
- якщо обидва отвори відкрито, маємо загальну інтенсивність $I_{12} \neq I_1 + I_2$;
- спостерігається явище інтерференції хвиль.

Такі висновки характерні для хвильових процесів.

Дослід з електронами

І, нарешті, розглянемо ще один дослід, схему якого подано на рис.11.6.

Цей уявний експеримент виконується з електронами, які генерує електронна гармата. Перед гарматою розміщено металічний екран з двома щілинами, а за ним - ще один екран, який поглинає електрони. Перед поглинаючим екраном розташовано рухомий детектор електронів, наприклад, лічильник Гейгера. Увімкнувши обладнання, ми почуємо клацання детектора. Усі сигнали однакові за інтенсивністю. Не буває слабших або сильніших сигналів. Сигнали можуть бути частішими або рідшими, але завжди однаковими за інтенсивністю (гучністю). Коли процес генерування електронів стабілізується, ми зможемо побачити, що кількість зареєстрованих лічильником Гейгера електронів за однакові проміжки часу буде майже однаковою. Тому можна говорити про середню частоту клацань детектора, тобто, певну кількість клацань на хвилину в середньому.

Якщо ми пересуваємо детектор по осі *x*, частота клацань змінюється, але величина (гучність) кожного клацання лишається постійною. Якщо б ми розмістили перед поглинаючим екраном другий детектор, ми б почули, що клацає то один детектор, то другий, і ніколи обидва разом (хіба що іноді наше вухо не в змозі розрізнити два клацання підряд). Тому ми можемо зробити висновок про те, що електрони, як кулі в першому досліді, приходять “порціями”. Усі “порції” однакові за величиною, в детектор або в поглинаючий

екран влучає лише одна “порція”. Тоді ми говоримо: “Електрони завжди приходять однаковими порціями”.

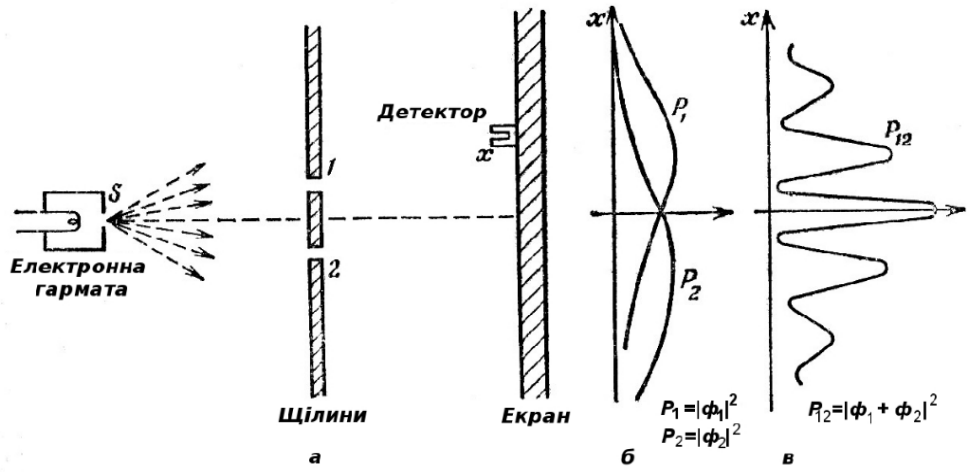


Рис. 11.6. Ілюстрація досліду з електронами: а – схема «установки»; б – інтенсивність від кожної щілини; в – інтенсивність від обох щілин

Як і в досліді з кулеметом, спробуємо знайти відповідь на питання: “Яка відносна ймовірність того, що електрон влучить у поглинаючий екран на різних віддалях від середини?”. Ми можемо отримати відповідь на це питання, підрахувавши частоту клацань лічильника під час стабільної роботи електронної гармати. В результаті такого уявного експерименту отримана дуже цікава крива, яка подана на рисунку 11.6 *в*. Слід відзначити, що саме так і поводять себе електрони в умовах реального експерименту. Спробуємо пояснити таку дивну поведінку електронів. Оскільки ці електрони, як ми бачили, приходять “порціями”, тобто, квантуються, то кожен з них проходить або крізь отвір 1, або крізь отвір 2. Тоді усі електрони, які влучили у поглинаючий екран, можна розділити на два класи: ті, що пройшли крізь перший отвір, і ті, що пройшли через другий. Це значить, що крива 11.6 *в* - це сума ефектів від електронів, які пройшли крізь різні отвори. Тепер закриємо отвір 2 і будемо вивчати тільки електрони, що пройшли крізь отвір 1. Результат таких вимірів подано на рисунку 11.6 *б*, крива P_1. Там же подано результати для отвору 2 – крива P_2. Вони виглядають звичним чином.

Крива P_{12}, яку отримано, коли обидва отвори відкрито, зовсім не узгоджується з сумою P_1+P_2. Очевидно, що вона аналогічна до такої в досліді з хвилями. Отже, ми чомусь бачимо інтерференційну картину. Що тут взаємодіє, щоб дати таку картину? Може, взаємодіють електрони, що пройшли крізь різні отвори? Але як це можливо, якщо вони “прибувають” до екрану дискретно і у різний час? Може вони рухаються складніше, проходячи спочатку через один отвір, а потім – через другий? Але тоді, закривши один отвір, ми маємо зменшити кількість електронів, що досягли екрану. Тим не менше, ми бачимо на кривій 11.6 точки, де кількість електронів більша за таку, коли відкрито обидва отвори. Виходить, що закриття одного отвору збільшує кількість електронів, що проходить крізь другий. І навпаки, середній пік кривої P_{12} більш, ніж удвічі

вищий за суму значень кривих P_1 та P_2. Очевидно, що ці два факти протирічать один одному! Пояснити обидва факти з точки зору складних траєкторій руху електронів неможливо. В усякому разі досі нікому це не вдалося зробити.

Математика, що описує зв'язок P_{12}, P_1 та P_2 , зовсім проста. Вона дуже схожа на ту, яка описувала дослід з хвилями. Такий результат не вдається отримати, якщо вважати, що електрони рухаються по складних траєкторіях. Отже, у нас немає іншого виходу, як визнати: електрони прибувають "порціями", тобто, виявляють при цьому властивості частинок (корпускул), а ймовірність їх влучання в екран розподіляється так само, як інтенсивність хвиль. Саме в такому розумінні електрон поводить себе частково як частинка, частково як хвиля. Також ми бачимо, що $P_{12} \neq P_1+P_2$. Очевидно, твердження про те, що електрони проходять або крізь отвір 1, або крізь отвір 2, неправильне.

Спробуємо модифікувати наш уявний експеримент. Відомо, що електрони добре взаємодіють зі світлом. Схема такого експерименту подана на рисунку 11.7. Якщо ми візьмемо деяке джерело світла і розмістимо його на шляху електронів, то за спалахами світла, наприклад, біля отвору 1 (чи 2) ми будемо знати, через який з них пройшов черговий електрон. А якщо ми побачимо спалах світла біля обох отворів, це буде свідчити, що ми невірно уявляємо собі реальність мікросвіту.

Тепер щоразу, коли ми чуємо клацання детектора, ми бачимо спалах або біля першого, або біля другого отвору, але ніколи біля обох одразу. Так відбувається при будь-якому положенні детектора. Звідси можна зробити висновок про те, що коли ми спостерігаємо за електроном, він з необхідністю проходить лише через один отвір: або через перший, або через другий. Продовживши спостереження, побачимо, що цього разу ми отримаємо ймовірність від двох щілин точно таку ж, як і в досліді з "кулеметом", тобто $P'_{12} = P'_1 + P'_2$. Чому ж тоді в попередньому експерименті ми отримали $P_{12} \neq P_1+P_2$? В чому різниця між експериментами? Єдина різниця полягає в тому, що в модифікованому експерименті ми спостерігали за електронами. Напевне, цей результат можна зрозуміти так: електрони настільки тендітна річ, що ми своїм джерелом світла збурюємо електрони, змінюючи їхню поведінку.

Перевіримо цей факт. Будемо зменшувати яскравість джерела світла. Тоді ми не будемо надавати електронам такого збурення, щоб впливати на їхню поведінку. Перше, що ми відзначимо, це те, що яскравість спалаху при взаємодії світла з електроном не зменшується. Це й зрозуміло, оскільки світло – це потік фотонів. При зменшенні яскравості світла параметри самих фотонів не змінюються, а лише зменшується їх кількість у потоці. Зменшуючи яскравість, ми, нарешті, прийдемо до такої ситуації, що фотонів для взаємодії з електронами не буде ставати, і ми будемо пропускати електрони, тобто, лічильник клацнув, зафіксувавши електрон, але ми не побачили спалаху, бо на тому місці, де перебував електрон, в той час не було фотона. Таким чином, у нас буде вже три типи електронів: ті, що пройшли крізь перший отвір, ті, що крізь другий, і ті, яких ми не побачили. Саме тоді явище інтерференції знову з'являється!

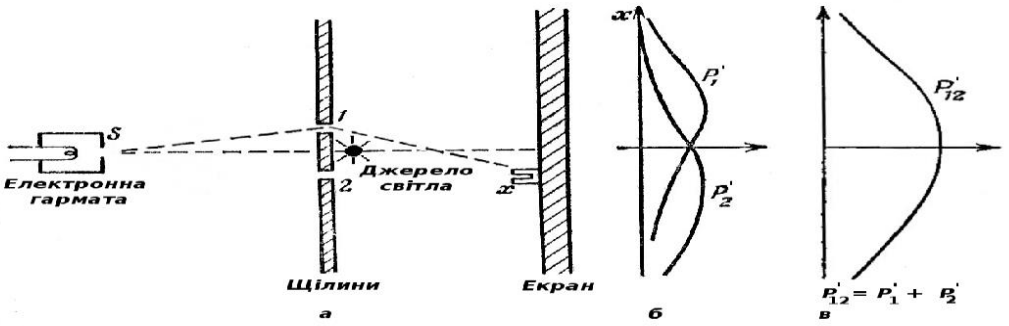


Рис. 11.7. Ілюстрація модифікованого досліду з електронами

Ну, що ж, це вже можна пояснити. Коли ми не побачили електрон, фотон не збурив його; а якщо ми його помітили, то фотон провзаємодіяв з електроном, змінивши його енергію якимось чином. Оскільки фотони усі однакові, значить степінь збурення електрона завжди одна й та ж, достатня для того, щоб зруйнувати явище інтерференції. Якщо ми міркуємо правильно, то існує ще один можливий спосіб "неруйнуючого" спостереження за електронами. Якщо взаємодія фотона з електроном, як це описується у фізиці, супроводжується передаванням електрону імпульсу фотона ($\boldsymbol{p = h/\lambda}$), очевидно, що чим більша довжина хвилі, тим менший імпульс ми йому передамо. Для цього треба подіяти світлом червоного (може, навіть, інфрачервоного) діапазону. Якщо ми будемо модифікувати дослід у такий спосіб, побачимо, що при збільшенні довжини хвилі спочатку нічого не зміниться. Однак потім трапиться дуже неприємна річ: як тільки довжина світла стане порядку віддалі між щілинами, спалахи стануть такими розмитими, що ми не зможемо визначити, біля якої з них він відбувся. Бачимо, що електрон проскочив, а крізь яку щілину – не зрозуміло. І знову, саме в цей момент, коли почнуть з'являтися "невраховані" електрони, крива $\boldsymbol{P_{12}}$ продемонструє появу інтерференції! При подальшому збільшенні довжини хвилі інтерференція стане все помітнішою, і, нарешті, коли довжина хвилі буде набагато більшою за відстань між щілинами, і вже зовсім незрозуміло, де відбувся спалах, картина інтерференції стане подібною на таку, яка дана на рисунку 11.7 *в*.

Як з'ясувалося, в роки становлення квантової механіки така поведінка мікрооб'єктів була помічена в багатьох експериментах. Вернер Гейзенберг припустив, що закони мікросвіту та макросвіту можна було б узгодити, якщо вважати, що існують деякі фундаментальні обмеження наших експериментальних можливостей, яких раніше не помічали. Він запропонував як загальний принцип так званий принцип невизначеності Гейзенберга. Цей принцип звучить так: "*Принципово неможливо одночасно виміряти з довільною точністю координати і імпульси квантового об'єкта*". У термінах нашого експерименту він звучить трохи інакше: "*Неможливо створити устаткування для визначення того, через яку щілину пройшов електрон, не збурюючи його настільки, що інтерференційна картина зникає*". Якщо устаткування здатне визначити, крізь яку щілину пройшов електрон, воно не здатне бути настільки делікатним, щоб не спотворити картину інтерференції. Жодній людині ніколи не

вдавалося винайти устаткування чи просто вказати спосіб, як обійти принцип невизначеності. Тому ми зобов'язані припустити, що він відображає одну з фундаментальних властивостей Природи.

Принцип невизначеності Гейзенберга надає нам єдиний спосіб, за допомогою якого ми повинні розмірковувати, щоб не зробити помилкових передбачень. Цей спосіб полягає в тому, що якщо у нас є прилад, здатний визначити, крізь яку щілину пройшов електрон, ми можемо говорити про те, пройшов він крізь першу (або другу). Але якщо ви не намагалися дізнатися, де пройшов електрон, то ви не можете навіть думати про те, чи електрон пройшов крізь першу, чи крізь другу щілину. Якщо ж почати розмірковувати на цю тему, а потім робити з цього якісь припущення, можна прийти до протиріч та помилкових фізичних висновків. Отже, ми повинні балансувати на цьому логічному лезі, щоб успішно описувати явища мікросвіту.

Якщо рух усієї речовини подібний до руху електронів в нашому уявному експерименті і його можна описувати за допомогою хвилевих понять, то як бути з кулями в нашому першому експерименті? Чому там ми не побачили інтерференційної картини? Спробуємо це пояснити.

Хвиля де Бройля, за допомогою якої ми можемо відповісти на це запитання, визначається з виразу: $\lambda = h/mv = 6.62 \times 10^{-34}/mv$. Припустимо, що швидкість кулі та електрона однакова -1000 м/с. Тоді отримаємо довжину хвилі де Бройля:

- Для електрона: $\lambda = 6.62 \times 10^{-34}/9.1 \times 10^{-31} \times 10^{3} \approx 0.7$ м;
- Для «кулі»: $\lambda = 6.62 \times 10^{-34}/10^{-2} * 10^{3} \approx 7 \times 10^{-35}$ м.

Чи існує такий прилад, який може виміряти сигнал з довжиною хвилі 10^{-34} м? З такою довжиною хвилі інтерференційні піки будуть настільки тонкими, що жоден детектор їх просто не зможе розрізнити, а буде просто показувати середнє значення або огинаючу такого коливного руху.

З проведених уявних експериментів можна зробити такі висновки:

1. Електрони, як і інші квантові об'єкти, поводять себе частково як частинки (вони прибувають до екрану дискретними порціями), частково, як хвилі (дають на екрані інтерференційну картину).

2. Надзвичайно важливим є висновок про те, що вимірювання в мікросвіті впливають на стан квантової системи. Цей висновок накладає просто екстремальні обмеження на усю квантову інформатику, адже читання чи запис даних еквівалентні процесу вимірювання. Отже, необхідно буде розробляти алгоритми, які працюють без проміжного читання/запису.

3. Цей висновок, до деякої міри, випливає з попереднього. Якщо неможливо чітко визначити траєкторію руху електрона без впливу на результат, то ми повинні говорити лише про ймовірність тієї чи іншої траєкторії. Це справедливо не тільки для руху електронів, але й для усіх квантових систем: ми можемо лише говорити про ймовірність того чи іншого значення в квантовій системі. Якщо ми захочемо дізнатися точно, яке значення міститься у квантовому ОЗП, ми повинні провести вимірювання, а значить, вплинемо на саме це значення. Чи отримаємо ми в результаті таких дій істинне

значення змінної – велике питання. Воно буде поставати перед нами усюди в квантовій інформатиці.

Незвичність подій, які виникають при виконанні квантових обчислень і не мають аналогів у класичних обчисленнях, ставить перед математиками незвичні завдання, які необхідно вирішити в процесі розробки квантових алгоритмів. Тим не менше, ці завдання успішно вирішуються, що говорить про те, що основні засади квантової інформатики розроблено, а на часі – технічна реалізація універсальних квантових комп'ютерів.

Додатково до розглянутих уявних експериментів див. Підрозділи (7.2.4 – 7.2.6).

Квантові біти

Опис простої квантової системи

Як же ми можемо математично описати таку систему, про яку ми говорили вище? Для коректного описання придумано спеціальний формалізм.

Позначимо стан квантової системи, коли частинка пройшла крізь верхній отвір як $|0\rangle = \binom{1}{0}$. Кутові дужки вказують на те, що це не просто "0", а стан квантової системи. Як бачимо, тут існує деякий зв'язок з реальністю, оскільки одиниця стоїть там, де пройшла частинка, а нуль – там, де її не було. Аналогічно можна записати для випадку, коли частинка пройшла крізь нижній отвір: $|1\rangle = \binom{0}{1}$. Для цього випадку міркування абсолютно аналогічні. Описані стани називаються базисними, оскільки вони описують ситуацію, коли частинка точно пройшла крізь один отвір, і не проходила крізь другий. Однак, ми з'ясували, що існують такі ситуації, коли ми не можемо напевне сказати, крізь який отвір пройшла частинка. Ми знаємо лише, що з деякою ймовірністю вона може проскочити крізь отвір 1, і з деякою – крізь отвір 2. Такий стан квантової системи ми можемо описати суперпозицією станів $|0\rangle$ та $|1\rangle$:

$$|Q\rangle = a|0\rangle + b|1\rangle. \qquad (11.1)$$

Коефіцієнти a та b описують ймовірності проходження частинки крізь відповідні отвори. Ці коефіцієнти повинні задовольняти деякі умови нормування. Вони описують той факт, що частинка точно пройшла або крізь перший, або крізь другий отвір, тобто, сума ймовірностей проходження крізь кожен отвір повинна дорівнювати одиниці: $|a|^2 + |b|^2 = 1$. Формула (11.1) є відображенням так званого *принципу суперпозиції*, одного з основних і найважливіших принципів квантової механіки. Цей принцип формулюється так: *якщо квантова система може перебувати в станах* $|0\rangle$ *та* $|1\rangle$*, то вона може перебувати й у будь-якій суперпозиції цих станів.*

Розглянемо простий приклад опису квантової системи. Нехай система перебуває в стані, що описується виразом:

$$|Q\rangle = \frac{4}{5}|0\rangle + \frac{3}{5}|1\rangle.$$

Якою буде ймовірність того, що при вимірюванні ми отримаємо нуль?

$$P(0) = \left|\frac{4}{5}\right|^2 = \frac{16}{25} = 0.64.$$

Ймовірність отримати при вимірюванні одиницю:

$$P(1) = \left|\frac{3}{5}\right|^2 = \frac{9}{25} = 0.36.$$

Це означає, що в 64% випадків вимірювань ми отримаємо “0”, а в 36% - “1”.

Які ж реальні фізичні системи, крім проходження частинки крізь щілини, можуть бути такою квантовою системою? Однією з найпростіших та найбільш популярних вважається спін мікрочастинки, наприклад, електрона. Коротко та дуже умовно спін частинки можна описати таким чином. Якщо уявити собі заряджену частинку як маленьку кульку, то її обертання, як обертання будь-якого зарядженого тіла, створює магнітний момент, як це показано на рисунку 11.8. Цей магнітний момент і називається спіном (від англ. *To spin* – обертатись) (див. підрозділ 7.2.4).

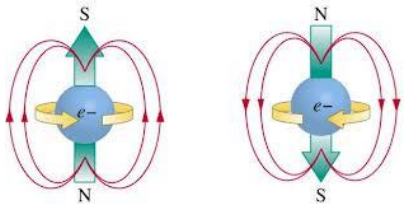


Рис.11.8. Ілюстрація до пояснення спіну електрона

І хоча насправді це зовсім не так, але така аналогія прижилася, і ми будемо нею користуватися. В залежності від напрямку обертання частинки напрям магнітного моменту може бути різним. Якщо спін частинки направлений вгору (як це показано на мал. 11.8, ліворуч), будемо позначати такий стан як $|0\rangle$, а стан, коли спін направлено вниз – через $|1\rangle$. Тоді знову стан, коли ми не знаємо, куди направлений спін частинки, тобто, стан суперпозиції, можна записати як

$$|Q\rangle = a|0\rangle + b|1\rangle.$$

І знов коефіцієнти при базисних станах вказують на ймовірність отримати при вимірюваннях нуль або одиницю. Звичайно, вони повинні задовольняти ту саму умову нормування: $|a|^2 + |b|^2 = 1$. Як буде реагувати така система на вимірювання? Припустимо, що ми при першому вимірюванні отримали нуль. В якому стані тоді перебуватиме система?

$$|Q\rangle = 1|0\rangle + 0|0\rangle = |0\rangle.$$

Як бачимо, система перейшла в стан $|0\rangle$ і при наступних вимірюваннях із 100% ймовірністю ми її знайдемо в такому стані. Це означає, що до вимірювання система знаходилася в стані суперпозиції, а після вимірювання перейшла у строго визначений стан $|0\rangle$. Таким чином, ми можемо припустити (і це підтверджується численними експериментами), що *вимірювання впливають на* ***стан квантової системи***. Більше того, скільки разів ми би не повторювали

вимірювання, квантова система більше не змінить свій стан, і ми отримуємо ***звичайну класичну систему з чітко визначеним станом***.

Часткові вимірювання

Раніше ми обговорювали той факт, що вимірювання впливають на стан кубітів, перетворюючи його з суперпозиційного у класичний, цілком визначений стан. При подальших вимірах цей стан вже не змінюється, залишаючись весь час однаковим. Тим не менше, ми можемо виконувати так звані *часткові вимірювання*, вимірюючи окремі біти квантової системи. Для кращого розуміння проблеми розглянемо систему з двох квантових бітів у стані:

$$|Q\rangle = \frac{3}{5}|00\rangle + \frac{2}{5}|01\rangle + \frac{2}{5}|10\rangle + \frac{2\sqrt{2}}{5}|11\rangle.$$

Спробуємо виміряти другий кубіт. З якою ймовірністю ми отримаємо одиницю? Другий кубіт не дорівнює нулю у другому та четвертому доданках. Тоді ймовірність того, що ми при вимірюванні отримаємо одиницю, можна записати так:

$$P(1) = \left(\frac{2}{5}\right)^2 + \left(\frac{2\sqrt{2}}{5}\right)^2 = \frac{4}{25} + \frac{8}{25} = \frac{12}{25}.$$

В якому стані опиниться система, якщо ми отримаємо одиницю в другому біті? Зрозуміло, що після вимірювання другий біт буде завжди дорівнювати одиниці, отже, перший та третій доданки зникнуть, оскільки там другий біт дорівнює нулю. Але просто так забрати ці доданки не можна, оскільки необхідно задовольнити умову нормування. Отже, треба обчислити нові коефіцієнти біля доданків з таких умов:

$$|a|^2 + |b|^2 = 1;\ \frac{a}{b} = \frac{2/5}{2\sqrt{2}/5} = \frac{\sqrt{2}}{2}.$$

Отже, для обчислення нових коефіцієнтів необхідно розв’язати систему:

$$\begin{cases} \dfrac{a}{b} = \dfrac{\sqrt{2}}{2}; \\ |a|^2 + |b|^2 = 1. \end{cases}$$

Обчисливши коефіцієнти, отримуємо такий стан системи:

$$|Q\rangle = \frac{1}{\sqrt{3}}|01\rangle + \sqrt{\frac{2}{3}}|11\rangle.$$

Як бачимо, часткові вимірювання хоч і не руйнують систему, однак також змінюють її квантовий стан.

Переплутані стани

Розглянемо систему з двох кубітів. Простір станів такої системи буде описуватися виразом:

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$$

або

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\},$$

де $\otimes$ - тензорний добуток, який визначається наступним чином:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}.$$

Як бачимо, тензорний добуток збільшує розмірність вектора до матриці. Тензорний добуток підкоряється стандартним для математики властивостям лінійності, асоціативності та комутативності, що дозволяє в деяких випадках працювати з ним аналогічно до звичайного добутку. У загальному випадку, як це було показано раніше, система з n кубітів має 2^n базисних станів, тобто, станів, тензорний добуток яких дає повний простір квантових станів системи.

Однак, не завжди усі стани квантової системи можна зобразити станами окремих кубітів. Наприклад, стан $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ не можна зобразити через стани окремих кубітів. Це означає, що не існує таких коефіцієнтів a_1, a_2, b_1, b_2, щоб:

$$(a_1|0\rangle + b_1|1\rangle)(a_2|0\rangle + b_2|1\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \qquad (11.2)$$

Дійсно:

$$(a_1|0\rangle + b_1|1\rangle)(a_2|0\rangle + b_2|1\rangle) =$$
$$= a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle.$$

Для того, щоб виконувалася рівність (11.2), необхідно, щоб a_1b_2=0 та b_1a_2=0. Це означає, що a_1a_2 = 0 та b_1b_2 = 0. Отже, стан $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ не може бути суперпозицією станів $(a_1|0\rangle + b_1|1\rangle)$ та $(a_2|0\rangle + b_2|1\rangle)$.

Таким чином, стан $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ є так званим *переплутаним станом* по відношенню до базису $(a_1|0\rangle + b_1|1\rangle)$ та $(a_2|0\rangle + b_2|1\rangle)$.

Іншим способом визначення *переплутаного* стану є такий. Ми знаємо, що тензорний добуток має властивості асоціативності та комутативноності. Отже, ми можемо виносити за знак тензорного добутку спільні множники. Наприклад, стан

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$$

можна зобразити як

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes (|0\rangle + |1\rangle)$$

або $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$ – у вигляді $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle$. Такі стани будемо вважати не переплутаними. А от квантовий стан $|Q\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

не можна подати у вигляді тензорного добутку двох однокубітових станів. Отже, “винести спільний множник” з такої конструкції не можна. Такий стан будемо вважати переплутаним. З такої точки зору квантовий стан $|\phi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ також вважається переплутаним.

Можна подивитися на переплутані стани ще з іншого боку, з точки зору вимірювань. Частинки не будуть переплутаними, якщо вимірювання однієї з них не вплинуть на стан іншої. Наприклад, стан $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = \frac{1}{\sqrt{2}}|0\rangle \otimes (|0\rangle + |1\rangle)$ не вважається переплутаним, оскільки при будь-якому вимірюванні першого біта ми отримаємо нуль незалежно від того, вимірювався другий біт чи ні. Аналогічно і вимірювання другого біта може дати нуль або одиницю з такою ж ймовірністю незалежно від того, чи вимірювався перший біт.

Цього не можна сказати про стан $|Q\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. В цьому випадку отримані значення будь-яких кубітів істотно залежать від того, чи проводилося вимірювання іншого біта. Наприклад, якщо при вимірюваннях першого біта ми отримали нуль, то значення другого біта може бути тільки нуль, і ніяк не одиниця. Якщо ж при вимірюванні першого біта отримано одиницю, то другий автоматично також дорівнює одиниці. Аналогічні міркування справедливі і для другого біта. Таким чином, цей стан буде переплутаним. Переплутаним буде також стан $|\phi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ або $|\theta\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$, який, по суті, є тим самим станом $|\phi\rangle$.

Підсумуємо викладену інформацію. Квантовий стан, наприклад двокубітний, будемо вважати переплутаним, якщо:

1. Його не можна зобразити у вигляді суперпозиції двох однокубітних станів;

2. З виразу, який характеризує стан квантової системи, не можна винести “спільний множник”;

3. Значення, отримані при вимірюванні одного з кубітів, істотно залежать від результатів вимірювання іншого.

Переплутані стани відіграють визначальну роль у квантових алгоритмах, безпосередньо відповідаючи за квантове прискорення обчислень.

Парадокс Ейнштейна-Подольські-Розена

Використовуючи поняття переплутаного стану, Альберт Ейнштейн, Борис Подольські та Натан Розен запропонували уявний експеримент, який, здається, порушує основні принципи теорії відносності. Цей експеримент було описано в їхній статті “Чи можна вважати квантово-механічне описання фізичної реальності повним?”, яка вийшла у друці 1935 року. Цей експеримент пізніше було названо парадоксом Ейнштейна-Подольські-Розена (ЕПР-парадоксом).

Суть цього парадоксу полягає в наступному.

Припустимо, у нас є джерело, яке генерує максимально переплутані частинки, наприклад, в стані $|Q\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Ці частинки називаються

ЕПР-парою. Припустимо, що учасники експерименту отримали по одній частинці з цієї пари і роз'їхалися на максимально можливу відстань.

Нехай власник першої частинки виконав вимірювання її параметрів і з'ясував, що його частинка знаходиться в стані $|0\rangle$. Це означає, що загальний стан системи повинен бути $|00\rangle$. Отже, коли власник другої частинки виконає вимірювання, він також отримає $|0\rangle$. Аналогічно, якщо перша частинка при вимірюваннях продемонструє стан $|1\rangle$, то друга неодмінно буде в такому ж стані $|1\rangle$. Зміна стану відбувається миттєво, не дивлячись на те, що частинки знаходяться на довільній відстані. Виходить, що учасники експерименту можуть миттєво обмінюватися інформацією. Однак, це протирічить спеціальній теорії відносності, яка доводить, що найбільша швидкість передавання інформації дорівнює швидкості світла у вакуумі.

Насправді суть парадоксу значно глибша. В загальному вона полягає в тому, чи може квантова механіка адекватно описувати фізичну реальність, тобто, чи це повна наука і вона не протирічить іншим, відомим та доведеним теоріям, чи квантова механіка є наукою, яка вимагає значного вдосконалення й узгодження з цими теоріями.

Сьогодні існує кілька інтерпретацій цього парадоксу.

1. Теорія прихованих параметрів Девіда Бома.
2. Теорема Белла та її експериментальні перевірки.
3. Багатосвітова інтерпретація.
4. Квантовомеханічна інтерпретація (інтерпретація Н. Бора).

Розглянемо ці тлумачення детальніше.

Теорія прихованих параметрів Девіда Бома.

У 1952 році Девід Бом у своїй книзі “Квантова теорія” запропонував реальний фізичний експеримент, який на той час здійснити було нереально, однак він дозволяв безпосередньо перевірити правильність різних тлумачень ЕПР-парадоксу. Суть експерименту полягає в тому, що деяке джерело випромінює два фотони у переплутаному стані в різні боки. Кожен фотон має якісь свої конкретні значення компонентів спіну в напрямках x, y, z. Ми можемо вимірювати ці значення, але тільки один компонент за раз. Нехай ми виміряли x-компоненту спіну першого фотона і отримали якусь величину x_0. Оскільки фотони знаходяться у переплутаному стані, то яке б не було значення y-компонента першого фотона, y-компонент спіну другого фотона обов'язково повинен бути протилежним за напрямом. Таким чином, вимірюючи y-компонент першого фотона, ми неявним чином вимірюємо такий самий компонент другого, як це й було описано в ЕПР-парадоксі. Якщо б така картина була справедливою для усіх вимірів та усіх напрямів випромінювання фотонів, то це протирічило б нерівності Гейзенберга, яка стверджує, що не можна достовірно виміряти дві величини однієї частинки. На основі таких міркувань, Д. Бом робить висновок про те, що квантова механіка як наука неповна, таким чином припускаючи існування прихованих параметрів, що можуть розв’язати цю проблему:

“… *сучасна форма квантової теорії вказує на те, що Всесвіт не можна однозначно відобразити жодною сукупністю строго визначених математичних*

величин, і що повна теорія завжди вимагатиме більш загальних понять, ніж розкладання на точно визначені компоненти …”

Що означає поняття “прихованих параметрів”? Розглянемо два таких твердження:

1) В момент часу $t = 1$ *с* частинка буде знаходитися в точці з координатою $x = 1$ м. Це – приклад класичного, детерміністичного твердження.

2) В момент часу $t = 1$ *с* частинка буде знаходитися між координатами $x = 1$ м та $x = 1{,}1$ м з ймовірністю $P = 0{,}8$. Це приклад того, що можуть дати квантово-механічні обчислення.

Як бачимо, квантово-механічні обчислення передбачають наявність деякого елемента випадковості у фізичних процесах. У зв’язку з цим фізики і розділилися на кілька таборів. Одні вважали, що випадковість притаманна самій фізичній природі. Інші були впевнені, що жодної випадковості в природі не існує, а ймовірності, які виникають у квантово-механічних розрахунках, насправді є наслідком недосконалості самої теорії та нашого неповного знання про природу речей. Загалом ці два підходи можна охарактеризувати таким чином.

Класичний підхід: поточний стан фізичної системи повністю та однозначно визначає її майбутній стан. Наприклад, якщо зараз система знаходиться в стані “А”, то через секунду вона буде знаходитися в стані “В”. І якщо квантова механіка стверджує, що через секунду квантова система буде знаходитися або в стані “В”, або в стані “С”, то це лиш значить, що вона при розрахунках не враховує якісь невідомі (приховані) параметри системи, врахування яких може однозначно визначати її майбутній стан.

Квантовий підхід: поточний стан системи несе в собі кілька (загалом – безмежну кількість) можливих варіантів її майбутнього стану. Який саме з цих можливих варіантів буде реалізовано – принципово неможливо передбачити, оскільки вибір тут відбувається абсолютно випадково, причому немає жодних прихованих параметрів, які б могли вплинути на цей вибір. Можна говорити лише про ймовірність того чи іншого вибору.

Прибічники класичного та квантового підходу сперечаються вже багато років, але жодна сторона не може надати вирішальних доказів на користь тієї чи іншої моделі. Тим не менше, часто результати експериментів, наприклад, таких, як запропонував Д. Бом, можна пояснити за допомогою понять квантової механіки досить просто, не залучаючи до розгляду теорії прихованих параметрів. Дійсно, нехай у нас є поляризатор, який випромінює фотони. Поляризатор має два канали “+” та “-”, направлені в різні боки. Ймовірність виявити фотон в будь – якому каналі – випадкова величина, причому рівноймовірна. Отже, $P_+(a)=P_-(a) = 0{,}5$. Аналогічно й для іншого фотона: $P_+(b) = P_-(b) = 0{,}5$.

Саме цей результат говорить про те, що ми не можемо сказати точно, яка поляризація у кожного фотона переплутаної пари, оскільки це процес ймовірнісний, і кожне вимірювання її дасть випадковий результат з ймовірністю 0,5. Тоді можна підрахувати ймовірності того, що фотони будуть однакової або різної поляризації:

$$P_{++}(a,b) = P_{--}(a,b) = ½ \cos^2 (a,b);$$

$$P_{+-} (a,b) = P_{-+} (a,b) = ½ \sin^2 (a,b).$$

Якщо фотони летять строго в різні боки (кут між a та b дорівнює 180°), то

$$P_{++}(a,b) = P_{--}(a,b) = ½;$$

$$P_{+-} (a,b) = P_{-+} (a,b) = 0.$$

Звідси слідує, що якщо ми виявляємо один фотон в каналі "+", то інший не може бути в каналі "-" і навпаки. Отже, тут спостерігається кореляція між двома випадковими подіями. Якщо ввести коефіцієнт кореляції, то у розглянутому випадку вона буде повною.

Теорема Белла та її експериментальні перевірки

У 1964 році з'явилася теорема, яка може примирити прибічників класичного та квантового підходів. Запропонував її Джон Стюарт Белл. Вона стверджує, що незалежно від того, містить квантова теорія деякі приховані параметри чи ні, можна провести серійний експеримент, статистичні результати якого підтвердять або спростують наявність таких параметрів. Коротше кажучи, Дж. Белл запропонував спосіб, яким можна було перевести перевірку існування прихованих параметрів з філософської площини в експериментальну. Белл вивів деяку формулу (нерівність) для математичної обробки запропонованого ним експерименту та довів таке:

1. Якщо нерівність виконується, то праві прибічники класичного підходу, тобто, квантова механіка не враховує деякі приховані параметри;

2. Якщо нерівність не виконується, то квантова механіка повна наука, не треба шукати прихованих параметрів, ймовірнісність притаманна фізичній реальності і є її властивістю.

Суть експерименту Белла полягає в такому.

Нехай у нас є джерело переплутаних частинок, які випромінюються ним у протилежні боки. Частинки проходять через поляризатори (якщо це фотони) або через прилади з магнітним полем (прилади Штерна-Герлаха, якщо це частинки з напівцілим спіном). Проходячи через поляризатори (прилади Штерна-Герлаха), частинки набувають певної поляризації (напряму спіна), а потім реєструються детекторами поляризації (спіна). Якщо вектор поляризації (спіна) частинки направлений вгору, таку частинку будемо позначати "+"; якщо вниз – "-". Таким чином, позначивши частинки, що випромінюються в протилежні боки, як "a" та "b", можна записати таке: $N_{++}(a,b)$ – кількість частинок, які мали поляризацію (спін) вгору в обох детекторах, тобто, кількість частинок, поляризація (спін) яких були: тип a – вгору; тип b – вгору. Аналогічно можна позначити $N_{--}(a,b)$ - кількість частинок з поляризацією "-" "-" в обох напрямках; $N_{+-} (a,b)$, $N_{-+} (a,b)$ — аналогічно.

Тоді коефіцієнт кореляції можна записати таким чином:

$$E(a,b) = 1/N \, [N_{++}(a,b) + N_{--}(a,b) — N_{+-}(a,b) — N_{-+}(a,b)],$$

де N – повна кількість частинок, що їх випромінило джерело. Виконавши такі вимірювання з різною орієнтацією (a,b); (a,b'); (a',b) та (a',b'), можна обчислити величину:

$$S(a,a',b,b') = E(a,b) + E(a',b') + E(a',b) — E(a,b')$$

та скласти нерівність Белла:

$$-2 \le S(a,a',b,b') \le 2.$$

З 1972 року нерівність Белла перевірялася неодноразово. Результати подано в таблиці 11.1.

Таблиця 11.1

Автори експериментів	Отримані значення S
Фрідман і Клаузер (1972 рік)	2.28 ± 0.04
А.Аспе зі співроб. (1982 рік)	2.23 ± 0.05
Дж. Вайс зі співр. (1998 рік)	2.23 ± 0.09
Шайдл зі співроб. (2010 рік)	2.37 ± 0.02

Як бачимо з таблиці 11.1, результати усіх експериментів доводять, що нерівність Белла не виконується. Це означає, що правильною виявляється квантова механіка, отже, класичний підхід до парадоксу ЕПР застосовувати не можна. Тим не менше, прибічники теорій прихованих параметрів весь час знаходять в експериментах слабкі місця, що дає їм право сумніватися в наведених результатах. На сьогодні питання все ще не закрито, повної ясності немає.

Багатосвітова інтерпретація

Іншим способом пояснення парадоксу ЕПР була й залишається багатосвітова інтерпретація. Суть її полягає в наступному.

Стан частинок переплутаної пари являє собою квантову суперпозицію усіх можливих станів. Отже, це можна інтерпретувати як суперпозицію станів однакових паралельних всесвітів, що не взаємодіють між собою. Кожен всесвіт містить альтернативну історію частинок пари і характеризується окремими значеннями фізичних величин, тобто, містить один визначений стан. Поки не проведено вимірювань, неможливо встановити, в якому саме всесвіті відбувається експеримент. В момент вимірювання обирається один з паралельних всесвітів, відбувається незворотне їх “розщеплення” й історія частинок стає визначеною.

Ось чому подальші вимірювання не впливають на частинки і протиріччя з принципом причинності відсутнє.

Квантовомеханічна інтерпретація

Цю інтерпретацію було сформульовано засновниками квантової механіки Нільсом Бором та Вернером Гейзенбергом приблизно у 1927 році під час спільної роботи у Копенгагені. Тому часто її називають копенгагенською

інтерпретацією. Вона стверджує, що ймовірнісного характеру передбачень квантової механіки принципово не можна позбавитися, причому він зовсім не свідчить про те, що наші знання обмежені, квантова наука не повна, що вона містить якісь приховані параметри. Це означає, що в квантовій механіці результат вимірювань принципово недетермінований. Н. Бор та В. Гейзенберг відзначають, що фізика взагалі та квантова механіка зокрема є наукою вимірювальних процесів. Тому питання типу "… де була частинка до того, як я зареєстрував її положення …" або "… якими були параметри частинки до вимірювання …" беззмістовні. Процес вимірювання приводить до випадкового вибору в точності одного з можливих варіантів значень параметрів, що вимірюються, які "містилися" у переплутаному стані. Автори цієї інтерпретації відзначають також, що тут немає ніякого протиріччя з теорією відносності, оскільки переплутаний стан квантової системи з кількох частинок описується однією математичною функцією, яка називається хвильовою, причому її не можна розділити на окремі функції, які б описували стан кожної частинки системи окремо.

Оскільки це так, то говорити про передавання якоїсь інформації між частинами хвильової функції всередині її також беззмістовно. Звичайно, така інтерпретація також не позбавлена недоліків. Наприклад, процесу вимірювання тут надається якийсь дуже особливий статус, але пояснення, що ж він таке є, які його визначальні риси, як його відділити від інших, невимірювальних процесів, немає.

Довгі роки копенгагенська інтерпретація була найпопулярнішою серед спеціалістів з квантової механіки. Зараз картина змінюється.

У 1997 році на симпозіумі в Мерилендському університеті було проведено опитування про підтримку тієї чи іншої інтерпретації квантової механіки. Результати подано в таблиці 11.2.

Як бачимо, популярність копенгагенської інтерпретації, яка раніше домінувала, поступово спадає.

Сьогодні багато фізиків схиляються до так званої "ніякої" інтерпретації квантової механіки, а значить, і ЕПР-парадоксу, яку вдало висловив Девід Мермін: "Shut up and calculate!", що в м'якому перекладі з англійської можна подати як: "Мовчи і працюй!"

Таблиця 11.2

Інтерпретація	Віддано голосів
Копенгагенська	13
Багатосвітова	8
Теорія прихованих параметрів	4
Критерій непротирічивості	4
Модифікована динаміка	1
Жодна з наведених	18

Підсумовуючи, можна сказати, що парадокс Ейнштейна-Подольські-Розена належить до такого класу задач, точного вирішення якого сьогодні не існує. Можливо колись, з розвитком квантово-механічної науки, це питання буде вирішено, однак відбудеться це, по всьому видно, в майбутньому.

Квантова телепортація

Детально квантова телепортація розглянута авторами в підрозділі 5.3 і в розділі 8.

Однією з найнеобхідніших операцій в комп'ютерній техніці є копіювання інформації. У класичних комп'ютерах все відбувається звичним для нас чином: зчитуються дані з однієї комірки пам'яті та записуються до іншої. У квантовому випадку читання інформації з кубіта прирівнюється до вимірювання його стану. Таким чином, існує певна підозра, що операцію копіювання стану кубіта в цьому випадку взагалі виконати неможливо.

Дійсно, припустимо, що нам треба скопіювати невідомий квантовий однокубітний стан. Припустимо, що ми маємо деякий унарний оператор, який виконує цю операцію. Тоді ми зможемо записати:

$$U\colon |a0\rangle = |aa\rangle. \tag{11.3}$$

Формула (11.3) справедлива для усіх станів $|a\rangle$. Нехай $|a\rangle$ та $|b\rangle$ - деякі ортогональні стани. Тоді справедливо: $U\colon |a0\rangle = |aa\rangle$ та $U\colon |b0\rangle = |bb\rangle$. Припустимо також, що ми маємо квантовий стан $|c\rangle == \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle)$. Тоді можна записати:

$$U\colon |c0\rangle = \frac{1}{\sqrt{2}}(U|a0\rangle + U|b0\rangle) = \frac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle). \tag{11.4}$$

З другого боку:

$$U\colon |c0\rangle = |cc\rangle, \tag{11.5}$$

оскільки U – клонуюче перетворення. Підставляючи в (11.3) значення $|c\rangle$, отримаємо:

$$|cc\rangle = \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle) \otimes \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle) =$$

$$= \frac{1}{2}(|aa\rangle + |ab\rangle + |ba\rangle + |bb\rangle) \neq \frac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle). \tag{11.6}$$

Звідси явно видно, що унарної операції, яка може копіювати (клонувати) невідомий стан, не існує. Цей факт отримав назву *теореми про заборону клонування*. Очевидно, що клонувати стан за допомогою вимірювання також неможливо, оскільки виміри руйнують квантовий стан. Відзначимо, що відомий квантовий стан легко копіюється або клонується. Проблема стосується лише невідомого стану.

Для подолання цієї проблеми розроблено схему *квантової телепортації*. Цей термін з'явився після публікації в 1993 році статті Чарльза Беннетта та Жіля Брассара в журналі *Physical Review Letters*. Там пояснюється протокол квантової телепортації та її відмінності від того, який зміст вкладається у це поняття звичайними людьми. Квантова телепортація не передає ні енергії, ні речовини на

відстані. Це гарантує, що під час квантової телепортації не порушуються закони фізики. Більше того, квантовий стан кубіта, який телепортується, після закінчення операції руйнується.

Припустимо, що сторона А хоче передати стороні В невідомий квантовий стан $|Q\rangle = a|0\rangle + b|1\rangle$. Сторони вже мають по квантовому біту з переплутаної пари $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Ідея полягає в тому, що сторона А підмішує невідомий кубіт до свого "переплутаного" кубіта, проводить вимірювання і відправляє результати стороні В. Та, у свою чергу, виконує допоміжні перетворення і отримує $|Q\rangle$.

Розглянемо протокол квантової телепортації детальніше.

Стартовий стан. Сторона *A* має один кубіт з "переплутаної" пари і кубіт з невідомим станом $|Q\rangle$. Сторона *B* має лише один "переплутаний" кубіт. Сторона А підмішує невідомий кубіт до пари, ставлячи його ліворуч:

$$|\psi\rangle = \frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|100\rangle + \frac{b}{\sqrt{2}}|111\rangle. \qquad (11.7)$$

Сторона *A* застосовує *CNOT* до перших двох кубітів стану (див. 11.7):

$$|\psi\rangle = \frac{a}{\sqrt{2}}|000\rangle + \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|110\rangle + \frac{b}{\sqrt{2}}|101\rangle. \qquad (11.8)$$

Якщо тепер сторона *A* виміряє середній кубіт, то вона отримає систему в стані $\frac{1}{\sqrt{2}}(a|00\rangle + b|11\rangle)$ або $\frac{1}{\sqrt{2}}(a|01\rangle + b|10\rangle)$.

Результати вимірювань відправляються стороні *B*. В разі, якщо цей результат дорівнює $|1\rangle$, сторона *B* інвертує свій кубіт. Таким чином, стан системи завжди буде однаковим і дорівнюватиме $|\psi\rangle = \frac{1}{\sqrt{2}}(a|00\rangle + b|11\rangle)$.

Тепер сторона *A* застосовує гейт Адамара до першого кубіта (це той самий невідомий кубіт, стан якого треба передати). В результаті отримаємо:

$$|\psi\rangle = \frac{a}{2}|00\rangle + \frac{a}{2}|10\rangle - \frac{b}{2}|11\rangle + \frac{b}{2}|01\rangle. \qquad (11.9)$$

Тепер сторона *A* вимірює перший кубіт. Якщо результат дорівнює $|0\rangle$, то стан системи буде $|\psi\rangle = \frac{1}{2}(a|0\rangle + b|1\rangle)$. Якщо ж він дорівнює $|1\rangle$, тоді маємо $|\psi\rangle = \frac{1}{2}(a|0\rangle - b|1\rangle)$. В останньому випадку сторона *B* змінює знак біля свого кубіта, і тоді ми завжди отримуємо стан системи $|\psi\rangle = \frac{1}{2}(a|0\rangle + b|1\rangle)$. Таким чином, ми телепортували стан квантового кубіта від сторони *A* до сторони *B*. Однак для того, щоб здійснити таку телепортацію, необхідно передати два класичних біти – результати вимірів сторони *A*. Підсумовуючи, приведемо таблицю 11.3, яка продемонструє вищенаведене.

Таблиця 11.3

Результати вимірювань сторони А	Стан кубіта сторони В	Операції сторони В над кубітом	Результуючий стан
0 0	$(a\|0\rangle + b\|1\rangle)$	Тотожна операція, Тотожна операція	$(a\|0\rangle + b\|1\rangle)$
0 1	$(a\|0\rangle - b\|1\rangle)$	Тотожна операція, Інверсія знаку кубіта	$(a\|0\rangle + b\|1\rangle)$
1 0	$(a\|1\rangle + b\|0\rangle)$	Інверсія кубіта, Тотожна операція	$(a\|0\rangle + b\|1\rangle)$
1 1	$(a\|1\rangle - b\|0\rangle)$	Інверсія кубіта, Інверсія знаку кубіта	$(a\|0\rangle + b\|1\rangle)$

Слід відзначити, що описаний протокол не порушує жодних принципів сучасної фізики, як це може здаватися, зокрема, принципів спеціальної теорії відносності. Дійсно, стан кубіта сторони B не несе ніякої інформації про стан кубіта сторони A, тому не можна стверджувати, що під час квантової телепортації інформація передається зі швидкістю, що перевищує швидкість світла. Більше того, для перенесення стану необхідно передати два класичних біти про результати вимірювань. Це здійснюється стандартними класичними каналами зв'язку, які також не порушують жодних фізичних законів.

Протокол квантової телепортації не порушує також теорему про заборону клонування, оскільки результуючий кубіт знаходиться в стані $|Q\rangle = a|0\rangle + b|1\rangle$, а вихідний кубіт - в одному з базисних станів: $|0\rangle$ або $|1\rangle$.

У практичному розумінні телепортація свідчить, що EPR-пара спільно з двома класичними бітами інформації утворюють ресурс для передавання одного кубіта інформації.

Експериментальну реалізацію квантової телепортації поляризаційного стану фотона було здійснено у 1997 році майже одночасно групами фізиків під керівництвом А. Цайлінгера з університету Інсбрука та Ф. Де Мартіні (університет Риму). У 2004 році було оголошено про успішну телепортацію квантового стану атомів кальцію (група М. Рібе) та атома берилію (група М. Барретта). У 2006 році науковою групою з Інституту Нільса Бора (Копенгаген) вперше було здійснено телепортацію між об'єктами різної природи: квантами лазерного випромінювання та атомами цезію.

У січні 2009 року вченим вперше вдалося телепортувати квантовий стан іона на один метр, а у травні 2010 року дослідниками з університету Сіньхуа виконано передавання квантового стану на 16 кілометрів. На сьогодні, однак, рекорд телепортації належить вченим Віденського університету та Академії наук Австрії, які здійснили передавання квантового стану на відстані 143 кілометри.

11.1 Протокол квантового щільного кодування

Протокол квантового щільного кодування використовує один кубіт разом з ЕПР-парою для кодування та передавання такої кількості інформації, яку можна передати лише двома класичними бітами. Оскільки бітами ЕПР-пари можна обмінятися завчасно, виходить, що у випадку маніпулювання квантовими бітами треба фізично передати лише один біт там, де в класичному випадку були б потрібні два (див. також підрозділ 7.8).

Щоб зрозуміти, в чому тут суть, розглянемо протокол квантового щільного кодування детальніше. Припустимо, що сторони інформаційного обміну *A* та *B* мають по одному кубіту з переплутаної пари:

$$|Q\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \qquad (11.10)$$

У такому разі вони можуть виконувати операції лише над своїми кубітами.

Припустимо, що сторона *A* хоче передати стороні *B* якусь інформацію, наприклад, здійснити вибір одного з чотирьох можливих варіантів або закодованих повідомлень. Для цього сторона *A* виконує деякі операції над своїм кубітом і відправляє його стороні *B*. Отримавши кубіт, сторона *B* виконує деякі перетворення над парою і дізнається, яку операцію виконала сторона *A*.

Отже, первісний стан описується співвідношенням (11.10).

Що може зробити сторона *A* зі своїм кубітом?

1. Може нічого не робити. Тоді стан системи не зміниться: $|Q_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

2. Може змінити знак біля $|1\rangle$. Отже, будемо мати: $|Q_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

3. Може переставити місцями $|0\rangle$ та $|1\rangle$. Тоді (11.10) матиме вигляд: $|Q_3\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$.

4. Може переставити місцями $|0\rangle$ та $|1\rangle$і змінити знак біля $|1\rangle$. В цьому випадку (11.10) перетвориться на: $|Q_4\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$.

Оскільки пара кубітів переплутана, то відповідних змін зазнає і кубіт, який знаходиться у сторони *B*.

Після виконання однієї з чотирьох дій сторона *A* відправляє свій кубіт стороні *B,* яка тепер матиме обидва кубіти переплутаної пари. Виконавши відповідні операції над нею та провівши вимірювання, сторона *B* може сказати, яку операцію було виконано над кубітом сторони *A*.

1. Сторона *B* застосовує до пари кубітів *CNOT*:

а) Для випадку першої операції (коли сторона *A* нічого не зробила зі своїм кубітом):

$$CNOT: |Q_1\rangle = CNOT: \left\{\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right\} = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) =$$

$$= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \vee |0\rangle \equiv |\psi_1\rangle.$$

Як бачимо, перший кубіт пари має переплутаний стан, а другий - дорівнює $|0\rangle$.

б) Для випадку другої операції будемо мати:

$$CNOT{:}\,|Q_2\rangle = CNOT{:}\left\{\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)\right\} = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) =$$

$$= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle \equiv |\psi_2\rangle.$$

Як бачимо, другий кубіт знову дорівнює $|0\rangle$.

в) У випадку, коли сторона *A* застосувала третю функцію, отримаємо:

$$CNOT{:}\,|Q_3\rangle = CNOT{:}\left\{\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)\right\} = \frac{1}{\sqrt{2}}(|11\rangle + |01\rangle) =$$

$$= \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)|1\rangle \equiv |\psi_3\rangle.$$

В цьому випадку вже другий кубіт дорівнює $|1\rangle$.

г) Для четвертої функції:

$$CNOT{:}\,|Q_4\rangle = CNOT{:}\left\{\frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)\right\} = \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle) =$$

$$= \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle)|1\rangle \equiv |\psi_4\rangle.$$

І в цьому випадку другий кубіт дорівнює $|1\rangle$.

2. Сторона *B* вимірює значення другого кубіта. Якщо воно дорівнює нулю, то сторона *A* застосувала або першу, або другу функції, якщо ж отримано одиницю - третю або четверту. Далі необхідно з'ясувати, яку саме функцію було застосовано.

3. Застосуємо гейт Адамара до першого кубіта:

$$H{:}\,|\psi_1\rangle = \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle)|0\rangle =$$

$$= \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] = \frac{1}{2}2|0\rangle = |0\rangle. \qquad (11.11)$$

Аналогічно отримаємо вирази для другої, третьої та четвертої функцій:

$$H{:}\,|\psi_2\rangle = \frac{1}{\sqrt{2}}(H|0\rangle - H|1\rangle)|0\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) -\right.$$
$$\left. - \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] = \frac{1}{2}2|1\rangle = |1\rangle. \qquad (11.12)$$

$$H{:}\,|\psi_3\rangle = \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle)|1\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) +\right.$$
$$\left. + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] = \frac{1}{2}2|0\rangle = |0\rangle. \qquad (11.13)$$

$$H{:}\,|\psi_4\rangle = \frac{1}{\sqrt{2}}(H|0\rangle - H|1\rangle)|1\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) -\right.$$
$$\left. - \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] = \frac{1}{2}2|1\rangle = |1\rangle. \qquad (11.14)$$

З формул (11.11 - 11.14) видно, що дія гейтом Адамара на перший кубіт дозволяє остаточно визначити, яку ж саме функцію було застосовано (табл. 11.4).

Таким чином, використовуючи квантову переплутану пару, ми можемо вибрати одну з чотирьох можливих альтернатив за допомогою передавання каналами зв'язку лише одного біта інформації. Такий факт називається *квантовим щільним кодуванням*.

Таблиця 11.4

Значення другого кубіта	Значення першого кубіта	Використана функція
$\|0\rangle$	$\|0\rangle$	$\|Q_1\rangle = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$- перша функція
	$\|1\rangle$	$\|Q_2\rangle = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$- друга функція
$\|1\rangle$	$\|0\rangle$	$\|Q_3\rangle = \frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$- третя функція
	$\|1\rangle$	$\|Q_4\rangle = \frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$- четверта функція

Очевидно, що для вирішення такої задачі класичними методами необхідно було б передати по каналах зв'язку два класичних біти (див. Табл 11.5):

Описаний протокол був запропонований Чарльзом Беннеттом та Стівеном Віснером у 1992 році. Це один з перших прикладів квантового прискорення обчислень та ущільнення передавання інформації.

Таблиця 11.5

Значення бітів	Використана функція
00	Перша функція
01	Друга функція
10	Третя функція
11	Четверта функція

Алгоритм Дойча

Алгоритм Дойча (часто відомий як алгоритм Дойча-Джоза) був запропонований Девідом Дойчем та Річардом Джозом у 1992 році і став одним з перших алгоритмів, призначених для квантових комп'ютерів. Ці алгоритми використовують суто квантові явища, такі, як квантова суперпозиція та квантова переплутаність, та демонструють значний приріст швидкості виконання порівняно з аналогічними класичними алгоритмами.

Суть задачі Дойча-Джоза полягає у визначенні, чи є вибрана функція двійкової змінної постійною (тобто, такою, що дорівнює 0 або 1 незалежно від значення аргументу) або збалансованою (тобто, дорівнює 0 для половини області визначення, а для іншої половини - 1). При цьому апріорі відомо, що функція може бути або константою, або збалансованою.

Для розв'язку цієї задачі класичному алгоритму необхідно виконати $2^{n+1}+1$ обчислень функції $f(n)$. Квантовий алгоритм Дойча-Джоза здатний дати правильну відповідь, один раз виконавши фазовий запит, що відповідає функції $f(n)$. Алгоритм Дойча-Джоза ґрунтується на алгоритмі, розробленому Девідом Дойчем у 1985 році. Він є частковим випадком першого і розглядає функцію однієї змінної. Ми розглянемо цей алгоритм якраз у варіанті функції однієї змінної.

Більш детально алгоритм Дойча розглянутий авторами у підрозділі 5.1.

Нехай у нас є деякий квантовий оракул, який реалізує деяку функцію $f(x)$.

$$U_f = (|x\rangle|0\rangle) \ \ |x\rangle|y\rangle \oplus f(x). \qquad (11.15)$$

Ця функція отримує на вході один кубіт та повертає також один кубіт. Які тут можуть бути варіанти? Таблиця 11.6 дає відповідь на це запитання.

Таблиця 11.6

	x=0	x=1	$f(x)$	Тип
f_0	0	0	I_O	Постійна
f_1	0	1	$CNOT_{i\,O}$	Збалансована
f_2	1	0	$NOT_O CNOT_{i\,O}$	Збалансована
f_3	1	1	NOT_O	Постійна

Як видно з таблиці 11.6, функції f_0 та f_3 не змінюють своїх значень для усього вхідного діапазону, тобто, вони є постійними, а f_1 та f_2 -збалансовані. Функцію f_0 можна вважати як тотожну операцію за виходом I_O, f_1 - $CNOT$ за входом та виходом (iO); f_2 — можна зобразити як $NOT_O CNOT_{i\,O}$, а f_3 - як заперечення за виходом.

Суть алгоритму Дойча зводиться до наступного.

Нехай ми маємо два кубіти в нульовому стані: $|0\rangle|0\rangle$. Подіємо на них гейтом NOT, щоб перевести у стан $|1\rangle|1\rangle$. Можна було б, звичайно, приготувати кубіти в одиничному стані, але простіше зробити це для нульового, а потім отримати одиничний за допомогою заперечення. Подіємо гейтом Адамара на кожен кубіт пари:

$$(H \otimes H)|1\rangle|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$= \tfrac{1}{2}(|00\rangle - |10\rangle - |01\rangle + |11\rangle) = |Q\rangle. \qquad (11.16)$$

Ми сформували вихідний стан і можемо діяти на нього квантовим оракулом.

$$U_f: |Q\rangle = \frac{1}{2}\{U_f|00\rangle - U_f|10\rangle - U_f|01\rangle + U_f|11\rangle\} =$$

$$= \frac{1}{2}\{|0\rangle|f(0)\rangle - |1\rangle|f(1)\rangle - |0\rangle|1 \oplus |f(0)\rangle + |1\rangle \vee 1 \oplus f(1)\rangle\}.$$

Згрупуємо доданки біля однакових кубітів. Будемо мати:

$$|Q\rangle = \tfrac{1}{2}\{|0\rangle|(f(0)\rangle - |1 \oplus f(0)\rangle) - |1\rangle|(f(1)\rangle - |1 \oplus f(1)\rangle)\}. \quad (11.17)$$

Формула (11.17) буде робочою для наступного матеріалу.

Розглянемо випадок постійної функції. В такому разі можна записати: $f(0) = f(1)$. Тоді формула (11.17) буде мати вигляд:

$$|Q\rangle = \tfrac{1}{2}\{|0\rangle|(f(0)\rangle - |1 \oplus f(0)\rangle) - |1\rangle|(f(0)\rangle - |1 \oplus f(0)\rangle)\} = \\ = \tfrac{1}{2}\{(|0\rangle - |1\rangle)|(f(0)\rangle - \vee 1 \oplus f(0)\rangle)\}. \quad (11.18)$$

У випадку збалансованої функції ми можемо записати такі співвідношення:

(а) якщо $f(0)=0$:

$$1 \oplus f(0) = 1 \oplus 0 = 1 = \overline{f(0)} = f(1);$$

(б) якщо $f(0)=1$:

$$1 \oplus f(0) = 1 \oplus 1 = 0 = \overline{f(0)} = f(1);$$

(в) якщо $f(1)=0$:

$$1 \oplus f(1) = 1 \oplus 0 = 1 = \overline{f(1)} = f(0);$$

(г) якщо $f(1)=1$:

$$1 \oplus f(1) = 1 \oplus 1 = 0 = \overline{f(1)} = f(0).$$

З урахуванням вище наведеного формула (11.17) для збалансованої функції буде мати вигляд:

$$|Q\rangle = \frac{1}{2}\{|0\rangle|(f(0)\rangle - |1 \oplus f(0)\rangle) - |1\rangle|(f(1)\rangle - |1 \oplus f(1)\rangle)\} =$$

$$= \frac{1}{2}\{|0\rangle|(f(0)\rangle - |f(1)\rangle) + |1\rangle|(f(0)\rangle - \vee f(1)\rangle)\}.$$

Винесемо спільний множник:

$$|Q\rangle = \tfrac{1}{2}\{(|0\rangle + |1\rangle)|(f(0)\rangle - |f(1)\rangle)\}. \quad (11.19)$$

Тепер подіємо на отримані результати гейтом Адамара.

Для постійної функції будемо мати:

$$H|Q\rangle = \frac{1}{2}H(|0\rangle - |1\rangle)|(f(0)\rangle - |1 \oplus f(0)\rangle) =$$

$$= \frac{1}{2}\frac{1}{\sqrt{2}}\{(|0\rangle + |1\rangle) - (|0\rangle - |1\rangle)\}\{|f(0)\rangle - |1 \oplus f(0)\rangle\} =$$

$$= \frac{1}{2}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle - \vee |0\rangle + |1\rangle)\{|f(0)\rangle - \vee |1 \oplus f(0)\rangle\} =$$

$$= \frac{1}{\sqrt{2}} \vee |1\rangle\{|f(0)\rangle - \vee |1 \oplus f(0)\rangle\}.$$

Для збалансованої функції отримаємо:

$$H|Q\rangle = \frac{1}{2}H(|0\rangle + |1\rangle)|(|f(0)\rangle - |f(1)\rangle) = \frac{1}{\sqrt{2}}|0\rangle\{|f(0)\rangle - |f(1)\rangle\}.$$

Відзначимо, що ми застосували квантовий оракул лише один раз.

Тепер нам потрібно виміряти перший кубіт. Якщо ми отримаємо в результаті вимірювання одиницю, то використовували постійну функцію, якщо ж нуль - збалансовану.

Якщо б ми використовували класичні обчислення, то нам необхідно було б використати оракул двічі: для $x = 0$ та $x = 1$. Однак, ми тоді могли б не тільки сказати, яку функцію використали, а й обчислити значення цієї функції. У квантовому випадку ми не можемо обчислити значення функції, однак можемо сказати тип використаної функції лише за одне її застосування. В цьому і є суть квантового прискорення обчислень.

Алгоритм факторизації цілих чисел (алгоритм Шора)

Алгоритм П. Шора - це квантовий алгоритм, призначений для швидкого розкладання цілих чисел на прості множники. Він дозволяє розкласти число N за час $O(\lg^3 N)$, використовуючи $O(\lg N)$ кубітів. Алгоритм було розроблено Пітером Шором 1994 року. Лише через сім років, 2001 року, змогли продемонструвати його працездатність: група спеціалістів з компанії IBM розклали число 15 на множники 3 та 5, використавши лабораторний зразок квантового комп'ютера з сімома кубітами.

Як бачимо, алгоритм Шора не тільки здатен розкласти числа на множники, але може це зробити, як довів сам Пітер Шор, за поліноміальний час. Вважається, що швидкодія алгоритму дозволить розкладати числа майже так само швидко, як і виконувати їх множення.

Що це означає?

В курсі криптографії вивчають, що стійкість асиметричних криптоалгоритмів ґрунтується на так званих односторонніх задачах, тобто, на таких математичних задачах, коли пряма функція знаходиться значно швидше за обернену. Найвідомішим прикладом такої задачі є факторизація великих цілих чисел, тобто, розкладання їх на прості множники. З математичної практики відомо, що задача розкладання великих цілих чисел на прості множники є проблемою надзвичайної обчислювальної складності. На цій задачі побудований криптоалгоритм RSA. Модуль криптоалгоритму утворюється множенням двох

великих простих чисел. Цю операцію виконати не складно. А ось обчислити приватний ключ, знаючи публічний, можна лише розклавши модуль на множники. При спробі виконати такі обчислення ми наштовхуємося на задачу надзвичайної обчислювальної складності.

Якщо квантовий алгоритм дозволить значно прискорити ці обчислення, асиметрична криптографія може припинити своє існування. Більше того, йдеться не тільки про RSA, а й про інші асиметричні алгоритми, які квантовий комп'ютер здатен зламати аналогічним чином.

Відмінністю алгоритму Шора порівняно з класичними алгоритмами факторизації є його ймовірнісний характер, оскільки задача факторизації зводиться до задачі знаходження періоду функції. Необхідно також спостерігати за квантовою пам'яттю, що теж призводить до випадкових результатів.

Припустимо, що нам треба розкласти ціле число N на прості множники. Це означає, що необхідно знайти таке просте число $a = (1 \div N\text{-}1)$, на яке N ділиться без залишку.

Для цього виберемо якесь ціле число a та за допомогою алгоритму Евкліда обчислимо найбільший спільний дільник двох чисел НСД (a, N). Якщо він більший за одиницю, то задачу розв'язано, число N розкладено на множники. Якщо ж числа a та N взаємно прості, то за теоремою Ейлера відомо, що існує такий степінь a, який ділиться на N із залишком 1.

Нехай r - найменший такий степінь. Тоді

$$a^r \equiv 1 \; mod \; N. \qquad (11.20)$$

Нагадаємо, що r називають порядком a за mod N. Покажемо, що інформація про N може нам надати інформацію про дільник N.

Розглянемо послідовність степенів двійки за модулем 15, тобто, $2^x mod\ 15$. Послідовність буде така:

2, 4, 8, 1, 2, 4, 8, 1, ...

Як бачимо, степені двійки за модулем 15 являють собою періодичну послідовність з періодом 4.

Інший приклад: $2^x mod\ 21$:

2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2, 4, ...

Бачимо, що ми також маємо періодичну послідовність з періодом 6.

Згідно теореми Ейлера, якщо число N є добутком двох простих чисел p та q, тоді, якщо a в послідовності (11.20) не ділиться націло ні на p, ні на q, то послідовність буде мати період, який є дільником числа $(p\text{-}1)(q\text{-}1)$.

Наприклад, якщо N=15, то числа p=3, q=5, $(p\text{-}1)(q\text{-}1)$=2x4=8. І дійсно, період такої послідовності дорівнює 4, що є дільником 8.

Аналогічно, якщо N=21, p=3, q=7, $(p\text{-}1)(q\text{-}1)$ = 2 x 6 = 12. В цьому випадку період послідовності дорівнює 6, що також є дільником 12.

Отже, ми бачимо, що знання періоду дійсно надасть нам корисні відомості про $(p\text{-}1)(q\text{-}1)$.

Тепер з'ясуємо, як ми можемо використати отриману інформацію.

Припустимо, що ми знайшли період r. Нехай він виявився парним. Тоді (11.20) можна записати таким чином:

$$a^r - 1 \equiv 0 \; mod \; N. \tag{11.21}$$

Якщо період парний, то ліву частину (11.21) можна розкласти на множники як різницю квадратів:

$$\left(a^{(r/2)} - 1\right)\left(a^{(r/2)} + 1\right) \equiv 0 \; mod \; N. \tag{11.22}$$

Нехай $A = \left(a^{(r/2)} - 1\right); B = \left(a^{(r/2)} + 1\right)$. Тоді добуток *AB* ділиться на *N*.

Якщо ні *A*, ні *B* окремо не діляться на *N*, тоді кожен з них обов'язково має з ним спільний множник. В такому разі, знайшовши НСД (*A, N*) та НСД (*B, N*) за допомогою того ж алгоритму Евкліда, ми знайдемо нетривіальний дільник числа *N*.

Продемонструємо це на простому прикладі. Розглянемо *N*=15. Оберемо взаємно просте з *N* число *a*=7.

Обчислимо:

$$7^1 mod \; 15 \equiv 7;$$
$$7^2 mod \; 15 \equiv 4;$$
$$7^3 mod \; 15 \equiv 343 \; mod \; 15 \equiv 13;$$
$$7^4 mod \; 15 \equiv 2401 mod \; 15 \equiv 1.$$

Отже, ми маємо порядок 7 за mod 15 = 4. Ми знайшли *r*. Воно парне. Використаємо формулу (11.22):

$$(7^2 - 1)(7^2 + 1) = 48 \times 50 = 2400.$$

Це число має ділитися на 15 без залишку: 2400 : 15 = 160. Оскільки ж ні 48, ні 50 окремо не діляться без залишку на 15, то вони повинні мати з ним спільні множники. Очевидно, що НСД (15, 48) = 3, а НСД (15, 50) = 5.

Таким чином, ми розклали число *N* на прості множники.

Що трапиться, якщо знайдене *r* буде непарним або взаємно простим з *N* ? Тоді вибирається інше число *a* й алгоритм повторюється спочатку.

Тепер нам залишилося з'ясувати, як можна обчислити період послідовності *a*. Для цього використовують квантове перетворення Фур'є.

Квантове перетворення Фур'є діє не на функцію, а на квантові стани так, що

$$\sum_x g\,(x)|x\rangle \rightarrow \sum_c G\,(c)|c\rangle,$$

де *G*(*c*) - перетворення функції *g*(*x*), *x*, *c* - цілі числа від 0 до *N*-1 у двійковому представленні. Якби ми виміряли значення квантового регістра, то ймовірність того, що ми б отримали в результаті *c*, буде $|G(c)|^2$. Застосовуючи квантове перетворення Фур'є до періодичної функції *g*(*x*) з періодом *r*, ми отримаємо $\sum_c G\,(c)|c\rangle$ з відмінним від нуля G(c) лише там, де значення *c* кратні *N*/*r*. Отже, коли стан виміряно, то результатом будуть лише значення *jN*/*r*. Квантове перетворення Фур'є дає лише наближений результат для періодів, які

не є степенями двійки, однак, чим більший степінь двійки використаний як база, тим точнішим буде результат.

Квантове перетворення Фур'є можна увести в такий спосіб:

$$QFT: |a\rangle \to \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} e^{\left(\frac{-2i\pi ab}{N}\right)} |b\rangle;$$

$$QFT^{-1}: |b\rangle \to \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} e^{\left(\frac{2i\pi ab}{N}\right)} |a\rangle.$$

Отже, якщо в квантовому регістрі буде двійкове представлення будь-якої функції з періодом, кратним N, то після перетворення в ньому будуть відмінні від нуля значення лише в точках, кратних N.

Квантове перетворення Фур'є з базою $N=2^m$ визначимо так:

$$QFT: |x\rangle \to \frac{1}{\sqrt{2^m}} \sum_{c=0}^{2^m-1} e^{\left(\frac{2i\pi cx}{2^m}\right)} |c\rangle.$$

Тепер розглянемо, як використовується квантовий алгоритм Фур'є для знаходження періоду функції $a^x \bmod N$ в алгоритмі Шора. Для його реалізації потрібно мати два квантових регістри, X та Y в нульовому стані.

Крок 1. На першому кроці за допомогою оператора Адамара регістр X переводиться до стану суперпозиції. Квантова система тоді буде знаходитися в стані:

$$|Q\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |\, x, 0\rangle.$$

Крок 2. Нехай ми маємо квантовий оракул, який діє звичним нам чином:

$$U_f: |x, 0\rangle \to |x, a^x mod\ N\rangle.$$

Подіявши на стан $|Q\rangle$, будемо мати:

$$U_f: |Q\rangle \to \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |\, x, a^x mod\ N\rangle.$$

Це значить, що в другому регістрі ми отримали значення функції, розраховані для усіх x одночасно. Оскільки функція в другому регістрі періодична, нам треба залучити квантове перетворення Фур'є для знаходження періоду.

Крок 3. Для цього подіємо на результуючий стан системи оракулом, який реалізує квантове перетворення Фур'є:

$$QFT: |Q\rangle \to \frac{1}{N} \sum_{x=0}^{N-1} \sum_{k=0}^{N-1} e^{\left(-2i\pi \frac{kx}{N}\right)} |k, a^x mod N\rangle.$$

Крок 4. Зі стандартного Фур'є-аналізу ми знаємо, що експонента буде відмінною від нуля лише там, де *k* кратне *N*, тобто, в точках *jN*/*k*. Вимірювання першого регістру нам дає стан $\left|k, a^k mod N\right\rangle$ з ймовірністю

$$\left(\left|\frac{1}{N}\sum_{x:a^x mod N\equiv a^k mod N} e^{\left(-2i\pi\frac{kx}{N}\right)}\right|\right)^2.$$

Отже, ми знайшли значення *k*. Решту часу працює класичний комп'ютер.

Крок 5. Нехай при цьому вимірюванні ми отримали в другому регістрі значення *v*. Якщо період дорівнює $r=2^k$, то ми можемо записати

$$v = j\frac{2^m}{r}$$

для деякого *j*. Тоді $\frac{v}{2^m} = \frac{j}{r}$, звідки знаходиться *r*.

Якщо період не дорівнює степені двійки або він непарний, або алгоритм видає саме число *N* як його дільник чи період *r* та множник *j* мають спільний множник, то операція знаходження періоду повторюється *O*(lglg *N*) разів з тим же самим *a*. Якщо й цього разу період не знайдений, змінюють *a* і алгоритм повторюється спочатку.

Пітер Шор довів, що невелика кількість повторень алгоритму знаходить множник числа *N* з великою ймовірністю.

Значення алгоритму Шора полягає в тому, що з його допомогою, використовуючи квантовий комп'ютер з кількома сотнями логічних кубітів, стає можливим злам криптосистем з публічним ключем однієї з найпопулярніших двоключових криптосистем - RSA. Відомо (теорема Рабіна), що криптостійкість такої системи дорівнює складності розкладання великого цілого числа (модуль криптосистеми) на прості множники. За допомогою такої факторизації можна, знаючи публічний ключ криптосистеми, обчислити приватний. Найкращий з класичних алгоритмів факторизації потребує часу порядку $N^{1/3}$. Алгоритм Шора, використовуючи можливості квантових комп'ютерів, здатен виконати факторизацію числа не просто за поліноміальний час, а й не на багато повільніше за множення цілих чисел. Таким чином, реалізація масштабованого квантового комп'ютера зробила б абсолютно безперспективною усю асиметричну криптографію, оскільки він здатен зламати аналогічним чином й інші схожі схеми.

Алгоритм пошуку у невпорядкованому масиві (алгоритм Гровера)

Алгоритм був розроблений американським математиком Ловом Гровером у 1996 році.

Припустимо, що ми маємо базу даних з *N* невпорядкованих записів, один з яких задовольняє певну умову. Ми хочемо знайти цей особливий запис. Будь-якому класичному алгоритму для знаходження такого запису потрібно не менше *O*(*N*) кроків. З елементарних міркувань зрозуміло, що якщо ми перевіримо *k* записів, то знайдемо потрібний з ймовірністю *k*/*N*. Ця ймовірність прямує до нуля зі зростанням *N*, якщо *k* не того ж порядку, що й *N*.

Квантовий алгоритм Гровера розв'язує цю задачу з прискоренням у $\sqrt{N}$ разів, тобто, за $O(\sqrt{N})$ кроків.

Неструктурованість даних дуже важлива, оскільки на сортованому масиві класичний алгоритм дихотомії гарантує швидкість $O(\lg N)$ кроків.

Переформулюємо задачу таким чином. Нам дано "чорну скриньку", яка обчислює потрібну нам функцію від N вхідних даних зі значеннями на виході, що дорівнюють нулю або одиниці. Крім того, відомо, що $f(x)=1$ лише в одному випадку для деякого вхідного значення x_0. Наша задача полягає в тому, щоб знайти x_0.

Нехай N таке, що $2^n \geq N$, U_P - квантовий оракул, який обчислює класичну функцію $P(x)$, що перевіряє істинність твердження (істині відповідає одиниця):

$$U_P: |x\rangle \to |x, P(x)\rangle.$$

Для того, щоб реалізувати в таких умовах алгоритм Гровера, необхідно виконати такі кроки.

Крок 1. Стандартний для квантових обчислень. Необхідно заповнити регістр нулями, тобто, отримати стан $|0\rangle$. Після цього ми діємо на цей квантовий регістр гейтом Адамара:

$$H: |0\rangle \to \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |\, x\rangle.$$

Таким чином ми отримаємо суперпозицію усіх можливих станів x в нашому регістрі.

Крок 2. Діємо квантовим оракулом на отриманий стан квантової системи:

$$U_P: |Q\rangle \to \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |\, x, P(x)\rangle.$$

Тепер задача полягає в тому, щоб отримати з цієї суперпозиції корисний результат. Очевидно, що тут буде присутній й потрібний нам результат: $|x_0, 1\rangle$. Оскільки амплітуда такого стану буде $2^{-n/2}$, то ніякої користі поки що немає. Значить, нам треба буде змінити квантовий стан таким чином, щоб амплітуда при потрібному нам результаті була значно більшою. Якщо нам вдасться це зробити, тоді при вимірюванні з великою ймовірністю ми зможемо отримати $|x_0, 1\rangle$.

Крок 3. Для збільшення амплітуди біля одиничних членів алгоритм пропонує такий спосіб. Перше - це зміна знаку біля одиничних членів, друге - інверсія відносно середнього значення.

Розглянемо простий приклад того, як можна змінити знак лише біля одиничних значень функції.

Нехай у нас є квантовий оракул, який виконує перетворення:

$$U_P: |x, b\rangle \to |x, b \wedge P(x)\rangle.$$

Застосувавши такий оракул до суперпозиції $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle$ з $b = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, отримаємо стан, де знак усіх x, при яких $P(x) = 1$, буде змінено, а b залишиться незмінним.

Дійсно,

$$|\psi, b\rangle \to \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, b \wedge P(x)\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \left(\sum_{x=X_0} |x, 0\rangle + \sum_{x=X_1} |x, 1\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) =$$

$$= \frac{1}{\sqrt{2^{(n+1)}}},$$

$$\left(\sum_{x=X_0} |x, 0 \wedge 0\rangle + \sum_{x=X_1} |x, 1 \wedge 0\rangle - \sum_{x=X_0} |x, 0 \wedge 1\rangle - \sum_{x=X_1} |x, 1 \wedge 1\rangle \right) =$$

$$= \frac{1}{\sqrt{2^{(n+1)}}} *$$

$$* \left(\sum_{x=X_1} |x, 0\rangle - \sum_{x=X_1} |x, 1\rangle \right).$$

Тут X_0 - ті стани, де $P(x) = 0$, а X_1 - ті стани, де $P(x)=1$. Ми бачимо, що знак біля одиниці змінився.

Інверсія відносно середнього. Щоб виконати інверсію відносно середнього значення, необхідно побудувати таке перетворення:

$$\sum_{i=0}^{n-1} a_i |x_i\rangle \to \sum_{i=0}^{n-1} (2A - a_i) |x_i\rangle.$$

Тут A - середнє значення, a_i - коефіцієнти біля x_i, які визначають ймовірність того чи іншого стану системи.

Почергове застосування цих операцій приведе до того, що ймовірність біля одиничних станів зросте порівняно з нульовими, як це показано на рисунку 11.9.

Верхній малюнок демонструє зміну знаку у елементів, де значення функції дорівнює одиниці, а нижній - інверсію відносно середнього значення A.

Лов Гровер довів, що повторення цих кроків $\frac{\pi}{4}\sqrt{2^n}$ разів дозволяє максимізувати ймовірність отримання при вимірюванні одиничного значення.

Цікавим є той факт, що подальше застосування вказаних операцій не тільки не приводить до збільшення ймовірності, але й зменшує її. Цей факт стає зрозумілим, якщо взяти до уваги, що процедури, які виконуються над регістром, є обертанням комплексного простору на певний кут. Отже, спочатку застосування квантового перетворення повертає стан до бажаного, а потім, очевидно, подальші повороти будуть віддаляти вектор від потрібного кута.

Таким чином, щоб отримати задовільні результати від застосування цього перетворення, необхідно чітко знати можливу кількість повторень.

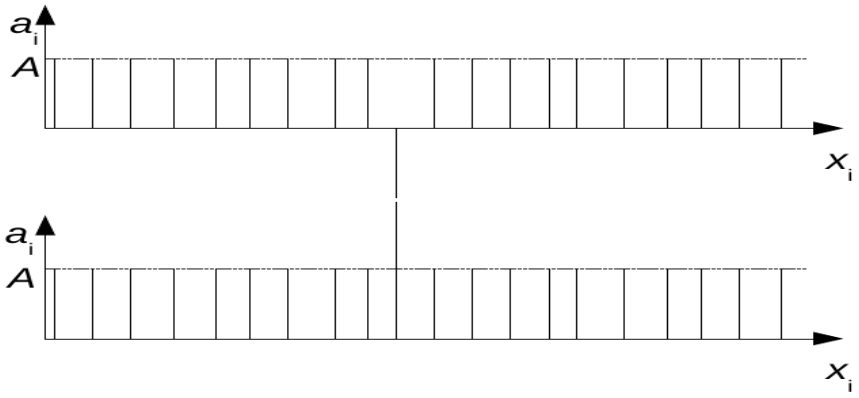


Рис. 11.9. Ймовірності одиничних станів

Для цього існують спеціальні алгоритми.

Алгоритми Шора, Гровера і квантове перетворення Фур'є будуть розглянуті авторами в подальших розділах більш детально.

12 ПРАКТИЧНА РЕАЛІЗАЦІЯ КВАНТОВИХ АЛГОРИТМІВ

12.1 Алгоритм факторизації цілих чисел

Для рішення задачі факторизації використаємо алгоритм знаходження на квантовій обчислювальній системі періоду (порядку) r числа x у мультиплікативній групі лишків за модулем числа N, що факторизується [26-28]. Ця група є циклічною.

Нехай необхідно факторизувати число N. При цьому досить знайти хоча б один множник факторизації. На першому етапі вибирається довільне число X, що є взаємно простим із числом N. Потім розглянемо послідовність, утворену функцією:

$$f(a) = x^a \, mod \, N. \tag{12.1}$$

Послідовності $\{x^a\}$ і $\{x^a \bmod N\}$ виглядають у такий спосіб:

$$\{x^a\} = 1, x, \dots, x^{r-1}, x^r, x^{r+1}, \dots, \tag{12.2}$$

$$\{x^a mod \, N\} = 1, \dots . x^r 1, x, \dots, x^r, 1, x, \dots, x^r. \tag{12.3}$$

Число r – мінімальна степінь, для якої

$$x^r = 1 \pmod N. \tag{12.4}$$

Представимо (12.4) з узагальнення теореми Ейлера, відповідно до якої для будь-яких взаємно простих чисел a і n

$$a^{L(n)} = 1 \, mod \, n, \tag{12.5}$$

де $L(n)$ – узагальнена функція Ейлера.

У залишковому підсумку задавання факторизації числа N полягає в складності пошуку періоду r числа x (12.4). При більших N необхідний перебір досить великої кількості варіантів a. Ефективний алгоритм пошуку періоду r, в основі якого лежить модель квантового паралелізму, наведений нижче.

Нехай період r знайдений. Якщо період непарний, вибирається нове число X і алгоритм пошуку r повторюється. Якщо алгоритм парний, приступаємо до виконання факторизації. Вираз (12.4) може бути записаний в наступному виді:

$$x^r - 1 = 0 \pmod N. \tag{12.6}$$

Звідси маємо:

$$\left(x^{r/2}\right)^2 - 1 = 0 \left(\bmod N\right). \tag{12.7}$$

Далі:

$$\left(\left(x^{r/2}\right) - 1\right)\left(\left(x^{r/2}\right) + 1\right) = 0 \left(\bmod N\right). \tag{12.8}$$

З виразу (12.8) необхідно, щоб добуток двох співмножників був кратний числу N. Таким чином, один із співмножників повинен мати з N загальний

множник. Остаточний етап факторизації полягає в пошуку найбільшого загального дільника співмножників:

$$\gcd\left(\left(\left(x^{r/2}\right)-1\right),N\right). \tag{12.9}$$

$$\gcd\left(\left(\left(x^{r/2}\right)+1\right),N\right). \tag{12.10}$$

Знайдене значення gcd є шуканим числом.

Нижче подано приклад факторизації. Нехай необхідно факторизувати число $N = 91$. Виберемо $x = 3$. Обчислимо послідовності

a: 0, 1, 2, 3, 4, 5, 6, 7,...

3^a: 1, 3, 9, 27, 81, 243, 729, 2187, ...

3^a mod 91: l, 3, 9, 27, 81, 61, 1, 3,...

Таким чином, шукане значення $r = 6$. Далі отримаємо:

$$\left(\left(x^{r/2}\right)-1\right)=3^3-1=26. \tag{12.11}$$

$$\left(\left(x^{r/2}\right)+1\right)=3^3+1=28. \tag{12.12}$$

Знаходимо за допомогою алгоритму Евкліда (алгоритм Евкліда має поліноміальну складність):

$$\gcd\left(\left(\left(x^{r/2}\right)-1\right),N\right)=\gcd(26,91)=13. \tag{12.13}$$

$$\gcd\left(\left(\left(x^{r/2}\right)+1\right),N\right)=\gcd(28,91)=7. \tag{12.14}$$

Отримані значення 7 і 13 є результатом факторизації числа $\boldsymbol{N} = 91$.

Розглянемо алгоритм пошуку періоду r за алгоритмом Шора

Для пошуку періоду r реалізується квантова система, що складається із двох регістрів [26-31]:

$$\phi(a,f(a)) = |0,0,0\ldots;0,0,0,\ldots\rangle. \tag{12.15}$$

Довжина регістрів вибирається таким чином, щоб перший регістр міг умістити максимально допустиме значення a (довжини m) виразу (12.1), а другий регістр міг умістити число N (довжини l).

Алгоритм пошуку періоду r наступний:

1. Для кубіту першого регістра виконується перетворення Уолша – Адамара, що може бути представлене наступною матрицею:

$$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{12.16}$$

тобто, квантові стани перетворяться в такий спосіб: $|0\rangle \to |0\rangle + |1\rangle, |1\rangle \to |0\rangle - |1\rangle$. При цьому перший регістр буде переводитися в суперпозицію станів всіх допустимих аргументів:

$$\varphi(a, f(a)) = \frac{1}{2^m} \sum_{a=0}^{2^m-1} |a,0\rangle \tag{12.17}$$

2. Використовуючи квантовий паралелізм, обчислюється значення функції (12.1) для можливого *a.* Результат (суперпозиція результатів) міститься в другий регістр. При цьому квантова система буде переводитися в стан:

$$\varphi(a, f(a)) = \frac{1}{2^m} \sum_{a=0}^{2^m-1} |a, f(a)\rangle. \tag{12.18}$$

В цьому випадку функція *f(a)* має вигляд, поданий на рис. 12.1.

3. Робиться вимір f_c одного отриманого результату функції *f(a).* При цьому хвильова функція проєктується в суперпозицію станів:

$$\phi(a, f(a)) = \frac{1}{2^m} \sum_{x=0}^{2^m-1} g(a) |a, f(a)\rangle. \tag{12.19}$$

де

$$g(a) = \begin{cases} 1, f(a) = f_c \\ 0, f(a) \neq f_c \end{cases}, \tag{12.20}$$

тобто, квантова система містить тільки стани аргументу *a,* для яких значення функції *f*(*c*) дорівнює обмірюваному f_c ; хвильова функція в цьому випадку є періодичною з періодом *r.*

4. Для отримання періоду необхідно виконати перетворення Фур'є хвильової функції (12.19). При цьому використаємо реалізацію квантового перетворення Фур'є, що має наступний вид:

$$U_{qft} : |c, f_c\rangle \to \frac{1}{\sqrt{2^m}} \sum_a e^{\frac{2\pi i ac}{2^m}} |a, f_c\rangle. \tag{12.21}$$

У випадку перетворення Фур'є дискретної періодичної функції отримуємо спектр, що містить спектральні «сплески», пов'язані з періодом вихідної періодичної функції. Для квантової системи отримаємо наступну хвильову функцію:

$$\varphi(c) = \sum_j^{2^m-1} c_j \left| j \frac{2^m}{r}, f_c \right\rangle. \tag{12.22}$$

Тепер квантова система переводиться в суперпозицію станів. При цьому перший регістр містить значення, обернено пропорційні *r.*

Спектр має вигляд, поданий на рис. 12.2.

5. Здійснюємо вимірювання першого регістра. Отримаємо значення:

$$v = j \frac{2^m}{r}, \tag{12.23}$$

де r – шуканий період, j – невідомий індекс спектрального сплеску. Вираз (12.19) може бути представлений в наступному виді:

$$rv - j2^m = 0. \qquad (12.24)$$

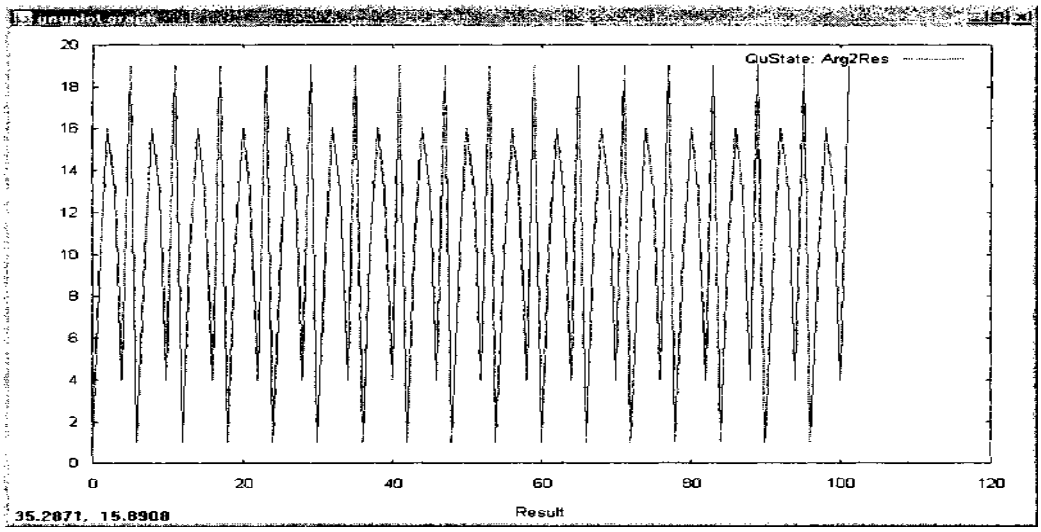


Рис. 12.1. Розподіл значення функції ***f(a)***

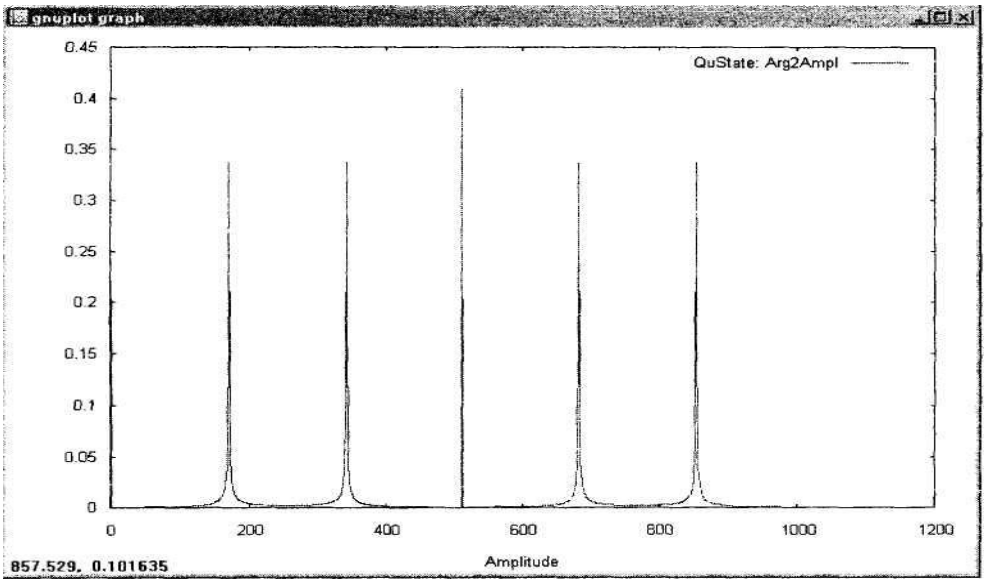


Рис. 12.2. Розподіл значення спектра *F(c)*

У виразі (12.24) v – обмірюване значення першого регістра, r – шуканий період, j – невідомий індекс обмірюваного регістра. У рівнянні (12.24) необхідно знайти r і j. Дане рівняння є діафантовим. Існує ефективний (поліноміальний) алгоритм рішення рівняння виду (12.24) з використанням ланцюгових дробів. Якщо знайдене значення r є парним, воно використається для факторизації числа N (12.8 – 12.10). У противному випадку алгоритм повторюється з іншим обраним значенням X (функція (12.1)).

Даний алгоритм має поліноміальну складність, тому що заснований на ефекті квантового паралелізму й не вимагає перебору варіантів аргументів функції (12.1).

Моделювання алгоритму квантових обчислень Шора

Наведений вище алгоритм був промодельований з використанням традиційних алгоритмічних і обчислювальних механізмів. На рис. 12.1 поданий розподіл значення функції $f(a) = x^a \bmod N$ (dependence of result on argument), а її спектра $F(c) = \sum_{a=0}^{q-1} e^{2\pi i c a} f(a)$ (dependence of amplitude module on argument) – на рис. 12.2.

У зв'язку з тим, що обчислення проводяться в гільбертовому просторі, система моделювання є досить вимогливою до ресурсів (до пам'яті й продуктивності процесора).

При факторизації чисел необхідно враховувати, що необхідна пам'ять для зберігання відображення системи розраховується в такий спосіб:

$$mem(N) = 2^{3len(N)}[3len(N) + 2len(double)], \qquad (12.25)$$

де *len* – функція, що повертає довжину аргумента в бітах.

Наприклад, при факторизації числа $N = 21$ $1en(21) = 5$ система моделювання використає не менш 2528·1024 біт. Для факторизації числа $len(N) = 16$ буде потрібно не менш 30064771072 Mb.

В даному підрозділі проведено аналіз алгоритму факторизації Шора, що дозволяє вирішувати задачі факторизації цілих чисел з поліноміальною складністю.

Висновок

Слід зазначити, що нині існує всього лише кілька алгоритмів, в основному для *NP-повних* завдань: алгоритм факторизації цілих чисел, алгоритм рішення задачі дискретного логарифмування і алгоритм пошуку елемента в несортованій базі. Як і раніше, на даному етапі є відкритою проблема можливості створення універсального рішення, що дозволяє вирішувати широке коло *NP-повних* завдань із поліноміальною складністю. Рішення даної проблеми може розширити коло можливостей обчислювальної техніки в областях, зв'язаних як з моделюванням квантово-механічних фізичних процесів [4], ефективним рішенням завдань штучного інтелекту, так і ефективним рішенням завдань криптоаналізу несиметричної криптографії.

12.2 Методи перехоплення інформації у системах квантової криптографії

У даному підрозділі запропоновано розширену класифікацію методів перехоплення інформації в системах квантової криптографії та розроблено модель порушника у таких системах. Отримані результати дозволяють

підвищити ефективність роботи систем квантової криптографії та формалізувати напрямки подальших досліджень щодо розробки нових ефективних систем захисту інформації з використанням квантових технологій.

На даний момент з усіх існуючих квантових технологій захисту інформації (ЗІ), до яких відносяться квантовий розподіл ключів (КРК), квантовий прямий безпечний зв'язок (КПБЗ), квантове розділення секрету, квантовий потоковий шифр, квантовий цифровий підпис та квантова стеганографія лише системи КРК є реалізованими практично, як окремі модулі та компоненти, що інтегровані в існуючі інформаційно-комунікаційні системи (ІКС) [27,31,32]. Системи КРК мають теоретико-інформаційну стійкість [27,31,32], що забезпечується законами квантової фізики. Якщо розподілені за допомогою квантової системи ключі будуть використані для шифрування за алгоритмом одноразового блокнота (Вернама), а також для аутентифікації користувачів з теоретико-інформаційною стійкістю [29], то можна створити повністю безпечний класичний канал зв'язку між двома користувачами. Фундаментальні закони квантової механіки [31, 32] з одного боку забезпечують виявлення атак на системи квантової криптографії (СКК), а з іншого - допускають можливість реалізації різного роду специфічних атак на такі системи. Зважаючи на те, що у квантовому каналі неможливо відрізнити природні завади від тих, що створюються порушниками при спробі перехоплення (знімання) інформації, необхідно передбачити цей факт при проєктуванні та розробці превентивних систем.

У роботі [28] проведено якісний аналіз атак у кіберпросторі, а робота [30] містить розширену класифікацію кібератак за ознаковим принципом. У роботі [31] наведена класифікація атак на канали КРК: виділено два класи таких атак – це атаки на кубіти та атаки, що використовують неідеальність компонентів системи. У роботі [32] проаналізована атака на протокол ВВ84 у припущенні, що криптоаналітик може управляти ймовірностями вибору базисних векторів для вимірювання станів кубітів, а також одночасною зміною базисів відправника і адресата. Базова класифікація атак на КРК за критерієм складності необхідного для проведення атаки обладнання наведена у роботах [31- 33]. Але загальна класифікація атак на СКК та методів знімання інформації в квантовому каналі зв'язку на даний момент відсутня у науковій літературі. Таким чином, виникають труднощі при побудові моделі порушника у СКК та оцінці можливостей таких систем протидіяти порушникам.

Розглянемо підвищення ефективності розробки систем квантової криптографії за рахунок аналізу можливих методів перехоплення інформації (МПІ) у квантовому каналі та побудови моделі порушника у квантових системах.

Найпростішим способом знімання інформації у звичайних оптичних телекомунікаційних мережах є розділення пучка фотонів. Однак у протоколах квантової криптографії передавання повинно відбуватися за допомогою одиночних фотонів, і в такому випадку порушник не може відвести частину сигналу. Тому подібні МПІ не можуть бути застосовані у СКК в ідеальних умовах однофотонних сигналів (до того ж такі джерела сигналів поки що не створені). На практиці наразі використовують слабкі когерентні імпульси, випромінювані лазерними світлодіодами. Число фотонів в імпульсі визначається

розподілом Пуассона, тобто, частина переданих імпульсів містить два й більше фотонів. Ймовірність зареєструвати в імпульсі більше одного фотона при передаванні їх каналом зі втратами визначається за формулою:

$$P_{n>1} = 1 - e^{-\eta\mu}(1 + \eta\mu), \qquad (12.26)$$

де **μ** - середнє число фотонів в імпульсі, **η** - коефіцієнт передачі каналу.

Таким чином, МПІ з розділенням пучка фотонів на цей час можливі і у квантовій криптографії. На рис. 12.3 наведена загальна класифікація МПІ у СКК. Розглянемо спочатку МПІ, які можуть бути використані зловмисником при умові, що легітимні користувачі реалізують квантовий протокол з ідеальними однофотонними джерелами сигналів. У цьому випадку МПІ за степенем складності можна поділити на *когерентні* та *некогерентні.* При некогерентних МПІ зловмисник обробляє кожний фотон, що передається квантовим каналом, окремо [32,33**].** У свою чергу, некогерентні МПІ бувають *непрозорими* та *напівпрозорими.* Непрозорі МПІ полягають у вимірюванні зловмисником безпосередньо квантового стану фотона і подальшій повторній посилці нового фотона у стані, який отримано в результаті вимірювання. Оскільки зловмисник не пропускає квантові стани відправника, а генерує нові і відправляє їх приймаючій стороні, то даний клас МПІ називається непрозорим.

МПІ при використанні легітимними користувачами ідеальних однофотонних джерел				**МПІ, зумовлені недосконалістю протоколів**		**МПІ, зумовлені недосконалістю обладнання**				
Когерентні		Некогерентні								
Об'єднані	Колективні	Непрозорі	Напівпрозорі	Атака типу "людина посередині"	Атака типу "відмова в обслуговуванні"	МПІ, пов'язані з часовою незбалансованістю детектора	МПІ, **пов'язані із** заміною існуючого квантового каналу на кращий	Розділення пучка фотонів	Розділення числа фотонів	Атаки типу "Троянський кінь"

Рис. 12.3. Узагальнена класифікація МПІ у СКК

Напівпрозорі МПІ [32, 33] передбачають використання зловмисником допоміжних квантових систем (квантових проб - КП) для переплутування їх з носіями, які суб'єкт А (відправник) пересилає суб'єкту Б (отримувачу) квантовим каналом. Після переплутування передані та допоміжні стани знаходяться у загальному переплутаному стані, потім перші передаються суб'єктові Б, а другі зберігаються у квантовій пам'яті у зловмисника. Після закінчення відкритого обміну інформацією між суб'єктами А та Б на етапі просіювання ключа, зокрема,

об'явлення базисів, в яких суб'єкт Б вимірював фотони, зловмисник визначає послідовність базисів, яку необхідно використати для вимірювання станів його проб, щоб отримати якомога більше інформації про ключ. Стани фотонів, що посилає суб'єкт А, змінюються після переплутування з пробами зловмисника, проте рівень помилок при даній атаці значно нижчий, ніж при непрозорій. Варто відмітити, що для реалізації подібного МПІ зловмиснику необхідно мати квантову пам'ять великого об'єму для зберігання проб до об'явлення базисів суб'єктом Б, а також складне обладнання для переплутування своїх проб з фотонами суб'єкта А. Напівпрозорі МПІ є також одним з основних видів знімання інформації в протоколах КПБЗ. У роботі [33] проаналізована атака з використанням КП на пінг-понг протокол КПБЗ з ГХЦ - триплетами [27, 31], а також обчислено повну ймовірність виявлення атаки зловмисника у залежності від кількості отриманої ним інформації для трьох варіантів пінг-понг протоколу. Аналогічний аналіз атаки на пінг-понг протокол з переплутаними парами кутритів виконано в роботі [33]. Доведено, що інформаційна місткість та стійкість різних варіантів даного протоколу є обернено пропорційними величинами.

При когерентних МПІ [31-34] зловмисник може будь-яким (унітарним) способом переплутати пробу будь-якого розміру з групою передаваних фотонів. Одним із підвидів даного класу МПІ є *колективна атака* [32, 34, 35]. Дана атака схожа з напівпрозорою в початковій стадії, тобто, кожен фотон, що посилає суб'єкт А, індивідуально переплутується з окремою пробою. Отже, зловмисник отримує проби в таких же станах, як і при напівпрозорій атаці. Але після закінчення відкритого обміну інформацією між легітимними користувачами зловмисник виконує так зване узагальнене вимірювання відразу на всіх КП, як на єдиній квантовій системі. Найефективнішим МПІ в квантових криптосистемах є об'єднана атака [31, 32] - це окремий випадок когерентних МПІ, коли зловмисник використовує єдину КП (з гільбертового простору [31] станів більшої розмірності) для переплутування з усією послідовністю фотонів, що суб'єкт А передає суб'єктові Б. Але цей МПІ є також і найбільш складним з технічної точки зору. Підводячи підсумки, слід відзначити, що відповідно до сучасного рівня технологій квантової інформатики, реалізація когерентних МПІ не є можливою (на відміну від некорегентних), так як на сьогодні не існує необхідних для цього квантової пам’яті великого об’єму та багатокубітного квантового комп’ютера (див. Мал. 12.3). Недосконаліть протоколів є серйозним чинником для реалізації різноманітних МПІ та інших атак в СКК. Найвідомішим МПІ цього класу є *атака “людина посередині” (man-in-the-middle attack).* Для реалізації цієї атаки зловмисник має повністю контролювати класичний канал зв’язку між легітимними користувачами, тобто, мати можливість замінювати усі повідомлення, що передаються класичним каналом зв’язку. Таким чином, зловмисник має можливість повністю зняти інформацію, якою обмінюються легітимні користувачі у квантовому каналі (наприклад, дізнатися про усі біти ключа), і при цьому не бути виявленим легітимними користувачами. Слід зазначити, що усі існуючі протоколи КРК і КПБЗ є вразливими до даної атаки

[27]. Захист від такої атаки є загальновідомим - автентифікація повідомлень легітимних користувачів у класичному каналі.

Наступна атака -*"відмова в обслуговуванні" {denial of service attack)* не є МПІ, це є метод порушення зв'язку між легітимними користувачами. Уперше для оригінального пінг-понг протоколу КПБЗ така атака була розглянута в роботі [35]. Суть її полягає у тому, що зловмисник не переплутує свою пробу з кубітом на шляху від суб'єкта Б до суб'єкта А, а просто вимірює стан кубіта на зворотному шляху від суб'єкта А до суб'єкта Б (тобто, після кодування інформації суб'єктом А) – тим самим порушуючи взаємну кореляцію кубітів у суб'єктів А та Б. У результаті зловмисник не отримає ніякої корисної інформації, проте зруйнує квантовий канал між легітимними користувачами. У випадку протоколу з ГХЦ-триплетами зловмисник може також вимірювати стани одного чи двох кубітів і порушувати таким чином переплутаність стану триплету [34]. Відзначимо, що до атаки "відмова в обслуговуванні" також вразливі практично всі протоколи квантової криптографії.

Розглянемо методи перехоплення інформації, які зумовлені недосконалістю обладнання. У класичній криптографії МПІ, що зумовлені недосконалістю обладнання, називають також МПІ*, що використовують витік інформації побічними каналами.* МПІ такого типу можливі також і в квантовій криптографії. *Атаки типу "Троянський кінь".* До атак даного типу уразливі так звані двосторонні (two-way) протоколи КРК та КПБЗ, тобто, протоколи, в яких фотони пересилаються від суб'єкта Б до суб'єкта А та назад. Прикладом такого протоколу є вищезгаданий пінг-понг протокол КПБЗ. Вперше атака типу "Троянський кінь" була запропонована в [32]. Зловмисник посилає світлові імпульси у квантовий канал, що з'єднує апаратуру легітимних користувачів, а потім аналізує відбите світло. Таким способом у принципі можливо виявити, який лазер або який датчик тільки що спрацював або параметри настроювання модуляторів поляризації й фази. Така атака не може бути просто відвернена використанням засувки, тому що легітимні користувачі повинні залишити "двері відкритими" для своїх фотонів. Але вони могли б виявити додаткові фотони зловмисника, так як при такій атаці відбувається збільшення енергії імпульсів. Тому зловмисник повинен використовувати світло іншої довжини хвилі, ніж використовують легітимні користувачі, а саме такої довжини хвилі, до якої їх датчики є нечутливими [32]. Інший спосіб для зловмисника приховати атаку полягає в тому, що він перехоплює сигнал, переданий від суб'єкта Б до суб'єкта А, і потім вставляє додатковий фотон у сигнал з часом затримки, коротшим ніж часове вікно датчика [32, 36]. Таким чином, суб'єкт А не може виявити цей додатковий фотон, оскільки він не спричиняє спрацювання його датчика. Після кодувальної операції, яку виконує суб'єкт А, зловмисник перехоплює сигнал знову й відокремлює додатковий фотон. Він може одержати повну інформацію про кодувальну операцію суб'єкта А, виконавши відповідне вимірювання. Такий варіант атаки отримав назву *атаки "Троянського коня із затримкою фотона"* [36].

Для протидії атаці "Троянський кінь" з використанням фотонів інших довжин хвиль, ніж використовують легітимні користувачі, вони повинні

встановити фільтр сигналів з іншими довжинами хвиль на вході свого обладнання [31, 36, 37]. На практиці легітимні користувачі повинні експлуатувати фільтр довжини хвилі для фільтрування фонового світла, особливо, коли у якості квантового каналу використовується вільний простір (бездротовий оптичний канал). Таким чином, для легітимних користувачів немає проблеми запобігти такій атаці [31]. Для атаки "Троянського коня із затримкою фотона" суб'єкт А повинен використовувати світлодільник 50/50 [36], щоб розділити кожний сигнал на дві частини й провести вимірювання станів у двох вимірювальних базисах. Якщо є тільки один фотон в оригінальному сигналі, то спрацює лише один з датчиків, інакше - спрацюють обидва. Таким чином, атаки типу "Троянський кінь" можуть бути відвернені технічними засобами. Але той факт, що цей клас атак існує, ілюструє, що безпека СКК не може гарантуватися тільки принципами квантової механіки, але обов'язково покладається також на технічні засоби [32].

До МПІ, пов'язаних з недосконалістю обладнання, відносяться також *атака розділення числа фотонів (photon number splitting attack – PNS attack)* та *атака розділення пучка фотонів (photon beam splitting attack - PBS attack).*

Як зазначено вище, однофотонні джерела сигналів поки не створені й на практиці використовують слабкі когерентні імпульси, в яких можуть міститися два та більше фотонів. Таким чином, на існуючі на ринку СКК [27, 38-40], які використовують протокол ВВ84, стає можливим атака поділу числа фотонів [32, 41-43]. Для проведення такої атаки для кожного імпульсу, що посилається суб'єктом А, зловмисник повинен виконати квантове неруйнуюче вимірювання числа фотонів в імпульсі, не впливаючи при цьому на їхню поляризацію. Відзначимо, що таке вимірювання дуже складно виконати, але на теперішній час це технічно можливо [41]. Якщо зловмисник виявляє в імпульсі більше одного фотона, то він відводить один, дозволяючи іншим безперешкодно пройти до суб'єкта Б. Потім зловмисник виконує переплутування перехопленого фотона зі своєю пробою і очікує, коли після завершення передавання легітимні сторони оголосять використані базиси. Виконуючи потім вимірювання стану проби, зловмисник отримує точне значення переданого біта, не вносячи при цьому ніяких помилок у просіяний ключ, тобто, його атака залишається невиявленою. Якщо ж імпульс несе один фотон, то стратегії зловмисника можуть бути різними. Наприклад, він може просто пропускати всі однофотонні імпульси (ОФІ), що дозволить йому залишитися невиявленим. Однак, при малому середньому числі фотонів в імпульсі (на практиці обладнання налаштовують так, щоб це число було порядка 0,1) кількість багатофотонних імпульсів (БФІ) буде невеликою, і це не дозволить зловмиснику отримати будь-яку суттєву інформацію про ключ. Інша стратегія полягає у тому, що зловмисник виконує некогерентну атаку на ОФІ. У цьому випадку, зрозуміло, він вносить помилки в просіяний ключ, кількість яких буде залежати як від типу атаки, так і від частки ОФІ при передачі ключа.

Ще одна стратегія для зловмисника полягає у блокуванні частини ОФІ - у результаті суб'єкт Б отримує порожній імпульс, тобто, його датчик не реєструє фотон. Таким блокуванням частки ОФІ зловмисник збільшує частку БФІ, що

дозволяє йому збільшити інформацію про ключ при тому ж рівні внесених у просіяний ключ помилок. Оскільки чутливість сучасних датчиків, які використовуються в комерційних СКК, невелика, і вони реєструють в середньому лише 20-30% одиночних фотонів, а, крім цього, також відбуваються втрати фотонів у каналі, то зловмисник теоретично може таким чином приховати свою атаку. Але суб'єкт Б, знаючи ймовірність отримати порожній імпульс при наявному обладнанні, може виявити значне перевищення кількості порожніх імпульсів над очікуваними. Відзначимо, що суб'єкт Б може також не тільки визначати кількість порожніх імпульсів, але й контролювати всю статистику отримуваних ним сигналів, виконуючи неруйнуюче вимірювання числа фотонів у імпульсі. У цьому випадку зловмисник змушений буде відводити фотон тільки у невеликої частини БФІ, а інші пропускати, не отримуючи ніякої інформації. Для захисту від атаки розділення числа фотонів можна використовувати вдосконалені протоколи КРК - протокол SARG04 та протоколи зі станами приманки (decoy states protocols) [27, 44].

Атака розділення пучка фотонів (PBS attack). Процедури вимірювання числа фотонів у імпульсі та відведення одного фотона (якщо в імпульсі їх два або більше), що використовуються в PNS-атаці, дозволені квантовою механікою, але їх виконання знаходиться поки що на межі можливостей сучасних технологій. Тому в ряді досліджень була запропонована більш проста та достатньо легко здійснювана з сучасними технологіями атака, що отримала назву атаки розділення пучка фотонів [45, 46]. Зловмисник контролює друге вихідне плече світлодільника й одержує повне знання бітів просіяного ключа (через відстрочене вимірювання), якщо БФІ розділений таким чином, що легітимні користувачі обидва одержують принаймні один фотон сигналу. Так, у роботі [45] було запропоновано використовувати для такої атаки метод адаптивної абсорбції, що дозволяє вилучити точно один фотон з моди. Потужність атаки, що отримала назву *умовної атаки розділення числа фотонів,* наближається до потужності PNS-атаки [45]. Один з методів захисту від PBS-атаки, як і від PNS-атаки - це контролювання суб'єктом Б усієї статистики одержуваних сигналів [45, 46], але для цього необхідно виконувати неруйнуюче вимірювання числа фотонів у імпульсі, що є дуже складним з технологічної точки зору. Тому більш практичним на теперішній час є використання у СКК, замість протоколу BB84, удосконалених протоколів – SARG04 або протоколів зі станами приманки.

Атака заміни існуючого квантового каналу на кращий. Удосконалення PNS- та PBS-атак можливо таким способом: зловмисник таємно замінює квантовий канал зі втратами між легітимними користувачами на ідеальний канал без втрат (або на канал зі значно меншими втратами) [41-43, 45, 46]. У такому випадку зловмисник зможе блокувати певну частину ОФІ, видаючи такі втрати за природні, тобто, суб'єкт Б отримає приблизно таку ж кількість *пустих* імпульсів, як до заміни каналу. Неважко помітити, що для початкового каналу з великими втратами зловмисник матиме можливість отримати майже весь ключ і залишитись непоміченим. Крім того, якщо рівень втрат у початковому каналі дуже значний, то зловмисник при заміні його на значно кращий зможе зберегти

не лише очікувану суб'єктом Б частку пустих імпульсів, а й усю статистику числа фотонів у імпульсі [41]. Відзначимо, що атаку заміни існуючого квантового каналу на кращий практично дуже важко здійснити. У будь-якому випадку для захисту від такого типу атаки легітимні користувачі повинні використовувати квантовий канал обмеженої довжини так, щоб його коефіцієнт передачі залишався достатньо високим [43].

Розглянемо деякі методи перехоплення інформації з використанням витоку інформації побічними каналами. У роботі [47] розглянута атака, за якої зловмисник вимірює просторові, спектральні або часові характеристики імпульсів, що передаються бездротовим оптичним каналом. Виконані в цій роботі експерименти з протоколом ВВ84 показують, що найбільшу інформацію про передані біти ключа - 6,6 х 10 біт/імпульс - зловмисник може отримати при вимірюванні спектральних характеристик. Але ця величина є достатньо малою і, таким чином, цю атаку не можна вважати потужною. Інша *атака, пов'язана з часовою незбалансованістю детекора {timing channel attack),* розглянута у роботі [48]. Дана атака, на відміну від попередніх, дозволяє зловмиснику отримати значну частину секретного ключа. Технічні методи захисту від цієї атаки також запропоновані у [48].

Взагалі, теоретичні аспекти безпеки квантової криптографії є на теперішній час дуже активною галуззю досліджень, але значно менше досліджень поки присвячено ретельному дослідженню практичних СКК. Однак, останнім часом спостерігається зростаючий інтерес до аналізу МПІ з використанням побічних каналів, що є результатом фізичної реалізації методів квантової криптографії в практичних СКК.

Розглянемо модель порушника у СКК. Як зазначалося раніше, метою зловмисника є *несанкціонований доступ* (*НСД*) [28] до ресурсів ІКС (у даному випадку СКК) з різною метою. Виключенням є випадок, коли зловмисник ненавмисно реалізує НСД - у такому випадку він є *порушником,* але не зловмисником [49]. Особливу небезпеку можуть нести порушники, які знаходяться під впливом кримінальних “угрупувань, бізнесових структур, політичних організацій, спецслужб тощо”. Припустимим характером дій таких порушників може бути прагнення отримання певних даних для їх подальшого використання, модифікації чи знищення з метою досягнення певних умов для себе чи структур, під впливом яких вони знаходяться. Варто також відзначити, що порушник може бути як внутрішнім (з числа співробітників чи легітимних користувачів), так і зовнішнім (перебуває за межами контрольованої зони ІКС або проникнув до неї несанкціонованим шляхом).

Кваліфікація порушника – це сукупність певних знань і вмінь порушника, які він використовує для реалізації несанкціонованого доступу до ІКС, зокрема, СКК. Можна відзначити кілька типів кваліфікації порушників, що дозволять успішно реалізувати загрози системам квантової криптографії:

- порушник володіє інформацією щодо функціональних особливостей СКК взагалі, уміє користуватися штатними засобами;

- порушник має високий рівень знань і досвід роботи в технічному обслуговуванні аналогічних СКК;

- порушник володіє високим рівнем знань в галузі обчислювальної техніки (зокрема, криптографії, теорії алгоритмів та паралельних обчислень [4] тощо) і програмування на мовах розробки програмного забезпечення (ПЗ) для СКК або їх аналогів;

- порушник досконало володіє знаннями квантової фізики, квантової оптики тощо, а також навиками роботи з обладнанням, що використовується у СКК [31];

- порушник має доступ до глобальних обчислювальних мереж, суперкомп'ютера чи квантового комп'ютера, за допомогою якого може реалізувати, наприклад, силову атаку, використовуючи відомі квантові алгоритми Шора, Гровера тощо [49-53].

Можливості порушника щодо впливу на СКК можна представити у вигляді такої ієрархічної класифікації:

1) порушник має можливість запуску певного обмеженого набору ПЗ, що реалізує певні функції з обробки класичної та квантової інформації;

2) порушник може створювати власне ПЗ та модифікувати існуюче, що дозволить створити нові функції обробки класичної і квантової інформації та подальше одержання частини необхідної порушнику інформації;

3) порушник має змогу управляти функціонуванням СКК, тобто, безпосередньо впливати на ПЗ, склад та конфігурацію технічного забезпечення ІКС;

4) порушник має весь обсяг можливостей легітимних користувачів – може розробляти та впроваджувати в експлуатацію технічні засоби ІКС, а також інтегрувати власні технічні засоби з метою подальшого отримання корисної йому інформації.

Підкреслимо, що теоретичний аналіз стійкості протоколів квантової криптографії, як правило, проводиться, виходячи з того, що порушник має технічні можливості, обмежені тільки законами квантової механіки, а не поточним рівнем розвитку технологій.

Цілі порушника у СКК - це створення нових та підвищення ефективності існуючих методів аналізу стійкості класичних криптографічних засобів та засобів КРК і КПБЗ. Підґрунтям цілеспрямованої реалізації порушником НСД до ресурсів ІКС є найчастіше корисливі мотиви, хоча іноді буває бажання самовираження чи нанесення моральної шкоди легітимним користувачам.

Порушник може використовувати сукупність *релевантних знань, умінь та навиків,* наприклад:

- *досконале знання математичного апарату* дозволить йому створити нові методи криптоаналізу відповідно до рівня криптографічного захисту;
- *знання мов програмування* дозволить порушнику реалізувати створені методи криптоаналізу, а також модифікувати існуюче ПЗ легітимних користувачів, в тому числі ПЗ СКК;
- *знання квантової фізики* дасть змогу порушнику підібрати відповідний МПІ та отримати корисну інформацію;
- *знання методів соціального інжинірингу* може дозволити порушнику без ґрунтовних знань математики, фізики та програмування обійти системи захисту як класичні, так і квантові.

За *характером дій* порушників можна класифікувати таким чином:

1) *випадковий порушник,* що помилково, ненавмисне і несвідомо порушив політику безпеки ІКС;

2) *терплячий порушник* безпеки, що порушив політику безпеки певного сегменту чи усієї ІКС свідомо, навмисно, але без рішучих дій, маскуючись, підбираючи атрибути доступу легітимних користувачів з метою подолання засобів управління доступом тощо;

3) *рішучий зловмисник,* що має на меті порушити одну із властивостей інформаційних ресурсів ІКС. Він прагне подолати усі існуючі засоби обмеження доступу і отримати можливість безпосереднього доступу до ресурсів ІКС з метою втручання у роботу системи, модифікації класичної чи квантової інформації, отримання необхідних даних тощо;

4) *віддалений порушник,* що аналізує технічні канали витоку інформації, впливає віддалено за допомогою спеціальних засобів на локальні та розподілені мережі ІКС, включаючи квантовий та класичний канали СКК.

Для більш детальної побудови моделі порушника, виходячи із конкретної СКК, рекомендується також класифікувати порушників за такими ознаками [50-52]:

- за підготовкою до подолання системи фізичного захисту ІКС;
- за характером поведінки;
- за інформованістю про об'єкт атаки (СКК);
- за використовуваними методами та засобами;
- за місцем реалізації атаки тощо.

Розглянута у даному підрозділі класифікація методів перехоплення інформації у СКК дозволяє чітко визначити напрямки подальших досліджень щодо розробки методів та побудови систем ЗІ, а також створити концептуальні аспекти квантової моделі попередження атак та формалізувати можливості превентивних систем для підвищення ефективності їх вибору і формування вимог при їх проєктуванні та розробці. Розроблена модель порушника в квантових системах ЗІ дозволяє визначити сукупність заходів різного характеру для організації комплексної системи ЗІ в ІКС.

Більш детально моделі порушника розглянуті авторами в розділі 14.6.

12.3 Алгоритм факторизації Шора

Алгоритм Шора дозволяє розкласти число N за час $O(log^2 N\ log^3(\log N))$, використовуючи $O\ (log\ N)$ логічних кубітів.

Значимість алгоритму полягає в тому, що при використанні квантового комп'ютера з декількома сотнями логічних кубітів він зробить можливим злом криптографічних систем з відкритим ключем. Наприклад, RSA використовує відкритий ключ N, що є добутком двох великих простих чисел. Один із способів зламати шифр RSA - знайти множники N. При досить великому N це практично неможливо зробити, використовуючи відомі класичні алгоритми. Найкращий з відомих класичних алгоритмів факторизації вимагає часу порядку $N^{\frac{1}{3}}$. Так як

алгоритм Шора працює тільки на квантовому комп'ютері, в даний час не існує технічних засобів, що дозволяють за поліноміальний час від довжини числа розкласти достатньо велике число на множники.

Взагалі, алгоритм Шора зводиться до знаходження періоду допоміжної функції, а для знаходження періоду застосовується швидке квантове перетворення Фур'є.

Алгоритм Шора і квантове перетворення Фур'є детально розглянуті авторами в 11 розділі.

Рядом Фур'є функції *f(t)*, яка визначена на інтервалі [-*T*,*T*], називається такий вираз:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos\frac{k\pi t}{T} + b_k cos\frac{k\pi t}{T}), \qquad (12.27)$$

де

$$a_k = \frac{1}{T}\int_{-T}^{T} f(t)\ cos\frac{k\pi t}{T}dt,\ b_k = \frac{1}{T}\int_{-T}^{T} f(t)\ sin\frac{k\pi t}{T}dt\,. \qquad (12.28)$$

Алгоритм Шора являється квантовою модифікацією атаки на RSA методом факторизації модуля шифрування. Особливістю даного алгоритму є використання квантового перетворення Фур'є для знаходження періоду числової послідовності. Наведемо основні кроки класичного алгоритму [51,52]:

1) Вибрати випадковий залишок *a* за модулем *N*;

2) Перевірити, що *НСД (a,N)*=1;

3) Знайти порядок *r* залишку *a* за модулем *N*;

4) Якщо *r* парне, обчислити *НСД* ($a^{r/2}$*-1, N).*

Нехай є число *N*. Будемо випадково підбирати число *a* так, щоб воно було взаємно простим з *N*. Після цього знайдемо порядок *r* залишку *a*. Отримаємо: $a^r \equiv 1 \pmod N$. Якщо *r* непарне, то повертаємось до першого кроку та починаємо алгоритм з початку.

Коли маємо $a^r \equiv 1 \pmod N$, *r* – парне, можна розписати $a^{r/2^2} \equiv 1\ (mod\ N)$ або $(a^{r/2^2} - 1)(mod\ N) = 0$ **,** а якщо розписати цей вираз, то отримаємо:

$$\left(a^{\frac{r}{2}} - 1\right)\left(a^{\frac{r}{2}} + 1\right) = cN. \qquad (12.29)$$

В цьому виразі *c* – деяке ціле додатне число. Неважко довести, що одна з цих дужок має з модулем спільний нетривіальний дільник. Тоді, взявши *НСД* ($x^{r/2}$-1, *N*), отримаємо один дільник *N* (може так скластися, що число *N* буде мати спільний нетривіальний дільник з іншою скобкою, тоді ми знову повинні будемо повторити вибір числа *a*).

Розділивши *N* на отримане число, знаходимо другий дільник [53].

12.3.1 Приклад факторизації за алгоритмом Шора

Припустимо, що треба факторизувати модуль шифрування *n = 15*.

Для цього візьмемо просте число *a = 7* і знайдемо лишки його степенем за модулем шифрування (див. табл. 12.1).

Таблиця 12.1

Основні кроки алгоритму

k	1	2	3	4	5	6	7
y^k	7	49	343	2401	16807	117649	823543
(mod 15)	7	4	13	1	7	4	13

З таблиці видно, що період ***t*** = 4, тоді

$$(7^2\text{-}1) = 48 \equiv 3 \ (mod\ 15)$$

або

$$(7^2+1) = 50 \equiv 5 (mod\ 15).$$

В даному випадку ми знайшли відразу два дільника модуля шифрування. Це був розв'язок задачі на класичному рівні, тепер подивимось, як вона вирішується на квантовому рівні. Для розв'язку задачі нам знадобиться два кубіти в початковому стані $|\psi_0\rangle = |0\rangle|0\rangle$.

Створимо з першого кубіта $|\psi_0\rangle$ рівномірну суперпозицію

$$|\psi_0\rangle \rightarrow |\psi_1\rangle = \frac{1}{\sqrt{16}}(|0\rangle|0\rangle + |1\rangle|0\rangle + |2\rangle|0\rangle + |3\rangle|0\rangle +$$
$$+|4\rangle|0\rangle + |5\rangle|0\rangle + |6\rangle|0\rangle + |7\rangle|0\rangle + |8\rangle|0\rangle + |9\rangle|0\rangle + |10\rangle|0\rangle +$$
$$+|11\rangle|0\rangle + |12\rangle|0\rangle + |13\rangle|0\rangle + |14\rangle|0\rangle + |15\rangle|0\rangle).$$

У другий кубіт $|\psi_0\rangle$ запишемо значення

$$\sum |x\rangle|0\rangle \rightarrow \sum |x\rangle|y^x (mod\ 15)\rangle$$

або

$$|\psi_0\rangle \rightarrow |\psi_1\rangle = \frac{1}{\sqrt{16}}(|0\rangle|1\rangle + |1\rangle|7\rangle + |2\rangle|4\rangle + |3\rangle|13\rangle +$$
$$+|4\rangle|1\rangle + |5\rangle|7\rangle + |6\rangle|4\rangle + |7\rangle|13\rangle + |8\rangle|1\rangle + |9\rangle|7\rangle + |10\rangle|4\rangle +$$
$$+|11\rangle|13 + |12\rangle|1\rangle + |13\rangle|7 + |14\rangle|4\rangle + |15\rangle|13\rangle).$$

Проведемо вимірювання другого кубіта, взявши вимірювальний прилад, що знаходиться в стані $П = |7\rangle\langle 7|$, після цього отримаємо:

$$|\psi_3\rangle = П|\psi_2\rangle = \frac{1}{\sqrt{16}}(|1\rangle + |5\rangle + |9\rangle + |13\rangle),$$

$$|\psi_3\rangle = П|\psi_2\rangle = \frac{1}{2}(|0001\rangle + |0101\rangle + |1001\rangle + |1101\rangle).$$

Візуально це можна спостерігати на рис.12.4.

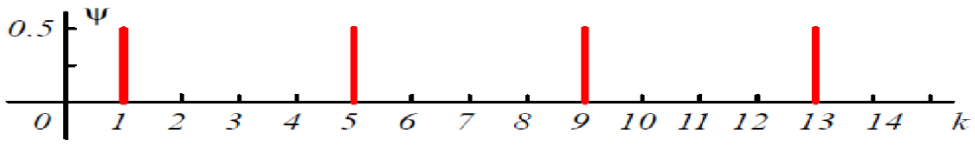


Рис. 12.4. График стану $|\psi_3\rangle$

Квантове перетворення Фур'є даного кубіта нам дає:

$$\left|\tilde{\psi}\right\rangle = U\left|\psi_3\right\rangle = \frac{1}{2}\frac{1}{\sqrt{16}}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{-i\frac{2\pi\ k\ j}{2^n}}\left|k\right\rangle\left\langle j\right|(\left|1\right\rangle + \left|5\right\rangle + \left|9\right\rangle + \left|13\right\rangle) =$$

$$= \frac{1}{8}\sum_{j=0}^{15}\sum_{k=0}^{15} e^{-i\frac{2\pi\ k\ j}{16}}\left|k\right\rangle(\delta_{j1} + \delta_{j5} + \delta_{j9} + \delta_{j13}) =$$

$$= \frac{1}{8}\sum_{k=0}^{15}(e^{-i\frac{2\pi k}{16}} + e^{-i\frac{2\pi k\ 5}{16}} + e^{-i\frac{2\pi k\ 9}{16}} + e^{-i\frac{2\pi k\ 13}{16}})\left|k\right\rangle =$$

$$= \frac{1}{2}(\left|0\right\rangle + i\left|4\right\rangle + \left|8\right\rangle + i\left|12\right\rangle).$$

Другий ненульовий стан $|k\rangle = |4\rangle$ визначає період послідовності згідно формули $t = \frac{2^n}{k} = \frac{16}{4} = 4$ [54].

12.3.2 Опис програми факторизації

Програма виконана у середовищі розробки Microsoft Visual Studio 2010 на мові **C#**. У програмі не використовувались жодні додаткові бібліотеки та модулі.

Основні можливості програми:

1) Здатна факторизувати модуль RSA-перетворення за алгоритмом Шора;
2) Може самостійно згенерувати модуль для RSA-перетворення;
3) Є можливість виведення простих чисел розміром до 12 бітів;
4) Присутня довідка та допомога користувача;
5) Присутня полоса прогресу обчислень.

Головне вікно програми можна побачити на рис. 12.5.

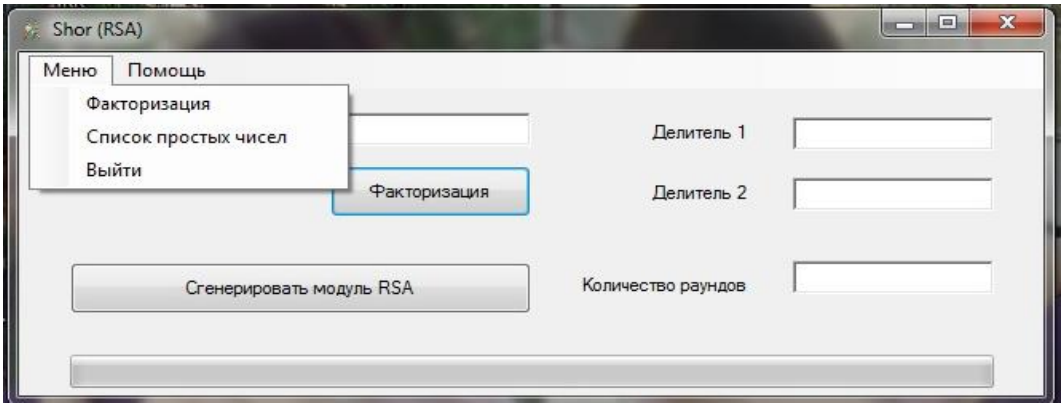


Рис. 12.5. Головне вікно програми

На рис. 12.6 можна побачити результат роботи програми. Як бачимо, програма повідомила про своє завершення за допомогою полоси прогресу.

Програма сама згенерувала модуль за допомогою кнопки «Згенерувати модуль RSA».

У випадку, коли програма не зможе факторизувати число, можна буде спостерігати повідомлення, яке зображене на рис. 12.7. Це повідомлення є мірою застереження, щоб не зациклилася програма. Кількість раундів вибрана по середнім оцінкам часу виконання програми та чисел, що подаються на факторизацію.

Генерацію простих чисел можна побачити на рис. 12.8.

Основні недоліки програми:

1) Працює з числами до 32 бітів (int);

2) Програма емулює класичний вигляд алгоритму Шора, а не квантовий, тобто, квантове перетворення Фур'є не застосовується;

3) Полоса прогресу є дуже суб'єктивна;

4) Програма може застосовуватися лише на ОС Windows з підтримкою .NET

5) Framework 3.0 та вище.

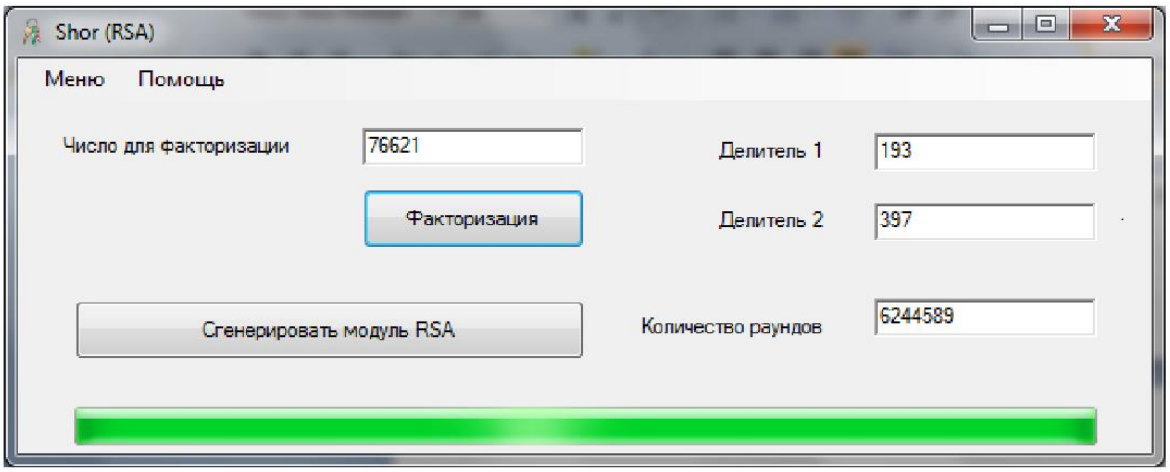


Рис.12.6. Результат роботи програми

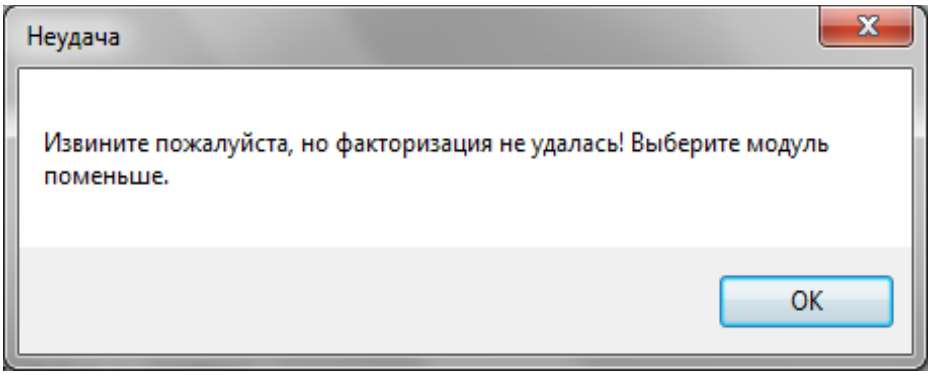


Рис. 12.7. Повідомлення про невдале завершення програми

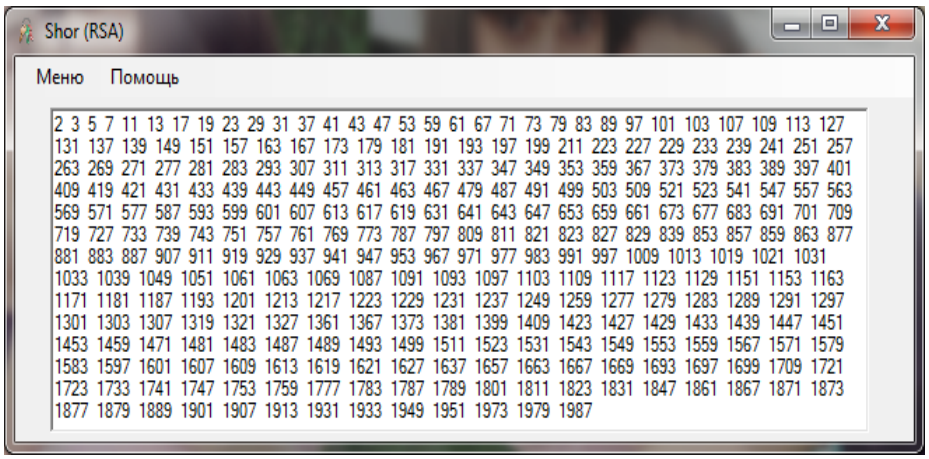


Рис. 12.8. Генерація простих чисел

12.4 Квантовий алгоритм Гровера

Нині в криптографічному загалі широко обговорюється та досліджується проблема створення та стандартизації перспективних криптографічних перетворень, в першу чергу для постквантового періоду [54,55]. Суттєві результати досягнені в частині розроблення, стандартизації та застосування симетричних криптоперетворень [56]. Разом з тим, продовжується розвиток та здійснюються спроби розробити більш ефективні методи криптоаналізу симетричних криптосистем – симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП) та функцій гешування (ФГ). Підтвердженням цьому є прийняття та застосування міжнародних стандартів ДСТУ ISO/IEC 18033 -3, 18034 – 4, ДСТУ 7624:2014, ДСТУ 7564:2014, FIPS – 197, FIPS – 202 тощо. При цьому великий інтерес до задач криптоаналізу проявляють як безпосередньо криптоаналітики, так і розробники стандартизованих симетричних та асиметричних криптосистем.

Зрозуміло, що криптоаналітики направляють свої зусилля на безпосередній злам відповідних криптосистем, а розробники та ті, що застосовують криптографічні перетворення, на перевірку їх криптографічних властивостей. Якщо раніше для спроб вирішення задач криптоаналізу застосовувались класичні спеціалізовані комп’ютерні системи, засоби та класична математика, то нині розробляються квантові комп’ютери та практично розроблені відповідні квантові математичні методи [54-56]. Але освоєння та застосування квантових систем та квантових математичних методів, а також програмування носять як методологічний аспект, так і психологічний – складність сприйняття.

Дослідження, що проведені, дозволяють прогнозувати використання для симетричного криптоаналізу квантового методу, що отримав назву методу Гровера [56,57]. Метою даного підрозділу є деталізація, освоєння для застосування, перевірка криптоаналітичних властивостей та демонстрація застосування методу Гровера при криптоаналізі СБП, в тому числі, з методичними цілями освоєння методу при навчанні з використанням прикладів,

але поки що на класичному комп'ютері. Нині важливу роль в сфері безпеки інформації відіграють функції гешування. Тому важливості набуває загроза з боку квантових комп'ютерів та методів квантового криптоаналізу, таких як метод пошуку невпорядкованою базою даних Гровера.

У сучасних системах криптографічного захисту інформації необхідно виділити наступні застосування функцій гешування (ФГ) [60]:

- протоколи електронного підпису, електронного штампу та мітки часу (ЕП);
- алгоритми генерації псевдовипадкових послідовностей;
- криптографічні протоколи узгодження та встановлення ключів;
- механізми контролю цілісності та справжності інформації та ресурсів;
- контроль цілісності баз даних, в тому числі, без використання спільного секрету;
- протоколи автентифікації з використанням паролів та біометричних ознак тощо.

Слід зауважити, що найбільш інтенсивним, та в ряді випадків безумовно необхідним, є застосування ФГ для надання електронних довірчих послуг, що пов'язані з електронним підписом, електронною міткою часу, електронним штампом та різними криптографічними протоколами [60].

Також все більшого значення набуває застосування функцій гешування в блокчейні при забезпеченні так званого «Proof of Work» («доведення виконання роботи»).

Однією з основних загроз для функцій гешування є знаходження колізії. Існування двох повідомлень з однаковим геш-значенням є незаперечним фактом, оскільки множина повідомлень довільної довжини нескінченна, а множина геш-значень скінченна. Відображення нескінченної множини в скінченну завжди приводитиме до появи пар різних повідомлень з однаковим геш-значенням. Таке явище називається колізією [60]. Складність знаходження колізії є однією з основних характеристик функції гешування. Це пов'язано з тим, що при знаходженні колізії створюється пара повідомлень, для яких вірним буде одне й те саме геш-значення, а, отже, один і той же самий електронний підпис. Також за допомогою знаходження колізії зловмисник може здійснити підбір повідомлення M', для якого буде дійсним вже існуючий електронний підпис. Таким чином, підпис для повідомлення M буде тим самим, що й для повідомлення M' при тому, що $M \neq M'$. Це дає зловмиснику можливість підміни повідомлення. При знаходженні для деякого повідомлення M деякого повідомлення $\mathrm{M}' \neq \mathrm{M}$ за умови $\mathrm{H}(\mathrm{M}) = \mathrm{H}(\mathrm{M}')$ повідомлення M' називається другим прообразом геш-значення $H(M)$. При атаці знаходження другого прообразу спільне для обох повідомлень геш-значення задане до атаки, в той час як при атаці здійснення атаки знаходження колізії стає відомим в результаті атаки. Таким чином, можна сказати, що атака знаходження другого прообразу спирається на результат атаки знаходження колізії. Знаходження колізії можна звести до пошуку несортованою базою даних, що є значною проблемою для класичних алгоритмів. Проте для квантових алгоритмів, а саме, алгоритму Гровера, ця задача має меншу складність.

12.5 Особливості методу Гровера при використанні в несортованій базі даних

В загальному вигляді сутність методу Гровера полягає в проведенні вичерпного пошуку специфічного (унікального) елемента у несортованій базі даних, що складається з $N = 2^n$ елементів, де n довжина квантового регістру (кількість кубітів) [61]. Для криптоаналізу СБП специфічність елементу може зводитись до сеансового чи довгострокового ключа, синхропослідовності тощо. Особливістю, наприклад, ключа є те, що при його застосуванні зашифровані дані можуть бути розшифрованими за поліноміальний час.

При знаходженні колізії для функції гешування таким елементом є повідомлення з ідентичним до заданого геш-значенням.

У порівнянні з найкращими класичними алгоритмами метод Гровера передбачає проведення пошуку з квадратичним прискоренням, замість *O(N)* всього за $O\left(\sqrt{N}\right)$ групових операцій. Для отримання такого прискорення використовується квантова суперпозиція станів. Метод Гровера носить узагальнений зміст, так як метод може бути реалізований у вигляді алгоритму криптоаналізу симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП), ФГ, асиметричного шифру в кільці поліномів тощо [42,43].

Наші обґрунтування методу Гровера ґрунтуються на його потенційних можливостях. Так, в таблиці 12.2 наведені результати розрахунку стійкості СБП проти квантового криптоаналізу [56,57]. Наведені в таблиці 12.2 дані дозволяють зробити висновок, що при розробленні та введенні в експлуатацію хоча б одного квантового комп'ютера значне число симетричних криптоперетворень буде зламаним, а деякі будуть під суттєвою підозрою.

Таблиця 12.2

Стійкість симетричних криптосистем проти квантового криптоаналізу на основі методу Гровера

№ п/п	Шифр	Параметри		Стійкість при атаці на	
		Розмір блока, біт	Розмір ключа, біт	Блок повідомлення	ключ
1	AES-128	128	128	2^{64} $(10^{19,2})$	2^{64} $(10^{19,2})$
2	AES-256	128	256	2^{64} $(10^{19,2})$	2^{128} $(10^{38,4})$
3	DES	64	56	2^{32} $(10^{9,6})$	2^{28} $(10^{8,4})$
4	TDES	64	168	2^{32} $(10^{9,6})$	2^{134} $(10^{40,2})$
5	ГОСТ-28147	64	256	2^{32} $(10^{9,6})$	2^{128} $(10^{38,4})$
6	Калина-128	128	128	2^{64} $(10^{19,2})$	2^{64} $(10^{19,2})$
7	Blowfish	64	448	2^{32} $(10^{9,6})$	2^{224} $(10^{67,2})$

Спочатку розглянемо сутність та зробимо відповідний аналіз методу Гровера та етапів його виконання.

1. На першому кроці квантовий регістр з n кубітів, необхідних для представлення пошукового простору розміру $N = 2^n$, встановлюється у стан $|0\rangle$ у вигляді:

$$|0\rangle^{\otimes n} = |0\rangle. \tag{12.30}$$

Після цього квантова система встановлюється в стан рівної суперпозиції станів. Для цього над квантовим регістром з $\boldsymbol{n}$ кубітами виконується перетворення Адамара $H^{\otimes n}$, що складається з використання n звичайних гейтів (вентилів) Адамара [58]:

$$|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \tag{12.31}$$

2. На другому кроці $\frac{\pi}{4}\sqrt{2^n}$ разів застосовуються ітерації Гровера. Така кількість застосування ітерації Гровера зумовлюється необхідністю отримання реальної оптимальної ймовірності того, що отриманий стан системи буде саме тим, який шукається, а також необхідністю того, що необхідний стан матиме загальне зміщення фази не більше $\frac{\pi}{4}$ радіан [58]. При цьому кожна ітерація Гровера проводить амплітудне підсилення ймовірності знаходження елемента бази, що шукається. Кожна ітерація Гровера складається із застосування квантового оракула *O* та застосування оператора дифузії.

Спочатку йде виклик квантового оракула *O*, що модифікує систему в залежності від того, чи знаходиться система в потрібному стані. Оракул характеризується тим, що може аналізувати та модифікувати систему без зведення її до класичного стану, тобто, продовжує знаходитися в квантовому стані. Конкретна реалізація квантового оракула залежить від кожного окремого випадку та задачі. Наприклад, у випадку криптоаналізу функцій гешування (пошуку колізії) оракул виконує пошук геш-значення та порівняння його з потрібним та повертає результат - успіх чи ні. Спільним для кожного оракула є те, що він розпізнає, чи знаходиться система в правильному стані та, якщо система знаходиться в правильному стані, оракул повертає фазу на π радіан, в іншому ж випадку він не робитиме нічого, позначивши правильний стан для подальших модифікацій наступними операціями. При цьому через зсув фаз можлива ситуація, коли правильний стан залишається тим самим, хоча його амплітуда матиме протилежний знак.

Ефект реакції оракула на систему (x) можна відобразити як:

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle\,, \tag{12.32}$$

де $f(x) = 1$, якщо система знаходиться в правильному стані (успішному) та $f(x) = 0$ в іншому випадку. Точна реалізація оракула залежить від $f(x)$, а $f(x)$ залежить від задачі пошуку. Як уже відмічалось, у випадку знаходження колізії функцій гешування $f(x) = 1$, якщо отримане геш-значення ідентичне потрібному.

Наступна частина ітерації Гровера називається оператором дифузії. Оператор дифузії проводить інверсію щодо середнього. Амплітуда кожного

стану перетворюється так, щоб вона була набагато вищою за середнє настільки, наскільки вона була нижчою за середнє значення, та навпаки.

Оператор дифузії складається з трьох послідовних операцій: застосування перетворення Адамара $H^{\otimes n}$, умовного фазового зсуву, який зсуває кожен стан, окрім $|0\rangle$, на -1, та ще одного застосування перетворення Адамара $H^{\otimes n}$. Умовний фазовий зсув відображається унітарним оператором $2\,|0\rangle\langle 0| - I$ [74]:

$$[2|0\rangle\langle 0| - I]|0\rangle = 2|0\rangle\langle 0|0\rangle - I = |0\rangle,$$
$$[2|0\rangle\langle 0| - I]|x\rangle = 2|0\rangle\langle 0|x\rangle - I = -|x\rangle. \qquad (12.33)$$

Згідно з виразом (12.33) стосовно $|\psi\rangle$ та з врахуванням (12.32) отримуємо оператор дифузії як:

$$H^{\otimes n}[2|0\rangle\langle 0| - I]H^{\otimes n} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - I = 2|\psi\rangle\langle\psi| - I. \qquad (12.34)$$

З урахуванням оракула O (12.32) та оператора дифузії (12.34) повну ітерацію Гровера можна подати у вигляді

$$[2|\psi\rangle\langle\psi| - I]\mathcal{O}\,. \qquad (12.35)$$

Аналіз показує, що основною вимогою до ітерації Гровера є вимога мінімізації її складності (часу виконання). Слід відмітити, що складність і відповідно точний час виконання оракула залежить від конкретних задач і реалізацій. Тому виклик оракула зазвичай розглядається як одна елементарна операція.

Загальна складність (час виконання) однієї ітерації Гровера становить $O(2n)$, що пояснюється необхідністю виконання двох перетворень Адамара, а також складності застосування $O(n)$ вентилів при виконанні умовного фазового зсуву зі складністю $O(n)$. У цілому складність виконання всього алгоритму Гровера можна оцінити як $O(\sqrt{N}) = O(\sqrt{2^n}) = O(2^{n/2})$ ітерацій, кожна зі складністю $O(n)$.

Схематичне зображення алгоритму Гровера з додатковим кубітом для оракула згідно [58] наведено на рис. 12.9.

Для визначення результату виконання ітерації Гровера виконуються класичні вимірювання, причому результат буде правильним з достатньо високою ймовірністю. На цьому виконання алгоритму Гровера завершується, його складність становить $O(1)$ [57,58].

Перетворення дифузії повторити $O(\sqrt{N}) \approx \frac{\pi}{4}\sqrt{N}$ разів.

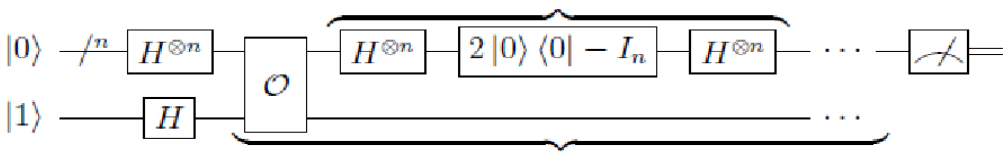


Рис. 12.9. Алгоритм Гровера з додатковим кубітом для оракула

Необхідно відмітити, що алгоритм Гровера є квантовим алгоритмом і вимагає застосування квантового комп'ютера та квантової математики. По суті,

квантова математика методу Гровера наведена вище, формули (12.31)-(12.35) та завершена рис. 12.2. Квантовий комп'ютер ще недоступний. Проте його можна реалізувати на класичному комп'ютері. Але слід зауважити, що реалізація його на класичному комп'ютері, що не підтримує квантові властивості, не є прийнятною передусім з точки зору складності (часу) виконання. Так, на квантовому комп'ютері, завдяки квантовим властивостям, наприклад, є можливість обстежувати весь регістр одразу без розгляду кожного окремого елемента. Оракул розглядається як одна елементарна операція, а її виконання займає набагато менше потужності (часу), ніж це потребується на класичному комп'ютері. Тобто, виконання алгоритму Гровера на класичному комп'ютері є суттєво складнішим. Так, по суті, лише реалізація оракула замість $\sqrt{N}$ звертань потребуватиме $N\sqrt{N}$ звертань. Відповідно, час застосування алгоритму зростає настільки, що використання методу Гровера буде повільнішим, навіть у порівнянні зі складністю пошуку ключа СБП методом «грубої сили» тощо.

З метою демонстрації практичної реалізації методу Гровера розглянемо його на прикладах алгоритму пошуку унікального елемента з певної бази даних.

Приклад 12.1. Припустимо, що система складається з $N = 65536 = 2^{16}$ станів, і стан, який ми шукаємо, x_0, має індекс 301.

Розглянемо алгоритм Гровера пошуку даного «унікального» елемента по крокам.

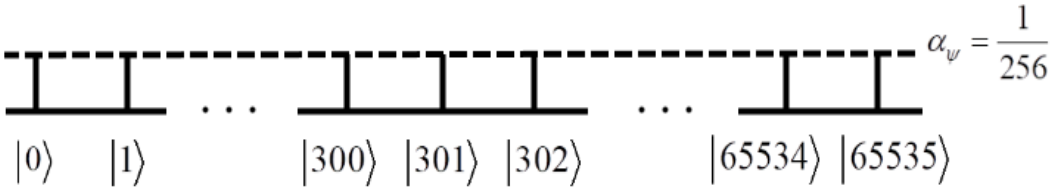
1. Для того, щоб описати цю систему, потрібно $n = 16$ кубітів. У відповідності з алгоритмом Гровера зробимо ініціалізацію квантового регістру з $n = 16$ кубітів, що необхідно для представлення пошукового простору розміру $N = 2^{16}$, встановивши регістр у початковий стан:

$$|\psi_0\rangle = |000\ldots000\rangle .$$

2. Проведемо перетворення Адамара, що дозволяє отримати значення амплітуди, яка пов'язана з кожним станом з рівною ймовірністю перебування в кожному з 16 можливих станів:

$$|\psi\rangle = H^{\otimes 16}|000\ldots000\rangle = (H|0\rangle)^{\otimes 16} = \frac{1}{256}\sum_{i=0}^{65535}|i\rangle .$$

Геометрично це можна зобразити як:



3. За вказаних даних оптимальним для отримання рішення є виконання ітерацій Гровера, кількість яких визначається таким чином:

$$\frac{\pi}{4}\sqrt{2^n} = \frac{\pi}{4}\sqrt{65536} \approx 201,06192983 .$$

В подальшому для використання округлимо число ітерацій до 201. З проведених досліджень можна зробити висновок, що округляти кількість ітерацій Гровера потрібно до меншого значення для досягнення найкращого результату. В кожній ітерації першим кроком є виклик квантового оракула *O*, потім проводиться інверсія середнього або ж оператор дифузії.

4. При пошуку елемента з індексом 7 оракул дає такі значення [56,57]:

$$U_f(|301\rangle|-\rangle) = -|301\rangle|-\rangle; \quad U_f(|i\rangle|-\rangle) = |i\rangle|-\rangle, \text{ if } i \neq 7;$$

5. Далі визначимо $|u\rangle$, використовуючи (12.34), в результаті маємо:

$$|u\rangle = \frac{1}{\sqrt{65535}} \sum_{\substack{i=0 \\ i\neq 301}}^{65535} |i\rangle = \frac{|0\rangle + |1\rangle + ... + |65534\rangle + |65535\rangle}{\sqrt{65535}}.$$

Також аналогічно (12.34) знаходимо, що $|\psi\rangle = \frac{\sqrt{65535}}{256}|u\rangle + \frac{1}{256}|301\rangle$.

6. Наступним кроком знайдемо

$$|\psi_1\rangle|-\rangle = U_f(|\psi\rangle|-\rangle) = \left(\frac{|0\rangle + |1\rangle + ... + |300\rangle - |301\rangle + |302\rangle + ... + |65534\rangle + |65535\rangle}{65536}\right)|-\rangle.$$

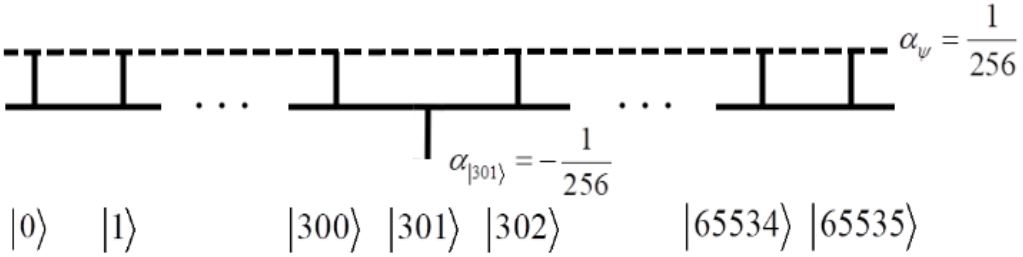
Зазначимо, що $|301\rangle$ є єдиним елементом, що має стан зі знаком *« − «*. Запишемо $|\psi_1\rangle$ у вигляді

$$|\psi_1\rangle = |\psi\rangle - \frac{1}{128}|301\rangle,$$

або у вигляді

$$|\psi_1\rangle = \frac{\sqrt{65535}}{256}|u\rangle - \frac{1}{256}|301\rangle.$$

Геометрично подамо отримані значення у вигляді позитивних та негативних:



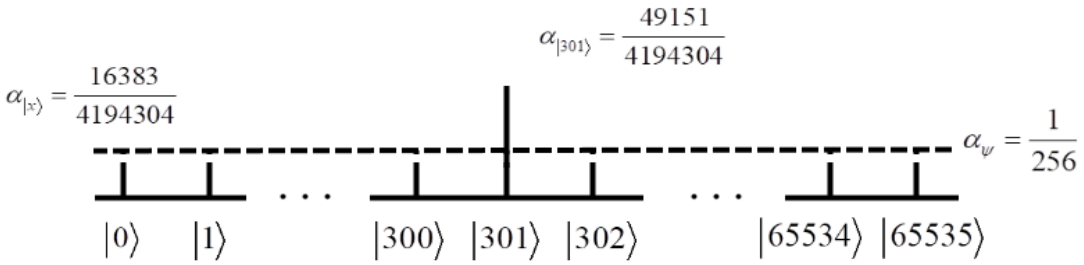
7. Далі ми обчислюємо

$$|\psi_2\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle; \quad |\psi_2\rangle = \frac{16383}{16384}|\psi\rangle + \frac{1}{128}|301\rangle;$$

$$|\psi_2\rangle = \frac{16383\sqrt{65535}}{4194304}|u\rangle + \frac{49151}{4194304}|301\rangle.$$

Слід відмітити, що $\langle\psi|\psi\rangle = 1$. На додаток до цього, так як $|301\rangle$ є одним з базисних векторів, ми можемо використати відповідність $\langle\psi|301\rangle = \langle 301|\psi\rangle = \frac{1}{256}$.

В результаті геометрично результат можна подати як:



Наведеним результатом завершується 1 ітерація ***G,*** залишилося ще 200. Далі отримуємо. що:

$$|\psi_3\rangle = 0{,}99993896|\psi\rangle - 0{,}01562452|301\rangle;$$

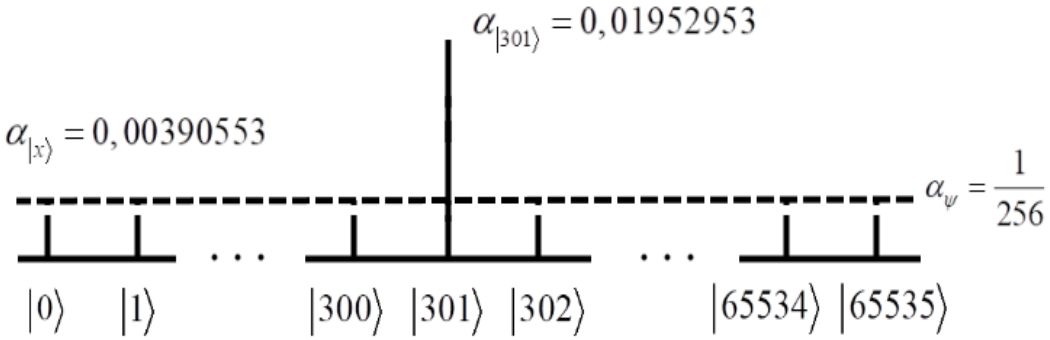
$$|\psi_3\rangle = 0{,}00390601\sqrt{65535}|u\rangle - 0{,}01171851|301\rangle.$$

Геометрично результат подаємо як :

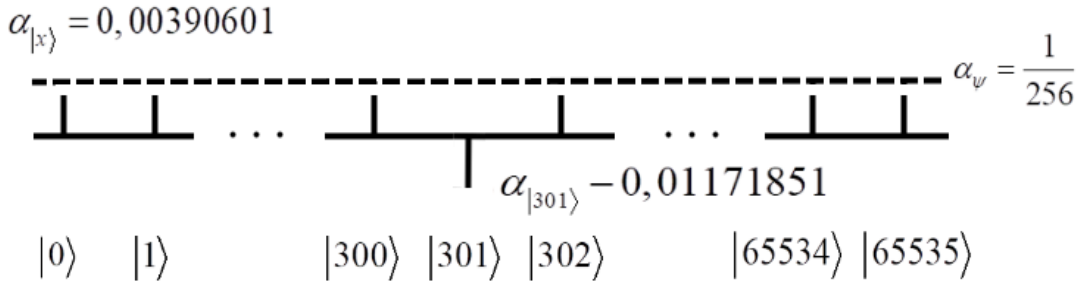
$$|\psi_4\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_3\rangle = 0{,}99981689|\psi\rangle + 0{,}01562453|301\rangle;$$

$$|\psi_4\rangle = 0{,}00390553\sqrt{65535}|u\rangle + 0{,}01952953|301\rangle.$$

Геометрично результат 2 ітерації подаємо як:



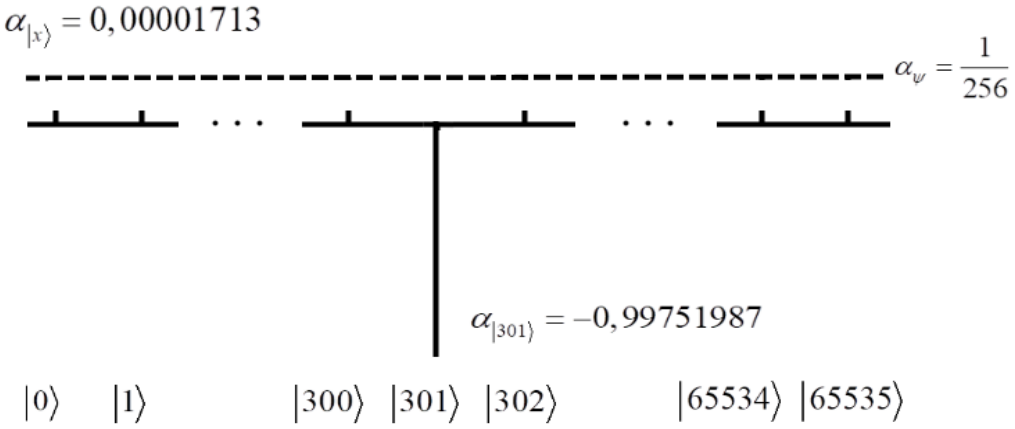
Отриманим значенням завершується 2 ітерація, залишається ще 199. … На останній ітерації маємо:



$$|\psi_{401}\rangle = 0{,}00438613|\psi\rangle - 0{,}99751987|301\rangle;$$

$$|\psi_{401}\rangle = 0{,}00001713\sqrt{65535}|u\rangle - 0{,}997537|301\rangle.$$

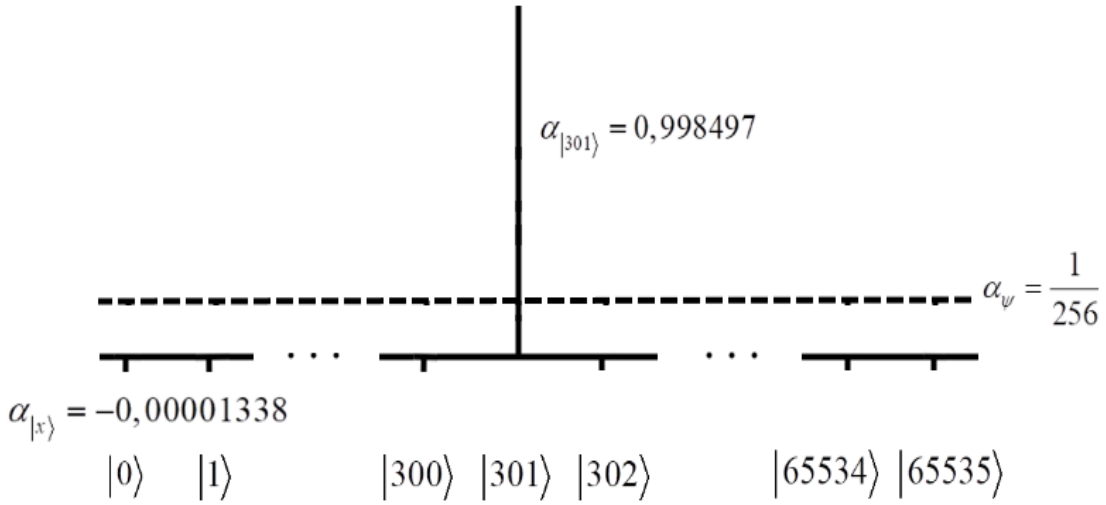
Геометрично результат подаємо як :



$$|\psi_{402}\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_{401}\rangle = -0,00342647|\psi\rangle + 0,99851038|301\rangle;$$

$$|\psi_{402}\rangle = -0,00001338\sqrt{65535}|u\rangle + 0,998497|301\rangle.$$

Геометрично результат останньої ітерації подаємо як:



Наведеним завершується виконання 201 ітерації *G*.

Вимірювання стану $|\psi_{302}\rangle$ дасть результат у вигляді стану $|301\rangle$ з ймовірністю, що становить:

$$P \approx 0,99998826 \approx 99,998826\% .$$

Таким чином, шанс отримання результату $|301\rangle$, який має індекс 301, близько 99,998826%. Як можна побачити на прикладі, зі зростанням N похибка стає незначною.

Приклад 12.2. Припустимо, що система складається з $N = 16 = 2^4$ станів, і стан, який ми шукаємо, x_0, має індекс 7 та представлений бітовим рядком $|7\rangle = |0111\rangle$. Розглянемо алгоритм Гровера пошуку даного «унікального» елемента по крокам.

1. Для того, щоб описати цю систему, потрібно $n = 4$ кубіти. У відповідності з алгоритмом Гровера зробимо ініціалізацію квантового регістра з

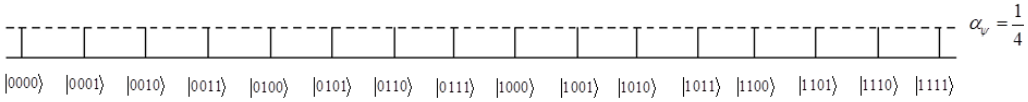
$n = 4$ кубітів, що необхідно для представлення пошукового простору розміру $N = 2^4$, встановивши регістр у початковий стан:

$$|\psi_0\rangle = |0000\rangle.$$

2. Проведемо перетворення Адамара, що дозволяє отримати значення амплітуди, що пов'язана з кожним станом з рівною ймовірністю перебування в кожному з 16 можливих станів:

$$|\psi\rangle = H^{\otimes 4}|0000\rangle = (H|0\rangle)^{\otimes 4} = \frac{1}{4}\sum_{i=0}^{15}|i\rangle.$$

Геометрично це можна зобразити як:



3. За вказаних даних оптимальним для отримання рішення є виконання ітерацій Гровера, кількість яких визначається таким чином:

$$\frac{\pi}{4}\sqrt{2^n} = \frac{\pi}{4}\sqrt{16} = \frac{4\pi}{4} = \pi \approx 3.14.$$

В подальшому для використання округлимо число ітерацій до 3.

В кожній ітерації першим кроком є виклик квантового оракула $\mathcal{O}$, потім проводиться інверсія середнього або ж оператор дифузії.

4. При пошуку елементу з індексом 7 оракул дає такі значення [56,57]:

$$U_f(|0111\rangle|-\rangle) = -|0111\rangle|-\rangle; \quad U_f(|i\rangle|-\rangle) = |i\rangle|-\rangle, \text{ if } i \neq 7;$$

1. Далі визначимо $|u\rangle$**,** використовуючи (2), в результаті маємо:

$$|u\rangle = \frac{1}{\sqrt{15}}\sum_{\substack{i=0 \\ i\neq 7}}^{15}|i\rangle = \frac{\begin{matrix}|0000\rangle+|0001\rangle+|0010\rangle+|0011\rangle+|0100\rangle+|0101\rangle+|0110\rangle+|1000\rangle+|1001\rangle+|1010\rangle \\ +|1011\rangle+|1100\rangle+|1101\rangle+|1110\rangle+|1111\rangle\end{matrix}}{\sqrt{15}}$$

Також аналогічно (2) маємо, що $|\psi\rangle = \frac{\sqrt{15}}{4}|u\rangle + \frac{1}{4}|0111\rangle$**.**

2. Наступним кроком знайдемо

$$|\psi_1\rangle|-\rangle = U_f(|\psi\rangle|-\rangle) = \left(\frac{\begin{matrix}|0000\rangle+|0001\rangle+|0010\rangle+|0011\rangle+|0100\rangle+|0101\rangle+|0110\rangle-|0111\rangle+|1000\rangle+ \\ |1001\rangle+|1010\rangle+|1011\rangle+|1100\rangle+|1101\rangle+|1110\rangle+|1111\rangle\end{matrix}}{4}\right)|-\rangle.$$

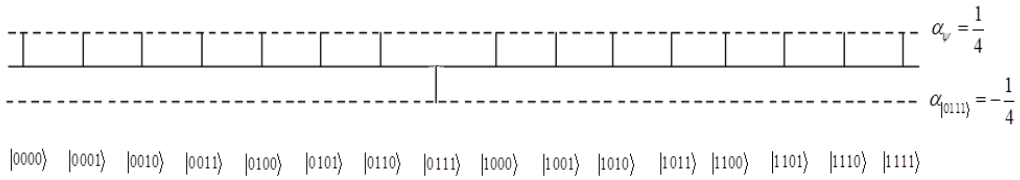
Зазначимо, що $|0111\rangle$ є єдиним елементом, що має стан зі знаком « − «. Запишемо $|\psi_1\rangle$ у вигляді:

$$|\psi_1\rangle = |\psi\rangle - \frac{1}{2}|0111\rangle,$$

або у вигляді:

$$|\psi_1\rangle = \frac{\sqrt{15}}{4}|u\rangle - \frac{1}{4}|0111\rangle\,.$$

Геометрично подамо отримані значення у вигляді позитивних та негативних:



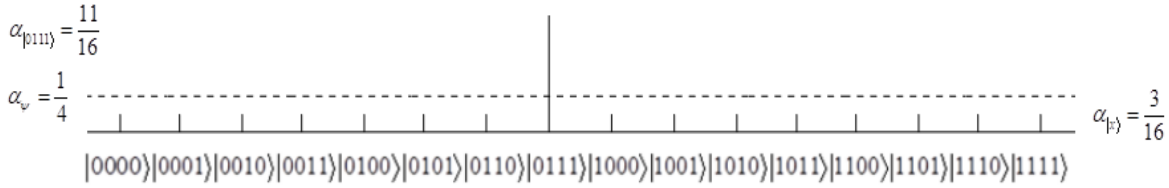
Далі ми обчислюємо

$$|\psi_2\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle; \quad |\psi_2\rangle = \frac{3}{4}|\psi\rangle + \frac{1}{2}|0111\rangle$$

$$|\psi_2\rangle = \frac{3\sqrt{15}}{16}|u\rangle + \frac{4}{16}|0111\rangle = \frac{3\sqrt{15}}{16}|u\rangle + \frac{1}{4}|0111\rangle.$$

Слід відмітити, що $\langle\psi|\psi\rangle = 16\frac{1}{4}\left[\frac{1}{4}\right] = 1$. На додаток до цього, так як $|0111\rangle$ є одним з базисних векторів, ми можемо використати відповідність $\langle\psi|0111\rangle = \langle 0111|\psi\rangle = \frac{1}{4}$.

Геометрично результат можна подати як:

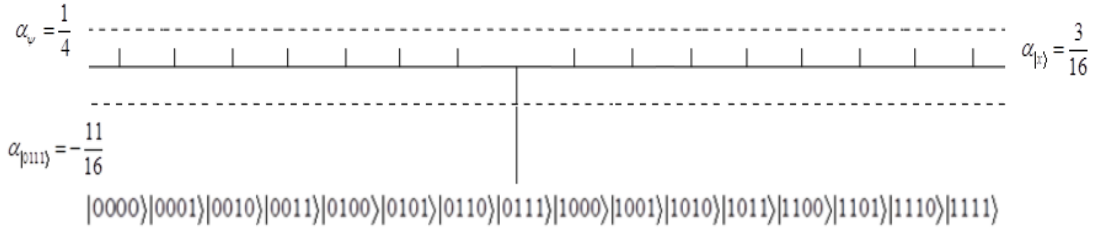


Наведеним результатом завершується 1 ітерація ***G***, залишилося ще 2. Далі отримуємо. що:

$$|\psi_3\rangle = \frac{3}{4}|\psi\rangle - \frac{7}{8}|0111\rangle\,,$$

$$|\psi_3\rangle = \frac{3\sqrt{15}}{16}|u\rangle - \frac{11}{16}|0111\rangle\,.$$

Геометрично результат подаємо як :

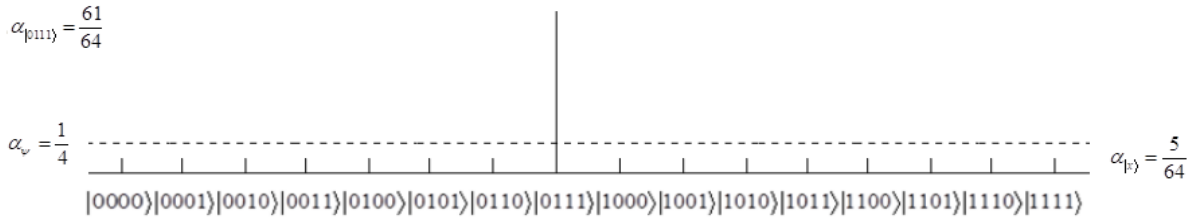


Далі отримаємо:

$$|\psi_4\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_3\rangle = \frac{5}{16}|\psi\rangle + \frac{7}{8}|0111\rangle\,,$$

$$|\psi_4\rangle = \frac{5\sqrt{15}}{64}|u\rangle + \frac{61}{64}|0111\rangle\,.$$

Геометрично результат 2 ітерації подаємо як:

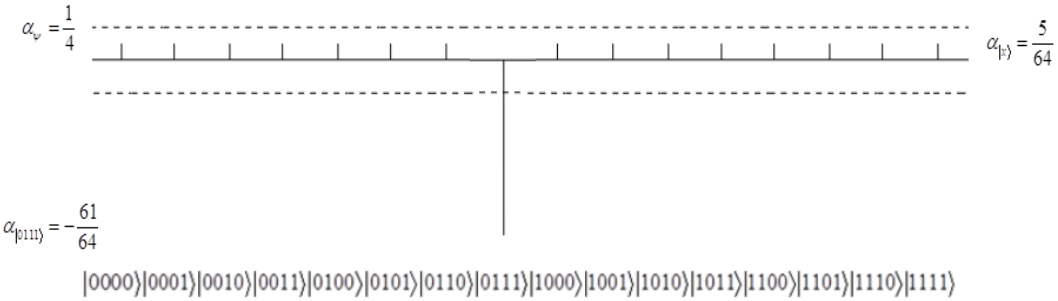


Отриманим значенням завершується 2 ітерація, залишається ще одна ***G*** ітерація. При її виконанні маємо:

$$|\psi_5\rangle = \frac{5}{16}|\psi\rangle - \frac{33}{32}|0111\rangle\,,$$

$$|\psi_5\rangle = \frac{5\sqrt{15}}{64}|u\rangle - \frac{61}{64}|0111\rangle\,.$$

Геометрично результат 3 ітерації подаємо як:

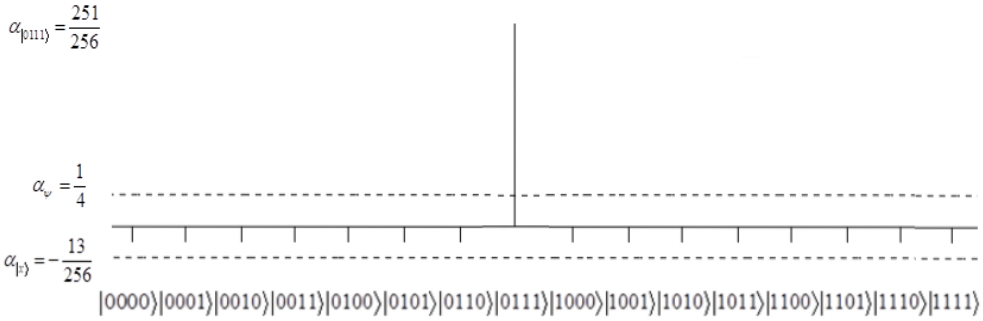


При виконанні третьої ітерації отримаємо:

$$|\psi_6\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_5\rangle = -\frac{13}{64}|\psi\rangle + \frac{33}{32}|0111\rangle$$

$$|\psi_6\rangle = -\frac{13\sqrt{15}}{256}|u\rangle + \frac{251}{256}|0111\rangle\,.$$

На завершення 3-ї ітерації маємо



Наведеним завершується виконання 3 ітерації *G*.

Вимірювання стану $|\psi_6\rangle$ дасть результат у вигляді стану $|0111\rangle$ з ймовірністю, що становить:

$$P = \left|\frac{251}{256}\right|^2 \approx 0{,}961.$$

Таким чином, шанс отримання результату $|0111\rangle$, який має індекс 7, близько 96,1%. Ймовірність отримання невірного стану становить близько 3,9%. Хоча метод Гровера є ймовірнісним, зі зростанням N похибка стає незначною.

Приклад 12.3. Припустимо, що система складається з $N = 256 = 2^8$ станів і стан, який ми шукаємо $\boldsymbol{x_0}$ **,** має індекс 15 та представлений бітовим рядком:

$$|x\rangle = |15\rangle = |0000000000001111\rangle.$$

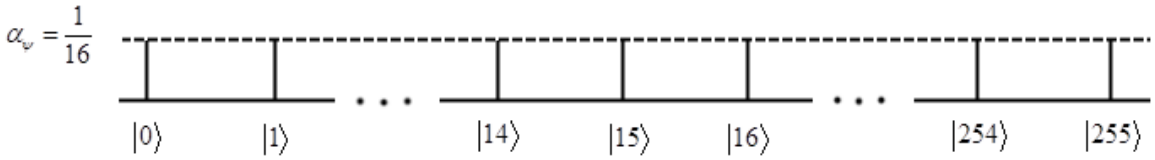
1. Розглянемо алгоритм Гровера пошуку даного «унікального» елемента. Для того, щоб описати цю систему, потрібно $n = 8$ кубітів. У відповідності з алгоритмом Гровера зробимо ініціалізацію квантового регістру з $n = 8$ кубітів, що необхідне для представлення пошукового простору розміру $N = 2^8$, встановивши регістр у початковий стан:

$$|\psi_0\rangle = |00000000\rangle\,.$$

2. Проведемо перетворення Адамара, що дозволяє отримати значення амплітуди, що пов'язана з кожним станом з рівною ймовірністю перебування в кожному з 16 можливих станів:

$$|\psi\rangle = \frac{1}{16}\sum_{i=0}^{255}|i\rangle.$$

Геометрично отримаємо результат



3. По аналогії з прикладом 12.1 маємо

$$\frac{\pi}{4}\sqrt{256} = \frac{16\pi}{4} = 4\pi \approx 12{,}56 \approx 13,$$

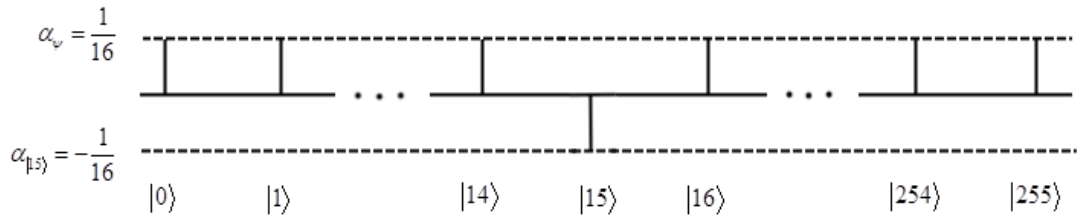
$$|u\rangle = \frac{1}{\sqrt{255}}\sum_{\substack{i=0\\ i\neq 15}}^{255}|i\rangle,$$

$$|\psi\rangle = \frac{\sqrt{255}}{16}|u\rangle + \frac{1}{16}|x\rangle.$$

4. Для першої ітерації маємо

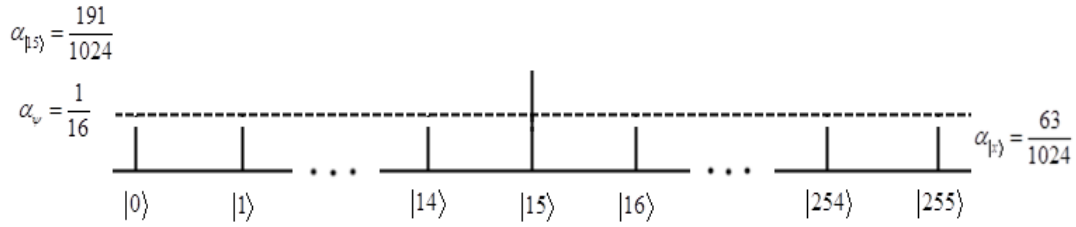
$$|\psi_1\rangle = |\psi\rangle - \frac{1}{8}|x\rangle,$$

$$P = 0.00390625.$$



$$|\psi_2\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle = \frac{63}{64}|\psi\rangle + \frac{1}{8}|x\rangle,$$

$$P = 0.03479099.$$

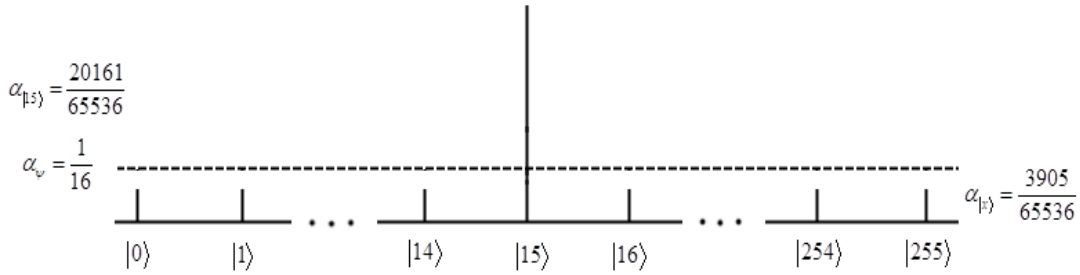


5. Друга ітерація

$$|\psi_3\rangle = \frac{63}{64}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{1}{8}|x\rangle = \frac{63}{64}|\psi\rangle - \frac{127}{512}|x\rangle\,,$$

$$|\psi_4\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_3\rangle = \frac{3905}{4096}|\psi\rangle + \frac{127}{512}|x\rangle,$$

$$P = 0.094637722\,.$$



6. Третя ітерація

$$|\psi_5\rangle = \frac{3905}{4096}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{127}{512}|x\rangle = \frac{3905}{4096}|\psi\rangle - \frac{12033}{32768}|x\rangle,$$

$$|\psi_6\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_5\rangle = \frac{237887}{262144}|\psi\rangle + \frac{12033}{32768}|x\rangle,$$

$$P = 0.17972063.$$

7. Четверта ітерація

$$|\psi_7\rangle = \frac{237887}{262144}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{12033}{32768}|x\rangle = \frac{237887}{262144}|\psi\rangle - \frac{1007999}{2097152}|x\rangle,$$

$$|\psi_8\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_7\rangle = \frac{14216769}{16777216}|\psi\rangle + \frac{1007999}{2097152}|x\rangle.$$

8. П’ята ітерація

$$|\psi_9\rangle = \frac{14216769}{16777216}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{1007999}{2097152}|x\rangle = \frac{14216769}{16777216}|\psi\rangle - \frac{78728705}{134217728}|x\rangle,$$

$$|\psi_{10}\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_9\rangle = \frac{831144511}{1073741824}|\psi\rangle + \frac{78728305}{134217728}|x\rangle\,.$$

9. Шоста ітерація

$$|\psi_{11}\rangle = \frac{831144511}{1073741824}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{78728305}{134217728}|x\rangle =$$

$$= \frac{831144511}{1073741824}|\psi\rangle - \frac{5869756031}{8589934592}|x\rangle$$

$$|\psi_{12}\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_{11}\rangle = \frac{47323492673}{68719476736}|\psi\rangle + \frac{5869756031}{8589934592}|x\rangle.$$

10. Сьома ітерація

$$|\psi_{13}\rangle = \frac{47323492673}{68719476736}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{5869756031}{8589934592}|x\rangle = \frac{47323492673}{68719476736}|\psi\rangle - \frac{422987878657}{549755813888}|x\rangle,$$

$$|\psi_{14}\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_{13}\rangle = \frac{2605715652415}{4398046511104}|\psi\rangle + \frac{422987878657}{549755813888}|x\rangle.$$

11. Восьма ітерація

$$|\psi_{15}\rangle = \frac{2605715652415}{4398046511104}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{422987878657}{549755813888}|x\rangle = \frac{2605715652415}{4398046511104}|\psi\rangle - \frac{29676939886463}{35184372088832}|x\rangle$$

$$|\psi_{16}\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_{15}\rangle = \frac{137088861868097}{281474976710656}|\psi\rangle - \frac{29676939886463}{35184372088832}|x\rangle\ .$$

12. Дев’ята ітерація

$$|\psi_{17}\rangle = \frac{137088861868097}{281474976710656}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{29676939886463}{35184372088832}|x\rangle =$$

$$= \frac{137088861868097}{281474976710656}|\psi\rangle - \frac{2036413014601729}{2251799813685248}|x\rangle$$

$$|\psi_{18}\rangle = \left(2|\psi\rangle\langle\psi| - I\right)|\psi_{17}\rangle =$$

$$= \frac{6737274144956479}{18014398509481984}|\psi\rangle + \frac{2036413014601729}{2251799813685248}|x\rangle$$

13. Десята ітерація

$$|\psi_{19}\rangle = \frac{6737274144956479}{18014398509481984}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{2036413014601729}{2251799813685248}|x\rangle =$$

$$= \frac{6737274144956479}{18014398509481984}|\psi\rangle - \frac{137067707079467135}{144115188075855872}|x\rangle$$

$$|\psi_{20}\rangle = \left(2|\psi\rangle\langle\psi| - I\right)|\psi_{19}\rangle =$$

$$= \frac{294117838197747521}{1152921504606846976}|\psi\rangle + \frac{137067707079467135}{144115188075855872}|x\rangle\ .$$

14. Одинадцята ітерація

$$|\psi_{21}\rangle = \frac{294117838197747521}{1152921504606846976}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{137067707079467135}{144115188075855872}|x\rangle =$$

$$= \frac{294117838197747521}{1152921504606846976}|\psi\rangle - \frac{9066451091283644161}{9223372036854775808}|x\rangle \quad ,$$

$$|\psi_{22}\rangle = \left(2|\psi\rangle\langle\psi| - I\right)|\psi_{21}\rangle =$$

$$= \frac{9757090553372197183}{7378697629483820646 4}|\psi\rangle + \frac{9066451091283644161}{9223372036854775808}|x\rangle .$$

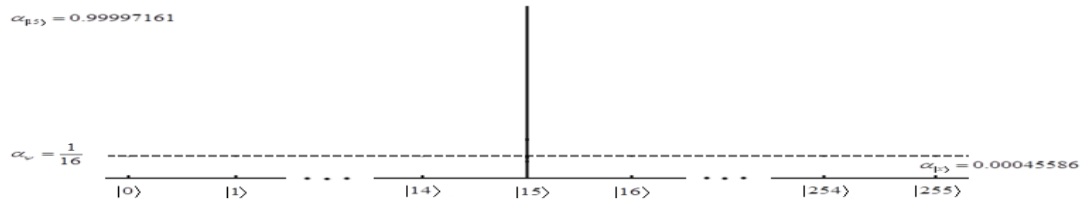
15. Дванадцята ітерація

$$|\psi_{23}\rangle = \frac{9757090553372197183}{7378697629483820646 4}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{9066451091283644161}{9223372036854775808}|x\rangle =$$

$$= \frac{9757090553372197183}{7378697629483820646 4}|\psi\rangle - \frac{590009960395525423487}{590295810358705651712}|x\rangle \quad ,$$

$$|\psi_{24}\rangle = \left(2|\psi\rangle\langle\psi| - I\right)|\psi_{23}\rangle =$$

$$= \frac{34443835020295196225}{4722366482869645213696}|\psi\rangle + \frac{590009960395525423487}{590295810358705651712}|x\rangle .$$



$$P = |0.99997161|^2 \approx 0.99994322 \approx 99.994322\%$$

16. Тринадцята ітерація

$$|\psi_{25}\rangle = \frac{34443835020295196225}{4722366482869645213696}\left(|\psi\rangle - \frac{1}{8}|x\rangle\right) - \frac{590009960395525423487}{590295810358705651712}|x\rangle =$$

$$= \frac{34443835020295196225}{4722366482869645213696}|\psi\rangle - \frac{37795081300333922299393}{37778931862957161709568}|x\rangle \quad ,$$

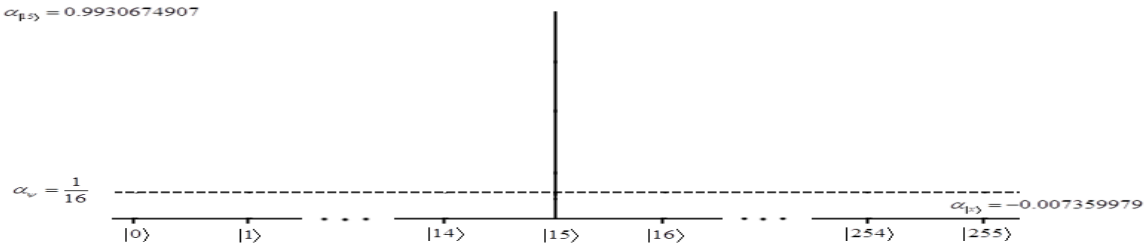
$$|\psi_{26}\rangle = \left(2|\psi\rangle\langle\psi| - I\right)|\psi_{25}\rangle =$$

$$= -\frac{35590675859035029740993}{30223145490365729367 6544}|\psi\rangle + \frac{37795081300333922299393}{37778931862957161709568}|x\rangle .$$

$$|\psi\rangle = \frac{\sqrt{255}}{16}|u\rangle + \frac{1}{16}|x\rangle ,$$

$$|\psi_{26}\rangle = -\frac{35590675859035029740993\sqrt{255}}{48357032784585166988247 04}|u\rangle + \frac{48021797305837070245813 11}{48357032784585166988247 04}|x\rangle$$

Наприкінці отримуємо значення ймовірності правильного визначення елемента 15.



$$P = \left|\frac{4802179730583707024581311}{4835703278458516698824704}\right|^2 \approx 0{,}98618305$$

З прикладу 12.3 можна зробити висновок, що для досягнення найкращого результату кількість ітерацій Гровера потрібно округляти до меншого значення. **$72057594037927936 = 2^{56}$** станів, що означає, що база для пошуку складається з **72057594037927936** елементів. Також припустимо, що стан системи, який ми хочемо отримати в результаті пошуку x_0, має індекс 234.

1) Для опису системи необхідно n = 56 кубітів. Згідно з алгоритмом Гровера ініціалізуємо квантовий регістр, що складатиметься з n = 56 кубітів, що є необхідною умовою для представлення пошукового простору, що має розмір $N = 2^{56}$, встановивши регістр у початковий стан, що має наступний вигляд: $|\psi_0\rangle = |000\ldots000\rangle$, де кількість нулів дорівнює 56.

2) Проведемо перетворення Адамара, що дозволяє встановити систему в стан квантової суперпозиції, що робить значення амплітуди, яка пов'язана з кожним станом таким чином, щоб ймовірність перебування в кожному з 2^{56} можливих станів була рівною. Цей крок матиме наступний вигляд:

$$|\psi\rangle = H^{\otimes 56}|000\ldots000\rangle = (H|0\rangle)^{\otimes 56} = \frac{1}{268435456}\sum_{i=0}^{2^{56}}|i\rangle .$$

Таким чином, ми маємо квантовий регістр, встановлений в стан суперпозиції, що геометрично можна представити так, як наведено на рис. 12.10.

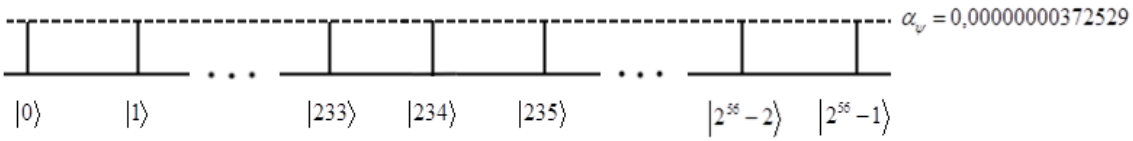


Рис. 12.10. Геометричне представлення регістру з 256 кубітів в стані суперпозиції

3) Визначаємо кількість необхідних ітерацій ***G***:

$$\frac{\pi}{4}\sqrt{2^n} = \frac{\pi}{4}\sqrt{2^{56}} = \frac{2^{28}\pi}{4} \approx 210828714{,}133156 .$$

Далі, для використання в розрахунках округлимо число ітерацій до **210828714** у зв'язку з округленням вниз.

4) Після цього визначимо $|\boldsymbol{u}\rangle$:

$$|u\rangle = \frac{1}{\sqrt{2^{56}-1}} \sum_{\substack{i=0 \\ i\neq 301}}^{2^{56}-1} |i\rangle = \frac{|0\rangle + |1\rangle + \ldots + |2^{56}-2\rangle + |2^{56}-1\rangle}{\sqrt{2^{56}-1}}.$$

Після перетворювань попередній вираз прийме вигляд:

$$|\psi\rangle = \frac{\sqrt{2^{56}-1}}{2^{28}}|u\rangle + \frac{1}{2^{28}}|234\rangle.$$

5) Далі, після звернення до оракула, починається перша ітерація. Після застосування оракула маємо:

$$|\psi_1\rangle = |\psi\rangle - \frac{1}{2 * 2^{28}}|234\rangle = |\psi\rangle - \frac{1}{536870912}|234\rangle.$$

Стан даного регістра після застосування оракула під час першої ітерації Гровера має геометричне відображення, що наведено на рис. 12.11.

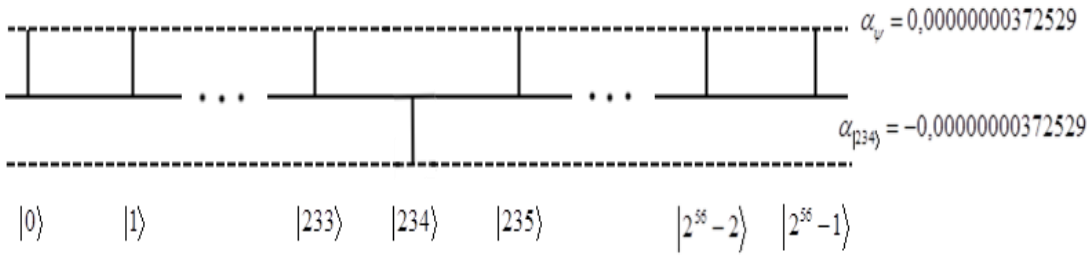


Рис. 12.11. Геометричне відображення стану регістру після застосування оракула під час першої ітерації *G*

Далі відбувається виконання оператора дифузії, в результаті чого отримуємо:

$$|\psi_2\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle = 0{,}9999999999999999861|\psi\rangle + 0{,}0000000018626|234\rangle;$$

Геометричне представлення стану регістру після першої ітерації зображено на рис. 12.12.

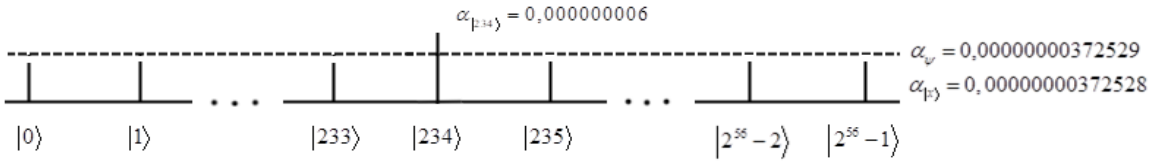


Рис. 12.12. Геометричне представлення стану квантового регістру після проведення першої ітерації *G*

Далі аналогічним чином проводиться ще **210828713** ітерацій.

Під час проведення останньої, **210828714**-ї ітерації отримаємо в результаті застосування оракула:

$$|\psi_{210828714}\rangle = 0{,}00000000000003\,|\psi\rangle - \\ - 0{,}9999999999999841|234\rangle.$$

Стан даного регістра після застосування оракула під час останньої ітерації Гровера має геометричне відображення, що наведено на рис. 12.13.

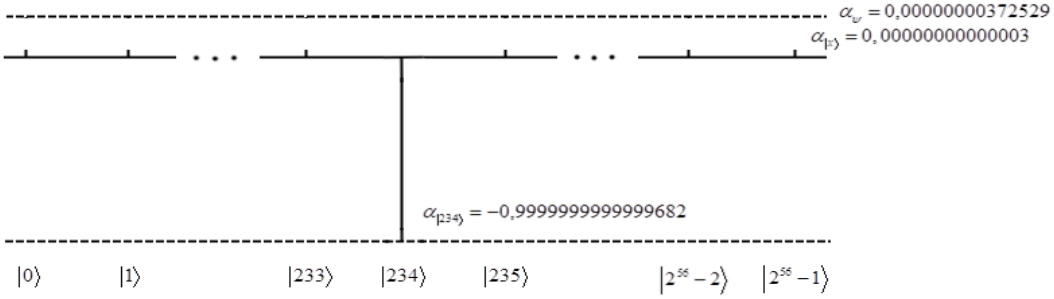


Рис. 12.13. Геометричне відображення стану регістру після застосування оракула під час останньої ітерації ***G***

Після застосування оператора дифузії під час останньої ітерації маємо:

$$|\psi_{210828714}\rangle=(2|\psi\rangle\langle\psi| - I)|\psi_{210828713}\rangle=-0{,}00000000000001|\psi\rangle +$$
$$+ 0{,}99999999999999841|234\rangle.$$

Геометричне представлення результатів останньої ітерації наведено на рис. 12.14.

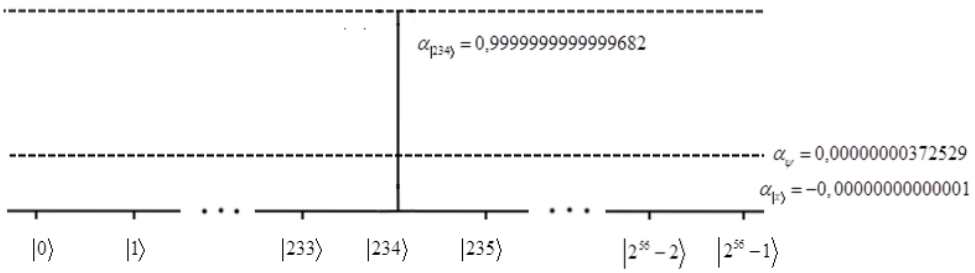


Рис. 12.14. Геометричне представлення стану квантового регістра після проведення останньої ітерації G

Ймовірність отримання необхідного елемента в результаті вимірювання регістра після проведення 210828714 ітерацій становить:

$$P \approx \left|0{,}99999999999999841\right|^2 \approx 0{,}9999999999999682 \approx 99{,}99999999999682\%.$$

Таким чином, на цьому прикладі, враховуючи попередні, можна побачити, що зі зростанням ***N*** похибка дійсно стає незначною.

Також слід зазначити, що приклад було розраховано з використанням програмної моделі проведення пошуку методом Гровера, що була модернізована за швидкодією. В загальному вигляді програмна модель, яка використана для розрахунків, відповідає загальній програмній моделі методу Гровера, але відрізняється тим, що оракул було спрощено до елементарної функції пошуку, що також проводилася повністю лише в частині, відповідальній за ініціалізацію. Така побудова моделі дозволила провести розрахунки без вирішення складної задачі пошуку, але зі збереженням достовірності результатів розрахунків на рівні, що відповідає рівню достовірності результатів програмної моделі, що проводить виконання складної задачі пошуку.

Висновки

1. Нині суттєві результати досягнені в частині розроблення, стандартизації та застосування симетричних криптоперетворень. Разом з тим, продовжується розвиток та здійснюються спроби розробити більш ефективні методи криптоаналізу симетричних криптосистем – симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП) та функцій гешування (ФГ). Одним із основних є метод Гровера.

2. Освоєння та застосування квантових систем та квантових математичних методів, а також відповідне їх програмування носить як методологічний аспект, так і психологічний – складність сприйняття.

3. Метод Гровера полягає в проведенні вичерпного пошуку специфічного (унікального) елемента у несортованій базі даних, що складається з $\boldsymbol{N = 2^n}$ елементів, де $\boldsymbol{n}$ довжина квантового регістра (кількість кубітів).

4. Для криптоаналізу СБП специфічність елемента може зводитись до сеансового чи довгострокового ключа, синхропослідовності тощо. Особливістю, наприклад, ключа є те, що при його застосуванні зашифровані дані можуть бути розшифрованими за поліноміальний час.

5. У порівнянні з найкращими класичними методами метод Гровера передбачає проведення пошуку з квадратичним прискоренням. Для отримання такого прискорення використовується квантова суперпозиція станів.

6. Конкретна реалізація квантового оракула залежить від кожного окремого випадку та задачі. Наприклад, у випадку криптоаналізу СБП оракул виконує розшифрування на ключі та повертає результат - успіх чи ні.

7. Метод Гровера може бути застосований як на класичному, так і квантовому комп'ютері, хоча реалізація на класичному комп'ютері не є рентабельною.

8. Наведені вище приклади підтвердили ефективність методу Гровера стосовно пошуку в несортованій базі: для: 4 – бітного числа з ймовірністю близько 96,1% зі складністю в 3 раунди; для 8 – бітного числа з ймовірністю близько 99% зі складністю в 13 раундів (силова атака вимагає 256 раундів).

9. Таким чином, застосування методу Гровера для пошуку специфічного елемента в несортованій базі дійсно дозволяє досягти квадратичне прискорення пошуку, яке вимагає $\sqrt{\boldsymbol{N}}$ раундів у порівнянні з **N** "грубої сили".

12.6 Практична реалізація методу Гровера

Для розуміння методу Гровера необхідно визначити квантову суперпозицію станів [59,62]. Нехай $|\psi\rangle$ - суперпозиція всіх станів (згідно нотації Дірака):

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

З урахуванням цього, алгоритм приймає такий вигляд:

1) Встановлення системи в стан суперпозиції $|\psi\rangle$.

2) Виконання «ітерації Гровера» (або ж G) $\frac{\pi}{4}\sqrt{N}$ разів, де N= 2^n та N становить розмір пошукового простору, a_n- розмір квантового регістру, що використовується для представлення пошукового простору. При цьому G включає в себе 2 етапи:

2.1) Застосування квантового оракула (O).

2.2) Застосування оператора дифузії, що здійснює «інверсію щодо середнього» та має вигляд $2|0\rangle\langle 0| - I$.

3) Виконання класичних вимірювань регістра для отримання результату роботи алгоритму, що з ймовірністю, близькою до 1, буде вірним.

З метою демонстрації дії методу Гровера на практиці розглянуті приклади використання алгоритму пошуку несортованою базою даних з невеликим розміром пошукового простору, як було зроблено в [62], в підрозділі 12.5.

12.7 Сутність та особливості реалізації методу Гровера на класичному комп'ютері для симетричного криптоаналізу

Нині в криптографічному загалі широко обговорюється та досліджується проблема створення та стандартизації перспективних криптографічних перетворень, в першу чергу для постквантового періоду [54,55]. Суттєві результати досягнені в частині розроблення, стандартизації та застосування симетричних криптоперетворень [56]. Разом з тим, продовжується розвиток та здійснюються спроби розробити більш ефективні методи криптоаналізу симетричних криптосистем – симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП) та функцій гешування (ФГ). Підтвердженням цьому є прийняття та застосування міжнародних стандартів ДСТУ ISO/IEC 18033 -3, 18034 – 4, ДСТУ 7624:2014, ДСТУ 7564:2014, FIPS – 197, FIPS – 202 тощо. При цьому великий інтерес до задач криптоаналізу проявляють як безпосередньо криптоаналітики, так і розробники стандартизованих симетричних та асиметричних криптосистем.

Зрозуміло, що криптоаналітики направляють свої зусилля на безпосередній злом відповідних криптосистем, а розробники та ті, що застосовують криптографічні перетворення, на перевірку їх криптографічних властивостей. Якщо раніше для спроб вирішення задач криптоаналізу застосовувались класичні спеціалізовані комп'ютерні системи, засоби та класична математика, то нині розробляються квантові комп'ютери та практично розроблені відповідні квантові математичні методи [54-56]. Але освоєння та застосування квантових систем та квантових математичних методів, а також програмування носить як методологічний аспект, так і психологічний – складність сприйняття.

Дослідження, що проведені, дозволяють прогнозувати використання для симетричного криптоаналізу квантового методу, що отримав назву методу Гровера [56,57]. При цьому на сучасному етапі розглядаються деталізація, освоєння для застосування, перевірка криптоаналітичних властивостей та демонстрація застосування методу Гровера при криптоаналізі СБП, в тому числі,

з методичним освоєнням методу при навчанні з використанням прикладів, але поки що на класичному комп'ютері.

12.7.1 Криптоаналіз з використанням методу Гровера

Квантовий алгоритм Гровера розроблено для вирішення задачі проведення пошуку неструктурованою базою даних певного унікального елемента.

В загальному вигляді ця проблема може бути сформульована так: неструктурована база даних складається з N елементів та містить один унікальний елемент, що має певну властивість, котра може бути перевіреною з поліноміальною складністю, та котрий потрібно знайти з мінімальними витратами часу та складністю. Класичні методи потребують виконання $O(N)$ запитів до бази для знаходження потрібного елементу, в той час як алгоритм Гровера дозволяє виконати лише приблизно $O(\sqrt{N})$ кроків, що є ітераціями виконання процедури, до бази та бути впевненим, що отриманий елемент буде саме потрібним елементом із ймовірністю, близькою до 1. Як і в квантових алгоритмах, націлених на криптоаналіз симетричних перетворень, в алгоритмі Гровера здійснюється повтор процедури (ітерації Гровера) для збільшення ймовірності отримання правильного результату. Також аналогічно до таких алгоритмів у методі Гровера, при продовженні повторення ітерацій після досягнення потрібної їх кількості, ймовірність отримання правильного результату зменшується. Відбувається це через те, що під час виконання ітерації Гровера здійснюється поворот в комплексному просторі. Так, кожна ітерація, здійснюючи поворот, наближає регістр до потрібного стану все ближче та ближче, але в певній точці досягається максимальна близькість, при котрій продовження застосування ітерацій призведе до повороту повз потрібний стан, що віддалить стан системи від потрібного стану.

Базою даних для цього алгоритму може бути будь-який пошуковий простір, що складається з певної кількості елементів. Так, наприклад, його можна застосувати для знаходження секретного ключа для симетричного криптоперетворення або ж для знаходження колізії для функції гешування. Окрім цього, існує безліч можливих застосувань для цього квантового методу.

В підрозділі 7.10 розглянутий авторами приклад квантового алгоритму Гровера пошуку деякого обраного елемента з великого набору *N* елементів.

12.7.2 Модель та реалізація методу Гровера на класичному комп'ютері

В якості програмної моделі застосування методу Гровера для криптоаналізу симетричних перетворень було розроблено програму для класичного комп'ютера, що здійснює пошук ключа та перевірку його правильності для алгоритму DES. Для цього було використано засоби мови програмування Java.

Алгоритм Гровера можна узагальнити наступним чином:

На вхід подаються:

- Квантовий оракул O. Він проводить операцію $O|x\rangle = (-1)^{f(x)}|x\rangle$, де $f(x) = 0$ для всіх елементів пошукового простору, окрім x_0, для якого $f(x) = 1$.

- Квантовий регістр, що складається з n кубітів, встановлений в стан $|0\rangle^{\otimes n}$.

На виході отримуємо лише елемент x_0, що є унікальним та відповідає вимогам пошуку.

Алгоритм потребує для виконання з ймовірністю успіху $O(1)$ $O\left(\sqrt{2^n}\right)$ операцій.

Послідовність виконання алгоритму можна звести до таких пунктів:

1) Початковий стан квантового регістру $|0\rangle^{\otimes n}$.

2) Встановлення квантового регістру в стан суперпозиції, що здійснюється завдяки виконанню над регістром перетворення Адамара:

$$H^{\otimes n}\left|0\right\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}\left|x\right\rangle = \left|\psi\right\rangle.$$

3) Виконання $R \approx \frac{\pi}{4}\sqrt{2^n}$ ітерацій Гровера, де R - кількість виконань ітерації Гровера для того, щоб ймовірність отримання правильного результату була оптимальною та відхилення фази не перевищувало $\frac{\pi}{4}$, а ітерація Гровера має вигляд $[(2|\psi\rangle\langle\psi| - I)O]^R|\psi\rangle \approx |x_0\rangle$.

4) Вимірювання квантового регістру та отримання унікального x_0, що відповідає вимогам пошуку.

Розглянемо поетапно структуру програмної моделі криптоаналізу симетричних перетворень за допомогою методу Гровера.

Спочатку необхідно отримати регістр бітів. Проте в мові програмування Java немає можливості записувати один біт окремо, а якщо виділити масив будь-якого типу, то за умови необхідності пошуку ключа, що складається з 56 бітів, що створює пошуковий простір у вигляді 2^{56} елементів, для зберігання значень цих бітів необхідно буде дуже багато пам’яті. Розглянувши алгоритм Гровера, можна помітити, що під час роботи методу ми маємо 2 значення, які змінюються протягом всієї роботи методу та впливають на результат: значення амплітуди потрібного елементу та значення амплітуди всіх інших елементів. Таким чином, можна запобігти складнощів з кількістю необхідної для представлення регістру пам’яті. Програма запам’ятовує індекс необхідного елемента та його амплітуду, а також амплітуду всіх інших елементів. На початку роботи програми значення наведених змінних встановлюються в 0. Таким чином, забезпечується виконання першого пункту.

Після цього необхідно встановити регістр в стан суперпозиції станів. На квантовому комп’ютері для цього використовуються квантові гейти Адамара. На класичному комп’ютері програма тримає в пам’яті амплітуди кубітів регістру. Саме амплітуди кубітів змінюються протягом всього використання методу Гровера. Таким чином, потрібно визначити, якими повинні бути амплітуди кубітів в стані суперпозиції та записати їх значення у відповідні змінні.

У випадку з DES та його 56-бітним ключем амплітуди всіх кубітів регістру приймуть значення $\frac{1}{\sqrt{2^{56}}}$ що й записується в обидві змінні, що відповідають за

збереження значень амплітуд кубітів. Таким чином, забезпечується виконання другого пункту.

Далі йде застосування ітерації Гровера, що у свою чергу складається із застосування оракула та оператора дифузії. Ітерація Гровера призначена для збільшення амплітуди потрібного елемента та зменшення амплітуди всіх інших елементів, що збільшує ймовірність отримання потрібного результату під час вимірювання.

Оракул є найскладнішою частиною методу, особливо для реалізації на класичному комп'ютері. У випадку з квантовим комп'ютером оракул змінює систему без зведення до класичного стану. При цьому виконується операція, що змінює знак амплітуди потрібного елемента на протилежний. Для того, щоб виявити потрібний елемент (в даному випадку - ключ), програма перебирає ключі, намагаючись розшифрувати з його допомогою зашифрований за допомогою DES текст та вже потім міняти знак його амплітуди на протилежний, що збільшує різницю між амплітудами потрібного елемента та інших. Слід зауважити, що без можливості здійснювати огляд всього регістру одночасно, як це може робити квантовий оракул, одне застосування оракула програмної моделі займає стільки ж часу, скільки й простий злам методом «грубої сили» (так як оракул, по своїй суті, його й реалізує). Таким чином, можна побачити, що в рамках квантового комп'ютера такі речі, як пошук елемента методом «грубої сили», можуть реалізовуватися одним оператором (виконання квантового оракула вважається за один оператор).

Для цілей тестування та налагоджування механізмів роботи було розроблено версію програми, що виконує пошук потрібного елемента лише один раз під час ініціалізації, а далі просто змінює значення амплітуди відповідно до вимог методу. Також, як вже було зазначено, була розроблена спеціальна версія програмної моделі, призначена для проведення розрахунків до практичних прикладів, наведених в підрозділі 12.5. В загальному вигляді вона відповідає наведеним програмним моделям, але відрізняється тим, що оракул було спрощено до елементарної функції пошуку, що також проводився повністю лише в частині, відповідальній за ініціалізацію. Така побудова моделі дозволила провести розрахунки без вирішення складної задачі пошуку, але зі збереженням достовірності результатів розрахунків на рівні, що відповідає рівню достовірності результатів програмної моделі, що проводить виконання складної задачі пошуку.

Наступною частиною ітерації Гровера є оператор дифузії. Він здійснює інверсію щодо середнього значення, зменшуючи таким чином амплітуди всіх елементів, окрім потрібного. Таким чином, оператор дифузії також збільшує ймовірність отримання в результаті вимірювання регістра потрібного елемента. В програмній моделі для класичного комп'ютера оператор дифузії існує як послідовність математичних операцій, що певним чином впливають на амплітуду тих елементів пошукового простору, що не задовольняють вимог пошуку.

Слід зауважити, що метод Гровера є ймовірнісним, тобто, потрібний результат при вимірюванні буде отриманий з певною ймовірністю. Для

збільшення ймовірності отримання правильного елемента і проводиться певна кількість ітерацій Гровера. В програмній моделі криптоаналізу для класичного комп'ютера виконується $R \approx \frac{\pi}{4}\sqrt{2^n}$ ітерацій Гровера у зв'язку із зазначеними вище причинами. Таким чином, забезпечується виконання третього пункту.

Останнім відбувається вимірювання стану регістра для визначення результату пошуку. У випадку з квантовим комп'ютером відбуваються класичні вимірювання. В програмній моделі достатньо вивести індекс ключа, тобто, вивести ключ, його амплітуду, котра визначає ймовірність отримати в результаті вимірювання саме його. Таким чином, забезпечується виконання четвертого пункту.

В результаті практичного застосування програми виявлено, що вона схильна виконувати дещо більшу кількість ітерацій Гровера, ніж потрібно для оптимального використання методу Гровера. Проте кількість зайвих ітерацій Гровера становить близько 500 ітерацій, що у порівнянні з потрібною кількістю ітерацій є дуже незначною похибкою, що була викликана використанням у розрахунках числа π.

Якщо детально розглянути Оракул та здійснити аналіз його реалізації та всіх його реалізацій, можливих для здійснення на класичному комп'ютері, можна зробити висновок про відсутність переваг перед класичними методами пошуку, особливо, з точки зору часу виконання. Перевага ж у кількості виконаних елементарних операторів не є такою на класичному комп'ютері, де квантовий Оракул, що на квантовому комп'ютері розглядається в якості одного елементарного оператора, насправді є сукупністю елементарних операторів, що своєю кількістю відповідає повному класичному алгоритму проведення пошуку.

Квантовий Оракул, перш за все, необхідний для зміни амплітуди елементів пошукового простору, при цьому зміна відбувається для всіх елементів одночасно. При реалізації на квантовому комп'ютері це не є проблемою через підтримку квантовим комп'ютером квантових властивостей. На класичному ж комп'ютері ми не маємо можливості «дивитися» одночасно на всі елементи, що входять до регістру, що не дозволяє виконувати квантовий Оракул на класичному комп'ютері з тією ж складністю, що й на квантовому комп'ютері.

Таким чином, на класичному комп'ютері, перебираючи елементи по одному, ми досягаємо складності класичного методу «грубої сили» для лише одного застосування Оракула. Це призводить до того, що з використанням $\frac{\pi}{4}\sqrt{N}$ Оракулів ми здійснюємо $\frac{\pi}{4}\sqrt{N}$ виконань пошуку методом «грубої сили». Так можна зробити однозначний висновок про недоцільність виконання пошуку квантовим методом Гровера на класичному комп'ютері через невигідність його використання в часовому розрізі.

12.8 Аналіз можливостей та особливості програмування задач криптології на квантовому комп'ютері

Нині обґрунтовано вважається, що застосування квантового комп'ютера при вирішенні задач криптоаналізу щодо існуючих криптоперетворень

асиметричного типу, скоріше всього, приведе до їх зламу. Але для цього ще потрібно розробити та виготовити квантові комп'ютери відповідної кубічної розрядності та розробити відповідне математичне та програмне забезпечення. Тому питання програмування задач криптології взагалі, особливо криптоаналізу з орієнтацією на квантовий комп'ютер, стає все більш актуальним. Такі можливості появились завдяки успіхам у справі створення робочого варіанту екземпляру квантового комп'ютера [61-63]. Але виявилося, що як математичне, так і програмне забезпечення квантового комп'ютера є суттєво специфічними і не вкладаються в наші звичні рамки. В той же час, необхідність вирішення задач криптоаналізу на квантовому комп'ютері пояснюється його суттєвою перевагою перед класичним у швидкодії. Така перевага здебільшого базується на використанні квантовими комп'ютерами квантових властивостей, що є недоступними для класичних комп'ютерів. Наприклад, завдяки квантовим властивостям існує можливість розглядати одразу весь регістр замість одного елемента, як це робиться в класичному комп'ютері. Вказане дає суттєву перевагу в швидкодії обчислень.

Наразі для загального доступу з використанням хмарних сервісів доступними є квантові комп'ютери компанії IBM на 5 та 14 кубітів та квантового симулятора (до 32 кубітів). Вони виконують квантові операції та можуть реалізовувати квантові гейти. Слід помітити, що ці гейти однокубітні та двохкубітні, а один з них трьохкубітний. Робоча реалізація методу Гровера, в свою чергу, потребує наявності квантових гейтів, що діяли б на більшу кількість кубітів одночасно. Тому багатокубітні гейти доводиться розкладати на послідовності гейтів, що є в наявності серед універсальних. Враховуючи вказане, необхідно виробити три схеми застосування методу Гровера на квантовому комп'ютері для квантового регістру з чотирьох кубітів для отримання квантового регістру.

Наразі актуальними є наступні задачі: обгрунтування підходів до аналізу можливостей та вивчення особливостей програмування задач криптоаналізу на квантових комп'ютерах, а також оцінки сучасного стану можливостей його реалізації на однокубітних, двохкубітних та трьохкубітному гейтах.

12.9 Особливості та можливості квантового комп'ютера

Квантові комп'ютери, як показує аналіз, суттєво відрізняються від класичних безпосередньо самими принципами роботи. В них використовується специфічна квантова інформатика. Вона цілком базується на властивостях квантових об'єктів [61-63]. В першу чергу, на здатності квантової частки приймати одночасно декілька станів завдяки квантовій суперпозиції станів. Також суттєвими є здатність систем, що складаються з декількох квантових часток, перебувати в переплутаних (корельованих) станах та нелокальність, що є їх наслідком. Важливим є вплив процесу вимірювання на стан вимірюваного об'єкта аж до його знищення, а також практична неможливість здійснити клонування квантових станів квантових часток.

Наведені вище властивості безпосередньо слідують з властивостей квантової фізики, вони є підґрунтям для квантової інформатики. Причому квантова фізика є складною в розумінні та відрізняється від класичної на принциповому рівні.

На основі квантових властивостей квантових об'єктів розроблено специфічну квантову математику, яку покладено в основу квантових алгоритмів криптоаналізу. Основними з них є [60]:

- алгоритм Шора для здійснення факторизації надвеликих цілих чисел;
- алгоритми розв'язку задач дискретного логарифму в скінченному полі;
- алгоритми дискретного логарифмування в групі точок еліптичних кривих;
- алгоритм пошуку в несортованій базі (Гровера);
- алгоритми криптоаналізу перетворень в кільцях поліномів та фактор-кільцях тощо.

Як правило, вказані алгоритми забезпечують вирішення задач, що є або неможливими для вирішення за допомогою класичних комп'ютерів, або нерентабельно складними з точки зору обчислювального ресурсу, що необхідний для отримання практичного результату [60-63]. Однією з специфічних властивостей квантових алгоритмів є те, що вони мають ймовірнісну природу, тобто, результат можна отримати тільки з певною ймовірністю. Тобто, проблемність вирішення задачі криптоаналізу в необхідності отримання результату з необхідною ймовірністю при вимірюванні стану квантового регістра після завершення роботи алгоритму.

Як теоретично, так і практично підтверджено, що застосування квантових алгоритмів на класичному комп'ютері не тільки не дає переваги, а є дуже невигідним з точки зору швидкодії [59]. Так, наприклад, реалізація методу Гровера на класичному комп'ютері є невигідною через те, що сам алгоритм передбачає для отримання потрібного результату при вимірюванні стану квантового регістру багаторазового повторення ітерації Гровера. Тоді як на квантовому комп'ютері повтори покращують результат, а квантові властивості нівелюють затрати на повтори. В той час, як на класичному комп'ютері ці повтори не є потрібними, а лише зайвими. Щодо властивостей, що нівелюють для квантового комп'ютера проведення повторів, то вони пояснюються тим, що квантовий комп'ютер дозволяє переглядати весь результат в регістрі всього однією операцією, в той час, як на класичному комп'ютері перегляд регістру здійснюється по одному елементу за одну операцію.

Необхідно також зазначити, що поява квантового комп'ютера, регістр якого матиме для здійснення квантового криптоаналізу достатню кількість кубітів, є суттєво необхідною умовою зламу існуючих асиметричних криптоалгоритмів з обмеженими розмірами системних параметрів. Достатність умови в тому, що, окрім власне квантового комп'ютера, необхідним ще є наявність відповідного математичного та програмного забезпечення, а також ще й реалізації на квантовому комп'ютері безпосередньо алгоритмів, з використанням яких можна провести криптоаналіз. Хоча, без сумнівів, створення квантового комп'ютера призведе до появи значної загрози для

сучасних криптографічних систем з боку вже розроблених квантових алгоритмів, таких як алгоритми Гровера та Шора тощо.

12.10 Аналіз можливостей та наявності забезпечень для вирішення задач криптоаналізу

Таким чином, завдяки сучасним успіхам в створенні квантового комп'ютера, існує можливість доступу до програмування квантового комп'ютера за допомогою хмарних сервісів. Так, наразі для загального доступу з використанням хмарних сервісів доступними є квантові комп'ютери компанії IBM на 1, 5 та 15 кубітів [62]. Вони виконують квантові операції та можуть реалізовувати квантові гейти. Також можна скористатися квантовим симулятором (до 32 кубітів) [62].

Практичний аналіз показав, що загальнодоступними для використання є однокубітні, двохкубітні та один з них трьохкубітний гейти. Але робочі реалізації, наприклад, методу Гровера потребують наявності квантових гейтів, що діяли б на більшу кількість кубітів одночасно. Тому багатокубітні гейти доводиться розкладати на послідовності гейтів, що є в наявності, тобто, універсальних.

Наш пошук показав [62], що серед доступних для загального доступу квантових комп'ютерів наявні: $ibmq\_16\_melbourne$ (15 кубітів), $ibmq\_london$ (5 кубітів), $ibmq\_burlington$ (5 кубітів), $ibmq\_essex$ (5 кубітів), $ibmq\_ourense$ (5 кубітів), $ibmq\_vigo$ (5 кубітів), $ibmq\_5\_yorktown-ibmqx2$ (5 кубітів), $ibmq\_armonk$ (1 кубіт). Квантовий симулятор $ibmq\_qasm\_simulator$ може використовуватись для реалізації алгоритмів, що передбачають використання регістрів з довжиною до 32 кубітів.

12.11 Особливість квантового програмування для методу Гровера

Одним з основних квантових методів, що є необхідними для вирішення задач криптології, є метод Гровера. Розглянемо його детальніше.

Алгоритм Гровера будується з використанням методу Гровера, він є квантовим алгоритмом, що призначений для проведення вичерпного пошуку унікального елемента в несортованій базі даних, що містить $N=2^n$ елементів, де n позначає довжину задіяного для представлення пошукового простору квантового регістра (кількість кубітів в ньому), а N є розміром пошукового простору [57,59].

Особливість алгоритму Гровера полягає в тому, що, завдяки квантовим властивостям та використанню функції «чорної скриньки» (у вигляді квантового оракула), він потребує лише $O\left(\sqrt{N}\right)$ групових операцій замість O (N) у класичних алгоритмів. Квадратичне прискорення, у порівнянні з класичними алгоритмами, досягається завдяки використанню квантових властивостей таких, як квантова суперпозиція станів.

Хоча інші квантові алгоритми при порівнянні з класичними аналогами можуть забезпечити експоненційне прискорення, а алгоритм Гровера може забезпечити лише квадратичне прискорення, слід зауважити, що навіть таке прискорення є дуже значним та його значущість збільшується зі зростанням $\boldsymbol{N}$. Для прикладу, методом Гровера 128-бітний криптографічний ключ можна зламати приблизно за 2^{64} звернень до функції «чорної скриньки», що можна визначити як 2^{64} звернень до ітерації Гровера, а отже 2^{64} ітерацій методу. В той же час як 256-бітний криптографічний ключ можна зламати за, приблизно, 2^{128} ітерацій. Виходячи саме з цього твердження, для збільшення стійкості проти квантових атак іноді пропонують збільшувати довжину криптографічних ключів в 2 рази [63].

Метод Гровера, як і більшість квантових методів, є ймовірнісним, тобто, правильна відповідь може бути виміряна з квантового регістру з певною ймовірністю, яка не повинна перевищувати 1. Також слід зауважити, що при виконанні більшого числа ітерацій, чим потрібно, ймовірність вимірювання правильного результату зменшується, тому це потрібно відповідним чином відслідковувати [59].

Метод Гровера має ряд можливостей для застосування, одним з яких є реалізація його як алгоритму криптоаналізу симетричних перетворень, функцій гешування, асиметричного шифрування в кільці поліномів тощо у зв'язку з його можливим узагальненим використанням [59,62]. Для криптоаналізу симетричних блокових перетворень метод можна звести до алгоритму пошуку сеансового чи довгострокового ключа тощо. У випадку функцій гешування метод можна застосувати для пошуку колізій тощо. Метод Гровера має значні потенційні можливості, які беруться в розрахунок з огляду на сучасний стан розробки квантового комп'ютера.

12.11.1 Квантовий Оракул

Метод Гровера передбачає використання оператора під назвою «квантовий оракул» O, що вбирає в себе найбільшу частину квантових властивостей з тих, що використовуються в алгоритмі Гровера. Оракул O розпізнає вирішення задачі пошуку та змінює їх знак на протилежний.

У зв'язку з тим, що кожна конкретна задача пошуку потребує окремої реалізації квантового оракула O, теоретично його дія розглядається, як функція «чорної скриньки». Щодо того, як здійснюється інверсія знаку, то оракул O містить в собі функцію f таку, що $f(x) = 1$, якщо стан $|x\rangle$ відповідає рішенню задачі пошуку, та $f(x) = 0$ в усіх інших випадках. Оракул O - унітарний оператор, що виконується з використанням додаткового кубіта [64]. В такому випадку:

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle,$$

де $|q\rangle$ позначає додатковий кубіт для оракула. Таким чином, у випадку, якщо $f(x) = 1$, додатковий кубіт інвертується та залишається незмінним в усіх інших випадках. Метод Гровера здійснює інверсію знаку потрібного стану $|x\rangle$, що може досягатися встановленням додаткового кубіту, призначеного для оракула, в стан

$(|0\rangle - |1\rangle)/\sqrt{2}$, що перетворюється в $(|1\rangle - |0\rangle)/\sqrt{2}$ у випадку, якщо стан $|x\rangle$ є рішенням задачі пошуку [65]. Якщо $|x\rangle$ вважається інвертованим, то додатковий кубіт для оракула не підлягає змінам, через що, за домовленістю, їх зазвичай не згадують під час опису. Таким чином, дія оракула O на систему в методі Гровера записується у вигляді [64]:

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle.$$

Зазвичай метод Гровера розглядає несортовану базу даних, що складається з N елементів, що може бути представлено $n = log_2 N$ кубітів. Функція f діє на систему аналогічно до попереднього випадку. В такому випадку унітарний оператор квантового оракула O постає у наступному вигляді [58]:

$$\begin{cases} O|x\rangle = -|x\rangle & \text{для } f(x) = 1, \\ O|x\rangle = |x\rangle & \text{для } f(x) = 0. \end{cases}$$

12.11.2 Реалізація методу Гровера на квантовому комп'ютері

Наразі для загального доступу з використанням хмарних сервісів доступними є квантові комп'ютери компанії IBM на 5 та 14 кубітів та квантового симулятора (до 32 кубітів). Вони виконують квантові властивості та можуть реалізовувати квантові гейти. Серед таких гейтів є:

- H - гейт (гейт Адамара, змінює стани $|0\rangle$ та $|1\rangle$ на стани $|+\rangle$ та $|-\rangle$ відповідно);
- cX - гейт $\oplus$, (контрольований X, контрольоване HI, змінює цільовий кубіт за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
- Id - гейт (позначає відсутність гейтів в одиницю часу);
- $U3$ - дозволяє утворити за допомогою трьох параметрів будь-який однокубітний гейт;
- $U2$ - за допомогою двох параметрів здійснюється два різних повороти всередині гейта за один проміжок часу;
- $U1$ - рівнозначно з гейтом Rz та використовується контролюючим ПЗ;
- Rx - поворот навколо осі x на сфері Блоха на заданий кут;
- Ry - поворот навколо осі y на сфері Блоха на заданий кут;
- Rz - поворот навколо осі z на сфері Блоха на заданий кут;
- X - гейт Паулі (змінює стан $|1\rangle$ на стан $|0\rangle$ та навпаки, також рівнозначно до гейта Rx за кутом π);
- Y - гейт Паулі (рівнозначно до гейта Ry за кутом π);
- Z - гейт Паулі (змінює стан $|+\rangle$ на стан $|-\rangle$ та навпаки, також рівнозначно до гейта Rz за кутом π);
- S - *гейт* $Clifford$ *(*рівнозначно до гейта Rz за кутом $\frac{\pi}{2}$*);*
- $S^{\dagger}$ - гейт $Clifford$ (рівнозначно до гейта Rz за кутом $-\frac{\pi}{2}$);
- T - рівнозначно до гейта Rz за кутом $\frac{\pi}{4}$;
- $T^{\dagger}$ - рівнозначно до гейта Rz за кутом $-\frac{\pi}{4}$;

-cH - контрольований гейт Адамара (виконує гейт Адамара на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-cY - контрольований гейт Y (виконує гейт Y на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-cZ - контрольований гейт Z (виконує гейт Z на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-cRz - контрольований гейт Rz (виконує гейт Rz на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-$cU1$ - контрольований гейт $U1$ (виконує гейт $U1$ на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-$cU3$ - контрольований гейт $U3$ (виконує гейт $U3$ на цільовому кубіті за умови, що управляючий кубіт знаходиться в стані $|1\rangle$);
-ccX - $Toffoli$ (діє подібно до гейта cX, але має два управляючих кубіти та впливає на цільовий кубіт лише за умови знаходження обох контролюючих кубітів в стані $|1\rangle$);
-$SWAP$ - (гейт, призначений для того, щоб здійснити обмін станами між двома кубітами);
-$|0\rangle$ - (перетворює будь-який стан кубіту на стан $|0\rangle$);
-IF - (дозволяє застосовувати гейти за певних умов, базується на стані класичного регістру, що існує паралельно з квантовим в цьому сервісі);
-$z-$measurement (проводить виміри на базисі z, інші види вимірів виконуються застосуванням цього гейта разом з іншими гейтами).

Також слід помітити, що ці гейти однокубітні та двохкубітні та один з них трьохкубітний. Робоча реалізація методу Гровера, в свою чергу, потребує наявності квантових гейтів, що діяли б на більшу кількість кубітів одночасно. Тому багатокубітні гейти доводиться розкладати на послідовності гейтів, що є в наявності (універсальних).

У зв'язку з тим, що, згідно з [64], оператор дифузії позначається як $2|\psi\rangle\langle\psi| - I$ та являє собою $H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$, окремо слід розглянути лише оператор $2|\psi\rangle\langle\psi| - I$. Його реалізація в загальному вигляді представлена на рис. 12.15.

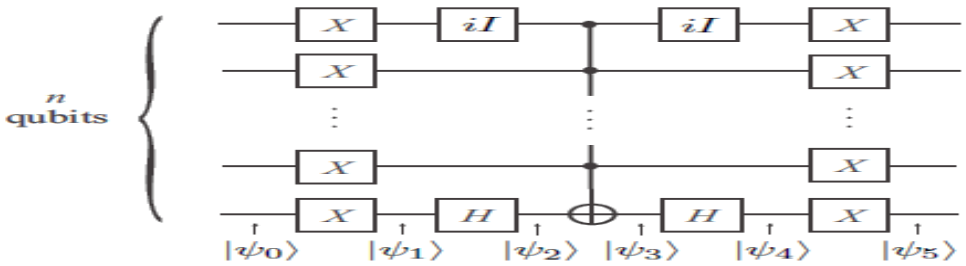


Рис. 12.15. Схема реалізації оператора $2|\psi\rangle\langle\psi| - I$ за допомогою елементарних гейтів

Також для реалізації алгоритму Гровера для, наприклад, 4-кубітного регістра необхідним є застосування відсутніх гейтів, серед яких можна відмітити

контрольований T гейт (cT) та потрійно-контрольований Z гейт ($cccZ$), реалізацію яких було наведено в [66]. Так, контрольований T гейт (cT) можна відтворити у вигляді $cU1\left(\frac{\pi}{4}\right)$, відповідно з вищенаведеним визначенням гейта T. В свою чергу, реалізація потрійно-контрольованого Z гейта ($cccZ$) базується на декількох фактах та припущеннях. Враховуючи, що гейт Z можна представити у вигляді T^4 та реалізацію потрійно-контрольованого гейту $U = V^4$ [66], що включає в себе серію гейтів cX та контрольованих гейтів V, потрійно-контрольований Z гейт ($cccZ$) може бути реалізовано наведеним на рис. 12.16 чином [66].

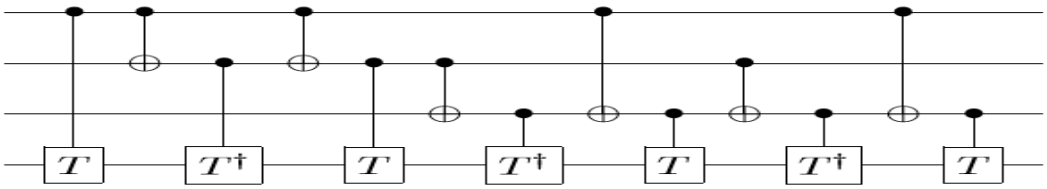


Рис. 12.16. Реалізація потрійно-контрольованого Z гейта ($cccZ$) через гейти cT та $cT^{\dagger}$

Оракул та оператор дифузії реалізуються згідно наведених на рис. 12.17 та рис. 12.18 схем відповідно [66].

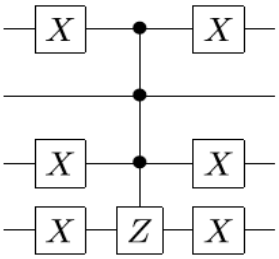


Рис. 12.17. Квантовий оракул для стану регістру $|0010\rangle$

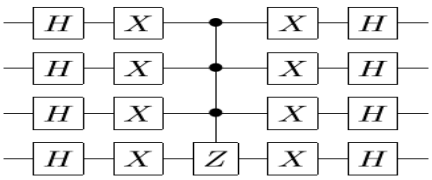


Рис. 12.18. Оператор дифузії для чотирьох кубітів

Враховуючи все наведене вище, було вироблено три схеми застосування методу Гровера на квантовому комп'ютері для отримання результату «0», що може бути представлено рядком бітів $|0000\rangle$, та три схеми для отримання результату 4, що може бути представлено рядком бітів $|0100\rangle$. Схеми відрізняються кількістю застосування ітерацій Гровера. Так, в перших схемах застосовується одна ітерація Гровера, в других – дві ітерації, в третіх – три

ітерації. Ці схеми було випробувано на доступних для загального доступу тестових квантових комп'ютерах компанії IBM та доступному через той же сервіс квантовому симуляторі. Серед квантових комп'ютерів, використаних в дослідженні, були: $ibmq\_burlington$ (5 кубітів), $ibmq\_16\_melbourne$ (14 кубітів), $ibmq\_5\_yorktown-ibmqx2$ (5 кубітів). Квантовий симулятор $ibmq\_qasm\_simulator$ може розраховувати схеми, що передбачають використання до 32 кубітів.

12.12 Схеми застосування методу Гровера на квантовому комп'ютері для квантових регістрів з використанням хмарних сервісів

Наразі для загального доступу з використанням хмарних сервісів доступними є квантові комп'ютери компанії IBM на 5 та 15 кубітів та квантового симулятора (до 32 кубітів). Вони виконують квантові операції та можуть реалізовувати квантові гейти. Слід відмітити, що ці гейти однокубітні та двохкубітні, а один з них трьохкубітний. Робоча реалізація методу Гровера, в свою чергу, вимагає наявності квантових гейтів, що діяли б на більшу кількість кубітів одночасно. Тому багатокубітні гейти доводиться розкладати на послідовності гейтів, що є в наявності серед універсальних. Як вже було зазначено, особливості реалізації методу Гровера для квантового регістру, що складається з чотирьох кубітів на квантовому комп'ютері, включають в себе необхідність застосування 3-кубітних квантових гейтів. В той же час інструментарій для взаємодії з доступними квантовими комп'ютерами не включає в себе квантові гейти, що оперують над необхідною кількістю кубітів. Необхідні квантові гейти можна замінити сукупностями наявних в інструментарії квантових гейтів, що дають той самий ефект.

Враховуючи наведене вище, розглянемо три схеми застосування методу Гровера на квантовому комп'ютері для квантового регістру з чотирьох кубітів для отримання результату «0», що може бути представлено рядком бітів $|0000\rangle$, та три схеми для отримання результату 4, що може бути представлено рядком бітів $|0100\rangle$. Схеми відрізняються кількістю застосувань ітерацій Гровера. Так, в перших схемах застосовується одна ітерація Гровера, в других – дві ітерації, в третіх – три ітерації. Ці схеми було випробувано на доступних для загального доступу тестових квантових комп'ютерах компанії IBM та доступному через той же сервіс квантовому симуляторі. Серед квантових комп'ютерів, що використані в досліджені, були: $ibmq\_16\_melbourne$, $ibmqx2$ (він же $ibmq\_5\_yorktown-ibmqx2$), $ibmq\_burlington$. Квантовий симулятор $ibmq\_qasm\_simulator$ може розраховувати схеми, що передбачають використання до 32 кубітів.

Результати виконаних випробувань можуть вказувати як на недосконаліть методів, використаних для представлення багатокубітних гейтів у вигляді сукупності одно- та двокубітних гейтів, так і на недосконаліть розроблених квантових комп'ютерів, що підлягає перевірці в майбутніх дослідженнях.

Результати застосування методу Гровера з однією ітерацією для отримання $|0000\rangle$ показали значні розбіжності між реальними результатами та отриманими за допомогою квантового симулятора. Так, симулятор вказував на те, що ймовірність отримання $|0000\rangle$ становить 46,875%, в той час як при проведенні реальних вимірювань цей результат було отримано лише 6,152% разів на $ibmq\_burlington$, 7,52% - на $ibmqx2$ та 8,789% - на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з двома ітераціями для отримання $|0000\rangle$ показали ще більші розбіжності між очікуваними та реальними результатами. Так, на симуляторі ймовірність отримання $|0000\rangle$ становила 91,211%, в той час як насправді було отримано лише 6,543% на $ibmq\_burlington$, 8,398% на $ibmqx2$ та 10,156% на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з трьома ітераціями (що є остаточною кількістю ітерацій для регістру цього розміру) для отримання $|0000\rangle$ підтвердили розбіжності, отримані на попередньому кроці. Так, симулятор спрогнозував ймовірність отримання $|0000\rangle$ в 96,387%, в той час як на реальних квантових комп'ютерах було отримано потрібний результат за значно меншу кількість разів: 7,227% - на $ibmq\_burlington$, 8,301% - на $ibmqx2$, 9,961% - на $ibmq\_16\_melbourne$ квантовому комп'ютері.

Результати застосування методу Гровера з однією ітерацією для отримання $|0100\rangle$ показали значні розбіжності між реальними результатами та отриманими за допомогою квантового симулятора. Так, симулятор вказував на те, що ймовірність отримання $|0100\rangle$ становить 46,387%, в той час як при проведенні реальних вимірювань цей результат було отримано лише 5,469% разів на $ibmq\_burlington$, 10,547% - на $ibmqx2$ та 6,506% - на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з двома ітераціями для отримання $|0100\rangle$ показали ще більші розбіжності між очікуваними та реальними результатами. Так, на симуляторі ймовірність отримання $|0100\rangle$ становила 90,259%, в той час як насправді було отримано лише 6,006% на $ibmq\_burlington$, 6,982% на $ibmqx2$ та 6,897% на $ibmq\_16\_melbourne$ квантових комп'ютерах.

Результати застосування методу Гровера з трьома ітераціями для отримання $|0100\rangle$ підтвердили розбіжності, отримані на попередньому кроці. Так, симулятор спрогнозував ймовірність отримання $|0100\rangle$ в 96,436%, в той час як на реальних квантових комп'ютерах було отримано потрібний результат за значно меншу кількість разів: 6,299% - на Бурлінгтонському, 6,152% - на Йорктаунському, 7,837% на Мельбурнському квантових комп'ютерах. Детальніше результати випробувань розроблених схем на квантових комп'ютерах наведено в таблицях 12.3 – 12.8.

Таблиця 12.3. Результати застосування методу Гровера з однією ітерацією для отримання $|0000\rangle$

	Симулятор (1024 повтори)	Бурлінгтон (1024 повтори)	Йорктаун (1024 повтори)	Мельбурн (1024 повтори)
0000	46,875%	6,152%	7,52%	8,789%
0001	4,395%	6,543%	6,934%	6,25%
0010	2,832%	6,738%	5,566%	7,031%
0011	3,809%	5,664%	8,691%	5,664%
0100	3,613%	8,594%	6,445%	5,566%
0101	2,637%	5,957%	4,688%	6,836%
0110	3,125%	6,543%	4,004%	4,98%
0111	3,906%	5,176%	10,742%	6,348%
1000	2,539%	7,422%	7,422%	7,52%
1001	4,199%	6,738%	4,688%	7,031%
1010	2,832%	7,031%	4,59%	6,641%
1011	2,637%	5,371%	7,715%	5,957%
1100	5,566%	6,934%	4,199%	5,273%
1101	3,809%	5,762%	4,492%	5,957%
1110	3,223%	3,906%	4,883%	5,566%
1111	4,004%	5,469%	7,422%	4,59%

Таблиця 12.4. Результати застосування методу Гровера з двома ітераціями для отримання $|0000\rangle$

	Симулятор (1024 повтори)	Бурлінгтон (1024 повтори)	Йорктаун (1024 повтори)	Мельбурн (1024 повтори)
0000	91,211%	6,543%	8,398%	10,156%
0001	0,781%	6,25%	6,152%	6,641%
0010	0,391%	6,25%	5,371%	6,348%
0011	0,391%	8,398%	8,008%	5,566%
0100	0,684%	5,762%	8,594%	7,227%
0101	0,586%	6,641%	5,176%	7,422%
0110	0,488%	6,152%	4,98%	5,762%
0111	0,586%	7,324%	7,813%	4,395%
1000	0,195%	6,445%	6,348%	6,738%
1001	0,586%	6,152%	4,004%	6,641%
1010	0,684%	5,273%	4,492%	5,273%
1011	0,586%	5,957%	7,715%	4,102%
1100	0,684%	5,273%	6,738%	7,422%
1101	0,879%	6,152%	3,906%	5,664%
1110	0,488%	5,566%	5,664%	6,348%
1111	0,781%	5,859%	6,641%	4,297%

Таблиця 12.5. Результати застосування методу Гровера з трьома ітераціями для отримання |0000⟩

	Симулятор (1024 повтори)	Бурлінгтон (1024 повтори)	Йорктаун (1024 повтори)	Мельбурн (1024 повтори)
0000	96,387%	7,227%	8,301%	9.961%
0001	0,391%	7,91%	2,93%	5.371%
0010	0,098%	6,836%	6,641%	8.496%
0011	0,586%	6,836%	6,641%	5.664%
0100	0,195%	5,664%	7,715%	5.859%
0101	0,098%	6,738%	3,613%	6.641%
0110	0,098%	6,543%	7,031%	5.762%
0111	0,195%	6,055%	7,422%	6.348%
1000	0,195%	5,859%	8,594%	10.742%
1001	0,293%	4,98%	3,613%	4.59%
1100	0,098%	5,566%	7,227%	5.859%
1101	0,195%	5,957%	3,223%	5.469%
1110	0,391%	5,078%	6,445%	4.395%
1111	0,293%	5,762%	6,934%	4.59%

Таблиця 12.6. Результати застосування методу Гровера з однією ітерацією для отримання |0100⟩

	Симулятор (4096 повторів)	Бурлінгтон (4096 повторів)	Йорктаун (4096 повторів)
0000	3,589%	8,691%	7,666
0001	3,247%	6,909%	6,348
0010	4,053%	5,859%	5,811
0011	3,003%	5,151%	4,541
0100	46,387%	5,469%	10,547
0101	3,125%	5,322%	8,398
0110	3,076%	5,957%	7,666
0111	3,906%	6,567%	7,153
1000	3,955%	9,839%	3,687
1001	4,321%	6,47%	6,274
1010	3,516%	6,03%	2,881
1011	3,198%	5,151%	6,396
1100	3,296%	5,859%	4,321
1101	3,54%	5,005%	8,252
1110	4,175%	5,591%	3,296
1111	3,613%	6,128%	6,763

Таблиця 12.7. Результати застосування методу Гровера з двома ітераціями для отримання |0100⟩

	Симулятор (4096 повтори)	Бурлінгтон (4096 повтори)	Йорктаун (4096 повтори)
0000	0,537%	7,861%	6,738%
0001	0,537%	5,981%	6,982%
0010	0,635%	8,179%	3,711%
0011	0,659%	6,934%	3,54%
0100	90,259%	6,006%	6,982%
0101	0,464%	5,64%	7,495%
0110	0,684%	6,616%	4,175%
0111	0,586%	5,981%	3,833%
1000	0,781%	7,3%	5,688%
1001	0,61%	5,884%	5,908%
1010	0,659%	5,688%	8,862%
1011	0,732%	6,274%	7,446%
1100	0,659%	5,493%	6,03%
1101	0,708%	5,566%	6,03%
1110	0,757%	5,396%	8,423%
1111	0,732%	5,2%	8,154%

Таблиця 12.8. Результати застосування методу Гровера з трьома ітераціями для отримання |0100⟩

	Симулятор (4096 повтори)	Бурлінгтон (4096 повтори)	Йорктаун (4096 повтори)
0000	0,22%	7,52%	6,299%
0001	0,244%	6,152%	6,177%
0010	0,244%	6,519%	3,784%
0011	0,122%	5,859%	9,131%
0100	96,436%	6,299%	6,152%
0101	0,244%	6,543%	6,323%
0110	0,22%	6,152%	3,467%
0111	0,171%	5,249%	7,91%
1000	0,391%	6,738%	5,762%
1001	0,244%	6,177%	5,957%
1010	0,269%	5,615%	3,711%
1011	0,22%	6,909%	8,862%
1100	0,195%	6,641%	6,738%
1101	0,269%	6,47%	6,055%
1110	0,269%	5,615%	3,76%
1111	0,244%	5,542%	9,912%

Наведемо основні висновки застосування методу Гровера на квантовому комп'ютері для квантових регістрів з використанням хмарних сервісів

1. Подальший розвиток вимагає вдосконалення представлення багатокубітних гейтів шляхом використання одно- та двокубітних гейтів, а також вдосконалення оснастки для роботи з квантовими комп'ютерами та розробки багатокубітних гейтів та впровадження їх у діючі зразки квантових комп'ютерів.

2. Хоча квантовий симулятор вказує на те, що схеми повинні надавати правильний результат із ймовірністю, близькою до максимальної, результати реальних експериментів не є навіть близько такими вдалими.

3. Реалізація методу Гровера на класичному комп'ютері є невигідною через те, що сам алгоритм передбачає для отримання потрібного результату при вимірюванні стану квантового регістру багаторазового повторення алгоритму Гровера. Тоді, як на квантовому комп'ютері повтори покращують результат, а квантові властивості нівелюють затрати на повтори. В той час, як на класичному комп'ютері ці повтори не є потрібними, а лише зайвими.

4. Щодо властивостей, що нівелюють для квантового комп'ютера проведення повторів, то вони пояснюються тим, що квантовий комп'ютер дозволяє переглядати весь результат в регістрі всього однією операцією, в той час, як на класичному комп'ютері перегляд регістру здійснюється по одному елементу за одну операцію.

5. Необхідно також зазначити, що поява квантового комп'ютера, регістр якого матиме для здійснення квантового криптоаналізу достатню кількість кубітів, є суттєво необхідною умовою зламу існуючих асиметричних криптоалгоритмів з обмеженими розмірами системних параметрів.

6. Достатність умови в тому, що, окрім власне квантового комп'ютера, необхідним ще є наявність відповідного математичного та програмного забезпечення, а також ще й реалізації на квантовому комп'ютері безпосередньо алгоритмів, з використанням яких можна провести криптоаналіз.

7. Можна стверджувати, що створення квантового комп'ютера призведе до появи значної загрози для сучасних криптографічних систем з боку вже розроблених квантових алгоритмів, таких, як алгоритми Гровера та Шора тощо.

8. Питання програмування задач криптології взагалі, особливо криптоаналізу, з орієнтацією на квантовий комп'ютер стає все більш актуальним. Такі можливості появились завдяки успіхам у справі створення робочого варіанту екземпляру квантового комп'ютера.

9. З метою демонстрації дії методу Гровера на практиці розглядаються приклади використання алгоритму пошуку несортованою базою даних з невеликим розміром пошукового простору.

10. Виявилось, що як математичне, так і програмне забезпечення квантового комп'ютера є суттєво специфічними і не вкладаються в наші звичні рамки. В той же час необхідність вирішення задач криптоаналізу на квантовому комп'ютері пояснюється його суттєвою перевагою перед класичним у швидкодії.

11. З отриманих результатів можна зробити висновок, що нинішні квантові комп'ютери ще не здатні на повноцінне контрольоване відтворення всіх

квантових властивостей. Хоча, можливо, з більшою кількістю кубітів метод Гровера і даватиме кращий результат, що буде перевірено подальшими дослідженнями.

12. Причини таких результатів та можливість отримання кращих результатів підлягають подальшим дослідженням.

Основні висновки з використання методу Гровера для пошуку унікального елемента на класичному комп'ютері при симетричному перетворенні

1. Нині суттєві результати досягнені в частині розроблення, стандартизації та застосування симетричного криптоперетворення. Разом з тим продовжується розвиток та здійснюються спроби розробити більш ефективні методи криптоаналізу симетричних криптосистем – симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП) та функцій гешування (ФГ). Одним із основних є метод Гровера.

2. Освоєння та застосування квантових систем та квантових математичних методів, а також відповідне їх програмування носять як методологічний аспект, так і психологічний – складність сприйняття.

3. Метод Гровера полягає в проведенні вичерпного пошуку специфічного (унікального) елемента у несортованій базі даних, що складається з $N = 2^n$ елементів, де n - довжина квантового регістра (кількість кубітів).

4. Для криптоаналізу СБП специфічність елемента може зводитись до сеансового чи довгострокового ключа, синхропослідовності тощо.

Особливістю, наприклад, ключа є те, що при його застосуванні зашифровані дані можуть бути розшифрованими за поліноміальний час.

5. У порівнянні з найкращими класичними методами метод Гровера передбачає проведення пошуку з квадратичним прискоренням. Для отримання такого прискорення використовується квантова суперпозиція станів.

6. Конкретна реалізація квантового оракула залежить від кожного окремого випадку та задачі. Наприклад, у випадку криптоаналізу СБП оракул виконує розшифрування на ключі та повертає результат - успіх чи ні. Метод Гровера може бути застосований як на класичному, так і квантовому комп'ютері, хоча реалізація на класичному комп'ютері не є рентабельною.

7. Наведені вище приклади підтвердили ефективність методу Гровера стосовно пошуку в несортованій базі: для: 4 – бітного числа з ймовірністю близько 96,1% зі складністю в 3 раунди; для 8 – бітного числа з ймовірністю близько 99% зі складністю в 13 раундів (силова атака вимагає 256 раундів).

8. Таким чином, застосування методу Гровера для пошуку специфічного елемента в несортованій базі дійсно дозволяє досягти квадратичне прискорення пошуку, вимагає $\sqrt{N}$ раундів у порівнянні з N раундів грубої сили.

13 АНАЛІЗ СКЛАДНОСТІ ТА МОЖЛИВОСТІ РЕАЛІЗАЦІЇ КВАНТОВОЇ АТАКИ СТОСОВНО ПОСТКВАНТОВОГО АЛГОРИТМУ ЕЛЕКТРОННОГО ПІДПИСУ RAINBOW

У розділі визначаються та аналізуються атаки, спрямовані на здійснення криптоаналізу постквантового алгоритму електронного підпису Rainbow. Нині, у зв'язку з успіхами в розробці квантових комп'ютерів, існує потреба в представленні нових стандартів електронних підписів, що були б стійкими до квантового та класичного криптоаналізу. З метою вирішення проблеми відсутності таких електронних підписів NIST США проводить конкурс NIST PQC. В межах цього конкурсу було представлено деякі електронні підписи, розроблені для протистояння квантовому криптоаналізу, серед них і Rainbow. Разом зі звичайним алгоритмом Rainbow було також представлено CZ-Rainbow та стислий алгоритм Rainbow. В розділі досліджуються атаки на всі три варіанти електронного підпису. Від наявності можливості реалізації квантової атаки стосовно електронного підпису Rainbow та складності здійснення такої атаки залежить можливість використання цього електронного підпису в постквантовий період.

Нині існує потреба в представленні та стандартизації нових постквантових електронних підписів. Через успіхи в розробці квантових комп'ютерів відбувається зростання інтересу до квантового криптоаналізу. У зв'язку з цим в межах конкурсу NIST PQC до третього раунду було відібрано 3 електронних підписи, що претендують на стійкість до класичного та квантового криптоаналізу та можливість їх застосування в постквантовий період. Згідно [67,68] до третього раунду в якості основних фіналістів увійшли такі електронні підписи: CRYSTALS-DILITHIUM, FALCON, RAINBOW.

13.1 Параметри RAINBOW

Для розгляду на конкурсі NIST PQC було представлено 3 варіанти схем електронного підпису Rainbow: звичайний алгоритм Rainbow, CZ- Rainbow та стислий алгоритм Rainbow [67] .

В [67] для кожного варіанту наводиться 3 набори параметрів, що відповідають різним категоріям безпеки NIST: набір параметрів I відповідає категоріям безпеки I та II, набір параметрів III відповідає категоріям безпеки III та IV, набір параметрів V відповідає категорії безпеки V. В таблицях 13.1, 13.2 та 13.3 наводяться розміри ключів та підписів для звичайного алгоритму Rainbow, CZ-Rainbow та стислого алгоритму Rainbow відповідно [67].

Таблиця 13.1.

Розміри ключів та підписів для звичайного алгоритму Rainbow

Набір параметрів	Параметри (F, v_1, o_1, o_2)	Відкритий ключ (кВ)	Закритий ключ (кВ)	Розмір гешу (біт)	Розмір підпису (біт)
I	(GF(16), 36, 32, 32)	157.8	101.2	256	528
III	(GF(256), 68, 32, 36)	861.4	611.3	576	1,312
V	(GF(256), 96,36, 64)	1,885.4	1,375.7	768	1,696

Таблиця 13.2.

Розміри ключів та підписів для CZ-Rainbow

Набір параметрів	Параметри (F, v_1, o_1, o_2)	Відкритий ключ (кВ)	Закритий ключ (кВ)	Розмір гешу (біт)	Розмір підпису (біт)
I	(GF(16), 36, 32, 32)	58.8	101.2	256	528
III	(GF(256), 68, 32, 48)	258.4	611.3	576	1,312
V	(GF(256), 96,36, 64)	523.6	1,375.7	768	1,696

Таблиця 13.3.

Розміри ключів та підписів для стислого алгоритму Rainbow

Набір параметрів	Параметри (F, v_1, o_1, o_2)	Відкритий ключ (кВ)	Закритий ключ (кВ)	Розмір гешу (біт)	Розмір підпису (біт)
I	(GF(16), 36, 32, 32)	58.8	0.06	256	528
III	(GF(256), 68, 32, 48)	258.4	0.06	576	1,312
V	(GF(256), 92,36, 64)	523.6	0.06	768	1,696

13.2 Атаки на Rainbow

Як зазначено в [67], всі відомі атаки на Rainbow на даний момент є, в основному, класичними атаками, деякі з яких можуть бути прискореними за рахунок використання алгоритму Гровера. До атак на Rainbow можна віднести [67]: атаки знаходження колізій на геш-функції, прямі атаки, атаки MinRank,

атаки HighRank, атаки “Rainbow-Band-Separation” (RBS), атаки UOV, атаки на диференціальне поле та квантові атаки «грубої сили».

Серед наведених атак передбачається використання квантових методів (а саме, алгоритму Гровера) в наступних атаках: прямі атаки, атаки HighRank, атаки UOV та квантові атаки «грубої сили». Розглянемо атаки, що можуть використовувати квантові методи.

Прямі атаки. Згідно з [67], найбільш прямолінійною атакою на Rainbow, як на мультиваріативну схему, є пряма алгебраїчна атака, що розглядає рівняння $\mathrm{P}(z) = h$ як приклад задачі MQ. Через те, що відкрита система Rainbow є невизначеною системою з $n \approx 1.5 \cdot m$, найефективнішим вважається отримання визначеної системи, що має рівно один розв’язок, шляхом фіксації $n - m$ змінних перед застосуванням такого алгоритму, як алгоритм обчислення базисів Гробнера (F_4) [70]. Для досягнення кращих результатів використовують «гібридний підхід» (відгадують додаткові змінні перед вирішенням системи), як в [71]. В [69] складність вирішення такої системи оцінюється як $Complexity_{direct,dassical} = min_k \left(q^k \cdot 3 \cdot \binom{m-k+d_{reg}}{d_{reg}} \cdot \binom{m-k}{2} \right)$ множень у полі, де d_{reg} – це так званий степінь регулярності системи. Застосування квантового комп’ютера дозволяє використати алгоритм Гровера для прискорення вгадування додаткових змінних при використанні «гібридного підходу». В [69] складність такої атаки оцінюється як $Complexity_{direct,quantum} = min_k \left(q^{\frac{k}{2}} \cdot \binom{m-k+d_{reg}}{d_{reg}}^2 \cdot \binom{m-k}{2} \right)$ множень у полі.

Атаки HighRank. Згідно з [72], метою атаки HighRank є виявлення змінних, що з’являються найменшу кількість разів у центральних поліномах (вони відповідають *Oil*-змінним останнього рівня Rainbow). Складність цієї атаки в [69] оцінюється як $Complexity_{HighRank;classikal} = q^{O_u} \cdot \frac{n^3}{6}$. Застосування квантового комп’ютера дозволяє використати алгоритм Гровера для прискорення пошуку [81] У такому випадку складність атаки становить $\mathrm{Complexity}_{\mathrm{HighRank;\ quantum}} = q^{\frac{O_u}{2}} \cdot \frac{n^3}{6}$ множень у полі.

Атаки UOV. Оскільки Rainbow можна розглядати як продовження добре відомої схеми підписів Oil and Vinegar [73], її можна атакувати, використовуючи всі відомі атаки UOV. Наприклад, атака UOV «OilSubspace» Кіпніса та Шаміра [74]. Можна розглядати Rainbow, як екземпляр UOV з $\upsilon = \upsilon_1 + \mathrm{o}_1$ та $\mathrm{o} = \mathrm{o}_2$. Метою цієї атаки є пошук попереднього зображення так званого Oil підпростору O афінного перетворення $\mathcal{T}$, де $\mathrm{O} = \left\{ \mathrm{x} \in \mathrm{F}^n : x_1 = \cdots = x_\upsilon = 0 \right\}$. Знаходження цього підпростору дозволяє відокремити Oil від змінних Vinegar та відновити закритий ключ.

Складність цієї атаки оцінюється як $\text{Complexity}_{\text{UOV-Attack; classical}} = q^{n-2_{o_2}-1} \cdot o_2^4$ множень у полі. Застосування квантового комп'ютера та алгоритму Гровера зменшує складність до $\text{Complexity}_{\text{UOV-Attack; classical}} = q^{\frac{n-2_{o_2}-1}{2}} \cdot o_2^4$ множень у полі [69].

Квантові атаки «грубої сили». За наявності квантових комп'ютерів атаку грубої сили проти схеми можна різко прискорити за допомогою алгоритму Гровера. У роботі [75] було показано, що можливо розв'язати систему $m-1$ двійкових квадратних рівнянь з $n-1$ двійкових змінних, використовуючи $m+n+2$ кубітів та обчислюючи схему з $2^{n/2} \cdot \left(2m\left(n^2+2n\right)+1\right)$ квантових логічних елементів. Також було наведено інший варіант, який вирішує систему, використовуючи менше кубітів, але з більшою схемою, що має приблизно в 2 рази більше квантових логічних елементів. Наприклад, коли $n=m$, то може бути вирішена двійкова система з m рівнянь у m змінних, використовуючи $2^{m/2} \cdot 2 \cdot m^3$ бітових операцій. Загалом, завдяки алгоритму Гровера, очікується квадратичне прискорення атаки грубої сили.

У таблицях 13.4, 13.5 та 13.6 наведено складність різних атак для наборів параметрів I, III та V відповідно [69]. У першому рядку наведено кількість необхідних для здійснення атаки класичних логічних елементів, а у другому – кількість необхідних для здійснення атаки квантових логічних елементів.

Таблиця 13.4.

Складність класичних та квантових атак на підпис з набором параметрів I

Набір параметрів	Параметри (F, v_1, o_1, o_2)	$\log_2$ (#гейтів)		
		прямі	HighRank	UOV
I	(GF(16), 36, 32, 32)	164	150	157
		122	86	91

Таблиця 13.5.

Складність класичних та квантових атак на підпис з набором параметрів III

Набір параметрів	Параметри (F, v_1, o_1, o_2)	$\log_2$ (#гейтів)		
		прямі	HighRank	UOV
III	(GF(256), 68, 32, 48)	234	410	437
		200	218	233

Таблиця 13.6.

Складність класичних та квантових атак на підпис з набором параметрів V

Набір параметрів	Параметри (F, v_1, o_1, o_2)	$\log_2$ (#гейтів)		
		прямі	HighRank	UOV
V	(GF(256), 96, 36, 64)	285	539	567
		243	283	299

Алгоритм електронного підпису **RAINBOW** буде розглянутий авторами в наступних розділах більш детально.

Висновки відносно складності та можливості реалізації квантової атаки стосовно постквантового алгоритму електронного підпису RAINBOW

Наведені параметри схеми електронного підпису Rainbow, відомості про атаки на нього та оцінки складності атак, набори параметрів, що відповідають різним категоріям безпеки NIST, вказують на наявність можливості реалізації атаки стосовно електронного підпису Rainbow та на те, що складність атак з використанням квантового комп'ютера значно менша за складність класичних реалізацій цих атак. Також слід зазначити, що вплив від застосування квантового комп'ютера на атаки на різні набори параметрів не є постійним. Для набору параметрів I найкращою атакою з використанням квантового комп'ютера є атака HighRank, а для наборів параметрів III та V – пряма атака. Також важливо відмітити, що використання квантових методів в наведених атаках зводиться до виконання певної частини атаки, окрім квантової атаки «грубої сили».

14 ПОСТКВАНТОВА КРИПТОГРАФІЯ ТА МЕХАНІЗМИ ЇЇ РЕАЛІЗАЦІЇ

Розглянемо результати аналізу досягнень в побудові квантового комп'ютера. Пропонується підхід до визначення моделей порушника та загроз постквантового періоду. Наведемо та проаналізуємо вимоги NIST та ETSI до квантово-захищених алгоритмів електронного підпису (ЕП). Розглянемо та порівняємо перспективні кандидати на стандарти ЕП у постквантовий період.

Вступ

В 2012 - 2015 роки зроблені суттєві кроки відносно побудови квантового комп'ютера [76 - 79]. В певній мірі розроблені та чекають свого часу методи та алгоритми квантового криптоаналізу. Такі досягнення викликали необхідність більш детальних оцінок криптографічної стійкості існуючих криптоперетворень та криптосистем, прогнозування їх стійкості в постквантовий період. Практичні висновки таких досліджень викликали стурбованість на самому високому світовому рівні. Так, в Інтернеті 01.11.2015 опублікована стаття «A RIDDLE WRAPPED IN AN ENIGMA [76], автори якої - видатні криптологи США - професори Ніл Кобліц та Альфред Менезес. В ній відмічається, що в серпні 2015 року агентство національної безпеки (АНБ) уряду США виступило із заявою про слабкість існуючих криптосистем відносно методів квантового криптоаналізу та необхідність розробки постквантових стандартів криптографічного захисту інформації. Подальшим підтвердженням стурбованості уряду США відносно необхідності розробки нових стандартів криптографічного захисту інформації є опублікування звіту «Report on Post – Quatum Cryptography NISTIR 8105 (DRAFT) [76], в якому повністю підтверджено вказані вище загрози. Більш того, в цьому звіті визначені основні проблеми та можливості і етапи їх вирішення.

Зважаючи на проблемність та необхідність розв'язання задач криптоаналізу нині діючих стандартизованих криптографічних перетворень, тобто, оцінки їх стійкості у постквантовий період, особлива увага нині зосереджена на визначенні можливостей розв'язання задач криптоаналізу з використанням квантових комп'ютерів та, в першу чергу, розроблення відповідних методів та алгоритмів квантового криптоаналізу. Таким чином, нині у світі вирішуються дві проблеми – створення квантового комп'ютера та розробка і реалізація математичних методів квантового криптоаналізу.

Скоріше всього, у випадку появи квантового комп'ютера, що може реалізувати уже розроблені квантові алгоритми, зокрема, алгоритми криптоаналізу Шора [52,78,79,81] та Гровера [78,80, 81,86,87], Ксіонга та Ванга [78,88,89] можуть спричинити великі загрози у інформаційній сфері відносно забезпечення криптографічної стійкості для існуючих асиметричних криптоперетворень. При цьому важливим є не тільки сам факт побудови такого комп'ютера, а й технічні характеристики, якими буде володіти квантовий комп'ютер. Вказане необхідно враховувати, так як існуючі квантові алгоритми для своєї «роботи» потребують значних технічних ресурсів, особливо просторових, у вигляді кількості кубітів регістрів [52,76-80,91-93].

Наприкінці, у грудні 2015 року фахівці компанії Google підтвердили, що згідно з їх дослідженням в комп'ютері D-Wave використовуються квантові ефекти. При цьому в «1000-кубітному» комп'ютері кубіти в дійсності організовані в кластери по 8 кубітів кожен. Якраз це і дозволило добитися в одному з алгоритмів [94] швидкодії в 100 млн разів більше, ніж у звичайному комп'ютері.

У цілому, нині в криптології існує ряд проблемних задач повного розкриття [78]. Детальний аналіз дозволив виділити основні методи криптоаналізу для застосування на квантовому комп’ютері (звичайно, якщо він буде побудований з відповідними характеристиками). До основних задач, які можуть бути вирішені на квантовому комп’ютері, необхідно віднести такі [79-90]:

- квантовий алгоритм факторизації Шора;
- квантовий алгоритм Гровера;
- квантовий алгоритм Шора розв’язку дискретного логарифму в скінченному полі;
- квантовий алгоритм Шора розв’язку дискретного логарифму в групі точок еліптичної кривої;
- квантовий алгоритм криптоаналізу для перетворень в фактор - кільці;
- квантовий алгоритм криптоаналізу Ксіонга та Ванга та його вдосконалення тощо.

Зрозуміло, що вирішення названих проблем займе ще значний час. Але уже сьогодні необхідно обґрунтовувати та розробляти моделі порушників постквантового періоду та моделі загроз. Перші спроби вирішення цих задач наведені в [78]. Також, на наш погляд, зважаючи на широке розповсюдження та застосування, в першу чергу нині необхідно вести дослідження по обгрунтуванню вимог, визначенні основних механізмів постквантових електронних підписів (ЕП) та їх первинному аналізу за критеріями стійкості та складності.

Метою цього розділу є розробка основних положень моделі порушника та загроз у постквантовий період, обгрунтування вимог криптографічних перетворень взагалі та до ЕП зокрема, при виконанні яких вони можуть бути стійкими у постквантовий період, а також визначення основних методів, з використанням яких можуть бути побудовані та стандартизовані міжнародні та національні ЕП.

14.1 Аналіз основних проблем створення квантового комп’ютера та розробки математичного забезпечення

14.1.1 Аналіз досягнень в побудові квантового комп’ютера

Таким чином, як вказувалось у вступі (розділ 14), нині вирішується дві важливі проблеми - створення квантового комп’ютера та розробки і реалізації математичних методів квантового криптоаналізу. Проведемо аналіз стану їх вирішення та відповідні пропозиції.

Аналіз показує, що основна проблема у побудові квантового комп'ютера полягає в тому, що необхідно створити систему, яка б задовольняла майже несумісним таким вимогам [78-93]:

- кубіти (елементи квантового комп'ютера) повинні бути максимально ізольовані один від одного та від навколишнього середовища;
- можливість корельованого впливу на пару кубітів, тобто, потрібно не тільки вміти змінювати стан одного кубіту, але й ще вмикати та вимикати взаємодію між парою сусідніх кубітів;
- система кубітів повинна бути досить стабільною, щоб зберігати корельованість станів, але одночасно й легко відновлюваною для нового циклу обчислень;
- реалізовувати сукупність зворотних перетворень над системою кубітів, які б дозволили виконати будь-яку потрібну логічну операцію;
- під час обчислень система повинна зберігати квантові властивості, але наприкінці треба зробити вимірювання, яке б однозначно визначило стан системи і у такий спосіб звело б квантову інформацію в класичну.

Дуже важливою властивістю квантових об'єктів є можливість здійснювати паралельні операції (див. розділ 7.6). Так, для системи із N кубітів, що перебуває в переплутаному стані, ефективно кодується відразу 2N чисел. Тому операція над нею, завдяки когерентності станів різних кубітів, впливає на всі доданки в сумі і це дозволяє обробляти відразу всі 2N чисел.

На сьогодні вже існують квантові алгоритми, які дають змогу проводити атаки на такі існуючі асиметричні криптосистеми [52,80-94]:

- системи, що базуються на складності факторизації великого цілого числа (RSA);
- системи, що базуються на складності вирішення дискретного логарифму в скінченному полі Галуа (DSA) ;
- системи, що базуються на складності вирішення дискретного логарифму в групі точок еліптичної кривої (ECC) ;
- системи на базі алгебраїчних решіток (NTRU).

Усі вказані криптосистеми відносяться до класу ймовірно-стійких.

А ця ймовірна стійкість якраз і визначається можливостями появи квантових комп’ютерів, і, як наслідок, вирішення задачі повного розкриття.

Значних здобутків досягла фірма D-Wave, яка стала першою компанією, що продала комерційну версію квантового комп’ютера [95]. Так, з 20 травня 2011 D-Wave Systems продає за $ 11 млн. квантовий комп'ютер D- Wave One з 128-кубітним чіпсетом, який виконує тільки одну задачу - дискретну оптимізацію. 25 травня 2011 р. Lockheed Martin підписала багаторічний контракт з D-Wave Systems, що стосується виконання складних обчислювальних завдань на квантових процесорах. Контракт також включає в себе технічне обслуговування, супутні послуги і купівлю квантового комп'ютера D - Wave One.

Квантовий комп'ютер – це пристрій, процеси обчислень та передачі даних у якому грунтуються на явищі квантової суперпозиції і квантової переплутаності. На нинішній час повноцінний квантовий комп'ютер є ще поки гіпотетичним пристроєм, можливість побудови якого пов'язана з вирішенням

складних теоретичних та практичних проблем квантової фізики та складних експериментів. Дослідження в цьому напряму знаходяться на передньому краї сучасної фізики. Важливою проблемою є також обгрунтування вимог та створення мови програмування для квантового комп'ютера.

Відносно поняття «квантовий паралелізм» в обчисленні можна трактувати так: «Дані в процесі обчислень є квантовою інформацією, яка після закінчення процесу перетворюється в класичну шляхом вимірювання кінцевого стану квантового регістра із заданим числом кубітів. При цьому виграш в квантових алгоритмах досягається за рахунок того, що при застосуванні однієї квантової операції велике число коефіцієнтів суперпозиції квантових станів, які у віртуальній формі містять класичну інформацію, перетворюється одночасно».

Вчені D-Wave опублікували статтю, в якій повідомляється, що за допомогою методу кубіто-тунельної спектроскопії ними було доведено наявність квантової когерентності і переплутаності між окремими підгрупами кубітів в процесорі під час проведення обчислень (розміром 2 і 8 елементів). Наприкінці, у грудні 2015 року фахівці компанії Google підтвердили, що, згідно з їх дослідженням, в комп'ютері D-Wave використовуються квантові ефекти. При цьому в «1000-кубітному» комп'ютері кубіти в дійсності організовані в кластери по 8 кубітів кожен. Якраз це і дозволило добитися в одному з алгоритмів швидкодії в 100 млн разів більше, ніж у звичайному комп'ютері [94].

В той же час програмна заява NIST або АНБ ретельно опрацьована протягом значного часу [76,77]. Комітет, відповідальний за складання його, обговорює кожне речення; нічого не залишено на волю випадку або недбалого редагування. Крім того, коли попросили роз'яснити звіт в серпні 2015 року, АНБ випустило оновлену версію, яка значно не відрізняється від першої. Таким чином, ми повинні почати з передумови, що АНБ мало намір в заяві передати, що воно і зробило.

Ще незрозуміло, коли масштабовані квантові комп'ютери будуть доступні, проте в минулому році дослідники, що працюють на побудову квантового комп'ютера, підрахували, що цілком ймовірно, що квантовий комп'ютер, здатний атакувати RSA-2048 в лічені години, може бути побудований до 2030 року, але це вимагає закласти в бюджет близько мільярда доларів. Це серйозна довгострокова загроза для криптосистем, що в даний час стандартизовані в NIST. Таким чином, перехід від 12 до 128 біт безпеки, можливо, менш актуальний, ніж перехід від існуючих криптосистем з постквантовою криптосистемою. Цей постквантовий перехід викликає багато фундаментальних проблем. Так, розробка стандартів для постквантової криптографії вимагає значних ресурсів для аналізу кандидата квантово-стійких схем, і зажадає значного залучення громадськості для запевнення довіри в алгоритмах NIST. Інтерес в областях квантових обчислень і квантової криптографії останнім часом збільшився в зв'язку з розвитком квантової обчислювальної техніки і останні зміни NIST це підтверджують [77]. Це дає можливість для взаємодії з науковою спільнотою, яка може не настати знову до практичних квантових обчислень, що є дійсно неминучим. Отже, NIST починає готуватися до переходу до квантово-стійкої криптографії. NIST США робить наступні кроки, щоб ініціювати стандартизацію

постквантової криптографії. У NIST планується вказати попередні критерії оцінки для “квантово-наполегливих” стандартів шифрування з відкритим ключем. Критерії включають безпеку та експлуатаційні вимоги. Проєкт критеріїв буде винесено на обговорення громадськості в 2016 році і сподіваються завершити до кінця року. У той час NIST почне приймати пропозиції щодо квантово-стійкої асиметричної криптографії, в першу чергу ЕП. NIST буде встановлювати останній термін подачі заявок в кінці 2017 року.

Хоча цей процес буде мати багато спільного з процесами, які привели до стандартизації AES і SHA-3 [78], це не змагання. NIST бачить свою роль в управлінні процесом досягнення консенсусу спільноти прозорим і своєчасним чином. В ідеалі, кілька алгоритмів будуть з'являтися у міру "правильності вибору". NIST може вибрати один або більше алгоритмів з ЕП, що пройдуть випробовування [93,94]. Коли стандарти для квантово-стійкої криптографії з відкритим ключем стануть доступні для аналізу, NIST буде переглядати загрози квантових комп'ютерів до існуючих стандартів, що нині застосовуються, та може відкликати такі стандарти. Тому установи повинні бути готові до переходу від існуючих алгоритмів до стійких у постквантовий період вже в найближчі 6- 8 років. Як показує аналіз, таких стандартизованих ЕП в даний час не існує.

Наведені вище результати аналізу стану розробки та можливості застосування квантового комп’ютера дозволяють обґрунтувати необхідність та важливість розробки постквантової криптографії [76,77]. Вирішення цієї задачі можливо при наявності моделі порушника. По суті вона зводиться до моделі квантового комп’ютера з конкретними характеристиками та можливостями по реалізації квантових алгоритмів криптоаналізу.

Моделі порушника також розглянуті авторами в підрозділі 12.2.

14.2 Аналіз основних загроз відносно криптографічних перетворень у постквантовий період

При побудові моделей загроз у постквантовий період за нинішніх умов в якості основних необхідно брати методи криптоаналізу, як основні моделі загроз, що можуть бути реалізованими на квантовому комп’ютері при вирішенні задач криптоаналізу. Причому усі вказані методи повинні бути орієнтовані на використання специфіки квантового комп’ютера та мову програмування на ньому.

До основних задач, які можуть бути вирішені на квантовому комп’ютері, в першу чергу необхідно віднести такі:

- квантовий алгоритм факторизації Шора [52,78,79];
- квантовий алгоритм Гровера [78,80,81,83];
- квантовий алгоритм Шора вирішення дискретного логарифму в скінченному полі [79,80];
- квантовий алгоритм Шора вирішення дискретного логарифму в групі точок еліптичної кривої [52,78,79,83,86];
- квантовий алгоритм криптоаналізу для перетворень в фактор – кільці [78,85,88-91].

Проблема криптоаналізу, на вирішення якої спрямовано метод Гровера, може бути сформульована наступним чином. Нехай дана неупорядкована база даних (список) з N елементів і нехай в ній існує один елемент, що володіє деякою властивістю, яка перевіряється з поліноміальною складністю. Потрібно знайти цей елемент з мінімально можливою складністю і, зрозуміло, за менший час (див. розділ 7.10).

Вирішення цієї задачі може бути виконане з використанням декількох класичних алгоритмів [81], в яких для підвищення ймовірності успіху процедура повторюється багатократно. Для того, щоб отримати при повторюваних квантових перетвореннях результат, що очікується, дуже важливо визначити, коли потрібно зупинитися і провести уточнення [78,80,83]. Наприклад, використовуючи алгоритм Гровера, можна знайти секретний ключ симетричного шифрування чи гешування за $\sqrt{N}$ ітерацій, де N- розмір простору ключів. В якості прикладу в таблиці 14.1 (див. підрозділ 14.3) наведені оцінки стійкості симетричних криптографічних систем проти квантового криптоаналізу. Аналіз даних таблиці показує, що стійкість симетричних шифрів при атаці з використанням квантового алгоритму Гровера суттєво зменшується.

Зроблені попередні оцінки показують, що з використанням квантового алгоритму Шора задачу факторизації модуля N можна звести до вирішення еквівалентної проблеми, сутність етапів якої полягає у наступному:

- вибрати випадково й рівномірно ціле число a , взаємно просте з N;
- для вибраного числа a, що є взаємно простим з N, знайти порядок r елемента $a \bmod N$.

Взаємну простоту чисел a та N виконати, використовуючи алгоритм Евкліда [52,78]. Якщо a не є взаємно простим з N, то потрібно повторно вибрати a, взаємно просте з N. Якщо a є взаємно простим з N, то порядок r елемента $a \bmod N$ буде дільником числа N. Збільшення розміру модуля перетворення і, відповідно, особистого ключа при застосуванні квантового алгоритму Шора не забезпечує необхідного збільшення складності дискретного логарифмування в скінченному полі, як при зломі електронного цифрового підпису, так і направленого шифрування. Наприклад, для модуля $P \geq 2^{3072}$ складність дискретного логарифмування в скінченному полі складає $1.4 \cdot 10^{31}$, а із застосуванням алгоритму Шора всього $2.9 \cdot 10^{10}$ операцій. Але, в той же час, при застосуванні квантового алгоритму проблемним є реалізація регістрів зі значним числом кубітів - не менше 9216 кубітів. Очевидно, досягти такого розміру буде ще певний час проблемною задачею. Збільшення розміру порядку базової точки при криптоаналізі з використанням квантового алгоритму не дає суттєвого збільшення криптографічної стійкості криптографічної системи на еліптичних кривих. Також показано [52,78,82], що при збільшенні модуля складність дискретного логарифмування класичними методами в групі точок еліптичної кривої зі збільшенням порядку базової точки збільшується суттєво. Але потрібно взяти до уваги, що реалізація квантового алгоритму пов’язана із застосуванням регістрів з великою кількістю кубітів, яка необхідна для проведення квантової атаки. Наприклад, для базової точки з порядком 2571 необхідно використовувати

реєстр з довжиною 4016 кубітів. Вважається, що така велика кількість кубітів ще певний час не буде реалізуєма.

Комбінуючи атаку зустріч посередині і квантовий алгоритм пошуку Гровера, можна підвищити ефективність атаки зустріч посередині на NTRU. Отримані оцінки показують, що часова складність $O\left(\sqrt{C_{\binom{N}{2}+1}^{\frac{d_f}{2}}}\right)$ різко скоротилася в порівнянні з класичним алгоритмом атаки зустріч посередині $\left(O\left(\frac{C_{N/2}^{d/2}}{\sqrt{N}}\right)\right)$ з тією ж просторовою складністю. В порівнянні з методом Ванга наведений метод також на багато кращий відносно часової складності. Атака Ванга має майже таку ж складність, як і класична атака зустріч посередині, проте вона має меншу складність відносно просторової складності. Із даних, наведених в таблицях 14.1 та 14.2, також видно, що системи на базі NTRU, навіть з використанням параметрів великих розмірів, не забезпечують суттєвої стійкості проти квантової атаки зустріч посередині. У цілому системи на базі NTRU стають уразливі до квантового криптоаналізу. Квантова атака зустріч посередині може нанести велику шкоду системам NTRU у разі появи квантових комп'ютерів.

На нинішньому етапі розробки квантового комп'ютера дуже важливо, паралельно з вирішенням питань створення апаратних засобів, розробити та дослідити математичні методи криптоаналізу з урахуванням вимог та можливостей безпосередньо квантового комп'ютера. Вказані задачі вирішуються, уже починаючи з 80 – х років 20 століття. Тому на нинішній час уже практично вирішені основні математичні та програмні проблемні питання створення квантового комп'ютера. Частина із вказаних проблемних питань перевірена засобом моделювання на класичних комп'ютерах.

В основу при розробленні моделі загроз у постквантовий період можуть бути покладені такі математичні методи [78-89]: - квантовий алгоритм факторизації Шора; -квантовий алгоритм Гровера; - квантовий алгоритм Шора вирішення дискретного логарифму в скінченному полі; - квантовий алгоритм Шора вирішення дискретного логарифму в групі точок еліптичної кривої; - квантові алгоритми Гровера та Ванга криптоаналізу для перетворень в фактор - кільці.

На даний момент атака зустріч посередині є найбільш ефективним методом проти NTRU, але часова складність атаки зустріч посередині до цих пір дуже велика. Так, метод Ванга можна розглядати як квантовий пошук «грубої сили», який зменшує часову складність від $O\left(C_N^{d_f}\right)$ до $O\left(\sqrt{C_{N+1}^{d_f}}\right)$. Вказане можна використати для комбінації квантових обчислень та атаки зустріч посередині. Такий метод був розроблений та наводиться у роботі Ксіонга та Ванга.

Наведені вище результати аналізу та певних досліджень дозволяють зробити висновок, що на нинішньому етапі основою створення практичних

моделей загроз у постквантовий період є розробка їх математичних та програмних основ, зрозуміло, з урахуванням апаратних практичних можливостей. Їх використання дозволить виконати детальний аналіз постквантових криптографічних примітивів, включаючи ЕП.

14.3 Вимоги NIST щодо можливих кандидатів постквантових криптопримітивів

NIST розуміє необхідність пошуку нових примітивів, які будуть актуальні у постквантовий період [94]. За планами NIST у 2016 році здійснюється підготовка до проведення конкурсу на нові постквантові криптографічні стандарти. На цей час вже підготовлений проєкт вимог до кандидатів. Об'ява конкурсу планується у 2017 році. А до кінця 2019 року є сподівання, що нові стандарти будуть затверджені. Ці роботи здійснюються у рамках відкритого конкурсу Post-Quantum crypto Project. У таблиці 14.1 наведені основні вимоги до кандидатів, а у таблиці 14.2 критерії відбору серед запропонованих кандидатів [94].

Таким чином, у 2016 році у США розпочалися активні роботи щодо підготовки до проведення конкурсів на майбутніх кандидатів квантово-захищених алгоритмів. Тобто, США активно приступили до виконання планів розроблення та прийняття постквантових стандартів криптографічного захисту інформації. По суті, наведені в таблиці 14.1 дані містять вимоги та рекомендації щодо першого етапу розроблення та тестування постквантових примітивів. Вони охоплюють широкий спектр та містять вимоги до безпеки постквантових криптографічних примітивів, техніко-економічні та техніко-експлуатаційні вимоги. Також в певній мірі запропоновані узагальнені критерії оцінки та порівняння можливих кандидатів. Також на період до прийняття нових стандартів органи зі стандартизації США уже надали рекомендації стосовно застосування існуючих криптографічних алгоритмів, в першу чергу в частині довжин (розмірів) загальних параметрів та ключів, а також конкретизації криптопримітивів.

Таблиця 14.1.

Вимоги NIST до постквантових криптографічних примітивів

Вимоги з безпеки	
Використання криптографії з відкритим ключем	Заміна стандарту цифрового підпису FIPS 186. Заміна стандартів розподілу ключів SP 800-56A, SP 800-56B. Використання нового стандарту в протоколах: TLS, SSH, IPsec, DNSSEC.
Модель безпеки для шифрування та розподілу ключів	Схема «семантично безпечного шифрування». Модель безпеки - IND-CCA2. Умови безпеки: доступ зловмисника менше, ніж до 2^{64} обраних шифротекстів.
Модель безпеки для електронного підпису	Модель безпеки EUF-CMA. Умови безпеки: доступ зловмисника менше, ніж до 2^{64} обраних повідомлень.

Продовження табл. 14.1.

Вимоги до стійкості	1) 128 біт класичної безпеки / 64 біт квантової захищеності (запас стійкості AES-128). 2) 128 біт класичної безпеки / 80 біт квантової захищеності (запас стійкості SHA-256, SHA3-256). 3) 192 біт класичної безпеки / 96 біт квантової захищеності (запас стійкості AES-192). 4) 192 біт класичної безпеки / 128 біт квантової захищеності (запас стійкості SHA-384, SHA3-384). 5) 256 біт класичної безпеки / 128 біт квантової захищеності (запас стійкості AES-256).
Додаткові властивості безпеки	«perfect forward secrecy» (удосконалена випереджаюча безпека). Стійкість до атак сторонніми каналами. Стійкість до мультиключових атак. Стійкість до відмов.
Інші вимоги	Прозорі математичні рішення при реалізації криптопримітивів. Обґрунтованість стійкості криптоперетворень.
Техніко-економічні вимоги	
Вимога	Сутність вимоги
Розміри (довжини) відкритого ключа, шифротексту, підпису	Орієнтація на розмір пакетів інтернет-протоколів. Ґешування ключової інформації. Для «perfect forward secrecy» використання менших довжин ключа.
Обчислювальна ефективність операцій відкритого шифрування (перевірки підпису) та особистих ключів (шифрування, підписування)	Забезпечення ефективності як апаратної, так і програмної реалізації.
Обчислювальна ефективність процесу генерування ключів	Відповідність розмірів ключа до обраної системи.
Помилки шифрування	Низький відсоток помилок шифрування. Багаторазове шифрування.
Техніко-експлуатаційні вимоги	
Гнучкість	Додаткові можливості схеми (оптимізація, неявний обмін ключами тощо). Кросплатформеність. Можливість розпаралелювання.
Простота	Зрозумілість та прозорість побудови.

Таблиця 14.2.

Критерії відбору серед запропонованих кандидатів

Технічна оцінка	
Перевірка на коректність	Перевірка правильності базових та оптимізованих реалізацій.
Перевірка на ефективність	Обчислення часу, необхідного для генерації ключа, шифрування, розшифрування, електронного підпису, перевірки підпису, встановлення ключів, а також розмір ключів, зашифрованого тексту і підпису (тестування проводиться на оптимізованих версіях).
Інші перевірки	Додаткові випробування.
Умови випробувань	Основна платформа: NIST PQC Reference Platform, Intel x64, Windows or Linux, the GCC compiler. Також можуть проводитися додаткові тестування за інших умов (8-бітових процесорів, цифрових сигнальних процесорів, виділених CMOS тощо)

В той же час, за нашими дослідженнями, орієнтація на критерій оцінки стійкості з довжиною симетричного ключа 128 біт є недостатньо обгрунтованою. Очевидно, потрібно збільшувати як довжину блока (при блоковому симетричному перетворенні), так і довжину ключа до 256 бітів. Разом з тим поява наведених вимог є важливим етапом постквантової стандартизації в криптографії.

14.3.1 Вимоги до квантово-захищених алгоритмів ЕП ETSI

Європейський Союз також розпочав активну роботу з підготовки нових постквантових стандартів. В деякій мірі зупиняються роботи, що пов’язані з виконанням завдань, що були визначені в Регламентах ЄС 2012 та 2014 років. Європейською організацією зі стандартизації ETSI у кластері «Безпека» сформований новий напрямок «Квантово - захищена криптографія» («Quantum-Safe Cryptography») [93,94].

Крім того, уже нині група спеціалістів активно веде дослідження в таких основних напрямках:

- оцінка потенційних загроз безпеці в постквантовий період;
- приклади та сценарії реалізації квантово-захищених примітивів;
- фундаментальні межі можливостей квантових обчислень стосовно існуючих та перспективних криптографічних перетворень;
- квантово-безпечні криптографічні протоколи в першу чергу встановлення ключів;
- електронний підпис, захищений у постквантовий період.

За результатами даних досліджень прогнозується прийняття групи стандартів для постквантового періоду. Вже зараз готові попередні версії кожного із проєктів.

Орган зі стандартизації ЄС (ETSI) опублікував груповий звіт «Квантово-захищена криптографія. Квантово-безпечна інфраструктура» [93], в якому закріплені основи перспективної інфраструктури, представлені механізми та наведені типи примітивів, що планується використовувати. Окремо висунуті вимоги та сформовані критерії оцінки майбутніх кандидатів.

За результатами попередніх досліджень визначені п’ять сімейств криптографічних примітивів, в основу яких покладені математичні методи, що, на погляд спеціалістів, забезпечать виконання прогнозуємих вимог. У таблиці 14.3 наведені дані відносно таких криптоперетворень та на чому базується їх стійкість. У таблиці 14.4 наведені дані відносно можливих криптографічних протоколів встановлення (узгодження, транспортування) ключів та відносно можливих механізмів автентифікації.

Таблиця 14.3.

Напрямки досліджень у постквантовій криптографії

Рекомендації ETSI Примітиви на основі решітки	Криптографічна стійкість (безпека) залежить від складності розв’язання рівняння на алгебраїчних решітках.
Багатоваріантні примітиви	Криптографічна стійкість (безпека) залежить від складності рішення системи багатовимірних поліноміальних рівнянь.
Примітиви на основі коду	Криптографічна стійкість (безпека) залежить від складності виконання завдання декодування лінійного коду.
Примітиви на основі гешування	Криптографічна стійкість (безпека) залежить від складності знаходження колізій або прообразів в криптографічних ґеш-функціях.
Ключові примітиви на основі ізогенії	Криптографічна стійкість (безпека) залежить від складності знаходження невідомої ізогенії між парою суперсингулярних еліптичних кривих.

Таблиця 14.4.

Класифікація постквантових примітивів встановлення ключів та механізмів ЕП

Примітиви узгодження ключів (key agreement primitives)	Дві сторони надійно генерують загальний симетричний ключ від інформації, що вноситься обома сторонами; наприклад, шляхом захищеного обміну відкритими ключами один з одним, наприклад, у вигляді сертифікатів відкритих ключів.
Примітиви транспортування ключів (key transport primitives)	Одна із сторін генерує симетричний ключ і надійно розділяє його з іншою стороною; наприклад, шляхом відправки його у зашифрованому вигляді у відкритому ключі (сертифікаті) іншій стороні.

Продовження табл. 14.4.

Схеми підпису типу Fiat-Shamir (Fiat-Shamir signature schemes)	Будуються на базі інтерактивних протоколів з доказово нульових знань.
Схеми підпису на основі ґеш - функцій (hash-and-sign signature schemes)	Будуються на основі використання односторонніх функцій.

Також ETSI розроблені вимоги, які запропоновано висунути до кандидатів на постквантові стандарти криптографічних перетворень, що наведені у таблиці 14.5. На їх основі в найближчий час планується оголосити Європейський конкурс на нові постквантові криптопримітиви з можливістю їх подальшої стандартизації. Важливим є той факт, що ETSI також висунув вимоги до розмірів параметрів та ключів ЕП. Відповідні дані наведені в таблиці 14.6. У таблиці 14.6 надані результати порівняльного аналізу виконання основних операцій криптоперетворень [77]. У якості базової одиниці виміру береться час виконання відповідного перетворення на алгоритмі RSA.

Таблиця 14.5.

Вимоги та параметри оцінювання кандидатів

Вимоги безпеки: – Проходження громадського контролю та визнання науковим співтовариством. – Надійне підтвердження криптографічної стійкості. – Актуальність моделі безпеки, якій відповідають криптопримітиви. – Стійкість криптопримітивів до існуючих атак, в тому числі на основі квантових алгоритмів. – Можливість використання криптопримітивів в безпечних протоколах розподілу ключів. – Можливість поєднання кількох функцій безпеки криптографічних протоколів (наприклад, одночасна автентифікація та встановлення). – Можливість кількісної оцінки та порівняння заявлених класичних та квантових рівнів безпеки. – Визначеність рекомендованих розмірів ключів для заданих рівнів безпеки (наприклад, 80 біт, 112 біт, 128 біт або 256 біт).	
Види безпек	
Класична безпека	Стійкість проти класичних атак.
Квантова безпека	Стійкість проти «квантових» атак. Зокрема, стійкість до алгоритму Гровера.
Доказова безпека	Базування на задачах, які мають необхідну складність криптографічного аналізу. Можливе ігнорування зниження рівня складності за умови, що практична стійкість зміниться несуттєво.
Довгострокова безпека	Можливість використання у протоколах типу TLS 1.3 з підтримкою forward secure cipher suites.

Продовження табл. 14.5.

Активна безпека	Стійкість проти атак з адаптивним підбором.
Ефективність	Використання рекомендованих розмірів загальних параметрів та ключів відповідно до заданого рівня безпеки. Незалежність швидкодії та кількості раундів перетворень від платформи реалізації. Допустима швидкодія генерації ключів і часу, необхідного для поширення нового ключа. Інші практичні вимоги (наприклад, стійкість до відмов).
Реалізація та розгортання	Простота впровадження стандартних примітивів нефахівцями. Відносно малий обсяг (у сенсі ресурсу) реалізації (зокрема, можливість реалізації на FPGA та вбудованих пристроях). Відносно малий (допустимий) обсяг необхідної пам'яті під час виконання (можливість реалізації на пристрої з обмеженими ресурсами). Практичність розмірів ключа і підпису для передачі або зберігання в ряді платформ, включаючи пристрої з обмеженими ресурсами. Простота інтеграції в існуючі криптографічні протоколи та криптографічні системи. Низька вартість заміни або модернізації захищених технологій. Можливість виконання криптопримітивів для декількох функцій, наприклад, для забезпечення автентифікації, розподілу ключів тощо. Сумісність, наприклад, гнучкість у виборі ґеш-функції в схемах дерева Мерклі тощо.

Таблиця 14.6.

Орієнтовні розміри параметрів та ключів для алгоритмів ЕП, що висунув NIST

Параметр Алгоритм	Час генерації ключів (RSA sign=1)	Час підпису (RSA sign=1)	Час перевірки (RSA sign=1)	Крипто період	Розмір відкритого ключа, біт	Розмір закритого ключа, біт	Розмір підпису (біт)
Підписи Зінтерніц-Мерклі	200 10000 500000	1 1 2	0.2 0.2 0.2	2^{20} 2^{30} 2^{40}	368 368 368	15200 22304 29344	17024 18624 20224
Підписи GLP (на основі решітки)	0.01	0.5	0.02		11800	1620	8950

Продовження табл. 14.6.

Підписи CFS (на снові коду)	5	2000	0.02		9437184	≈15000 000	144
Підпис Psflash (багатова-ріантний)	50	1	0.1		576992	44400	296

Попередній аналіз вимог до криптографічних примітивів постквантового періоду, що висунуті NIST США та ETSI ЄС (див. пункти 14.3 та 14.3.1), дозволяє зробити такі попередні висновки:

- в основному вимоги NIST США та ETSI ЄС дозволяють розпочати конкурс із створення криптографічних примітивів, що будуть стійкими у постквантовий період;
- вимоги як NIST США, так і ETSI ЄС не містять конкретних критеріїв оцінки та порівняльного аналізу потенційних кандидатів;
- нечіткими є визначення вимог до систем управління ключами, вони формулюються тільки відносно механізмів узгодження та транспортування ключів;
- в пропозиціях з оцінки стійкості (безпечності) використовується тільки довжина ключа симетричного криптографічного перетворення, допустимі значення довжин блоків відсутні, що, на наш погляд, не дозволяє отримати об’єктивних оцінок стійкості;
- на наш погляд, допустимі довжини ключів у 128 бітів симетричних криптографічних перетворень є замалими, за рахунок цих особливостей можуть бути закладені вразливості;
- для оцінки стійкості та, взагалі, оцінки криптографічних примітивів при порівнянні та вибиранні кандидатів необхідно використовувати методики, що базуються на системах безумовних та умовних критеріїв, які дозволяють суттєво зменшити суб’єктивність прийняття рішень [78].

У цілому висунуті NIST США та ETSI ЄС вимоги дозволяють, на наш погляд, зробити тільки попередні оцінки постквантових механізмів ЕП та приймати певні рішення відносно подальших їх досліджень з метою вибору найбільш ефективних по критеріям стійкості та техніко-економічним і техніко-експлуатаційним критеріям.

14.4 Порівняльний аналіз перспективних механізмів ЕП

В процесі попереднього аналізу та досліджень в якості постквантових механізмів, згідно таблиць 14.3 та 14. 4, були вибрані такі [76,77, 97-106]:

- механізми ЕП на основі ґеш-функцій (НВ-криптографія);
- механізми ЕП на збиткових кодах (СВ-криптографія);

- механізми ЕП на основі перетворення на алгебраїчних решітках (LB-криптографія) (див. п. 16.1);

- механізми ЕП на основі мультиваріативного квадратичного перетворення (MQ-криптографія);

- механізми ЕП на основі ізогеній еліптичних кривих (див. п. 16.6).

Розглянемо в узагальненому вигляді сутність названих вище кандидатів на стандарт постквантових механізмів ЕП та порівняємо їх, орієнтуючись на вимоги, що сформульовані в пунктах 14.3 та 14.3.1.

Сутність ЕП на основі ґеш-функцій (HB-криптографія)

Механізми ЕП на основі ґеш-функцій грунтуються на використанні стандартної криптографічної ґеш-функції H, наприклад, в [97-99] запропоновано формувати геш - значення довжиною від 28 до b = 128 бітів. Якщо для цього обрати ґеш - функцію SHA-256, то відкритий ключ підписувача в ній буде мати $8\,b^2$ бітів. Для вказаного значення b=128 відкритий ключ буде мати довжину у 16 кілобайт та складається з $4b$ рядків:

$$Y_1[0],\ Y_1[1],\ Y_2[0],\ Y_2[1],\ \dots,\ Y_{2b}[0],\ Y_{2b},$$

кожний з яких має довжину $2b$ бітів. За цих умов, тобто, для $b=128$ ЕП повідомлення m буде мати довжину $2b$ ($2b$+1) бітів, тобто, 8 кілобайт. Підпис складається із $2b$-бітових рядків $r,\ x_{1,\dots},\ x_{2b}$, таких, що біти $(h_{1,\dots},\ h_{2b})$ ґеш-значення $H(r,\ m)$ визначаються як:

$$Y_1[h_1] = H\,(x_1),\ Y_2[h_2] = H(x_2)\ \dots\ Y_{2b}[h_{2b}] = H(x_{2b}).$$

З початку підписувач генерує секретний параметр x, а потім обчислює

$$Y=H(x).$$

Особистий ключ підписувача має $8b^2$ бітів, а саме – $4b$ незалежних однорідних випадкових рядків:

$$x_1[0],\ x_1[1],\ x_2[0],\ x_2[1],\ \dots\ ,\ x_{2b}[0],\ x_{2b}[1],$$

кожний з яких містить $2b$ бітів.

Потім підписувач обчислює відкритий ключ

$$Y_1[0],\ Y_1[1],\ Y_2[0],\ Y_2[1],\ \dots,\ Y_{2b}[0],\ Y_{2b}[0]$$

у вигляді:

$$H(x_1[0]),\ H(x_1[1]),\ H(x_2[0]),\ H(x_2[1]),\ \dots,\ H(x_{2b}[0]),\ H(x_{2b}[1]).$$

При підписуванні повідомлення m підписувач генерує однорідний випадковий рядок T, обчислюючи біти $(h_{1,\dots},\ h_{2b})$ ґеш-значення $H(r,m)$ та формує значення:

$$(r,\ x_1[h_1],\ \dots\ ,\ x_{2b}[h_{2b}]),$$

що і є електронним підписом повідомлення m. Далі підписувач знищує значення x і, значить, більше не використовує його. Тобто, цей механізм схожий на механізм одноразового ЕП Lomport-Diffie [101]. Якщо необхідно підписати більше, ніж одне повідомлення, то можна використовувати «зчеплення». В

цьому випадку для підписання наступного повідомлення підписувач знову використовує згенерований відкритий ключ. При верифікації перевірник перевіряє перше підписане повідомлення, а також включає до нього новий відкритий ключ. Після того він зможе здійснити перевірку підпису наступного повідомлення. Підпис *n-го* повідомлення містить всі *n*-1 попередньо підписані повідомлення.

Згідно поглядів [101], криптографія на основі ґеш-функцій досить перспективний апарат, що може бути використаний у постквантових системах цифрового підпису. Криптографія на основі ґеш-функцій може використовувати будь-яку важко обернену функцію в безпечну систему підпису на відкритих ключах. Це дозволить ефективно протистояти атаці, що грунтується на основі використання алгоритму Гровера [81].

Механізми криптографічного перетворення з використанням збиткових кодів (СВ-криптографія)

Вважатимемо, що ціле b є степенем двійки. Запишемо згідно [99-101] параметри збиткового коду:

$$N = 4b \lg b\,;\ d = \lceil \lg n \rceil;\ t = 149.$$

Відкритим ключем в такій системі є матриця *K* розмірністю $dt \times n$ з коефіцієнтами у полі F_2. Повідомлення представляє собою *n*-бітовий рядок з вагою *t*, тобто, *n*-бітовий рядок у своєму складі має точно *t* бітів, які дорівнюють 1. Для цього необхідно здійснити попереднє форматування повідомлення. Далі, при зашифруванні повідомлення *m* відправник перемножує матрицю *K* на *m* та формує таким чином *dt*-бітовий шифротекст:

$$\mathrm{C} = \mathrm{K}m.$$

Головною проблемою для зловмисника при криптоаналізі є пошук синдрому декодування матриці *K*. Тобто, для того, щоб розв'язати цю систему рівнянь, необхідно знати вхідний вектор з вагою *t*. Це можна зробити методами лінійної алгебри. Для цього необхідно виконати зворотню операцію відновлення *Km* для деякого *n*-бітного вектору ν, такого, що $K\nu = Km$. Але існує експоненційно велика кількість векторів *νБ*, тому пошук серед них *t*-вектора є також експоненційно складним. Найменш складною відомою атакою є атака зі складністю e^b для більшості матриць *K*.

При верифікації отримувач генерує відкритий ключ *K* з таємною структурою, а саме, із структурою прихованого коду Гоппи. Це дозволить отримувачу провести декодування у прийнятний час. Зрозуміло, що порушник може виявити структуру прихованого коду Гоппи у відкритому ключі, але на цей час така атака невідома.

Більш конкретно. Для випадку шифрування отримувач починає з особливих елементів $\lambda_1, \lambda_2, \ldots, \lambda_n$ у полі F_{2^d} та таємного незвідного поліному системи *t* у полі $g \in F_{2^d}[x]$. При цьому основною складовою отримувача є формування синдрому декодування, тобто, матриці розмірністю $dt \times n$:

$$H = \begin{pmatrix} 1/g(\lambda_1) & \cdots & 1/g(\lambda_n) \\ \vdots & \ddots & \vdots \\ \lambda_1^{t-1}/g(\lambda_1) & \cdots & \lambda_1^{t-1}/g(\lambda_n) \end{pmatrix}.$$

В такій матриці кожен елемент є елементом поля F_{2^d} та розглядається як стовпчик із d елементів поля F_2 у стандартному базисі F_{2^d}. Ця матриця є частково контрольною матрицею для незвідного двійкового коду Гоппи і може бути декодована за допомогою алгоритму Патерсона або іншого швидкого алгоритму [99-101]. Відкритий ключ K є версією матриці H. Таємний (особистий) ключ також містить інвертуєму матрицю S розмірністю $dt \times dt$ і матрицю перестановки P розмірністю $n \times n$. Відкритий ключ є результатом спеціального перетворення. Отриманий шифротекст $Km = SHPm$ отримувач, для того, щоб отримати HPm, перемножує на S^{-}, а потім декодує H, щоб отримати Pm. Наостанок він перемножує результат на матрицю P^{-1} та отримує відкритий текст m.

Особливості реалізації ЕП розглянемо на прикладі криптосистеми Нідеррайтера. Вперше вона запропонована в [101] та отримала назву CFS ЕП. Доведення стійкості такої схеми зводиться до оцінки складності розв'язку задачі синдромного декодування. Так, при відомому таємному ключі при декодуванні стійкість можна вирішити тільки для деякої долі випадкових слів. Для цього використовується багаторазове гешування повідомлення, яке рандомізоване засобом цього гешування з використанням лічильника з довжиною в r бітів. При цьому підписувач повинен використати свій ключ для визначення відповідного вектора помилки. Далі таємний ключ використовується для визначення відповідного вектора помилок. На останок підписувач використовує цей вектор разом із значенням лічильника в якості підпису.

Розглянемо сутність та дамо оцінку стійкості такого механізму ЕП [101].

1. Генеруються загальні системні параметри: m, $t \in N$;

2. Генерується ключ генерації пари ключів, як в системі Нідеррайтера. Для цього використовується алгебраїчний код на основі ($n = 2m$, $k = n - mt$, $2t + 1$) двійкових незвідних поліномів Гоппи [101]. Далі для нього породжуються такі матриці:

- матриця H: $(n - k) \times n$, що є матрицею перевірки алгебраїчного коду, що має t - виправляючу спроможність;
- матриця X: $(n - k) \times (n - k)$ – випадкова обернена матриця;
- матриця P: $n \times n$ – випадкова матриця перестановок.

Відкритим ключем є матриця $H_x = X \cdot H \cdot P$ та t - виправляюча спроможність. Таємним ключем є матриці X, P та швидкий алгоритм декодування алгебраїчного коду.

3. Обчислення підпису:

Вхідними даними є такі:

а) h – функція гешування, застосовується гешування вхідних даних x, результатом є геш – значення $h(x)$ з довжиною $n - k$ біт;

б) поліноміально – складний алгоритм декодування алгебраїчного коду, що застосовується до синдромної послідовності $s = (s_0, s_1, \ldots, s_{n-k-1})$;

в) відкритий текст M, для якого потрібно обчислити ЕП.

Вихідними даними є ЕП згідно механізму CFS для відкритого повідомлення M.

Сутність алгоритму ЕП:

1. Обчислення геш – значення *h(M)* та присвоєння змінній *i* значення *i* =1.

2. Обчислення геш – значення *h(h(M))* та конкатенація його зі значенням *i*-ї змінної.

3. Значення *h(h(M))* розглядається як синдром у вигляді послідовності $S_X = (S_0, S_1, \cdots, S_{n-k+1})$, що обчислена для деякого довірного кодового слова та вектора помилок $e = (e_0, e_1, \cdots, e_{n-1})$.

4. Обчислюється значення вектора $S_X^{*T} = X^{-1} \cdot S_X^T$ та знаходиться вектор $e^{-T} = P \cdot e^T$.

5. Для послідовності (синдрому) S_X^* здійснюється швидке декодування і реалізується виконання швидкого (алгебраїчного) алгоритму декодування та, у випадку успіху, змінній *i* присвоюється значення *i=i+1* та здійснюється перехід до п.2.

6. Обчислюється значення вектора $e^T = P^{-1} \cdot e^{-T} = P^{-1} \cdot P \cdot e^T$.

7. Обчислюється ЕП *Y =(e,i)* для повідомлення *M* згідно схеми CFS.

8. Формування ЕЦП за схемою CFS *Y = (e,i)* для відкритого тексту M.

Алгоритм перевірки (верифікації) здійснюється таким чином:

Вхідними даними є такі.

1. Відкритий ключ у вигляді матриці $H_x = X \cdot H \cdot P$ та число *t*.

2. Відповідна функція гешування *h*.

3. Швидкий з поліноміальною складністю алгоритм декодування.

4. Значення ЕП *Y = (e,i)*;

5. Відкритий текст M.

Алгоритм верифікації зводиться до виконання наступного:

1. Обчислення вектора здійснюється згідно механізму CFS за такими кроками:

2. Обчислюється значення:

$$s''_X = h(h(M) \| i).$$

3. Приймається рішення про правильність чи помилковість у вигляді:

якщо $S'_X = S''_X$, то підпис правильний, інакше - помилковий.

Відносно стійкості механізму необхідно відмітити таке – згідно сучасних поглядів вона є експоненційно складною [101].

14.5 Сутність механізму ЕП на основі перетворення на алгебраїчних решітках (LB-криптографія)

Вперше сутність криптоперетворення на алгебраїчних решітках у 1996 році запропонував M. Aitai [78,83,88,89]. Решітка - це дискретна підмножина векторів (точок) у евклідовому просторі R^n, яка є замкненою за операціями додавання та віднімання векторів. Решітка має розмірність n, якщо вона розміщуються у будь-якому підпросторі простору R^n. При геометричному

поданні решітка - це множина, що рівноймовірно розміщена у просторі R. За своїми властивостями алгебраїчна решітка є абелевою групою. Основою побудування алгебраїчної решітки L є множина векторів B, таких, що будь-яка точка (вектор) L може бути представлена у вигляді лінійної комбінації елементів B. Стійкість LB-криптосистем базується на складності вирішення двох комбінаторних проблем [88,89]:

- проблеми найкоротшого вектора, вирішення якої зводиться до пошуку найкоротшого вектора у решітці L з базисом B;
- проблеми найближчих векторів, коли для заданого базису B решітки L і деякого вектору $\boldsymbol{v} \in \boldsymbol{L}$ необхідно знайти вектор $\boldsymbol{v}' \in \boldsymbol{L}$, що є найближчим до вектора $\boldsymbol{v}$.

Вказані проблеми у загальному випадку є експоненційно складними, наприклад, для більшості базисів B. Якщо базисні вектори є короткими та ортогональними, то обидві проблеми можна ефективно вирішити. Пошук таких базисів на решітці називається редукцією базису решітки і для цього можна використати відомий алгоритм LLL [83].

На нинішній час LB-криптосистеми можна поділити на дві такі групи:

- імовірно не стійкі, але ефективні криптосистеми, до яких можна віднести LB-криптосистеми, що грунтуються на перетвореннях в фактор – кільцях, наприклад, NTRU-алгоритм [83];
- імовірно стійкі, але не ефективні LB-криптосистеми. В якості прикладу такої криптосистеми можна назвати LWE-проблему (Learning with Errors). Складність такої криптосистеми (Ring-LWE) грунтується на чисто комбінаторній проблемі, яка дозволяє будувати криптосистеми, які є відносно ефективні і стійкі до атак із застосуванням квантових комп'ютерів. Механізми побудування таких криптосистем та оцінки їх властивостей (попередні) можна знайти в [83].

14.5.1 Сутність механізму ЕП на основі мультиваріативного квадратичного перетворення (MQ-криптографія)

Розглянемо сутність механізму ЕП на основі мультиваріативного квадратичного перетворення, спираючись на [76,77,102,103]. Відкритий ключ в мультиваріативній криптосистемі є послідовністю:

$$P_1, P_2, \dots, P_{2b} \in F_2[w_1, \dots, w_{4b}]$$

із $2b$ поліномів з $4b$ змінних $w_1, \dots, w_{4b}$ з коефіцієнтами у полі $F_2 \in \{0,1\}$. Кожний поліном повинен мати що найбільше степінь 2, без квадратичних термів та представлений як послідовність:

$$1, w_1, \dots, w_{4b}, w_1w_2, w_1w_3, \dots, w_{4b-1}w_{4b}.$$

В цілому, відкритий ключ має 16b3+4b2+2b бітів. Наприклад, для b=128 об'єм ключа складатиме 4 Mbyte.

Суттєвою перевагою ЕП на основі MQ-криптографії, наприклад, над HB-підписом є те, що підпис є коротким. Інші MQ-системи мають ще коротші підписи і, у ряді випадків, більш короткий відкритий ключ. Основною

проблемою для криптоаналітика відносно MQ-криптографії є пошук послідовності з 4b ($w_1, \ldots w_{4b}$) бітів, що породжує 2b вище зазначених вихідних бітів:

$$(P_1(w_1, \ldots w_{4b}), \ldots, P_{2b}(w_1, \ldots w_{4b})).$$

Імовірність вгадування послідовності з 4b бітів можна оцінити як 2^{-2b}.

ЕП виконується таким чином.

Зафіксуємо стандартний незвідний поліном $\varphi \in F_2(t)$ степені 3b. Визначимо L як поле $F_2(t)/_{\varphi}$ розмірністю 2^{3b}. Критичним кроком під час формування підпису є пошук кореня таємного одновимірного (*uniraviate*) поліному малої степені над L, а саме, поліному в $L[x]$ зі степенем не більше, ніж $2b$. Існує декілька стандартних алгоритмів для вирішення цієї задачі за час $b^{O(1)}$. Таємний поліном обирається таким чином, щоб мати всі ненульові експоненти виду 2^i+2^j або 2^i.

Якщо елементи $x \in L$ представлені у вигляді:

$$x_0 + x_1 t_1 + \cdots + x_{3b-1}, \text{ де } x_i \in F_2,$$

тоді

$$x_2 = x_0 + x_1 t^2 + \cdots + x_{3b-1} t^{6b-2},$$

$$x_4 = x_0 + x_1 t^4 + \cdots + x_{3b-1} t^{12b-4}.$$

і так далі.

Таким чином, $x^{2^i+2^j}$ є квадратичним поліномом зі змінними $x_{0,\ldots,}x_{3b-1}$. Деякі прості перетворення приховують структуру цього поліному і породжують відкритий ключ.

Таємний ключ підписувача має три компоненти:

1. Оборотну матрицю S розмірністю $4b \times 4b$ з коефіцієнтами в F_2.

2. Поліном $Q \in L[x, \nu_1, \nu_2, \ldots, \nu_b]$, де кожний терм має одну з шести можливих форм:

$$l\, x^{2^i+2^j},$$

де $$l \in L,\ 2^i < 2^j,\ 2^i + 2^j \leq 2b,$$

де $$l\, x^{2^i} \nu_j,$$

де $$l \in L,\ 2^i \leq 2b,$$

$$l \nu_I \nu_j,$$

$$l\, x^{2^i},$$

$$l \nu_j,$$

$$l.$$

Якщо $b=128$, то ми маємо $944b$ можливих термів, кожний з яких має 384-бітний коефіцієнт l, а загальний об'єм складатиме 443 Kbytes.

3. Матрицю T розмірністю $2b \times 3b$ рангу $2b$ з коефіцієнтами у F_2. Підписувач обчислює відкритий ключ таким чином:

$(x_0, x_1, \ldots, x_{3b-1}, v_1, v_2, \ldots, v_b)$, як S раз вектора $(\omega_1, \ldots, \omega_{4b})$ всередині фактор-кільця $L[\omega_1, \ldots, \omega_{4b}]/(\omega_1^2 - \omega_1, \ldots, \omega_{1b}^2 - \omega_{4b})$.

Далі обчислює:

$$x = \sum x_i t^i$$

та

$$y = Q(x, v_1, v_2, \ldots, v_b).$$

Подає y вигляді:

$$Y_0 + Y_1 t + \cdots + Y_{3b-1} t^{3b-1},$$

де кожний $Y_i \in F_2\ [\omega_1, \ldots, \omega_{4b}]$

та обчислює $(P_1, P_2, \ldots, P_{2b})$.

Підписування є зворотним. Воно здійснюється таким чином:

1. Починаючи з величини $P_1, P_2, \ldots, P_{2b}$, вирішуємо таємне лінійне рівняння $T(Y_0, Y_1, \ldots, YP_{3b-1}) = (P_1, P_2, \ldots, P_{2b})$ для того, щоб отримати значення $(Y_0, Y_1, \ldots, Y_{3b-1})$. Існує 2^b можливих варіантів рішення для $(Y_0, Y_1, \ldots, Y_{3b-1})$. Обираємо випадковим чином одне із них.

2. Обираємо випадково значення $v_1, v_2, \ldots, v_b, \in F_2$ і підставляємо його у таємний поліном $Q(x, v_1, v_2, \ldots, v_b)$, отримуючи поліном $Q(x) \in L[x]$.

3. Обчислюється $Y = Y_0 + Y_1 t + \cdots + Y_{3b-1} t^{3b-1} \in L$ та розв'язується $Q(x) = Y$, отримуючи $x \in L$. Якщо існує декілька коренів, починаємо процес з початку.

4. Записується x як:

$$x_0 + x_1 t + \cdots + x_{3b-1} t^{3b-1},$$

де

$$x_0, x_1, x_{3b-1} \in F_2.$$

Розв'язується таємне рівняння:

$$S(\omega_1, \ldots, \omega_{4b} = (x_0, \ldots, x_{3b-1}, v_1, \ldots, v_b)$$

та отримується підпис.

Далі, *HFE є* приховане рівняння в полі $Q(x) = Y$ з пропуском декількох бітів. Тобто, $Q(x) = V$ є еквівалентом $3b$ рівнянь, але публікується тільки $2b$ рівнянь. де «v»- означає «vinegar» - змінні $v_1, v_2, \ldots, v_b$.

Необхідно відмітити, що HFE перетворення, тобто, без пропусків бітів та без v-змінної, може бути атаковано за $2^{(lgb)^2}$ операцій при застосуванні атаки Grobner, але HFE^v - перетворення буде протистояти такій атаці.

14.5.2 Порівняльний аналіз ЕП для постквантового періоду

В цьому підпункті наводяться попередні результати порівняльного аналізу криптографічних перетворень, що є кандидатами постквантового періоду. У таблиці 14.7 надаються загальні характеристики математичного апарату, на яких грунтуються механізми ЕП, з використанням яких можуть бути розроблені

квантово-захищені алгоритми ЕП. Швидше всього, перспективними алгоритмами ЕП є методи на основі HB та MQE-криптографії. У той час як для механізмів направленого (асиметричного) шифрування швидше всього перспективними є методи на основі LB та CB-криптографії. Наведені в таблиці 14.8 механізми ЕП були запропоновані робочою групою ETSI для подальшого вивчення і дослідження у якості можливих кандидатів на квантово-захищені схеми ЕП.

Таблиця 14.7.

Характеристики напрямків, у яких можуть бути розроблені квантово-захищені алгоритми

Криптографічна схема	Підпис	Шифрування	Розмір ключа	Тип даних	Core Ops	Cryptographic Maturity
Hash-Based	Yes	No	≈20k	Hash outputs	Hashing	High
Multivariate Quadratic	Yes	No	≈10k	$GF(2^m)$	Matrix mult. LSE solving	Low, medium for conservative schemes
Lattice-Based: NTRU General lattice	Maybe Maybe	Yes Yes	<0.1k ≈100k	Z_q $GF(2^m)$	Convolution Matrix mult.	Medium Medium
Code-Based	Expensive	Yes	≈100k	$GF(2^m)$	Matrix mult. decoding	High, with precautions to implementation

Таблиця 14.8.

Порівняння довжин ключів та підписів для квантово-захищених алгоритмів

Тип	Схема	Безпечність (біти)	Відкритий ключ (байти)	Підпис (байти)
Lattice	Lyubashevsky	-------	1 664	2 560
	NTRU-MLS	128	988	988
	Aguilar et al	128	1 082	1 894
	Guneysu te al	80	1 472	1 120
	BLISS	128	896	640
	Ducas et al	80	320	320
	HIMMO	128	32	-----

Продовження табл. 14.8.

MQ	Quartz	80	72 237	16
	Ding	123	142 576	21
	UOV	128	413 145	135
	Cyclic-UOV	128	60 840	135
	Rainbow	128	139 363	79
	Cyclic-Rainbow	128	48 411	79
Code	Parallel-CFS	120	503 316 480	108
	Cayrel et al	128	10 920	47 248
	Cyclic- Cayrel et al	128	208	47 248
	RankSign	130	7 200	1 080
	Cyclic- RankSign	130	3 538	1 080
Hash	Merkle	128	32	1 731
	Leighton-Micali	128	20	668
	XMSS	256	64	8 392
	SPHINCS	256	1 056	41 000
Isogeny	Jao-Soukharev	128	768	1 280
	Sun-Tian-Wang	128	768	16

Таким чином, на цей час поки що важко визначити конкретну схему постквантового ЕП. Кожний з поданих варіантів має свої переваги та недоліки. У перспективі доцільно провести дослідження та уведення більш обґрунтованої методики та критеріїв порівняння ЕП постквантового періоду. На наш погляд, для порівняльного аналізу може бути застосована методика, що грунтується на застосуванні системи безумовних та умовних критеріїв, та обчислення на їх основі безумовного та умовного інтегральних критеріїв.

Висновки та рекомендації

1. При побудові моделей загроз у постквантовий період, за нинішніх умов, в якості основних необхідно брати методи криптоаналізу як основні моделі загроз, що можуть бути реалізованими на квантовому комп’ютері при вирішенні задач криптоаналізу. Причому усі вказані методи повинні бути орієнтовані на використання специфіки квантового комп’ютера та мови програмування на ньому.

2. До основних задач, які можуть бути вирішені на квантовому комп’ютері при реалізації загроз, в першу чергу необхідно віднести такі:

- квантовий алгоритм факторизації Шора;
- квантовий алгоритм Гровера;
- квантовий алгоритм Шора вирішення дискретного логарифму в скінченному полі;
- квантовий алгоритм Шора вирішення дискретного логарифму в групі точок еліптичної кривої;
- квантовий алгоритм криптоаналізу для перетворень в фактор – кільці.

3. У 2015 році фахівці компанії Google підтвердили, що, згідно з їх дослідженням, в комп'ютері D-Wave використовуються квантові ефекти.

Причому в «1000-кубітному» комп'ютері кубіти в дійсності організовані в кластери по 8 кубітів кожен. Якраз це і дозволило добитися в одному з алгоритмів швидкодії в 100 млн. разів більше, ніж у звичайному комп'ютері.

4. До можливих кандидатів на ЕП сформульовані такі мінімальні вимоги:

- відкритість алгоритму ЕП для криптографічної спільноти та аналізу їх;
- можливості реалізації ЕП у широкому діапазоні існуючих платформ;
- забезпечення, як мінімум, функції ЕП та її застосування у криптографічних протоколах, а також для направленого шифрування;
- наявність теоретичних та емпіричних доказів щодо забезпечення вимог безпеки (стійкості).

5. Більш конкретні вимоги формулюються у трьох напрямках: це вимоги з безпеки (вимоги до стійкості до криптографічного аналізу), техніко-економічні вимоги (обчислювальні витрати та витрати на пам’ять) та технічні характеристики реалізації алгоритмів. У таблиці 14.1 наведено вимоги NIST до постквантових криптографічних примітивів. По суті, наведені в таблиці 14.1 дані містять вимоги та рекомендації щодо першого етапу розроблення та тестування постквантових примітивів. Вони охоплюють широкий спектр та містять вимоги до безпеки постквантових криптографічних примітивів, техніко-економічні та техніко-експлуатаційні вимоги. Також в певній мірі запропоновані узагальнені критерії оцінки та порівняння можливих кандидатів.

6. На сьогодні ще остаються відкритими питання встановлення конкретних значень вказаних показників та визначення критеріїв порівняння. На протязі 2016 року ці параметри будуть визначені відповідною групою експертів. Але вже сьогодні NIST також надає деякі орієнтовні значення для окремих параметрів.

7. Наведені в таблицях 14.1 та 14.6 результати дозволяють зробити попередні оцінки постквантових механізмів ЕП та приймати певні рішення відносно подальших їх досліджень з метою вибору для стандартизації більш перспективних. У таблиці 14.3 наведені дані відносно таких криптоперетворень та на чому базується їх стійкість.

8. В процесі попереднього аналізу та досліджень в якості постквантових механізмів були вибрані такі:

- механізми ЕП на основі ґеш-функцій (HB-критографія);
- механізми ЕП на основі збиткових кодів (CB-криптографія);
- механізми ЕП на основі перетворення на алгебраїчних решітках (LB-криптографія) (див. Пункт 16.1);
- механізми ЕП на основі мультиваріативного квадратичного перетворення (MQ-криптографія);
- Механізми ЕП на основі ізогеній еліптичних кривих (див. Пункт 16.6).

9. У таблицях 14.7 та 14.8 наведені загальні характеристики математичного апарату, на яких грунтуються механізми ЕП, з використанням яких можуть бути розроблені квантово-захищені алгоритми ЕП. На наш погляд, швидше всього, перспективними алгоритмами ЕП є методи на основі HB та MQE-криптографії. У той час, як для механізмів направленого (асиметричного) шифрування швидше всього перспективними є методи на основі LB та CB-криптографії.

14.6 Модель порушника систем електронних цифрових підписів в умовах постквантового криптоаналізу

Розглянемо модель порушника в системах електронного цифрового підпису в постквантовому середовищі. Проаналізуємо можливість дій зловмисника із доступом до квантового випадкового оракула. Також проаналізуємо апарат теорії ігор та його застосування для формування моделі порушника в умовах квантового криптоаналізу. Наведемо модель загроз системи електронного цифрового підпису в постквантовому середовищі.

Вступ

Нині досить суттєве значення мають автоматизовані засоби автентифікації та ідентифікації, що знайшли своє застосування в різних криптографічних протоколах [46,106-119]. Ключову роль в забезпеченні контроля *цілісності* та *неспростовності* для вказаних технологій відіграють електронні цифрові підписи (ЕЦП). Із появою квантового комп'ютера очікується значна зміна поглядів на криптоаналіз та, як наслідок, на визначення кандидатів постквантового ЕЦП [60,109]. Оцінку основних результатів щодо квантового криптоаналізу проведено в [60,109-112]. Найбільш перспективні розробки схем цифрового постквантового підпису представлені в роботах [116,117,119]. Проте, немає точного та *усталеного* сприйняття моделі загроз та моделі порушника в умовах появи квантового комп'ютера. Метою даного розділу є підвищення ефективності розробки систем квантової криптографії за рахунок аналізу можливих методів квантового криптоаналізу на системи ЕЦП та побудови моделі порушника в постквантовому середовищі. Для досягнення поставленої мети вирішуються наступні задачі:

– розробка моделі порушника ЕЦП та управління ключовими даними із доступом до квантового комп'ютера;

– розробка моделі порушника ЕЦП із доступом до квантового випадкового оракула;

– розробка моделі загроз на системи ЕЦП в умовах появи квантового комп'ютера.

При побудові моделі порушника та загроз застосовується методика, що ґрунтується спочатку на побудові моделі порушника, виявленні усіх можливих загроз та визначенні способів їх реалізації, а на останок – на побудові моделі загроз ЕЦП для конкретних умов застосування. Представлені умови застосування ЕЦП моделюють використання квантового комп'ютера, що впливає на характер дій зловмисника та особливості загроз. В третьому пункті отримано кількісні оцінки захищеності ЕЦП від основних загроз. Побудовані моделі порушника та загроз дозволяють сформулювати вимоги до системи захисту інформації при автентифікації повідомлень засобами ЕЦП.

14.6.1 Загальна модель порушника та загроз ЕЦП

Моделі порушників розглядаються авторами також в підрозділі 12.2 – Методи перехоплення інформації у системах квантової криптографії.

Орієнтуючись на [60], під моделлю порушника будемо розуміти абстрактний формалізований чи неформалізований опис дій порушника, який відображає його практичні та теоретичні можливості, апріорні знання, час та місце дії тощо. Також може бути врахована наступна інформація, доступна противнику:

- криптографічні дані, доступні атакуючому (наприклад, атака з підібраним шифротекстом);
- визначені аспекти криптографічних алгоритмів (наприклад, чи доступно атакуючому визначення хеш-функції, використовуваної в алгоритмі – це може бути важливим фактором в моделі випадкового оракула);
- визначені аспекти виконання (наприклад, чорний ящик);
- обчислювальні ресурси, доступні атакуючому.

Проведений аналіз [60] показав, що порушник (супротивник) може здійснювати по відношенню до криптографічних примітивів пасивні або активні атаки. При пасивних атаках порушник може здійснювати:

- аналіз даних, перехоплених в процесі виконання криптографічного примітиву, наприклад, визначення таємного сеансового ключа за перехопленим відкритим ключем;
- розв’язання задач повного або універсального розкриття по відношенню до статичного, сеансового або одночасно і статичного, і сеансового ключів.

При активних атаках порушник може створювати та передавати хибні повідомлення та підписи, що є імітацією роботи істинного криптографічного примітиву.

Поряд з цим, в моделі порушника необхідно задавати інші його можливості [60]:

- міру доступу до каналу передачі даних між підписувачем та перевіряючим;
- можливості записування даних, що передаються в процесі виконання криптопримітиву ЕЦП;
- можливість зміни даних криптопримітиву (модифікування);
- можливість знищення даних;
- багаторазове використання даних примітиву;
- несанкціоновану ініціалізацію виконань алгоритму, перезапускання тощо.

Зловмисник в процесі протидії може знаходитися в різних умовах [60]. Враховуючи це, будемо розглядати наступних зловмисників:

- зовнішній, що має тільки загальнодоступні дані та відомості, наприклад, повідомлення, підписи ЕЦП, відкриті ключі, загальносистемні параметри;
- зловмисник, що має постійний чи тимчасовий доступ до критичної інформації, наприклад, до ключів сеансів, таємних даних тощо.

Крім вказаного, згідно [60], будемо також розглядати порушника з чотирма рівнями можливостей:

- нульовий рівень – випадкове навмисне ознайомлення зі змістом інформації (випадкове прослуховування в каналі);

- перший рівень – порушник має обмежені кошти та самостійно створює засоби і методи атак на засоби криптографічного захисту інформації (КЗІ), а також інформаційно-телекомунікаційні системи із застосуванням широко розповсюджених програмних засобів та електронно-обчислювальної техніки;

- другий рівень – порушник корпоративного типу має змогу створення спеціальних технічних засобів, вартість яких співвідноситься з можливими фінансовими збитками при втраті, спотворенні та знищенні інформації, що захищається. У цьому разі для розподілу обчислень при проведенні атак можуть застосовуватися локальні обчислювальні мережі;

- третій рівень – порушник має науково-технічний ресурс, який прирівнюється до науково-технічного ресурсу спеціальної служби економічно розвинутої держави.

Цілі порушника - створення нових та підвищення ефективності існуючих методів аналізу стійкості класичних та постквантових ЕЦП. Існує класифікація моделей порушника за мотивом. Підґрунтям цілеспрямованої реалізації порушником криптоаналізу є, найчастіше, корисливі мотиви. Раціональна модель мотивації зловмисника описує дії порушника в залежності від вигоди та витрат, які він отримує в результаті криптоаналізу. Наприклад, вибір стратегії дій зловмисником визначається прагненням одержати особисто максимальну вигоду та нанести максимальні втрати користувачеві (власникові). Існує також альтруїстична модель поведінки зловмисника, коли мотивація противника виходить за межі поняття вигоди і є більш руйнівною. Модель раціонально вмотивованого зловмисника представляється більш релевантною до обставин реального світу, ніж альтруїстична.

Захищеність ключів від дій зловмисників називається криптографічною живучістю. При цьому необхідно враховувати, що компрометація тільки довгострокових (статичних) ключів не приводить до компрометації ключів сеансів, що уже були застосовані раніше. В цілому, при здійсненні аналізу криптографічних примітивів типу ЕЦП необхідно зважати на наступне:

- якщо довгострокові особисті ключі ЕЦП скомпрометовані, то наступні сеанси здійснення протоколів не захищені від підробки;

- при компрометації статичних (довгострокових) ключів компрометація минулих ключів сеансів не відбувається.

В наукових колах розповсюджено декілька різних підходів до моделювання дій порушника. Розглянемо деякі з них. Відповідно до моделі загроз Долева - Яо [60], будь-яке повідомлення, що передається у мережі, може бути прослухане зловмисником. Із цього слідує, що будь-яке повідомлення, отримане користувачем із мережі, спочатку було перехоплено зловмисником, а потім передано адресату.

Зауваження 1. За даними твердженнями зробимо припущення, що зловмисник має контроль над всією мережею. Аналіз мережі, в якій використовуються різні протоколи, що включають криптопримітиви ЕЦП, виходить за рамки даного підрозділу. Посилені припущення щодо дій зловмисника в мережі робляться для того, аби результати даної роботи могли бути застосовані для моделі загроз будь-якої мережі, а структура доведення

загального протоколу включала би примітив ЕЦП у вигляді модуля, тобто, без врахування конкретних особливостей моделі загроз ЕЦП.

Необхідно зазначити також межі можливостей зловмисника. Якщо примітиви ЕЦП безпечні, то зловмисник не може вгадати випадкове число, обране із достатньо великого простору. Не маючи правильного особистого ключа, зловмисник не може відтворити підпис за його відкритим ключем або підписати повідомлення. Зловмисник не може знайти особистий ключ, знаючи тільки відкритий ключ.

Спочатку виділимо основну методику оцінки дій зловмисника, а потім застосуємо математичний апарат оцінки дій зловмисника щодо зламу основних постквантових ЕЦП. Визначимо такі кандидати ЕЦП на постквантовий період [60,109-112, 116, 117, 119]:

- на основі хеш-функцій (Hash Based);
- на основі завадостійких кодів (Code Based);
- на основі решіток (Lattice Based);
- на основі факторизації поліномів (Multivariate Polynomial Based).

Розгляд основних концепцій автентифікації з метою захисту від НСД ведеться, зазвичай, в частині криптографічних протоколів. Наведені концептуальні положення можуть бути застосованими й до примітивів ЕЦП.

Розглянемо спрощену модель середовища взаємодії [60] відправника та отримувача в умовах дій зловмисника. Ця узагальнена модель включає найкритичніший випадок, коли кожна сторона може бути зловмисником.

Таким чином, гарантії безпеки взаємодії засновані не на довіреній стороні чи не на припущенні, що більшість учасників є чесними, а на гарантії криптографічної стійкості ЕЦП. Таким чином, дана модель включатиме модель загроз ЕЦП, яку буде розглянуто далі. Такий підхід дасть змогу поширити результати моделі загроз ЕЦП і на середовище, де сторони не довіряють один одному. Учасники моделі середовища взаємодії це:

- D1, D2 - відправник та отримувач відповідно;
- КРА – криптоаналітик (зловмисник);
- випадковий оракул.

Оскільки сторони $D_1 D_2$ можуть виявитися зловмисниками, проаналізуємо основні погрози:

1) D_1 відмовляється від факту передачі повідомлення M_i в мережу;

2) D_1 намагається обманути, що він сформував та передав деяку інформацію M_i ;

3) підміна джерелом D_1 часу передачі інформації M_i ;

4) підміна передачі інформації M_i на інформацію M_i';

Основні погрози, що може реалізувати джерело D_2 :

1) D_2 формує деяку інформацію M_i' та стверджує, що він отримав її від D_1;

2) підміна джерелом D_2 часу передачі інформації M_i ;

3) D_2 відмовляється від факту отримання повідомлення M_i .

Основні погрози, що може реалізувати КРА:

1) імітація помилкового повідомлення M_i КРА у момент часу, коли D_1 пасивний, створює помилкову M_i інформацію і передає її D_2 ;

2) модифікація правильної інформації M_i у випадку, якщо D_1 передає D_2 деяку інформацію M_i, КРА модифікує інформацію M_i в M_i' і передає її D_2;

3) нав'язування раніше створеної інформації (повтор), тобто, КРА в будь-який момент часу t_j передає її ще раз D_2 , коли D_1 пасивний.

В даній моделі представлений випадковий оракул. Всі учасники моделі мають доступ до оракула, який обчислює випадкову функцію. При кожному новому запиті значення функції на даному аргументі вибирається випадковим чином. При цьому оракул запам'ятовує пару (аргумент, значення) і при повторному запиті для цього аргументу, незалежно від того, хто з учасників його видав, буде повернуто те саме запам’ятоване значення. Модель із випадковим оракулом була введена для протоколів електронного підпису, де випадкова функція замінила хеш-функцію. Згодом модель було узагальнено на криптографічні протоколи різних типів, і в даний час в криптології значна частина результатів про стійкість доводиться в цій моделі. В класичному варіанті сам алгоритм підпису та всі інші учасники, крім зловмисника, здійснюють доступ до випадкового оракула за допомогою класичних бітових рядків. Під класичним входом/виходом розуміється звичайний бітовий рядок. При цьому на вхід випадкового оракула подається один рядок, що представляє аргумент, а інший представляє вихідне значення.

Класичний комп’ютер працює зі скінченним числом біт. Кожен біт може знаходитися в одному з двох станів 0 чи 1. Стан всієї системи задається значенням всіх бітів. Тому множина станів класичного комп'ютера $B^n = \{0,1\}^n$ скінченна та має потужність 2^n. Під позначенням $\{0;1\}^*$ будемо розуміти множину скінченних бінарних рядків, а під $\{0\,;1\}^\infty$ будемо розуміти нескінченні рядки. Тоді R - це відображення із $\{0\,;1\}^*$ на $\{0\,;1\}^\infty$, визначене вибором кожного біта $R(x)$ випадково та незалежно для кожного x. Відповідно, ЕЦП – це кортеж $(G, Sign, Verify)$ поліноміальних алгоритмів генерації ключів, підпису та перевірки. Перші два з них імовірнісні, а останні два мають доступ до випадкового оракула. Отримуючи на вхід рядок 1^k (максимального обсягу інформації із довжиною k, де k - параметр безпеки), генератор створює пару (pk, sk) публічний та секретний ключ за поліноміальний час. Зловмисник ЕЦП A - це невипадковий поліноміальний алгоритм F із доступом до R та оракула підписання. Вихід F - це пара (m, σ) повідомлення та підпис, причому, m не є результатом роботи оракула підписання.

Ґрунтуючись на наведеному, розглянемо модель порушника ЕЦП в умовах появи квантового комп’ютера, на основі якої побудуємо модель загроз ЕЦП.

14.6.2 Модель порушника та загроз ЕЦП в умовах появи квантового комп'ютера

Модель порушника ЕЦП в умовах появи квантового комп’ютера включає в себе наступні аспекти. Метою зловмисника є підробка підпису повідомлення та компрометація особистих ключів ЕЦП [60]. На відміну від протоколів

автентифікації, ми виключаємо випадки, коли зловмисник ненавмисно реалізує криптоаналіз [60]. Кваліфікація порушника – сукупність певних знань і вмінь порушника, які він використовує для порушення роботи криптопримітиву ЕЦП. Можна визначити кілька типів кваліфікації порушників, що дозволять успішно реалізувати загрози системам постквантової криптографії [79,119 - 121]:

- порушник володіє інформацією щодо математичного базису криптографічних перетворень конкретного ЕЦП, що дозволить створити нові методи криптоаналізу, відповідно до рівня криптографічного захисту;

- порушник володіє високим рівнем знань в галузі обчислювальної техніки (зокрема, криптографії, теорії алгоритмів та паралельних обчислень тощо) і програмування криптографічних засобів автентифікації;

- порушник досконало володіє знаннями квантової фізики, квантової інформатики тощо, а також навичками роботи з обладнанням, що складає квантовий комп'ютер [123];

- порушник має доступ до глобальних обчислювальних мереж, суперкомп’ютера чи квантового комп’ютера, за допомогою якого може реалізувати, наприклад, силову атаку, використовуючи відомі квантові алгоритми Шора, Гровера тощо [107].

Можливості порушника в умовах появи квантового комп'ютера представлені нижче.

1) Порушник має можливість запуску програмного забезпечення, що реалізує певні функції з обробки класичного та квантового криптоаналізу. Знання мов програмування дозволить порушнику реалізувати створені методи криптоаналізу, а також модифікувати існуючий програмний код легітимних користувачів.

2) Порушник може створювати нові алгоритми класичного та квантового криптоаналізу для подальшого одержання необхідної порушнику інформації шляхом створення програмного забезпечення та модифікування існуючого.

Теоретичний аналіз стійкості примітивів постквантової криптографії будемо проводити, виходячи з того, що порушник використовує технічні можливості не тільки квантової механіки та інформатики, а й класичні методи криптоаналізу або їх поєднання, що поширює картину загроз, які треба враховувати.

Розглянемо розробку моделі порушника ЕЦП із доступом до квантового випадкового оракула. В останні роки в літературі, особливо закордонній, робиться спроба знайти зручний та “суворий” математичний апарат доказу постквантової стійкості криптографічних примітивів за допомогою отримання доказів класичної стійкості до доказів постквантової стійкості, використовуючи апарат теорії ігор [122], формулювання умов, коли такі відомості є ефективними. Також вивчається модель зловмисника із доступом до квантового випадкового оракула та відмінність від моделі порушника із доступом до класичного випадкового оракула [103]. Ці роботи також розглядають методику оцінки стійкості ЕЦП. В [124] продемонстровано, як поняття криптографічної безпеки можуть бути представлені за допомогою використання принципів формалізації теорії ігор.

Особливістю ЕЦП, як криптографічних перетворень, є те, що ключі асиметричної пари генеруються особисто кожним власником цієї пари у складі особистого та відкритого ключів. Це означає, що власник ключової пари може згенерувати її, використовуючи спеціальні засоби, в тому числі такі, що створюються порушником для шахрайства. Будемо вважати, що ці шахрайські дії здійснює порушник 1, 2 або 3 рівнів [60]. Будемо також вважати, що ЕЦП здійснюється з використанням криптографічних перетворень, які є постквантово стійкими. В якості моделі порушника виберемо таку, що базується на спробі селективної або екзистенційної підробки повідомлень, що підписуються. Селективна підробка представляє загрозу, спрямовану на створення правильного ЕЦП для попередньо вибраного повідомлення M_i. Екзистенційна підробка представляє загрозу, яка полягає в створенні порушником для повідомлення M_i, можливо навіть такого, що не має змісту, правильного ЕЦП.

Розглядається санкціонований користувач-порушник, який здійснює такі зловмисні дії. Робиться спроба для повідомлень M_i та M_j виробити ідентичний ЕЦП. Далі порушник (зловмисник) може маніпулювати цими підписаними повідомленнями, пред'являючи або передаючи при реалізації загроз те чи інше повідомлення. Для моделювання дій зловмисника будемо оперувати поняттям модель випадкового оракула. Також вважатимемо, що всі сторони (підписувач, перевіряючий та зловмисник) можуть посилати запити до випадкового оракула, який обчислює хеш-функції для них та забезпечує послідовні та несуперечливі відповіді для всіх учасників взаємодії. Питання моделювання випадкового оракула для квантового зловмисника досить важливе та складне. Для цього поняття суттєвим є визначення квантового комп'ютера та відмінності квантових обчислень від класичних. Квантовий комп'ютер працює зі скінченним набором елементарних станів, що називаються кубітами. Кожний кубіт має два виділених стани (якщо кубіти вважати спінами, то ці стани "спін вгору", "спін вниз"). Визначення виділених станів для кожного кубіта системи задає не всі можливі стани системи, а тільки базисні. Можливі також будь-які лінійні комбінації базисних станів із комплексними коефіцієнтами. Базисні стани будемо позначати $|x_1, \dots, x_n\rangle$, де $x_n \in B$ або $|x\rangle$, де $x \in B^n$. Довільний стан системи може бути представлений у вигляді:

$$|\psi\rangle = \sum_{(x_1,\dots,x_n)\in B^n} c_{x_1,\dots,x_n} |x_1,\dots,x_n\rangle,$$

де

$$\sum_{(x_1,\dots,x_n)\in B^n} \left|c_{x_1,\dots,x_n}\right|^2 = 1.$$

Простір станів для такої системи - скінченний (розмірності 2^n) простір над полем комплексних чисел. Визначимо відмінність станів класичного та квантового комп'ютера [121]:

- стан класичного комп'ютера, що складається із бітів, виражається як $x_1 x_2 \dots x_n x_j \in B$;

- стан квантового комп'ютера складається із кубітів. Існує базисний стан, що виражається як $|x_1, \dots, x_n\rangle, x_n \in B$, і довільний стан, який виражається як $\sum_{x \in B^n} c_x |x\rangle$, де $\sum_{x \in B^n} |c_x|^2 = 1$.

Обчислення можна представити, як послідовність перетворень над множиною станів системи. Опишемо, які перетворення можливі в класичному, а які - в квантовому випадку [121]:

- для класичного випадку перетворення - це функція із B^n в B^n;

- для квантового випадку перетворення - це унітарні оператори, тобто, оператори, які зберігають довжину вектора $\sum_{x \in B^n} |c_x|^2$.

Більшість сучасних моделей випадкових оракулів є доказово стійкими по відношенню до зловмисника із класичним доступом до випадкового оракула. В цих моделях зловмисник має доступ до оракула хеш-функції $O : \{0,1\}^* \to \{0,1\}^*$, він може дізнатися значення $O(x)$ за допомогою подачі запитів до оракула O в класичному стані x. Для вивчення конкретної системи випадковий оракул заміняється конкретною хеш-функцією, що змушує зловмисника оцінювати цю хеш-функцію через квантовий стан x [107].

Для того, щоб змоделювати цю властивість, ми дозволимо зловмиснику оцінювати випадкового оракула в суперпозиції (або через явище суперпозиції). Це означає, що зловмисник може відправляти квантові стани $\langle\varphi| = \sum c_x |x\rangle$ до оракула O та отримувати назад "оцінені" стани $\sum c_x |O(x)\rangle$ (у відповідному представленні для того, щоб зберегти унітарність перетворення). На вхід випадкового оракула подаються запити в суперпозиції, а на виході отримують суперпозицію відповідей. Суперпозиція - це лінійна комбінація 2^n можливих векторів, взятих із комплексними коефіцієнтами, де кожен з 2^n векторів є точним аналогом класичного бітового рядка. Таку модель будемо називати *модель випадкового оракула із квантовим доступом* (ВОКД). У той же час сам алгоритм підпису та всі інші учасники, крім зловмисника, здійснюють доступ до випадкового оракула за допомогою класичних бітових рядків. Під класичним входом/виходом розуміється звичайний бітовий рядок, а під квантовим - набір кубітів.

Доказ безпеки в моделі ВОКД значно важчий, ніж в класичній моделі. В якості прикладу розглянемо цифровий підпис. Стандартна стратегія доказу безпеки в класичних налаштуваннях - зловмисник обирає запити до випадкового оракула, а відповіді оформляє як результат вирішеної ним задачі. Таким чином, ця відповідь може бути оформлена, як вирішення задачі підробки підпису. Якщо зловмисник зробить q випадкових запитів до випадкового оракула, то задача вирішується із вірогідністю $\frac{1}{q}$, а оскільки q є поліноміальним алгоритмом, то вірогідність "успішного" результату є досить великою для доказу безпеки в класичному налаштуванні випадкового оракула. На жаль, така стратегія оцінки дій зловмисника в ВОКД повністю втрачає сенс [87]. Кожний запит до ВОКД

потенційно оцінює випадкового оракула в експоненційно великій кількості точок.

Якщо в класичній схемі випадковий оракул може бути змодельований безпосередньо за допомогою ідеальної хеш-функції або абстрактної алгоритмічної моделі (запит - перевірка попереднього значення - вибір генерації випадкового числа із записом або читанням раніше існуючого значення - відповідь), то в квантовому випадковому оракулі вся функція повинна бути згенерована заздалегідь, адже у відповідь, навіть на одноразовий запит, квантовий оракул видає безліч відповідей одразу. Таким чином, неможливо об'єднати відповіді від одного запиту. Це показує, що доказ безпеки в класичному випадковому оракулі не обов'язково доводить постквантову безпеку.

Нижче подані відмінності між квантовим та класичним оракулом.

Класичні випадкові оракули є адаптивними. Це дозволяє програмувати їх відповіді для зловмисника, який користується адаптивно підібраними відповідями. Оскільки ВОКД забезпечує експоненційно велику множину відповідей в суперпозиції їх станів на запит, то це ускладнює програмування адаптивного ВОКД. Однією з відмінностей класичного випадкового оракула, порівняно з квантовим, є те, що в його моделі симулюються прообрази (pre-images), потрібні криптоаналітику. В ВОКД відповідь на запит може бути схована в суперпозиції в експоненційно великому наборі квантових станів і не зрозуміло, як отримати необхідні відповіді.

В класичному випадку можна ефективно змоделювати експоненційного випадкового оракула, просто обираючи випадкові відповіді. У випадку оракула з квантовим доступом зловмисник може отримати всі відповіді одночасно на всі вхідні запити. Це ускладнює стратегію запит-відповідь, яка застосовується в класичних випадкових оракулах і яка є більш узгодженою.

Деякі докази, засновані на випадковому оракулі, застосовують тактику повернення в попередній стан алгоритму (Rewinding/Partial Con-sistency), коли певні хеш - значення залишаються незмінними й, принаймні, одне хеш - значення серед них змінюється. Для ВОКД існує чимало труднощів впровадження такої техніки, зокрема, неможливо змінювати хеш - значення непомітно.

Далі наведемо модель зловмисника при екзистенційній підробці. Нехай є зловмисник A та чесна сторона C. Модель зловмисника у вигляді взаємодії цих двох сторін назвемо моделлю гри G^{FOR} (див. алгоритм 1) [106].

Алгоритм 1

Схема підпису: $\Pi = (KGen, Sign, Vrfy)$.

1. C генерує $(pk, sk) \leftarrow KGen(1^n)$. Відправляє pk зловмиснику A.

2. A робить запити на формування підписів від повідомлень $\{m_i\}$. C повертає $\sigma_i := Sign(sk, m_i)$. Ці повідомлення обираються адаптивно самим A.

3. A формує $(m*, \sigma*)$. Якщо $Vrfy(pk, (m*, \sigma*)) = 1$ та $m* \notin \{m_i\}$, C оголошує успіх. Інакше оголошує невдачу.

В літературі широке розповсюдження знайшла наступна техніка. Якщо розглядати послідовність ігор, то кожна наступна гра схожа на попередню з невеликою різницею в постановці експерименту. Буквально моделюється декілька схожих моделей гри зловмисника і, на підставі схожості цих ігор, можна дійти до причино-наслідкових зв'язків у різних моделях зловмисника при доказі безпечності ЕЦП щодо всіх можливих варіантів поведінки зловмисника. Врахуємо, що існує можливість вразливостей постквантових ЕЦП до класичного криптоаналізу. Тоді доцільно розглядати не послідовність ігор, як в роботах [106-108], а їх паралельну комбінацію. Тобто, нехай є класична гра G_c^{FOR} та квантова гра G_q^{FOR}, які обчислювально незначною мірою відрізняються від алгоритму 1. Тоді загальна вірогідність успіху зловмисника дорівнює 1, якщо він має успіх принаймні в одній з ігор, тобто $Pr(G_c^{FOR}) + Pr(G_q^{FOR}) = 1$. При цьому, в першій грі зловмисник використовує класичний комп'ютер, а в другій - квантовий. Відповідна модель зловмисника має формалізований вигляд алгоритму 2.

Алгоритм 2

Схема підпису: $\Pi = (KGen, Sign, Vrfy)$.

1. C генерує $(pk, sk) \leftarrow KGen(1^n)$. Відправляє pk зловмиснику A.

2. A робить запити на формування підписів від повідомлень $\{m_i\}$. C повертає $\sigma_i := Sign(sk, m_i)$. Ці повідомлення обираються адаптивно самим A.

3. A формує $(m*, \sigma*)$. Якщо $Vrfy(pk, (m*, \sigma*)) = 1$ та $m* \notin \{m_i\}$, то $C_1 = 1$. Інакше $C_1 = 0$.

4. A формує $(m*, \sigma*)_{QuantComp}$. Якщо $Vrfy(pk, (m*, \sigma*)_{QuantComp}) = 1$, $C_2 = 1$. Інакше $C_2 = 0$.

5. Якщо $C_1 \vee C_2 = 1$, C оголошує успіх. Інакше оголошує невдачу.

Аналіз моделі загроз, наведеної в алгоритмі 2, показав, що криптопримітиви ЕЦП потрібно конструювати з врахуванням вимог часу та використовувати математичний базис квантовостійкої однопрохідної функції. При цьому потрібно оцінювати загальну захищеність від зламу за допомогою класичних обчислень, що ускладнює і модель зловмисника.

Доказ різних випадків моделей зловмисника та зведення моделі із класичного випадку до квантового потребують подальших досліджень.

Розглянемо модель загроз ЕЦП, яка випливає із відомих класичних та квантових моделей загроз, та з'ясуємо особливості вищезазначених підходів за умови виникнення квантового комп'ютера. Для цього зробимо класифікацію за типом загроз, що несуть дії порушника:

1. *Атака на основі відомого відкритого ключа* (key-only attack). Практично завжди доступна криптоаналітику (порушнику), оскільки може виконуватися при знанні загальносистемних параметрів й діючих відкритих ключах.

2. *Атака на основі відомих підписаних повідомлень* (known-message attack). В розпорядженні криптоаналітика, разом із даними з попередньої атаки, є деяке число

пар $(m, signature)$ підписаних повідомлень m, при цьому відсутня можливість вибору повідомлення m зловмисником.

3. *Проста атака з вибором підписаних повідомлень* (generic chosen-message attack). Криптоаналітик має можливість вибрати деяку кількість підписаних повідомлень, знає загальносистемні параметри і має після вибору підписаних повідомлень доступ до відкритих ключів.

4. *Спрямована атака з вибором повідомлення* (direct chosen-messa-ge attack). Криптоаналітик знає загальносистемні параметри, може на свій розсуд вибрати відкритий ключ і після цього вибрати підписані повідомлення.

5. *Адаптивна атака з вибором підписаного повідомлення* (adaptive chosen-message attack). При здійсненні атаки криптоаналітик може вибирати відкритий ключ, а також підписане повідомлення. При цьому вибір наступного підписаного повідомлення він може робити на основі знання припустимого підпису попередньо обраного повідомлення.

Проведений аналіз показав, що кожна атака спрямована на досягнення визначеної мети. З урахуванням цього, для всіх схем ЕЦП у порядку зростання небезпеки можна виділити такі види загроз [60]:

1. Екзистенційна підробка (existential forgery). Загроза полягає в створенні підпису для будь-якого, можливо безглуздого, повідомлення m', що відрізняється від дійсного повідомлення, що підписується.

2. Селективна підробка (selective forgery) - це можливість створення для заздалегідь обраного повідомлення m правильного ЕЦП.

3. Універсальна підробка (universal forgery) - знаходження криптоаналітиком алгоритму формування підпису, функціонально еквівалентного дійсному алгоритму ЕЦП, без використання або із використанням дійсного особистого ключа.

4. Повне розкриття (total break). При цій загрозі криптоаналітик може обчислити таємний ключ, який відповідає відкритому ключу *Q*.

Це дозволяє криптоаналітику формувати цифрові підписи для будь-яких повідомлень і надалі нав'язувати ці повідомлення кореспондентам.

Розглянемо модель загроз ЕЦП у відповідності до наступної методики редукції доказів. Атакуючий намагається досягти певних зловмисних цілей (які складатимуть компрометацію системи). Компрометація може включати обчислення особистого ключа або підробку двох певних повідомлень. Редукція, яка становить доказ, це доказ того, що якщо атакуючий може досягти мети компрометації, то атакуючий (з невеликим додатковим обсягом роботи) здатний досягти і деякої іншої мети, додаткової компрометації, можливо пов'язаних з "примітивом" нижчого рівня, на якому заснована схема. Твердження, що доводиться, має вигляд: “Якщо противник може зробити 'X', то він також повинен вміти робити 'Y'“. Зазвичай конструюється такий доказ, щоб 'Y' було будь-яким видом обчислень, яке вважається важко здійснюваним. Якщо приймається припущення, що противник не може зробити 'Y', тоді слідує логічний висновок доказу, що приймається також і те, що противник не може зробити 'X' (принаймні, якщо він діє в рамках даної моделі атаки).

Відповідно до вищевказаних методів редукції доказів безпеки досить просто класифікувати вищевказані погрози ЕЦП за додатковою складністю криптоаналізу в порівнянні із криптоаналізом більш загальної задачі зламу. Так, якщо противник може зробити атаку типу повне розкриття, то він також повинен вміти робити всі інші види атак із додатковою поліноміальною складністю. Очевидно, якщо противник може зробити атаку типу універсальна підробка, то він також повинен вміти робити всі інші види атак підробки із додатковою поліноміальною складністю. У свою чергу, селективна підробка включає, за логікою редукції, можливість виконання екзистенційної підробки не вище, ніж із додатковою поліноміальною складністю. Дану редукційну модель представимо в вигляді однонаправленого графа, вершини якого представляють тип атаки (рис. 14.1). Стрілки позначають редукцію переходу із одного типу атаки до іншого.

Зв'язки в редукційній моделі для класичного криптоаналізу позначають відношення більш загального виду атаки до часткового випадку. Але якщо противник може зробити атаку типу екзистенційна підробка, то він не обов'язково зможе зробити атаку повного розкриття із додатковою поліноміальною складністю. В противному випадку підпис є нестійким.

Ця схема справедлива для загального випадку і є, швидше, ідеальною моделлю, тому вона може змінюватися в залежності від різних умов. Порушення чи видозмінення цієї схеми залежать від існування тих чи інших атак на конкретний підпис та від базових параметрів ЕЦП. З'ясуємо ці зміни в редукційній моделі у випадку атак із застосуванням квантового алгоритму Гровера в загальному вигляді. Припустимо, зловмиснику доступний квантовий комп'ютер, який використовує алгоритм Гровера квантового пошуку із часовою складністю [120] $O\left(\sqrt{N}\right)$, що зменшилась по відношенню до складності класичних алгоритмів як корінь квадратний від об'єму обчислень класичних алгоритмів.

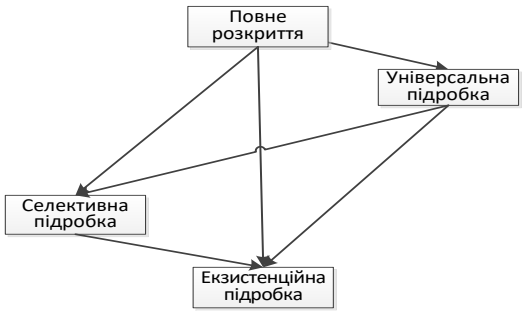


Рис.14.1. Схема редукційної моделі атак на ЕЦП

Оскільки алгоритм Гровера є атакою повного перебору, то відмінність в складності різних типів атак, що досліджуються, буде залежати від вхідних даних алгоритму. Множина вхідних даних для атаки повного розкриття визначається областю допустимих значень множини ключів, що кількісно виражається довжиною ключа. Множина вхідних даних для атак підробки визначається областю допустимих значень повідомлень чи підписів (довжина

повідомлення чи підпису). Із цього слідує, що за умови, якщо конкретний алгоритм ЕЦП має однакову довжину ключа та підпису, то для атак на квантовому комп'ютері із використанням алгоритму Гровера змінюється схема зв'язків в редукційній моделі (рис. 14.2). Оскільки алгоритм грубого пошуку не оптимізований під яку-небудь конкретну задачу, то складність атак типу повне розкриття та універсальна підробка буде тією самою.

Так, повний перебір ключів алгоритмом Гровера змінює складність $O(N) \to O(\sqrt{N})$ у порівнянні з класичним повним перебором, а складність знаходження колізій зменшується, у порівнянні із класичною атакою, як $2^{n/2} \to 2^{n/3}$. Тобто, для алгоритму Гровера повний перебір ключа має складність $2^{n/2}$, яка не набагато більша, ніж складність знаходження колізій $2^{n/3}$. Відповідно до вищесказаного, атаки у загальному вигляді мають наступні оцінки:

- атака повного розкриття має складність $2^{n/2}$;
- атака універсальної підробки має складність $2^{n/2}$;
- атака селективної підробки має складність $2^{n/2}$;
- атака екзистенційної підробки має складність $2^{n/3}$;

Таким чином, редукційна модель буде мати вигляд, представлений на рис. 14.2.

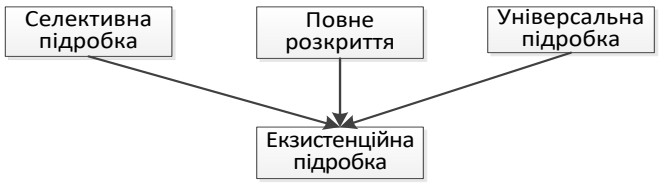


Рис. 14.2. Схема редукційної моделі атак на ЕЦП у випадку застосування квантового алгоритму Гровера

Запропонована у даному підрозділі класифікація методів криптоаналізу дозволяє чітко визначити напрямки подальших досліджень щодо розробки методів та побудови систем захисту інформації. Врахування даних зауважень може бути корисним при розробці нових чи оцінці існуючих моделей захисту ЕЦП. Розроблена модель порушника в квантових системах захисту інформації дозволяє визначити напрямки підвищення безпеки постквантових криптопримітивів ЕЦП.

14.6.3 Оцінка загроз схеми ЕЦП для класичних та квантових атак

З метою завершення аналізу моделі порушника ЕЦП, що діє в умовах появи квантових обчислень та їх практичного застосування, зробимо основні кількісні оцінки загроз класичних і постквантових ЕЦП на основі складності відомих атак. Для цього розглянемо оцінки стійкості перетворень в кільці (RSA), скінченному полі (DSA) та групі точок еліптичних кривих (EC-DSA). Спочатку

визначимо складність криптоаналізу для перетворень у кільці. Очевидно, найменш складним буде криптоаналіз, що застосовується на факторизації модуля перетворення N з використанням загального решета числового поля [60]:

$$I_K = \exp\left\{\delta\left(\log N\right)^{\nu}\left(\log\log N\right)^{1-\nu}\right\}. \quad (14.1)$$

При криптоаналізі криптографічних перетворень у полі Галуа $GF(p)$ найбільш складною є задача розв'язання дискретного логарифмічного рівняння:

$$Y_i = \theta_v^{X_i} (\mathrm{mod}\, p). \quad (14.2)$$

Складність розв'язання (14.2) також може бути оцінена з використанням формули [60]:

$$I_K = \exp\left\{\delta\left(\log N\right)^{\nu}\left(\log\log N\right)^{1-\nu}\right\}. \quad (14.3)$$

Якщо розв'язок базується на використанні решета числового поля, то для дискретного логарифмування у (14.3) треба застосовувати $\delta = 1/92, \nu = 1/3$, а для випадку факторизації у (14.1) застосовуються $\delta = 1,90, \nu = 1/3$ [60].

Складність криптоаналізу в групі точок ЕК при використанні оптимального методу ρ −Полларда можна оцінити як:

$$I^2 - 1 + n\ \ ln(1 - P_k) = 0 \quad (14.4)$$

або за спрощеною формулою

$$I_\rho = \sqrt{\frac{\pi \cdot n}{4}} \text{ або } I_\rho \approx \sqrt{-2n\ln\left(1-P_k\right)},$$

де n – порядок базової точки G в групі точок ЕК.

На основі вище перерахованих оцінок приведемо аналіз існуючих ЕЦП.

Приклад. Необхідно порівняти криптоперетворення в кільцях (RSA), полях (DSA) та групі точок ЕК (EC-DSA) за критерієм складності виконання та визначити вартість криптоаналізу методом повного розкриття, при якому криптоаналітик знаходить секретний (особистий) ключ абонента, якщо довжини модулів криптоперетворень в кільці, полі та групі точок ЕК відповідно дорівнюють $l = 336$ бітів.

Розв'язання. Для випадку факторизації маємо:

$$I_K = \exp\left[1.90\left(\ln 2^{336}\right)^{\frac{1}{3}}\left(\ln\ln 2^{336}\right)^{\frac{2}{3}}\right] = \exp\left(1.90 \cdot 6{,}17 \cdot 3{,}1\right) =$$

$$= \exp\left(39{,}78\right) = 8 \cdot 10^{15}.$$

Для випадку дискретного логарифмування маємо:

$$I_{П} =$$

$$= \exp\left[1{,}92\left(\ln 2^{336}\right)^{\frac{1}{3}}\left(\ln\ln 2^{336}\right)^{\frac{2}{3}}\right] = \exp\left[1{,}92\left(\frac{\log_2 2^{336}}{\log_2 e}\right)^{\frac{1}{3}}\left(\ln\left(\frac{\log_2 2^{336}}{\log_2 e}\right)\right)^{\frac{2}{3}}\right] =$$

$$= \exp\left[1{,}92\left(\frac{336}{1{,}43}\right)^{1/3}\left(\ln\left(\frac{336}{1{,}43}\right)\right)^{2/3}\right] = \exp\left[1{,}92\cdot\left(234{,}96\right)^{1/3}\left(\ln 234{,}96\right)^{2/3}\right] =$$

$$= \exp(1{,}92 \cdot 6{,}17 \cdot 3{,}1) = \exp(36{,}7) = 8{,}9 \cdot 10^{15}.$$

Для дискретного логарифмування в групі точок ЕК маємо:

$$I = \sqrt{\frac{\pi \cdot n}{4}} = \sqrt{\frac{3{,}14 \cdot 2^{336}}{4}} = 0{,}886 \cdot 2^{168} = 8{,}86 \cdot 10^{49}.$$

Порівняння отриманих результатів дозволяє зробити такі висновки:

- криптографічна стійкість проти атаки «повне розкриття» при застосуванні криптоперетворення в групі точок еліптичної кривої має експоненційний характер і суттєво перевищує стійкість криптоперетворень у кільці (факторизація) та в скінченному полі Галуа (дискретне логарифмування);

- стійкість криптоперетворень у кільці та в скінченному полі має субекспоненційний характер і оцінки значень складності криптоаналізу практично співпадають.

Для постквантових застосувань більшість відомих та стандартизованих алгоритмів ЕЦП є непридатними, бо складність квантового криптоаналізу дуже низька [113]. Тому основні перспективні дослідження і розробки криптографічних засобів захисту інформації в нинішній час спрямовані на пошуки рішень, які не мали б вразливостей по відношенню до квантових обчислень, будучи одночасно стійкими до атак за допомогою звичайних комп'ютерів.

В результаті аналізу стійкості криптопримітивів виділяють такі перспективні постквантові кандидати:

- Криптографія на основі хеш-функцій (Hash Based);
- Криптографія на основі завадостійких кодів (Code Based);
- Криптографія на основі решіток (Lattice Based);
- Криптографія на основі факторизації поліномів (Multivariate Polynomial Based).

Існують різні схеми ЕЦП, зокрема, на основі завадостійких кодів (CFS, Paralell CFS); підписи на основі апарату хеш-функцій (Lamport та XMSS); схеми ЕЦП на решітках (BLISS та NTRU); ЕЦП на основі факторизації поліномів (Rainbow).

Висновки

В ході проведених досліджень розглянуто загальну модель порушника та загроз ЕЦП і спрощену модель середовища взаємодії відправника та отримувача в умовах дій зловмисника. В розглянутій моделі застосовується поняття випадкового оракула, до якого мають доступ всі учасники взаємодії. При кожному новому запиті значення функції вибирається випадковим чином і оракул запам'ятовує пару (аргумент, значення). При повторному запиті для цього аргументу, незалежно від того, хто з учасників його видав, буде повернуто те саме запам’ятоване значення. Модель із випадковим оракулом застосовується для протоколів ЕЦП, де випадкова функція замінила хеш-функцію. В класичному варіанті сам алгоритм підпису та всі учасники, крім зловмисника, здійснюють доступ до випадкового оракула за допомогою класичних бітових рядків.

Ґрунтуючись на розглянутій моделі середовища взаємодії відправника та отримувача в умовах дій зловмисника із випадковим оракулом, в підрозділі було досліджено модель порушника ЕЦП в умовах появи квантового комп’ютера. Для врахування особливостей квантових обчислень та, зокрема, можливостей щодо здійснення квантового криптоаналізу в дослідженій моделі порушника ЕЦП застосовано поняття *випадкового оракула із квантовим доступом.* Відповідно до розглянутої моделі, зловмисник може оцінювати випадкового оракула через явище суперпозиції, тобто, зловмисник може відправляти квантові стани $\langle\varphi| = \sum \mathrm{c}_x \, |x\rangle$ до оракула O та отримувати назад "оцінені" стани $\sum c_x \, |O(x)\rangle$. У той же час сам алгоритм підпису та всі інші учасники, крім зловмисника, здійснюють доступ до випадкового оракула за допомогою класичних бітових рядків (без використання явища суперпозиції). Однією з відмінностей класичного випадкового оракула, порівняно з квантовим, є те, що в його моделі симулюються прообрази, потрібні криптоаналітику. У випадку випадкового оракула з квантовим доступом відповідь на запит може бути схована в суперпозиції в експоненційно великому наборі квантових станів і не зрозуміло, як отримати необхідні відповіді. В класичному випадку можна ефективно змоделювати експоненційного випадкового оракула, просто обираючи випадкові відповіді. У випадку оракула з квантовим доступом зловмисник може отримати всі відповіді одночасно на всі вхідні запити. Це ускладнює стратегію запит-відповідь, яка застосовується в класичних випадкових оракулах і яка є більш узгодженою.

Із застосуванням моделі середовища взаємодії відправника та отримувача формалізовано поведінку зловмисника при екзистенційній підробці. При цьому застосовано моделі та методи теорії ігор. Зокрема, модель зловмисника у вигляді взаємодії двох сторін (зловмисника A та чесної сторони C) розглядається на основі паралельного об'єднання різних ігор: класичної гри G_c^{FOR} та квантової гри G_q^{FOR}.

Загальна вірогідність успіху зловмисника дорівнює 1, якщо він має успіх принаймні в одній з ігор, тобто, $Pr\ (G_c^{FOR}) + Pr\ \left(G_q^{FOR}\right) = 1$. При цьому в першій грі зловмисник використовує класичний комп'ютер, а в другій - квантовий.

Відповідну модель зловмисника подано у формалізованому вигляді обчислювального алгоритму.

Аналіз моделі загроз, заснованої на паралельному об'єднанні різних (класичних та квантових) ігор, показав, що криптопримітиви ЕЦП потрібно конструювати з врахуванням вимог часу та використовувати математичний базис квантовостійкої однопрохідної функції. При цьому потрібно оцінювати загальну захищеність від зламу за допомогою класичних обчислень, що ускладнює і модель зловмисника. Доказ різних випадків моделей зловмисника та зведення моделі із класичного випадку до квантового потребують подальших досліджень.

Розглянута модель загроз ЕЦП, яка випливає із відомих класичних та квантових моделей загроз, дозволила увести певну класифікацію різних атак за типом загрози, що несуть дії порушника. Відповідно до розглянутих методів редукції доказів безпеки зазначені загрози ЕЦП класифіковано за додатковою складністю криптоаналізу в порівнянні із більш загальною задачею криптоаналізу. Таку класифікацію подано як редукційну модель у вигляді однонаправленого графа, вершини якого представляють тип атаки, а стрілки позначають редукцію переходу із одного типу атаки до іншого. Зв'язки в редукційній моделі для класичного криптоаналізу позначають відношення більш загального виду атаки до часткового випадку.

Розглянута редукційна модель загроз ЕЦП є ідеалізованою моделлю, яка може змінюватися в залежності від різних умов, наприклад, від існування тих чи інших атак на конкретний підпис та від базових параметрів ЕЦП. У випадку існування атак із застосуванням квантового алгоритму Гровера часова складність квантового пошуку зменшується по відношенню до складності класичних алгоритмів як корінь квадратний від об'єму обчислень класичних алгоритмів. Тобто, повний перебір ключів алгоритмом Гровера змінює складність $O(N) \to O\left(\sqrt{N}\right)$ у порівнянні з класичним повним перебором, а складність знаходження колізій зменшується у порівнянні із класичною атакою, як $2^{n/2} \to 2^{n/3}$. Тобто, для алгоритму Гровера повний перебір ключа має складність $2^{n/2}$, яка не набагато більша, ніж складність знаходження колізій $2^{n/3}$. Відповідно до цього, складність атак у загальному вигляді змінюється, що призводить і до зміни редукційної моделі переходу із одного типу атаки до іншого. Урахування даних зауважень можуть бути корисними при розробці нових чи оцінці існуючих моделей захисту ЕЦП, зокрема, розроблена модель порушника в квантових системах захисту інформації дозволяє визначити напрямки підвищення безпеки постквантових криптопримітивів ЕЦП.

З метою завершення аналізу моделі порушника ЕЦП, що діє в умовах появи квантових обчислень та їх практичного застосування, в підрозділі отримано основні кількісні оцінки загроз класичних і постквантових ЕЦП на основі складності відомих атак. Порівняння отриманих результатів дозволило зробити такі висновки:

- криптографічна стійкість (в умовах класичних обчислень) проти атаки «повне розкриття» при застосуванні криптоперетворення в групі точок еліптичної кривої має експоненційний характер і суттєво перевищує стійкість криптоперетворень у кільці (факторизація) та в скінченному полі Галуа (дискретне логарифмування); стійкість криптоперетворень у кільці та в скінченному полі має субекспоненційний характер і оцінки значень складності криптоаналізу практично співпадають;

- для постквантових застосувань більшість відомих та стандартизованих алгоритмів ЕЦП є неприйнятними, бо складність квантового криптоаналізу дуже низька. Тому перспективним є дослідження і розробка таких криптопримітивів, які не мали б вразливостей по відношенню до квантових обчислень та були одночасно стійкими до атак за допомогою звичайних комп'ютерів. До перспективних постквантових алгоритмів ЕЦП слід віднести: ЕЦП на основі завадостійких кодів (CFS, Paralell CFS); підписи на основі апарату хеш-функцій (Lamport та XMSS); схеми ЕЦП на решітках (BLISS та NTRU); ЕЦП на основі факторизації поліномів (Rainbow).

15 МЕТОДИ ТА МЕХАНІЗМИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ ПОСТКВАНТОВИХ КРИПТОСИСТЕМ

15.1 Порівняльний аналіз одноразових підписів на базі геш-функцій

Вступ

Стійкість значної частини сучасних асиметричних систем базується на складності рішення задач факторизації (RSA), дискретного логарифму в простому полі (DSA) або групі точок еліптичної кривої (EC-DSA). Квантові комп'ютери, що можуть з'явитися в найближчі десятиліття, здатні вирішити ці задачі.

Актуальною є проблема аналізу та стандартизації постквантових криптосистем. З цією метою NIST проводить відкритий конкурс [121], в якому представлені алгоритми, побудовані на різних алгебраїчних структурах. В тому числі були представлені два алгоритми цифрового підпису на базі геш-функцій: Gravity SPHINCS [122] (брав участь тільки у першому раунді) та SPHINCS+ [123] (успішно пройшов до третього раунду [124]), які є незалежними модифікаціями раніше розробленого алгоритму SPHINCS [125].

Усі алгоритми сімейства SPHINCS мають складну структуру та включають декілька простіших алгоритмів підпису, серед яких одноразовий підпис Вінтерніца.

Метою розділу є аналіз існуючих алгоритмів одноразового підпису, зокрема, підписів Лампорта [126] та Вінтерніца [127]. Запропоновано та проаналізовано модифікації цих алгоритмів.

Огляд алгоритмів ЕЦП на базі геш-функцій

Загальна ідея підписів, що базуються на геш-функціях, полягає в тому, що, у відповідність до повідомлення, ставиться масив бітових рядків, елементи якого обрані та, можливо, прогешовані певну кількість разів відповідно до бітів повідомлення. Стійкість цього класу ЕЦП базується на основних властивостях криптографічних геш-функцій - односпрямованості та стійкості до колізій. Згідно з [128], основними вимогами до функцій гешування є:

1) Складність знаходження колізії

$$C_{col} \geq 2^{h_{len}/2}.$$

2) Складність відновлення прообразу

$$C_{preim} \geq 2^{h_{len}}$$

3) Складність знаходження іншого прообразу

$$C_{sec\_preim} \geq 2^{h_{len}}.$$

Розглянемо переваги підписів на базі геш-функцій:

1) постквантовість – сучасні криптографічні геш-функції, наприклад, SHA-2 та SHA-3 стійкі до квантових атак і можуть безпечно використовуватись і після створення квантових комп'ютерів;

2) легка модифікація – алгоритми ЕЦП цього класу не детермінують використовувані геш-функції, що дозволяє легко замінити механізм гешування, якщо в ньому будуть знайдені вразливості;

3) гнучкість – доповнює попередню властивість тим, що, в залежності від вимог до системи, можуть бути обрані геш-функції та системні параметри, що найкраще їх задовольняють, та досягається компроміс між стійкістю, швидкодією та пам'яттю.

До недоліків слід віднести:

1) Обмежена кількість підписів – усі алгоритми цього класу накладають обмеження на максимальну кількість підписів, що можуть бути створені з використанням однієї пари ключів, однак, у сучасних криптосистемах ця кількість може бути дуже високою, що фактично знімає це обмеження;

2) великий розмір підпису – сучасні підписи мають складну структуру, що включає багато елементів.

ЕЦП на основі геш-функцій можна класифікувати наступним чином:

- одноразові (Лампорта [126], Вінтерніца [127 – 129]);
- багаторазові з використанням дерев Мерклі [127];
- багаторазові з поступовим зниженням стійкості (HORS [130], HORST [144], FORS [123], PORS [124]);
- багаторазові з використанням гіпердерев (SPHINCS [125], SPHINCS+ [123], Gravity-SPHINCS [124], XMMSMT [131]).

Алгоритми одноразового підпису

Перші алгоритми цифрового підпису на базі геш-функцій мали серйозний недолік – одна пара ключів могла бути використана для підпису тільки одного повідомлення, оскільки при підписуванні розкривалася значна частина приватного ключа і його подальше використання ставало небезпечним; ці алгоритми отримали назву одноразових підписів (One-Time Signature – OTS).

До цього класу належать:

- підпис Лампорта (L-OTS);
- підпис Лампорта – Діффі (LD-OTS);
- підпис Вінтерніца (W-OTS);
- модифікації підпису Вінтерніца (WOTS+);
- інші варіанти.

Зауваження 1. В подальшому будемо розглядати узагальнений випадок, вважаючи, що для гешування повідомлення та елементів ключа використовуються різні геш-функції з довжинами вихідних значень m та n відповідно. На практиці, зокрема, у найбільш перспективних криптосистемах сімейства SPHINCS для цього використовується одна і та сама функція з довжиною виходу n.

Зауваження 2. Далі під повідомленням мається на увазі геш-значення від фактичного повідомлення, отримане з використанням криптографічної геш-функції $H(M):\{0,1\}^* \to \{0,1\}^m$, якщо не вказано інше.

Розглянемо більш детально перераховані вище алгоритми підписів.

Підпис Лампорта (L-OTS)

Перший алгоритм використання геш-функції для створення підпису запропонований Лампортом [126] в 1979 р. Схема була побудована наступним чином.

Секретний ключ. Два масиви по m елементів в кожному масиві заповнені випадковими бітовими послідовностями довжини n.

$$SK = \begin{pmatrix} X = (x_0, x_1, \dots, x_{m-1}) \\ Y = (y_0, y_1, \dots, y_{m-1}) \end{pmatrix}.$$

Відкритий ключ. Два масиви геш-значень елементів секретного ключа:

$$PK = \begin{pmatrix} H(X) = \left(H(x_0), H(x_1), \dots, H(x_{m-1})\right) \\ H(Y) = \left(H(y_0), H(y_1), \dots, H(y_{m-1})\right) \end{pmatrix}.$$

Створення підпису. Підпис – масив з m значень елементів секретного ключа, обраних за таким правилом: якщо поточний біт повідомлення, що підписується, дорівнює 0, в підпис включається відповідний елемент з першого масиву секретного ключа, якщо 1, то з другого. Таким чином розкривається половина елементів секретного ключа, інша половина повинна бути знищена.

Перевірка підпису. Елементи підпису гешуються і порівнюються з відповідними елементами відкритого ключа.

Зауваження 3. Підпис Лампорта має два недоліки, що заважають його широкому використанню:

- може бути використаний для підпису тільки одного повідомлення;
- має великий розмір.

Підпис Лампорта-Діффі

Діффі незначно модифікував алгоритм Лампорта, уточнивши вимоги до геш-функцій, що використовуються, а саме:

- для гешування повідомлення повинна використовуватися криптографічна геш-функція;
- геш-функція, що використовується для створення відкритого ключа та перевірки підпису, повинна бути односпрямованою.

Було змінено позначення елементів секретного ключа:

$$SK = \begin{pmatrix} X_0 = \left(x_{0,0}, x_{0,1}, \dots, x_{0,m-1}\right) \\ X_1 = \left(x_{1,0}, x_{1,1}, \dots, x_{1,m-1}\right) \end{pmatrix}.$$

Інші складові алгоритму залишилися без змін.

Підпис Вінтерніца (W-OTS)

З метою зменшення розмірів підпису Вінтерніц запропонував [127] використовувати ланцюгову (ітеративну) геш-функцію, що дозволяє підписувати одним значенням декілька біт повідомлення. Розглянемо цей алгоритм.

Параметр Вінтерніца – кількість біт повідомлення, що будуть одночасно підписані. В деяких джерелах параметром Вінтерніца називають значення $2w$. Будемо дотримуватися першого варіанту.

Вводяться наступні додаткові параметри:

t1 – блочна довжина повідомлення;
t2 – блочна довжина контрольної суми;
t – блочна довжина повідомлення з контрольною сумою.
Секретний ключ. Масив з t випадкових n-бітових послідовностей

$$SK = (sk_0, sk_1, \dots, sk_{t-1}), sk_i = \{0,1\}^n.$$

Для економії пам'яті, в якості секретного ключа, можна використовувати одну випадкову послідовність $x = \{0,1\}^n$, яка при необхідності буде розгорнута у SK за допомогою генератора псевдовипадкових послідовностей. Це незначно ускладнює створення підпису.

Відкритий ключ. Ітеративно прогешовані $(2w-1)$ разів елементи секретного ключа.

$$PK = (pk_0, pk_1, \dots, pk_{t-1}), pk_i = f^{2^w-1}(sk_i).$$

Створення підпису.
Вхідні дані: M, H, SK.
Вихідні дані: σ.

1) Повідомлення розбивається на $t1$ блоків довжиною w біт

$$H(M) = \left(h_0, h_1, \dots, h_{t_1-1}\right).$$

2) Обчислюється контрольна сума

$$S = \sum_{i=0}^{t_1} 2^w - 1 - h_i = (s_0, s_1, \dots, s_{t_2-1}).$$

3) Контрольна сума додається до повідомлення

$$B = H(M)||S = (h_0, h_1, \dots, h_{t_1-1}, s_0, s_1, \dots, s_{t_2-1}) = (b_0, b_1, \dots, b_{t-1}),$$

де || означає конкатенацію.
4) Обчислюється підпис

$$\sigma = (\sigma_1, \sigma_0, \dots, \sigma_{t-1}) = \left(f^{b_0}(sk_0), f^{b_1}(sk_1), \dots, f^{b_{t-1}}(sk_{t-1})\right).$$

Перевірка підпису.

Вхідні дані: M, H, PK, σ.
Вихідні дані: true або false.
1) Аналогічно пунктам 1-3 алгоритму, створення підпису обчислюється

$$B' = (b'_0, b'_1, \dots, b'_{t-1}).$$

2) Обчислюється

$$Z = (z_0, z_1, \dots, z_{t-1}) =$$
$$= \left(f^{2^w-b_1}(\sigma_0), f^{2^w-b_0}(\sigma_1), \dots, f^{2^w-b_{t-1}}(\sigma_{t-1})\right).$$

3) Якщо $Z = PK$, підпис коректний.

Існують різні модифікації ЕЦП Вінтерніца, зміни в основному стосуються ітеративної функції, що використовується, серед них:

- перед черговим гешуванням на число операцією XOR накладається бітова маска, що дозволяє знизити вимоги до геш-функції [129 – 132];

- використання ключової геш-функції, де результат попередньої ітерації використовується в якості ключа [128].

Модифіковані алгоритми

Розглянемо декілька модифікацій класичних алгоритмів.

Підпис Лампорта-Вінтерніца

В [132] запропоновано модифікацію класичних схем Лампорта та Вінтерніца, яка потребує більш детального аналізу.

Розглянемо дві варіації вдосконаленого підпису:

- Альтернативний підпис Лампорта;
- Альтернативний підпис Вінтерніца.

Альтернативний підпис Лампорта.

Сутність даного алгоритму полягає в зменшенні розмірів ключів та підпису за рахунок зменшення кількості біт повідомлення, що підписуються.

Загальносистемні параметри:

m – довжина геш-значення повідомлення;

w – кількість бітів, які підписуються одним елементом ключа;

n – довжина елементів ключів та підпису;

$t = \left\lceil \frac{m}{w} \right\rceil -$ кількість блоків, на які буде розбито повідомлення, а також кількість елементів ключів, що будуть для цього використані;

$H_c :\{0,1\}^* \to \{0,1\}^m -$ криптографічна геш-функція;

$H :\{0,1\}^n \to \{0,1\}^n -$ односпрямована геш-функція.

Секретний ключ.

Два масиви з t випадкових n-бітових рядків.

$$SK = \begin{pmatrix} X_0 = (x_{0,0}, x_{0,1}, \dots, x_{0,t-1}) \\ X_1 = (x_{1,0}, x_{1,1}, \dots, x_{1,t-1}) \end{pmatrix}.$$

Відкритий ключ.

Два масиви геш-значень секретного ключа SK.

$$PK = \begin{pmatrix} H(X_0) = \left(H(x_{0,0}), H(x_{0,1}), \dots, \mathrm{H}(x_{0,t-1})\right) \\ H(X_1) = \left(H(x_{1,0}), H(x_{1,1}), \dots, H(x_{1,t-1})\right) \end{pmatrix}.$$

Створення підпису:

Геш-повідомлення розбивається на t блоків по w біт (якщо m не кратно w, у кінець додається необхідна кількість нулів).

$$H(M) = (h_0, h_1, \dots, h_t).$$

Відповідно до старшого біта h_i в підпис додається відповідний елемент з X_0 чи X_1.

Перевірка підпису:

Елементи підпису гешуються та порівнюються з відповідними елементами відкритого ключа.

Альтернативний підпис Вінтерніца

Сутність полягає в об'єднанні ідей Лампорта і Вінтерніца: секретний та відкритий ключі складаються з двох масивів і відкритий обчислюється з приватного за рахунок циклічного гешування. Оскільки використовується циклічне гешування для захисту від підробки, використовується контрольна сума, аналогічно WOTS.

Опис алгоритму.

Загальносистемні параметри:

H, f, w, t, t1, t2 – аналогічно звичайному підпису Вінтерніца.

Секретний ключ.

Два масиви з t–1 випадкових n-бітових послідовностей

$$SK = \begin{pmatrix} (sk_{0,0}, sk_{0,1}, \dots, sk_{0,t-1}) \\ (sk_{1,0}, sk_{1,1}, \dots, sk_{1,t-1}) \end{pmatrix},\ sk_{i,j} = \{0,1\}^n.$$

Відкритий ключ.

Ітеративно прогешовані $2^{w-1} - 1$ разів елементи секретного ключа

$$PK = \begin{pmatrix} (pk_{0,0}, pk_{0,1}, \dots, pk_{0,t-1}) \\ (pk_{1,0}, pk_{1,1}, \dots, pk_{1,t-1}) \end{pmatrix},\ pk_{i,j} = f^{2^{w-1}-1}(sk_{i,j}).$$

Створення підпису.

Вхідні дані: M, H, SK.

Вихідні дані: σ.

1) Повідомлення розбивається на t1 блоків довжиною w біт (з метою, щоб зробити довжину повідомлення, кратну w , до повідомлення додається необхідна кількість нульових біт)

$$H(M) = (h_0, h_1, \dots, h_{t_1-1}).$$

2) Обчислюється контрольна сума

$$S = \sum_{i=0}^{t_1} 2^w - 1 - h_i = (s_0, s_1, \dots, s_{t_2}).$$

3) Контрольна сума додається до повідомлення

$$B = H(M)||S = (h_0, h_1, \dots, h_{t_1-1}, s_0, s_1, \dots, s_{t_2-1}) = (b_0, b_1, \dots, b_{t-1}),$$

де || означає конкатенацію.

4) Для кожного з блоків виділяють старший та молодші біти: $b_i = bh_i||bl_i$

5) Обчислюється підпис: в залежності від значення старшого біта обирається відповідний елемент з першої чи другої частини приватного ключа та гешується у відповідності до значення молодших біт.

$$\sigma = (\sigma_1, \sigma_0, \dots, \sigma_{t-1}) =$$

$$= (f^{bl_0}(sk_{bh_0,0}), f^{bl_1}(sk_{bh_1,1}), \dots, f^{bl_{t-1}}(sk_{bh_{t-1},t-1})).$$

Перевірка підпису.

Вхідні дані: M, H, PK, σ.

Вихідні дані: true або false.

1) Аналогічно пунктам 1-3 алгоритму створення підпису обчислюється

$$B' = (b'_0, b'_1, \dots, b'_{t-1}).$$

2) Обчислюється

$$Z = (z_0, z_1, \dots, z_{t-1}) =$$

$$= \left(f^{2^{w-1}-bl_1}(\sigma_0), f^{2^{w-1}-bl_0}(\sigma_1), \dots, f^{2^{w-1}-bl_{t-1}}(\sigma_{t-1})\right).$$

3) Обчислюється $T = \left(pk_{bh'_0,0}, pk_{bh'_1,1}, \dots, pk_{bh'_{t-1},t-1}\right)$.

4) Якщо $Z = T$, підпис коректний.

Розширений підпис Лампорта

В класичному варіанті алгоритму Лампорта кожному біту відповідає один з двох можливих станів (елементів підключів), що призводить до значних витрат: кількість елементів підпису дорівнює кількості бітів в геш-значенні повідомлення, що підписується. Зменшення довжини геш-значення призводить до зниження стійкості. Пропонується об'єднувати біти геш-значення повідомлення в групи та ставити кожній групі бітів у відповідність по одному елементу з секретного ключа. Для цього необхідно збільшити кількість підключів до 2w, де w – довжина групи. Вибір підключа для поточного елементу здійснюється відповідно до бітового значення конкретної групи бітів. Назвемо це розширеним алгоритмом Лампорта.

Загальносистемні параметри.

Криптографічна геш-функція $H_c : \{0,1\}^* \to \{0,1\}^m$.

Односпрямована геш-функція $H : \{0,1\}^n \to \{0,1\}^n$.

w – довжина групи бітів, яка підписується одним значенням.

Секретний ключ.

2w масиви по *m/w* випадкових n-бітових рядків.

$$SK = \left(X_0, X_1, \dots, X_{2^w-1}\right);$$
$$X_i = (x_{i,0}, x_{i,1}, \dots, x_{i,(m-1)/w}).$$

Відкритий ключ.

2w масиви по m / w геш-значень елементів секретного ключа.

$$PK = H(SK) = \left((H(X_0), H(X_1), \dots, H(X_{2^w-1}))\right);$$
$$H(X_i) = (H(x_{i,0}), H(x_{i,1}), \dots, H(x_{i,(m-1)/w})).$$

Створення підпису.

Вхідні дані: SK.

Вихідні дані: σ.

Підпис – масив з m / w значень секретного ключа, які обираються відповідно до пар бітів повідомлення.

$$H_c(M) = \{h_0, h_1, \ldots, h_{(m-1)/w}), \quad h_i = \{0,1\}^w;$$
$$\sigma = \left(x_{h_0,0}, x_{h_1,1}, \ldots, x_{h_{(m-1)/w},(m-1)/w} \right).$$

Перевірка підпису.

Вхідні дані: РК, σ.

Вихідні дані: true або false.

Елементи підпису гешуються і порівнюються з відповідними елементами відкритого ключа.

$$H'_c(M) = \{h'_0, h'_1, \ldots, h'_{(m-1)/w});$$
$$H(\sigma) = \left(H(x_{h_0,0}), H(x_{h_1,1}), \ldots, H(x_{h_{(m-1)/w},(m-1)/w}) \right);$$
$$Z = \left(H(x_{h'_0,0}), H(x_{h'_1,1}), \ldots, H(x_{h'_{(m-1)/w},(m-1)/w}) \right).$$

Якщо $H(\sigma) = Z$, підпис коректний.

Аналіз параметрів одноразових підписів

До основних параметрів, за якими можливо порівнювати алгоритми ЕЦП незалежно від їх математичної бази, відносяться:

- стійкість до криптоаналізу;
- розміри ключів та підписів;
- обчислювальна складність генерації ключів, створення та перевірки підпису.

У контексті одноразових підписів узагальнено можна виділити два класи атак:

- при відомому публічному ключі;
- при відомому публічному ключі та підписі.

Зауваження 4. Більш складні атаки, зокрема, із вибором або адаптивним вибором повідомлення розглядатися не будуть, оскільки наявність декількох підписів, створених із використанням однієї ключової пари, є неможливою за визначенням одноразового підпису. Також виникнення такої пари підписів свідчить про порушення у роботі схеми підпису і призводить до миттєвого зниження стійкості системи.

До другого класу належить атака типу «екзистенційна підробка», що може бути застосовано до довільного алгоритму підпису, а саме, знаходження другого прообразу до геш-значення підписаного повідомлення. При використанні криптографічної геш-функції її складність складає m біт та не може бути зменшена за рахунок накопичення аналітиком великої кількості повідомлень, оскільки, за визначенням, ключі одноразового підпису використовуються лише один раз. З іншого боку, особливості підписів на основі геш-функцій дозволяють зменшити складність пошуку підходящого повідомлення, геш якого співпадає з

підписаним лише частково. Детальний аналіз стосовно кожного з алгоритмів наводиться далі.

Аналіз підпису Лампорта

1) Стійкість.

а) Атака при відомому відкритому ключі.

Для успішного втілення подібної атаки зловмиснику необхідно знайти перший чи другий прообраз для *m n*-бітних геш-значень (можливістю колізій між елементами приватного ключа знехтуємо, так як для його генерації використовувався генератор з необхідними криптографічними властивостями). Якщо використовується криптографічна геш-функція, то складність такої атаки складає 2m*n, що відповідає стійкості m*n біт. Наприклад, якщо m = n = 256, стійкість алгоритму складає 65536 біт.

б) Атака при відомому відкритому ключі та підписі.

Зловмисник може спробувати знизити складність атаки «екзистенційна підробка», а саме, пошуку другого прообразу для підписаного повідомлення, підібравши повідомлення, геш якого співпадає з підписаним лише частково, наприклад, співпадають (m–k) біт. Однак в цьому випадку йому буде необхідно знайти прообраз (перший чи другий) для кожного невідомого елемента відкритого ключа, тобто, зниження складності пошуку повідомлення на 1 біт призводить до збільшення складності створення підробного підпису на n біт. Складність такої атаки буде становити (m – k + n*k = m + (n – 1)*k) біт. Таким чином, стійкість криптоалгоритму складає не менше m біт. Якщо невикористані елементи секретного ключа не були знищенні під час створення підпису і частина з них стала відома зловмиснику, складність підробки зменшується. Кожен зкомпрометований елемент фактично знижує ентропію геш-значення повідомлення на 1 біт, якщо криптоаналітику відомі k додаткових елементів секретного ключа, то складність підробки складає (m–k) біт.

2) Розміри підпису та ключа.

Довжина секретного та відкритого ключів у схемі підпису Лампорта визначається як 2*m*n біт, довжина ЕП m*n біт. Результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для механізмів Лампорта та Лампорта – Діффі в залежності від параметрів безпеки наведені в табл. 15.1.

Розміри підпису Лампорта становлять від одиниць до десятків кілобайт в залежності від рівня стійкості.

3) Обчислювальна складність.

Створення пари ключів підпису Лампорта вимагає генерації випадкових значень та їх гешування, створення підпису вимагає тільки відбору необхідних значень, перевірка підпису вимагає тільки гешування. У випадку, якщо підписувач за якихось причин не зберігає публічний ключ, для створення підпису йому необхідно прогешувати відповідні елементи приватного ключа. Обчислювальну складність механізму наведено в табл. 15.2.

Усе зазначене також актуально для підпису Лампорта – Діффі, що фактично відрізняється тільки позначеннями та вимогами до геш-функцій.

Таблиця 15.1.

Розміри секретних та відкритих ключів
та ЕП для механізму Лампорта в кілобайтах

Параметр безпеки (n)	Розмір секретного ключа	Розмір відкритого ключа	Розмір ЕП
128	4	4	2
192	9	9	4.5
256	16	16	8
384	36	36	18
512	64	64	32

Таблиця 15.2.

Обчислювальна складність механізму Лампорта

Параметр	Генерація ключів	Створення підпису	Перевірка
Генерація випадкового значення	2m	-	-
Обчислення геш-функції	2m	-/m	m

Аналіз підпису Вінтерніца

Детальний аналіз стійкості підпису Вінтерніца та його модифікацій проведено у роботах [128,129,133]. В даній роботі ми будемо використовувати спрощену модель.

1) Стійкість.

а) Атака при відомому відкритому ключі.

Розглянемо три можливі для зловмисника випадки: найгірший, звичайний та найкращий. Складність пошуку повідомлення, геш якого має необхідні властивості, не враховуємо.

В найгіршому випадку геш-значення підробного повідомлення містить тільки нульові біти, в такому випадку зловмиснику необхідно знайти прообрази для декілька разового гешування t1 елементів (ймовірністю колізій знехтуємо), тобто, стійкість складає не менше n * t1 біт.

У звичайному (найбільш ймовірному) випадку значення груп пікселів будуть рівномірно розподілені в діапазоні [0; 2w–1], тобто, максимальне значення 2w–1, якому відповідає елемент відкритого ключа, приймуть близько (t1/(2w–1)). Складність підробки при цьому складає не менше (t1 – (t1/(2w–1))*n) біт, а з урахуванням елементів, що відповідають контрольній сумі, стійкість стає вищою.

В найкращому для зловмисника випадку геш-значення підробного повідомлення містить тільки одиничні біти, в такому випадку t1 елементів підпису дорівнюють елементам відкритого ключа, що відомий зловмиснику. Однак залишаються t2 елементів, що відповідають контрольній сумі, яка в цьому випадку дорівнює нулю. Таким чином, зловмиснику необхідно знайти прообрази

для декілька разового гешування t2 елементів. Стійкість до цієї атаки дорівнює не менше t2*n біт.

б) Атака при відомому відкритому ключі та підписі.

Зловмисник може знизити складність пошуку другого прообразу, підібравши повідомлення, геш якого співпадає з підписаним лише частково. Якщо зловмисник має коректний підпис, то через певну кількість обчислень він зможе створити повідомлення з геш-значенням $H(M') = \left(h'_0, h'_1, \dots, h'_{t_1-1}\right)$ таким, що $h'_i \geq h_i$. Таким чином, після додаткових ітерацій початковий підпис стане коректним для повідомлення M'. Використання контрольної суми дозволяє захистити підпис від підробки: при збільшенні h_i значення контрольної суми зменшується, отже, елементи ключа, використані для підписування контрольної суми, повинні бути прогешовані меншу кількість разів, ніж у початковому повідомленні, що зловмисник зробити не може через односторонню природу функції *f*. Таким чином, зловмиснику необхідно знайти прообраз хоча б одного *n*-бітного значення, що призводить до стійкості не менше *n* біт.

2) Розміри підпису та ключа.

Розмір підпису в схемі Вінтерніца дорівнює розміру відкритого ключа та складає SignSize = t * n біт. Нехай до повідомлення та елементів ключа застосовується одна й та ж сама криптографічна геш-функція з довжиною геш-значення $n = \{128; 192; 256; 384; 512\}$ біт, параметр Вінтерніца $w = [2,3,\cdots,16]$. Залежність розмірів підпису від n та w наведено у табл. 15.3.

Таблиця 15.3.

Залежність розміру підпису Вінтерніца від n та w в кілобайтах

n / w	128	192	256	384	512
2	1.063	2.367	4.156	9.234	16.313
3	0.719	1.570	2.813	6.188	10.938
4	0.547	1.195	2.094	4.641	8.188
5	0.438	0.984	1.719	3.750	6.625
6	0.375	0.797	1.406	3.094	5.563
7	0.328	0.703	1.219	2.672	4.750
8	0.281	0.609	1.063	2.344	4.125
9	0.266	0.563	0.969	2.109	3.688
10	0.234	0.516	0.875	1.922	3.375
11	0.219	0.469	0.813	1.734	3.063
12	0.203	0.422	0.750	1.594	2.813
13	0.188	0.398	0.688	1.500	2.625
14	0.188	0.375	0.656	1.406	2.438
15	0.172	0.352	0.625	1.313	2.313
16	0.156	0.328	0.563	1.219	2.125

3) Обчислювальна складність.

Для генерації приватного ключа підпису Вінтерніца необхідно згенерувати t випадкових n-бітних значень, для обчислення публічного ключа необхідно виконати N = t * (2w – 1) операцій гешування. Створення та перевірка підпису разом фактично повторюють процедуру обчислення публічного ключа, однак обчислювальна складність виконання кожної з цих процедур окремо залежить від повідомлення, що підписується. Залежність кількості гешувань для обчислення відкритого ключа схеми Вінтерніца від n та w наведено у табл. 15.4.

Зауваження 5. З таблиці 15.4 видно, що на практиці можна використовувати лише невеликі значення параметра Вінтерніца, оскільки кількість ітерацій для генерації відкритого ключа, створення та перевірки підпису експоненційно залежить від даного параметру. Оптимальним, на наш погляд, є значення w = 4, розмір підпису в такому випадку складає 0,5 – 8 кілобайт та потребує близько 500 – 2000 операцій гешування в залежності від рівня стійкості, що є прийнятною величиною. Також розбиття даних, що підписуються, на 4-бітні блоки легко реалізується як програмно, так і апаратно.

Аналіз альтернативного підпису Лампорта

Вдосконалений алгоритм відрізняється від класичного підпису Лампорта тим, що підписуються не всі біти геш-значення, а лише кожен w-й, що є еквівалентним до використання у підписі геш-функції з більш коротким вихідним значенням.

1) Стійкість.

Для цього алгоритму справедливі ті ж самі припущення, що і для стандартного підпису Лампорта.

Таблиця 15.4. Залежність кількості гешувань для обчислення відкритого ключа підпису Вінтерніца від n та w

n / w	128	192	256	384	512
2	204	303	399	591	783
3	322	469	630	924	1225
4	525	765	1005	1485	1965
5	868	1302	1705	2480	3286
6	1512	2142	2835	4158	5607
7	2667	3810	4953	7239	9652
8	4590	6630	8670	12750	16830
9	8687	12264	15841	22995	30149
10	15345	22506	28644	41943	55242
11	28658	40940	53222	75739	100303
12	53235	73710	98280	139230	184275
13	98292	139247	180202	262112	344022
14	196596	262128	344043	491490	638937
15	360437	491505	655340	917476	1212379
16	655350	917490	1179630	1703910	2228190

а) Атака при відомому відкритому ключі.

В цьому випадку зловмиснику необхідно *t* разів знайти прообраз для *n*-бітової геш-функції, що за умови стійкості до пошуку прообразів еквівалентно вгадуванню t*n випадкових біт.

б) Атака при відомому відкритому ключі та підписі.

Для класичної схеми Лампорта складність пошуку другого прообразу дорівнює 2n, в модифікованій схемі при підписі враховуються лише t бітів геш-значення. Відповідно, складність пошуку повідомлення, що дасть геш зі співпадаючими відповідними бітами, дорівнює 2t.

Криптоаналітик може спростити задачу пошуку повідомлення, яке має відповідний геш, дозволивши відмінності у «критичних» t бітах. При цьому, як і у звичайному підписі Лампорта, відмінність у одному біті вимагає пошуку прообразу для одного з елементів відкритого ключа. Складність цієї задачі 2n. Стійкість при ігноруванні k критичних біт складатиме t + (n – 1)*k.

Зауваження 6.

а) Найнебезпечнішою є друга атака, яка унеможливлює використання великих значень w.

б) Доцільність використання альтернативного підпису Лампорта викликає сумніви через його уразливість.

2) Розміри підпису та ключа.

Альтернативний підпис Лампорта повністю еквівалентний звичайному підпису Лампорта з довжиною геш-значення повідомлення t біт.

Розміри приватного та публічного ключів становлять 2*t*n біт. Розміри підпису t*n біт.

3) Обчислювальна складність.

Аналогічно стандартному агоритму Вінтерніца, генерація ключа включає генерацію 2t випадкових значень та їх гешування. Створення та перевірка підпису вимагає t гешувань кожне.

Аналіз альтернативного підпису Вінтерніца

1) Стійкість.

Властивості альтернативного підпису Вінтерніца в цілому схожі на властивості звичайного підпису Вінтерніца з деякими уточненнями.

а) Атака при відомому відкритому ключі.

Аналогічно звичайному підпису Вінтерніца, можна виділити декілька випадків складності підробки: відносно бітів геш-значення підробного повідомлення – усі біти 0, біти рівномірно розподілені - усі біти 1. Як і у стандартній схемі, стійкість до підробки у найгіршому випадку складає n * t1 біт, у найкращому n * t2 біт. Оскільки обидві частини відкритого ключа відомі зловмиснику, він може обирати довільний елемент у кожній парі, відповідно до повідомлення, що підписується.

б) Атака при відомому відкритому ключі та підписі.

Аналогічно механізму Лампорта, криптоаналітик може спростити пошук другого прообразу для геш-значення повідомлення, ігноруючи відмінності у бітах, що визначають вибір елемента з першої чи другої підмножини. В цьому випадку для створення коректного підпису необхідно знайти невідомі елементи

приватного ключа. Знаходження k елементів приватного ключа означає знаходження прообразів для k значень ітеративної геш-функції.

Аналогічно механізму Вінтерніца, можна підібрати повідомлення, усі блоки геш-значення якого будуть більші за блоки підписаного повідомлення. Як було зазначено раніше, від цього захищає використання контрольної суми. Тобто, стійкість складає не менше n біт.

Таким чином, зміна одного біта геш-значення у частині блоку, що відповідає за Лампортову складову, або у частині, що відповідає за складову Вінтерніца, неодмінно вимагає знаходження хоча б одного прообразу для геш-функції і стійкість залишається не нижче n біт.

2) Розміри підпису та ключа.

Розмір підпису альтернативного алгоритму Вінтерніца залишився незмінним і складає SignSize = t * n. Розміри ключів збільшилися вдвічі і дорівнюють KeySize = 2 * t * n. Залежність розмірів ключів від n та w наведено у табл. 15.5.

Таблиця 15.5.

Залежність розміру ключів (в кілобайтах) альтернативного підпису Вінтерніца від n та w

n / *w*	128	192	256	384	512
2	2.125	4.734	8.313	18.469	32.625
3	1.438	3.141	5.625	12.375	21.875
4	1.094	2.391	4.188	9.281	16.375
5	0.875	1.969	3.438	7.500	13.250
6	0.750	1.594	2.813	6.188	11.125
7	0.656	1.406	2.438	5.344	9.500
8	0.563	1.219	2.125	4.688	8.250
9	0.531	1.125	1.938	4.219	7.375
10	0.469	1.031	1.750	3.844	6.750
11	0.438	0.938	1.625	3.469	6.125
12	0.406	0.844	1.500	3.188	5.625
13	0.375	0.797	1.375	3.000	5.250
14	0.375	0.750	1.313	2.813	4.875
15	0.344	0.703	1.250	2.625	4.625
16	0.313	0.656	1.125	2.438	4.250

3) Обчислювальна складність.

Кількість ітерацій ланцюгової функції для обчислення публічного ключа в альтернативному підписі Вінтерніца Nkey = Nkey_WOTS – t. Кількість ітерацій ланцюгової функції для створення та перевірки підпису Nsign = (Nsign_WOTS – t) / 2. Залежність кількості гешувань для створення та

перевірки підпису та обчислення відкритого ключа від n та w наведено у табл. 15.6 та 15.7.

Зауваження 7.

1. Вдосконалений алгоритм потребує менше обчислень для створення та перевірки підпису, однак більше пам’яті для зберігання секретного та відкритого ключів.

Таблиця 15.6.

Залежність кількості гешувань для підпису від n та w

n / *w*	128	192	256	384	512
2	68	101	133	197	261
3	138	201	270	396	525
4	245	357	469	693	917
5	420	630	825	1200	1590
6	744	1054	1395	2046	2759
7	1323	1890	2457	3591	4788
8	2286	3302	4318	6350	8382
9	4335	6120	7905	11475	15045
10	7665	11242	14308	20951	27594
11	14322	20460	26598	37851	50127
12	26611	36846	49128	69598	92115
13	49140	69615	90090	131040	171990
14	98292	131056	172011	245730	319449
15	180213	245745	327660	458724	606171
16	327670	458738	589806	851942	1114078

Таблиця 15.7.

Залежність кількості гешувань для обчислення відкритого ключа від n та w

n / *w*	128	192	256	384	512
2	136	202	266	394	522
3	276	402	540	792	1050
4	490	714	938	1386	1834
5	840	1260	1650	2400	3180
6	1488	2108	2790	4092	5518
7	2646	3780	4914	7182	9576
8	4572	6604	8636	12700	16764
9	8670	12240	15810	22950	30090

Продовження табл. 15.7.

10	15330	22484	28616	41902	55188
11	28644	40920	53196	75702	100254
12	53222	73692	98256	139196	184230
13	98280	139230	180180	262080	343980
14	196584	262112	344022	491460	638898
15	360426	491490	655320	917448	1212342
16	655340	917476	1179612	1703884	2228156

2. Оскільки відкритий ключ неможливо повністю обчислити з підпису, цей ЕЦП не так зручно використовувати в схемах з геш-деревами, разом з підписом необхідно передавати іншу половину елементів відкритого ключа, що подвоює реальний розмір підпису.

Аналіз розширеного підпису Лампорта

1) Стійкість.

а) Атака при відомому відкритому ключі.

Аналогічно звичайному підпису Лампорта, для підробки розширеного алгоритму необхідно знайти прообрази для усіх елементів відкритого ключа, що будуть використані. Стійкість дорівнює (m / w) * n біт.

б) Атака при відомому відкритому ключі та підписі.

Так само, як і для класичної схеми, криптоаналітик повинен або виконати атаку типу «екзистенційна підробка» зі складністю 2m, або віднайти прообрази до елементів ключа, що відповідають зміненим бітам.

2) Розміри ключів та підпису.

Довжина секретного та відкритого ключів у модифікованому алгоритмі Лампорта визначається як 2w * (m / w) * n, довжина ЕП (m / w) * n. Результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для розширеного алгоритму Лампорта в залежності від параметрів безпеки наведені в табл. 15.8.

Таблиця 15.8.

Залежність розміру ключів та підписів
(в кілобайтах) розширеного підпису Лампорта від n та w

	Ключ					Підпис				
n / *w*	128	192	256	384	512	128	192	256	384	512
2	4	9	16	36	64	1,00	2,25	4,00	9,00	16,00
3	5,333	12	21,333	48	85,333	0,67	1,50	2,67	6,00	10,67
4	8	18	32	72	128	0,50	1,13	2,00	4,50	8,00
5	12,8	28,8	51,2	115,2	204,8	0,40	0,90	1,60	3,60	6,40
6	21,333	48	85,333	192	341,333	0,33	0,75	1,33	3,00	5,33

Продовження табл. 15.8.

7	36,571	82,286	146,286	329,143	585,143	0,29	0,64	1,14	2,57	4,57
8	64	144	256	576	1024	0,25	0,56	1,00	2,25	4,00
9	113,778	256	455,111	1024	1820,44	0,22	0,50	0,89	2,00	3,56
10	204,8	460,8	819,2	1843,2	3276,8	0,20	0,45	0,80	1,80	3,20
11	372,364	837,818	1489,45	3351,27	5957,82	0,18	0,41	0,73	1,64	2,91
12	682,667	1536	2730,67	6144	10922,7	0,17	0,38	0,67	1,50	2,67
13	1260,31	2835,69	5041,23	11342,8	20164,9	0,15	0,35	0,62	1,38	2,46
14	2340,57	5266,29	9362,29	21065,1	37449,1	0,14	0,32	0,57	1,29	2,29
15	4369,07	9830,4	17476,3	39321,6	69905,1	0,13	0,30	0,53	1,20	2,13
16	8192	18432	32768	73728	131072	0,13	0,28	0,50	1,13	2,00

3) Обчислювальна складність.

Генерація приватного ключа розширеної схеми Лампорта еквівалентна генерації 2m * (m / w) n-бітних значень. Для обчислення відкритого ключа необхідно прогешувати кожен з елементів приватного ключа. У табл. 15.9 наведена залежність кількості гешувань для обчислення відкритого ключа від n та w.

Таблиця 15.9.

Залежність кількості гешувань для обчислення відкритого ключа від n та w

N / *w*	128	192	256	384	512
2	256	384	512	768	1024
3	342	512	683	1024	1366
4	512	768	1024	1536	2048
5	820	1229	1639	2458	3277
6	1366	2048	2731	4096	5462
7	2341	3511	4682	7022	9363
8	4096	6144	8192	12288	16384
9	7282	10923	14564	21846	29128
10	13108	19661	26215	39322	52429
11	23832	35747	47663	71494	95326
12	43691	65536	87382	131072	174763
13	80660	120990	161320	241980	322639
14	149797	224695	299594	449390	599187
15	279621	419431	559241	838861	1118482
16	524288	786432	1048576	1572864	2097152

Висновки

Підписи на основі геш-функцій є перспективним класом постквантових асиметричних криптоалгоритмів. Важливим компонентом криптосистем сімейства SPHINCS є одноразові підписи, зокрема, підпис Вінтерніца. В рамках підрозділу були проаналізовані загальновідомі одноразові ЕЦП - Лампорта, Лампорта - Діффі, Вінтерніца та ряд можливих модифікацій, які за певних умов дозволяють досягнути кращих результатів. Особливо цікавим є розширений підпис Лампорта, що зберігає обчислювальну складність та розміри ключів оригінального алгоритму і при цьому дозволяє вдвічі зменшити розмір підпису. Проте він не має важливої переваги підпису Вінтерніца – можливості повного обчислення публічного ключа з підпису, що є особливо важливим при використанні алгоритму у багаторівневих структурах гіпердерев, зокрема, в підписах сімейства SPHINCS.

15.2 Методи обчислення системних параметрів для електронного підпису «CRYSTALS-DILITHIUM» 128, 256, 384 ТА 512 біт рівнів безпеки

Вступ

Наразі спостерігається стійкий прогрес у створенні квантових комп'ютерів різних призначень та можливостей, одним із основних призначень є здійснення криптоаналізу існуючих асиметричних криптосистем [134, 136]. Практично завершується створення математичних основ та програмного забезпечення для таких квантових комп’ютерів [79, 137 – 141]. Особлива увага приділяється розробці великомасштабних квантових комп'ютерів, що призначаються для криптоаналізу існуючих стандартизованих криптосистем з відкритим ключем – електронних підписів, асиметричних шифрів та криптографічних протоколів різного призначення [79, 141]. Залишаються певні сумніви щодо симетричних криптоперетворень типу гешування, блокового та потокового симетричного шифрування. Але, як показують дослідження, збільшення при їх використанні розмірів параметрів та ключів, згідно з нинішніми поглядами, дозволяє забезпечити криптографічний захист на далеку перспективу. Зважаючи на можливості зламу існуючих асиметричних криптосистем, Національний інститут стандартів і технологій (NIST) США прийняв рішення у вигляді проєкту стандарту NIST 8309 щодо другого раунду конкурсу на перспективні стандартні алгоритми електронного (цифрового) підпису (ЕП) [135,136].

Прийняті нові стандартизовані алгоритми ЕП дозволять реалізувати системи ЕП, що будуть здатні захистити конфіденційну інформацію уряду США в доступному для огляду майбутньому, в тому числі, після появи квантових комп'ютерів. Таким чином, однією з основних проблем сучасної криптографії на міжнародному рівні є створення стандартизованих криптографічних систем (схем), які були б безпечними у постквантовий період. Таким чином, на світовому рівні зусилля значного числа криптологів, математиків та криптологів-практиків зосереджені на відкритому конкурсі NIST PQC [134–137, 142–148], одним із основних завдань якого є розробка та прийняття постквантового чи постквантових стандартів ЕП. Його підсумком є визначення фіналістів другого

етапу конкурсу у вигляді проєктів CRYSTALS-DILITHIUM, FALCON та RAINBOW [135,136]. Окрім цього, були визначені три альтернативні кандидати, які потребують більш детальних досліджень уже на четвертому етапі конкурсу – GeMSS, Picnic та SPHINCS+ [135].

Аналіз показує, що в Україні є розуміння існування загроз стандартизованим існуючим асиметричним криптоперетворенням. Так, розроблено та прийнято національний стандарт «Алгоритми асиметричного шифрування та інкапсуляції ключів» [135,136], що побудований на основі застосування алгебраїчних решіток. Особливістю цього стандарту (ДСТУ 8961-2019) [123] є суттєве підвищення криптографічної стійкості асиметричного шифрування та інкапсуляції ключів у перехідний та постквантовий період. На відміну від пропозицій та можливостей, що затверджені та прийняті NIST 8309 [122], є можливість використовувати його з рівнями безпеки 384 та 512 бітів. Найвищий рівень безпеки проєктів CRYSTALS-DILITHIUM, FALCON та RAIN-BOW 256 бітів класичного та 128 бітів квантового криптоаналізу. В той же час наші дослідження показали, що з урахуванням можливостей щодо забезпечення безпеки з використанням уже прийнятих в Україні симетричних криптоперетворень ДСТУ 7564-2014, ДСТУ 7624-2014 та ДСТУ 8845-2019 [123], національні стандарти ЕП повинні забезпечити в перспективі до 512 біт класичної та 256 біт квантової безпеки. Аналіз показав [135, 147, 148], що серед схем ЕП на решітках суттєві переваги надано проєктам ЕП CRYSTALS – DILITHIUM та FALCON, вони рекомендовані для подальшого дослідження та стандартизації в процесі третього етапу конкурсу. Причому кращими визначено як математичні основи, так і алгоритми ЕП CRYSTALS – DILITHIUM [136, 146 – 148]. ЕП CRYSTALS-DILITHIUM досліджувався у другому раунді як один із трьох проєктів ЕП на основі решітки. Його безпека грунтується на складності MLWE задачі криптоаналізу та задачі модульного короткого цілого рішення (MSIS) [147, 148]. Основою побудови алгоритму ЕП CRYSTALS–DILITHIUM є використання алгоритму Fiat-Shamir [147] з перериваннями. В ньому використовуються один і той же модуль і кільце поліномів для всіх наборів параметрів, а ентропія забезпечується за допомогою рівномірного розподілу. Це призводить, у порівнянні з гаусовим розподілом проєкту ЕП FALCON, до більш простої реалізації.

Загалом DILITHIUM має високі, збалансовані показники щодо розміру ключів та підписів, а також щодо ефективності алгоритмів генерації ключів, підпису та перевірки. DILITHIUM добре працює в реальних експериментах. Також у другому раунді до реалізації DILITHIUM додано опцію випадкової генерації підпису, що заснована на використанні AES. Це дало майбутні переваги при апаратних реалізаціях інструкцій ЕП. Крім того, було опубліковано нове дослідження безпеки в QROM [174], яке стосується DILITHIUM.

NIST рекомендував розробникам ЕП DILITHIUM додати набір параметрів 5-го рівня безпеки (5-й – за нашою класифікацією це 1-й рівень безпеки із 4-х рівнів). Також необхідні додаткові дослідження розуміння конкретної безпеки, оскільки DILITHIUM має найнижчий набір параметрів безпеки CoreSVP [155,136] з усіх схем на основі решітки, які все ще знаходяться в процесі

досліджень та порівняння. В той же час NIST вибрав DILITHIUM як фіналіста і очікує, що або DILITHIUM, або FALCON будуть стандартизовані як основна схема постквантового підпису в кінці третього раунду.

Необхідно відмітити, що в процесі формування вимог до ЕП NIST у рамках конкурсу був зацікавлений тільки в наборах загальносистемних параметрів до 256 біт класичної безпеки включно. Проте, на нашу думку, на перспективу доцільним є використання в DILITHIUM 384 і 512 біт безпеки проти класичного криптоаналізу та 192 та 256 біт безпеки проти квантового криптоаналізу. Але, як показали дослідження, як з точки зору теорії, так і практики, генерація загальносистемних параметрів для використання 256, 384 і 512 біт безпеки проти класичного криптоаналізу та 128, 192 та 256 біт безпеки проти квантового криптоаналізу в фіналі CRYSTALS-DILITHIUM не реалізована. Будемо також вважати, що відносно побудування ЕП CRYSTALS-DILITHIUM для 128 класичної та 64 біт квантової стійкості проблем немає [136, 137].

Під час ухвалення рішення стосовно фіналістів на прийняття стандарту ЕП в третьому раунді кращими та отримали рекомендації визначені проєкти ЕП CRYSTALS-DILITHIUM та FALCON. В оцінці NIST ці проєкти на основі структурованої решітки представляються найбільш перспективними та універсальними алгоритмами для електронного підпису, асиметричного шифрування та протоколу інкапсуляції ключів [135, 142 – 148].

Метою даного підрозділу є обгрунтування моделі безпеки, класифікація, первинний аналіз та оцінка відомих атак на криптосистему ЕП CRYSTALS–DILITHIUM, встановлення обмежень та розробка практичних алгоритмів обчислення (генерації) загальносистемних параметрів для забезпечення 128, 256, 384 і 512 біт безпеки проти класичного та 64, 128, 192 та 256 біт проти квантового криптоаналізу.

1 Сутність криптографічних перетворень фіналіста ЕП конкурсу NIST США «CRYSTALS-DILITHIUM»

У [135, 136, 146] наведено пропозиції та результати досліджень щодо сутності, властивостей та умов застосування кандидата на постквантовий стандарт ЕП Crystals-Dilithium (в подальшому Dilithium), що були сформовані та подані на конкурс NIST. На першому етапі досліджень в 2018 р. виявлені певні проблемні питання, щодо яких авторами проєкту стандарту були обґрунтовані та запропоновані певні удосконалення механізму ЕП Dilithium [136]. У цьому розділі розглянемо сутність та основні математичні та криптографічні положення Dilithium, що будуть потрібні для оцінки криптографічної стійкості. Причому особливу увагу звернемо на рівні криптографічної стійкості, тобто 1 – 5 рівні стійкості [134,135], що рекомендуються та можуть бути реалізованими поки що в проєкті стандарту ЕП Dilithium. Відмітимо, що одним із основних проблемних питань будемо мати на увазі обґрунтування необхідності та розробки удосконаленої версії ЕП Dilithium, що може забезпечувати в постквантовий період 128, 256, 384 і 512 біт безпеки проти класичного та 64, 128, 192 та 256 біт проти квантового криптоаналізу від найбільш загрозливих атак [135, 146 – 148].

Метод (схема) ЕП Dilithium ґрунтується на підході, що отримав назву "Fiat-Shamir з перериваннями" [136, 147]. Він в певній мірі схожий на схему, що запропонована з послідовним удосконаленням в [126, 128]. Для спрощеного та узагальненого подання механізму розглянемо спрощену його версію на рис. 15.1 [136, 146], на якому наведено алгоритми генерації ключа, вироблення та перевірки ЕП. Основні положення та вирішення задачі обчислення (генерування) системних параметрів удосконаленого ЕП Dilithium наведені нижче в цьому розділі.

Генерація основних складових ключа

Спочатку (рядок 01, рис. 15.1) генерується матриця поліномів A розміру $k \times \ell$, кожен з елементів якої є поліномом у кільці $R_q = A_q[X]/(X^n+1)$. В процесі попереднього розгляду будемо вважати, що модуль q=223−213+1, а степінь полінома n=256. Потім генеруються (обчислюються) випадкові вектори, тобто, множини поліномів секретного ключа $\boldsymbol{s_1}$ і $\boldsymbol{s_2}$ (рядок 02) відповідно з числом поліномів k та l. Коефіцієнти цих векторів (поліномів) є елементами поля R_η, тобто, з (малими) коефіцієнтами з розміром не більше ŋ (від – ŋ до ŋ). Далі з використанням матриці A та секретного ключа $\boldsymbol{s_1}$ і $\boldsymbol{s_2}$ обчислюється друга частина відкритого ключа $\boldsymbol{t=A\,s_1+s_2}$ (рядок 03), а перша частина відкритого ключа задається значенням ρ. Всі алгебраїчні операції з поліномами в механізмі виконуються над кільцем полінома $R_q = A_q[X]/(X^n+1)$. Четвертий рядок показує вихідні значення відкритого $\boldsymbol{P_k}$ та секретного $\boldsymbol{S_k}$ ключів. Детально алгоритм генерації ключа наводиться в [136].

Узагальнений алгоритм вироблення ЕП

В алгоритм вироблення ЕП вводяться значення секретного ключа $\boldsymbol{S_k}$ та повідомлення M, що підписується. Далі обчислюється вектор поліномів маскування $\boldsymbol{y}$ з коефіцієнтами, що є меншими, ніж γ_1 (рядок 07), а також обчислюється значення вектора поліномів $\boldsymbol{Ay}$. На основі отриманого значення Ay обчислюються старші біти $\boldsymbol{w_1}$ ("біти високого порядку") коефіцієнтів у цьому векторі поліномів (рядок 08); $\boldsymbol{w_1}$ є вектором, що містить всі поліноми $\boldsymbol{w_1}$. Потім обчислюється поліном c (рядок 09), що є поліномом у полі R_q з точно 60 символами ±1, а решта 0. Безпосередньо ЕП обчислюється у вигляді вектора поліномів $\mathbf{z=y+c}\,\boldsymbol{s_1}$.

Якщо значення ЕП z вивести безпосередньо після його обчислення (рядок 10), то механізм ЕП Dilithium не буде безпечним через те, що при певних значеннях секретний ключ може бути компроментований. Щоб уникнути залежності z від секретного ключа та його витоку, використовується відхилення вибірки. Для цього встановлюється значення параметра $\boldsymbol{\beta}$ як максимально можливий коефіцієнт cs_1. Оскільки c має значення 60 ±1, а максимальний коефіцієнт в s_1 дорівнює s_1, то легко побачити, що $\boldsymbol{\beta \le 60\eta}$. Якщо будь-який коефіцієнт z перевищує $\gamma_1-\beta$, то процес ЕП відхиляється, а процедура ЕП повторюється. Також, якщо коефіцієнт бітів низького порядку вектора $\boldsymbol{Az-ct}$ більше, ніж $\gamma_2-\beta$, то процес ЕП відхиляється, а процедура ЕП знову повторюється (рядок 11).

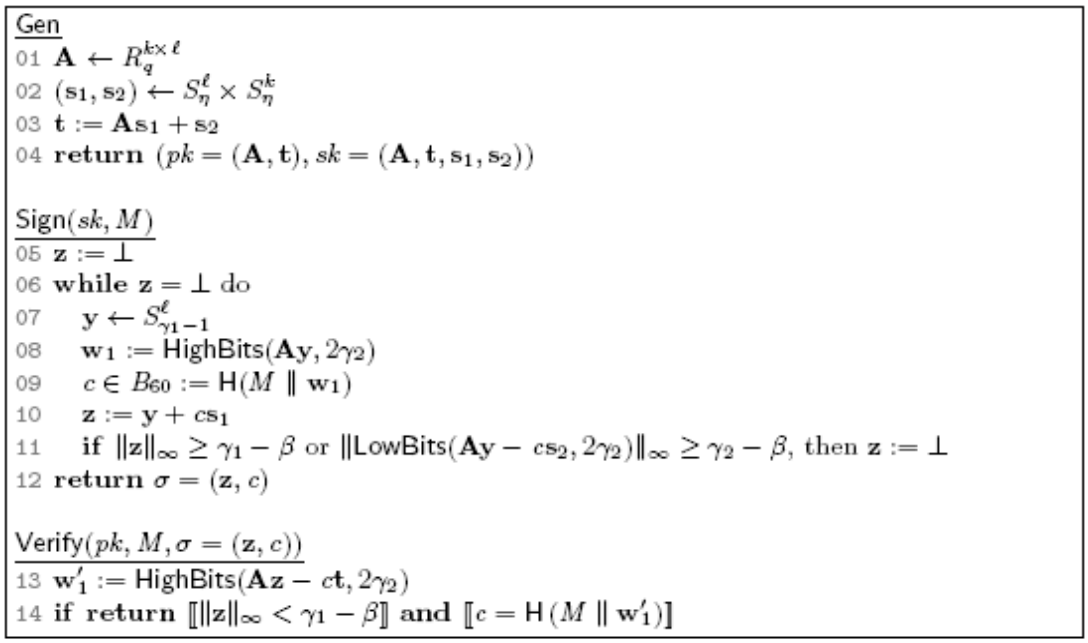


Рис. 15.1. Шаблон для механізму ЕП без стиснення відкритого ключа

Перша перевірка необхідна для безпеки ЕП, а інша – для його безпеки та правильності. Таким чином, процес ЕП повторюється, доки не будуть виконані дві наведені умови. Необхідно відмітити, що параметри $\boldsymbol{\beta}$ та $\boldsymbol{\gamma_1}$ і $\boldsymbol{\gamma_2}$ повинні бути вибрані, щоб очікувана кількість повторень ЕП була не надто висока (наприклад, від 4 до 7). Детально алгоритм ЕП наводиться в [136].

Узагальнений алгоритм перевірки ЕП

При перевірці ЕП перевірник обчислює $\mathbf{w'}_1$, як біти високого порядку вектора $\boldsymbol{Az-ct}$. Далі ЕП приймається, якщо всі коефіцієнти z менше, ніж $\gamma_1 - \beta$ і якщо c є геш-значенням повідомлення $\boldsymbol{M}$, що перевіряється, та значенням $\mathbf{w'}_1$ (рядок 13). Перевірка спрацьовує за умов, якщо

$$\text{HighBits}(Az-ct, 2\gamma2) = \text{HighBits}(Ay, 2\gamma2). \quad (15.1)$$

Дійсно, причиною цього є те, що для дійсного ЕП буде завжди виконуватися умова

$$||\text{LowBits}(Ay-cs2, 2\gamma2)|| \infty<\gamma2-\beta. \quad (15.2)$$

А так як коефіцієнти $\mathbf{cs_2}$ менше, ніж $\boldsymbol{\beta}$, то додавання $\mathbf{cs_2}$ недостатньо для того, щоб викликати будь-які перенесення, збільшивши будь-який коефіцієнт низького порядку до величини, щонайменше, γ_2. Таким чином, рівняння (15.1) є правильним, і ЕП перевіряється правильно. Детальніше ця умова розглядається нижче.

Загальні оцінки щодо рівня безпеки ЕП Dilithium

Якщо дотримуватись основного підходу, що викладений в [146 – 148], то безпеку механізму ЦП, що наведено на рис. 15.1, можна довести в моделі випадкового оракула (ROM), ґрунтуючись на складності двох проблем. Перша - це стандартна задача LWE (над кільцями поліномів), в якій пропонується відрізняти (A, t:= A s_1+s_2) від (A, u), де u рівномірно випадкове. Інша проблема

полягає в тому, що в [136,148] було названо проблемою **SelfTargetMSIS**, тобто, проблемою пошуку вектора:

$$\begin{bmatrix} z \\ c \\ v \end{bmatrix} \tag{15.3}$$

з малими коефіцієнтами і повідомлення (дайджесту) $\boldsymbol{\mu}$, що задовольняє умові

$$H\left(\mu \| [A \mid t \mid I] \cdot \begin{bmatrix} z \\ c \\ v \end{bmatrix}\right) = c, \tag{15.4}$$

де A і t рівномірно випадкові, а I – одинична матриця.

У ROM можна отримати (не жорстке) скорочення, використовуючи розгалужену лему [136,146] зі звичайної задачі SIS знаходження z' з малими коефіцієнтами, що задовольняють Az'=0, до **SelfTargetMSIS.** Можна дотримуватися цього точного підходу, щоб довести безпеку Dilithium в ROM на основі складності LWE та SIS.

У моделі квантового випадкового оракула (QROM), де зловмисник може запитувати H у суперпозиції, ситуація дещо інша. У [136,148] було показано, що Dilithium все ще базується на LWE та **SelfTargetMSIS** у QROM, навіть при жорсткому скороченні, коли механізм є детермінованим. Але більше не можна безпосередньо використовувати розгалужену лему (оскільки це інший тип розгляду), щоб дати квантове скорочення від **SIS до SelfTargetMSIS**. Є ще вагомі підстави вважати, що задача **SelfTargetMSIS**, а, отже, і Dilithium є безпечною в QROM. По-перше, не існує «природних» механізмів ЕП, побудованих із Σ-протоколів, що використовують перетворення Fiat-Shamir [147], які безпечні в ROM і небезпечні в QROM. Крім того, можна встановити параметри Dilithium (залишивши структуру механізму незмінною), щоб проблема **SelfTargetMSIS** стала інформаційно теоретично складною, що робить цю версію Dilithium безпечною в QROM на основі тільки LWE. Також зовсім недавно дві нові роботи ще більше звузили розрив між безпекою в ROM та QROM. Так, у [136] показано, що якщо покладений в основу Σ-протокол руйнується, але відрізняється особливою надійністю, то його перетворення Фіата – Шаміра є безпечним підписом в QROM.

2 Модель безпеки щодо ЕП на основі фіналіста ЕП «DILITHIUM»

Введемо поняття комплексної моделі безпеки криптографічних перетворень типу ЕП, орієнтуючись на [149 – 153]. Приймемо в якості часткових складових комплексної моделі безпеки щодо асиметричних криптоперетворень типу ЕП такі приватні моделі:

- порушника щодо асиметричних криптоперетворень типу ЕП;
- загроз щодо асиметричних криптоперетворень типу ЕП;
- безпеки щодо асиметричних криптоперетворень типу ЕП.

Нижче наводяться результати обгрунтування вказаних моделей безпеки щодо перспективних асиметричних криптоперетворень типу ЕП «DILITHIUM».

Сутність моделі порушника щодо перспективного ЕП

Побудова моделі порушника необхідна для того, щоб розробити комплекс заходів із забезпечення захищеності алгоритму. Така модель може бути побудована з урахування різних критеріїв.

По суті модель порушника – це опис можливих дій порушника, який формується на основі аналізу типу зловмисника, рівня його повноважень, знань, теоретичних та практичних можливостей. У якості порушника розглядається особа, що може отримати доступ до роботи з включеними до складу відповідної комп'ютерної системи засобами.

Порушники класифікуються за рівнем можливостей, що надаються їм штатними засобами КС. Виділяються чотири рівні таких можливостей. Класифікація є ієрархічною – кожний наступний рівень включає в себе функціональні можливості попереднього [154].

Вважатимемо, що порушник використовує усі доступні йому ресурси – найпотужніші комп'ютери і необмежений час.

Таким чином, у найгіршому випадку, порушник знає все про метод синтезу перспективного ЕП та про всі механізми безпеки, що виконуються під час синтезу та застосування, виключенням є те, що криптоаналітик не знає особистого ключа чи відповідним чином обґрунтовану частину особистого ключа. У найкращому випадку порушник не знає нічого про системні параметри та ключі. У нашому випадку це рівноймовірні варіанти.

Обгрунтування та сутність моделі загроз щодо ЕП

Модель загроз ЕП (далі – Модель загроз) повинна бути документом, яким закріплено найбільш повний перелік загроз щодо існуючих та перспективних ЕП. Відповідно до Законів, інформація у основних інформаційних ресурсах поділяється на відкриту і конфіденційну. Інформація у підтримуючих інформаційних ресурсах є технологічною інформацією.

При застосуванні ЕП, незалежно від видів додатків, використовуються асиметричні пари ключів; для кожної пари особистий та відкритий [78,136,159]. В подальшому, при реальному застосуванні ЕП, відкритий ключ, як правило, є сертифікатом відкритого ключа та є доступним усім користувачам інфраструктури відкритого ключа (ІВК).

Оскільки сертифікат відкритого ключа є відкритою інформацією, то під час обробки згідно з [78,134 – 136] він повинен зберігати цілісність, справжність, доступність, неспростовність та бути захищеним від несанкціонованих дій, які можуть привести до випадкової чи умисної модифікації, нав'язування хибного значення чи знищення. Усім користувачам, наприклад ІВК, має бути забезпечений доступ до ознайомлення з відкритою інформацією, в даному випадку, у вигляді сертифіката відкритого ключа [78,159].

Під час обробки (застосування) конфіденційна інформація, в нашому випадку особистий ключ, повинна зберігати цілісність, справжність, доступність, неспростовність та бути захищеною від несанкціонованих дій, має бути практично забезпечений її захист від несанкціонованого та

неконтрольованого ознайомлення, модифікації, знищення, копіювання та поширення. Тобто, безумовно має бути забезпечена конфіденційність особистого ключа кожного користувача. Також технологічна інформація повинна бути відома тільки авторизованим на це особам та зберігати цілісність.

Таким чином, в усіх відомих додатках, у яких використовується ЕП, стосовно відкритого ключа ЕП повинна бути можливість забезпечувати його цілісність, справжність, доступність, неспростовність та бути захищеною від несанкціонованих дій, які можуть привести до випадкової чи умисної модифікації, нав'язування хибного значення чи знищення. Стосовно особистого ключа мають бути забезпеченими його цілісність, справжність, доступність, неспростовність та бути захищеним від несанкціонованих дій, а також повинен бути забезпеченим його захист від несанкціонованого та неконтрольованого ознайомлення, модифікації, знищення, копіювання та поширення, тобто, конфіденційність.

Тобто, як існуючі ЕП, так і перспективні ЕП повинні дозволяти гарантовано захищати їх асиметричні пари ключів у відповідності із вказаними вище вимогами, незалежно від їх подання при використанні – в апаратному, програмному чи апаратно-програмному вигляді. Причому, незалежно від виду їх обробки, реалізація загроз, що спрямовані на вказані ресурси, може призводити до порушення вимог безпеки первинних інформаційних ресурсів, вплинути на інше ПЗ, що підписується, та, в окремих випадках, на функціонування апаратних ресурсів.

Загальні загрози щодо ЕП

Перелік можливих загроз безпеці застосування існуючих та перспективних ЕП був сформований з числа загроз, наявних у IT-Grund-schutz Catalogues з урахуванням апаратних, програмних та апаратнопрограмних ресурсів, технологій обробки даних та механізмів криптографічного захисту при застосуванні ЕП.

За результатами аналізу IT-Grundschutz Catalogues щодо методів синтезу та застосування системи визначено такі загрози: атака "Людина посередині"; атака Clickjacking; викрадення даних за допомогою мобільних носіїв інформації; викрадення пристроїв, носіїв чутливої інформації та документів; витік каналами побічних електромагнітних випромінювань і наведень; відмова від дій; відмова криптомодулю; відсутнє або недостатнє оповіщення при виникненні інцидентів безпеки; відсутність дозволів для обробки персональних даних; відсутність прозорості для особи, що зацікавлена та уповноважена контролювати захист даних; відсутня або неповна документація; втрата цілісності інформації, яка повинна бути захищена; старіння криптографічних методів; зловживання повноваженнями; зловживання правами адміністратора; зловживання правами користувачів; компрометація криптографічних ключів; крадіжка чутливих даних; не виявлені інциденти інформаційної безпеки; невірне тлумачення події інформаційної безпеки; недооцінювання актуальності виправлень і змін; неналежне зберігання носіїв інформації в разі виникнення надзвичайної ситуації; необережне знищення обладнання або даних; неправильне використання криптомодулів; несанкціоноване використання криптомодулів; несанкціоноване

використання прав; нестійкі криптографічні алгоритми; неякісна або відсутня автентифікація; підробні сертифікати; порушення законів або правил; проблеми при автоматизації поширення виправлень і змін; розголошення чутливої інформації; систематичний перебір паролів; троянський кінь; уразливості або помилки ПЗ; шкідливе програмне забезпечення. При синтезі та застосуванні перспективних ЕП повинно бути зроблено перекриття названих загроз.

Обгрунтування та сутність моделі безпеки щодо ЕП

В якості моделі безпеки стосовно асиметричних постквантових криптоперетворень ЕП пропонується застосовувати EUF-CMA модель [149 – 153]. EUF-CMA модель визначає екзистенційну непідроблюваність від атак на основі адаптивно вибраних повідомлень. Зокрема, безпека в сенсі EUF-CMA не дозволяє криптоаналітику (зловмиснику) виробляти ЕП для повідомлень, що залежать від ключів, наприклад, ЕП при застосуванні особистого s_k ключа. По суті, при застосуванні механізму безпечного ЕП, згідно з моделлю EUF-CMA, що є безпечною для EUF-CMA у випадку, коли не застосовуються запити повідомлення. Але при наявності хоча б одного запиту повідомлення, що залежить від ключів, безпека механізму ЕП порушується.

Існує два загальних формальних визначення для забезпечення безпеки схеми ЕП. Кожне з цих визначень представлено як "гра" або експеримент, який виконується між атакуючим (attacker) та деяким чесним претендентом (challenger).

Неформально експеримент EUF-CMA (екзистенційна непідроблюваність при атаці на основі підібраних повідомлень) виконується так:

1. Претендент генерує дійсну пару ключів (p_k, s_k) і надає p_k атакуючому.

2. Атакуючий тепер може повторно запросити підписи на підібраних (вибраних) повідомленнях $\boldsymbol{(M_1,..., M_q)}$ за своїм вибором і отримує дійсні підписи $\boldsymbol{(\sigma_1,...., \sigma_q)}$ у відповідь.

3. По завершенню експерименту зловмисник повинен вивести повідомлення та підпис **M*, σ*** такі, щоб одне повідомлення було не одним із повідомлень, які вимагали попередніх кроків (1) і (2), а повідомлення/підпис перевіряється правильно з відкритим ключем.

Схема вважається безпечною, якщо жоден (ефективний) зловмисник не має ні найменшої переваги у виконанні зазначених умов. Зазвичай кількість повідомлень q обмежується лише часом дій атакуючого, однак, для спеціального випадку одноразових ЕП зловмисник обмежується запитом лише одного підпису на кроці (2).

Це визначення досить сильне, але не настільки сильне, наскільки це можливо. Дещо сильнішим визначенням є визначення та відповідна вимога SUF-CMA.

Неформально, експеримент SUF-CMA (Сильна екзистенційна непідроблюваність при атаці на основі підібраних (вибраних)) повідомлень виконується так:

1. Те саме, що і в попередньому експерименті.

2. Те саме, що і в попередньому експерименті.

3. Після завершення експерименту атакуючий повинен вивести повідомлення та підпис M*, σ* такі, щоб пара (1) (M*, σ*) не була одним із запитаних повідомлень, а підпис повернувся на попередньому кроці, повідомлення/підпис (2) перевіряється правильно з відкритим ключем.

Атакуючий виграє, якщо вона задовольняє наведеним умовам:

1. Головна відмінність полягає в тому, що це більш сильне визначення гарантує, що атакуючий не зможе підібрати підпис. Наприклад, схема, в якій атакуючий може повторно рандомізувати дійсний підпис, щоб він залишався дійсним, але виглядав інакше, ніж вихідне значення, не задовольнила би SUF-CMA [149, 152, 153].

2. Вищезгадане визначення безпеки EUF-CMA здійснюється одноразовими схемами підпису в очевидний спосіб – єдиною модифікацією є те, що Aeuf може запитувати оракул підпису OS лише один раз. Зокрема, безпека в сенсі EUF-CMA не дозволяє зловмиснику отримувати підписи повідомлень, залежних від ключів, як підпис на повному секретному ключі s_k. Фактично, враховуючи EUF-CMA-безпечну схему ЕП, легко скласти схему підписів, яка все ще є EUF-CMA-безпечною, але один запит на повідомлення, залежного від ключів, порушує безпеку схеми.

3 Огляд та аналіз атак щодо стійкості ЕП «DILITHIUM»

Класифікація атак та загальна оцінка стійкості ЕП «DILITHIUM»

В постквантовій криптології актуальними є завдання забезпечення криптографічної стійкості щодо класичних та квантових атак. Проблема зводиться до навчання з помилками (LWE), сутність якої у наступному.

Нехай n, q є деякими натуральними числами, χ – деякий ймовірнісний розподіл над Z та S – секретний вектор (множина поліномів) у Z_q^n. Ймовірнісний розподіл $L_{s\cdot\chi}$ над $Z_q^n \times Z_q$ отримується обчисленням [136, 146]

$$(a, c) = (a, \langle a, s\rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q, \tag{15.5}$$

де $a \in Z_q^n$ отримується з рівномірного розподілу та $e \in \mathbb{Z}$ з розподілу χ. Атака Decision-LWE полягає у тому, щоб визначити, чи отримана пара $(a,c) \in Z_q^n \times Z_q$ з розподілу $L_{s\cdot\chi}$ або рівномірного розподілу. Її Search-LWE складова полягає у знаходженні S з пари $(a,c) \in Z_q^n \times Z_q$. Вважається, що як проблема Decision-LWE, так і Search-LWE [136, 146] з точки зору складності є еквівалентними та можуть бути зведені одна до одної за поліноміальний час і фактично є різними поглядами на одну і ту ж задачу. Розподіл χ для цих задач, зазвичай, є дискретним нормальним розподілом над кінцевим полем з математичним очікуванням, рівним 0, та дисперсією, що характеризується параметром a. При цьому більшість атак на LWE полягають у знаходженні деякого вектора ν з певною нормою на решітці L з фіксованим об'ємом $vol(L)$, але з різною розмірністю m, яка фактично характеризує оптимальну кількість пар $(a_i, c_i) \in Z_q^n \times Z_q$, необхідних для атаки.

Аналіз показав, що складність проблеми навчання з помилками точно знайдена лише асимптотично. Так, доведено [160,161], що за певних умов

складність вирішення LWE в просторі розмірності n становить щонайменше $2^{O(n)}$. Цей результат зручно використовувати для оцінки загальносистемних параметрів, проте конкретні оцінки складності криптостійкості досі не відомі. Це пов'язано з тим, що атаки на LWE зводяться в кінцевому випадку до редукції решіток.

За останні 10 років помітний суттєвий прогрес у цьому напрямку, що призводить до постійного уточнення та зміни оцінок. Він стосується більшості сучасних криптосистем, у яких використовуються варіанти LWE над поліноміальними кільцями (PR-LWE), тобто, розподіл розглядається не над Z_q, а над $\boldsymbol{Z_q}[\boldsymbol{X}]/(\boldsymbol{f}(\boldsymbol{x}))$. При аналізі криптоперетворень засобом множення використовується поліном виду $f(x) = x^{2^n} + 1$ і відповідне поле $\boldsymbol{R_q = Z_q}[\boldsymbol{X}]/(\boldsymbol{x^{2^n} + 1})$.

Причому, якщо $(a_i, c_i) \in R_q \times R_q$, то задача має назву R-LWE. Коли $(a_i, c_i) \in R_q^d \times R_q$ – M-LWE відповідно.

Відмітимо, що поліном $f(x) = x^{2^n} + 1$ обраний невипадково. Його властивості дозволяють здійснити доказ про стан захищеності асиметричного криптоперетворення щодо квантових атак. Також його властивості дозволяють використати для операцій множення поліномів швидке NTT – перетворення і, як наслідок, дозволяють створювати швидкодіючі реалізації криптоперетворень. Однак з теорії Галуа відомо [79,136,139], що поле $\boldsymbol{R_q = Z_q}[\boldsymbol{X}]/(\boldsymbol{x^{2^n} + 1})$ має складну структуру підполів, що може бути використано для здійснення криптоаналізу. Проте, на нинішній час на практиці такі атаки носять більше обмежений теоретичний характер, ніж практичний. Фактично сучасними криптологами ігноруються додаткові можливості, а R-LWE та M-LWE розглядаються як LWE. Це пояснюється тим, що для полінома $f(x) = x^{2^n} + 1$ доведено, що R-LWE та M-LWE є складнішими за LWE атаки.

На основі аналізу визначено [162 – 170], що стосовно атак на LWE можливо виділити та необхідно розглядати наступні:

1) атака грубої сили, тобто, повного перебору;
2) традиційна атака зустріч посередині;
3) атака на основі алгоритму Arora-Ge;
4) BKW, коли LWE зводиться до SIS атаки;
5) диференційні атаки на помилки на детермінований ЕП на решітках;
6) Primal attack (Search-LWE зводиться до BDD атаки);
7) Dual attack (Decision-LWE зводиться до SIS);
8) зведення до uSVP атаки пошуку короткого вектора.

Розглянемо детальніше вказані атаки та зробимо їх оцінки та порівняння. Відмітимо, що атаки (1 – 4) наведеного переліку є експоненційно складними, аналіз їх складності та можливості застосування у загальному виді наведено в [162 – 168]. Попередні оцінки дозволили зробити висновки про неможливість їх використання, оскільки їх часова та просторова складності є експоненційно складними (для випадку застосування в обґрунтовано вибраних розмірах системних параметрів та ключів).

В [171] запропоновано диференційні атаки на помилки на детермінований ЕП на решітках. По суті, вони зводяться до розширення застосування диференційних атак на помилки на криптографію на основі решітки. Показано, чи вразливі до таких атак дві детерміновані схеми ЕП на основі решітки – Dilithium та qTESLA. Показано, що одиничні випадкові помилки можуть спричинити сценарій повторного використання початкового стану (нонсу), який дозволяє відновити ключ. Також зроблено розширення до атак, викликаних помилками з частковим повторним використанням нонсу, які не пошкоджують дійсність обчислених ЕП, тому їх важче виявити.

Використовуючи лінійну алгебру та методи зведення базису решітки, зловмисник після успішного внесення помилки може отримати один із елементів секретного ключа. Деякі інші частини ключа неможливо відновити, але показано, що алгоритм підпису з підробленим підписом все ще може успішно підписувати будь-яке повідомлення. В роботі [171] зроблено експериментальні перевірки наведених атак, наприклад, виконуючи збої годинника на мікроконтролері ARM Cortex-M4 [32]. Зокрема, показано, що до 65,2 % часу виконання Dilithium вразливе до непрофільованої атаки, де випадкова помилка вводиться в будь-якому місці під час процедури підпису і все ще призводить до успішного відновлення ключа. Але віднесемо ці можливості зловмисника до атак сторонніми каналами, їх аналіз є окремою задачею досліджень.

Атаки на основі решіток

В цьому підрозділі розглядаються такі основні атаки на основі решіток:

- із застосуванням зведення LWE до BDD атаки;
- Dual Attack зі зведенням LWE до SIS атаки;
- Primal Attack виду (LWE ≥ uSVP);
- на основі алгоритму SIS.

Розглянемо ці атаки детальніше.

Атака із застосуванням зведення LWE до BDD.

Сутність атаки у наступному. Припускається, що відомі m пар:

$$(a_i, c_i) = (a_i, \langle a_i, s_i \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Зробимо запис наведеного у більш зручному вигляді:

$$(A, c) = (A, A * s + e) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times 1}. \qquad (15.6)$$

Для (15.6) побудуємо решітку

$$L = \{Ax \bmod q : x \in \mathbb{Z}_q^m\}.$$

Очевидно, що s є вектором на решітці та є найближчим до вектора $As + e$. Задача знаходження найближчого вектора на решітці до деякого довільного вектора має назву BDD та вирішується за допомогою алгоритму Бабаї [162]. Алгоритм є поліноміально складним, він працює за поліноміальний час, проте рішення знаходиться тільки з деякою ймовірністю. Для LWE цю ймовірність можна оцінити як:

$$\prod_{i=0}^{m-1} erf\left(\frac{\left\|b_i^*\right\|\sqrt{\pi}}{2\alpha q}\right), \tag{15.7}$$

де $\| b_i^* \|$ – норми ортогоналізованих за Граммом – Шмідтом векторів базису решітки (тобто, стовпців матриці A). Для того, щоб ймовірність вирішення BDD була близька до одиниці, потрібно зменшити $\| b_i^* \|$, тобто, редукувати базис. Одним з найкращих алгоритмів для редукції базису є алгоритм блочної редукції Коркіна – Золотарьова (BKZ) та його модифікаціїї (BKZ 2.0) [136,167]. Фактор Ерміта δ_0 для редукції можливо отримати із співвідношення:

$$\begin{aligned} \left\|b_0\right\|_2 &= \delta_0^n q^{\frac{1}{n}}, \\ \left\|b_i^*\right\|_2 &\approx \delta_0^{-2i+n} * q^{\frac{1}{n}}. \end{aligned} \tag{15.8}$$

Алгоритм BKZ 2.0 залежить від натуральних параметрів β і m, що позначають так звані довжину блоку та кількість ітерацій відповідно і дозволяє будувати редукований за Коркіним – Золотарьовим базис повної решітки вимірності n за $2^{E(\beta,m,n)}$ операцій, де:

$$\begin{aligned} E(\beta,\ m,\ n) = \\ = 0{,}000784314\ \beta^2 + 0{,}366078\ \beta + log(\ (n)m) + 0{,}875. \end{aligned} \tag{15.9}$$

В [166] описано симулятор алгоритму BKZ 2.0, який дозволяє обчислювати за вхідним параметром $\delta_0 > 1$ такі значення параметрів β і m, що застосування алгоритму BKZ 2.0 з цими параметрами до будь-якого вхідного базису повної решітки вимірності n приводить до її редукованого базису з кореневим фактором Ерміта δ_0.

Dual Attack зі зведенням LWE до SIS

Існують атаки на дуальній решітці засобом зведення LWE до SIS задачі. Сутність зведення у наступному. Побудуємо спочатку решітку $\widehat{\boldsymbol{L}} = \{\boldsymbol{x} \in \boldsymbol{Z_q^m} | \boldsymbol{A}^* \boldsymbol{x} = \boldsymbol{0 mod q}\}$. Задача SIS полягає у знаходженні такого найменшого цілого $\boldsymbol{x} \in \boldsymbol{Z^n}$, щоб $\boldsymbol{A}^* \boldsymbol{x} = \boldsymbol{0}$. Припустимо, що такий вектор знайдений, тоді можна вирішити задачу Decision-LWE. Нехай дано m пар

$$(A,c) = (A,\ A^* s + e) \in Z_q^{m \times n} \times Z_q^{m \times 1}.$$

Обчислимо скалярний добуток $\langle x, c \rangle$:

$$\left\langle x,c \right\rangle = x * a * s + x * e = 0 * s + x * e = x * e = \left\langle x,e \right\rangle. \tag{15.10}$$

Оскільки вектор $x \in Z^n$ відомий, то з цієї рівності можна знайти значення вектора помилок e, проте простір помилок залишається досить великим. У [169] показано, що, якщо вектор x має норму

$$\|x\|_2 = \frac{1}{\alpha} * \sqrt{\frac{\ln(\frac{1}{\varepsilon})}{\pi}}, \tag{15.11}$$

то з ймовірністю, близькою до 1, можливо вирішити цю задачу, при цьому знадобиться $e^{1/2}$ запусків вирішувача SIS. Вирішувач фактично знаходить достатньо малий вектор на решітці, тобто, вирішує задачу SVP. У [162] було показано, що при цьому фактор Ерміта δ_0 має бути не більше

$$\log \delta_0 = \frac{\log^2(\frac{1}{\alpha}\sqrt{\frac{\ln(\frac{1}{\varepsilon})}{\pi}})}{4 * n \log q}. \tag{15.12}$$

Атаки такого типу називаються Dual Attack. Точна оцінка атаки потребує вибрати певний вирішувач. У якості вирішувача можливо взяти BKZ 2.0 і виконати оцінку як для атаки на BDD.

Primal Attack (LWE ≥ uSVP)

Нехай в атаці, що розглядається, решітка містить вектор *s*. Сутність Primal Attack полягає у тому, щоб побудувати таку решітку, на якій буде лежати вектор (*s*,*e*,1) і він буде найменшим унікальним вектором, тобто, вона зводиться до задачі uSVP [136]. Такою решіткою буде

$$\Lambda = \{x \in \mathbf{Z}^{m+n+1} : (A | I_m | -c) * x = 0 \bmod q\}.$$

Для пошуку вектора можливо скористатися вирішувачем BKZ 2.0 і редукувати решітку; як наслідок, b_0 буде рішенням, що шукається. Далі, для вдалої редукції оцінити фактор Ерміта можливо виразом [162, 163]:

$$\log \delta_0 = \frac{1}{4n^2 \ln^2 q}\left(W\left((-2n \ln q) * (\sqrt{n \log q}) * \frac{(\tau\alpha)^2}{2\pi}\right)\right)^2. \tag{15.13}$$

Зауваження стосовно алгоритму SIS

Як вище вказувалось, задачу SIS можливо звести до пошуку найменшого вектора на дуальній решітці, тобто, у вигляді:

$$v \approx \gamma\lambda_1(\hat{L}), \hat{L} = \{x \in \mathbb{Z}_q^m | A * x = 0 \, mod \, q\}. \tag{15.14}$$

Проте автори Dilithium запропонували інший підхід – шукати рішення як найменший вектор на решітці з обмеженням на коефіцієнти

$$v \approx \gamma\lambda_1(L), \|\gamma\lambda_1(L)\|_\infty < \beta, L = \{Ax \, mod \, q : x \in \mathbb{Z}_q^m\}. \tag{15.15}$$

Але автори не вказують, чому вони вважають, що такий підхід є кращим за класичний підхід.

Аналіз атаки щодо SUF-CMA безпеки

Згідно з [172], схема підпису забезпечує SUF-CMA безпеку, яка може бути визначена як:

$$Adv^{SUF-CMA}_{Dilithium}(A) \le Adv^{MLWE}_{k,l,D}(B) + Adv^{SelfTargetMSIS}_{H,k,l+1,\xi}(C) + Adv^{MSIS}_{k,l,\xi'}(D) + 2^{-254}, \quad (15.16)$$

де D є рівномірним розподілом над відповідним полем значень,

$$\xi = \max(\gamma_1 - \beta, 2 * \gamma_2 + 1 + 2^{d-1} * h),$$

$$\zeta' = max(2 * (\gamma_1 - \beta), 4 * \gamma_2 + 2).$$

Згідно (15.16) безпека залежить від трьох атак: MLWE, SelfTargetMSIS та MSIS.

Захист від MLWE потрібен для захисту від атак на відтворення ключа. Атаки SelfTargetMSIS та MSIS направлені на підробку повідомлення. Варто зазначити, що до MLWE є дуальна атака, яка іноді може мати кращі результати. Таким чином, для удосконаленого ЕП «DILITHIUM» потрібно знайти параметри, що є стійкими до атак Primal MLWE, Dual MLWE, SelfTargetMSIS та MSIS.

У [172,173] для оцінки безпеки застосовувалася методика “core SVP hardness”. Вона базується на тому факті, що досі не знайдено ефективних методів експлуатації структури модульних решіток, тож можна звести MLWE та MSIS до LWE та SIS без втрати точності аналізу. Безпеку останніх можна звести до проблеми SVP, яка є добре вивченою (порівняно з іншими проблемами на решітках). Проте з кожним роком з’являються нові, більш ефективні, алгоритми для вирішення цієї проблеми. Найкращим відомим алгоритмом для редукції решіток є BKZ. Він мінімізує кожен вектор решітки у ортогональному підпросторі розмірності *b*. Найкраща відома складність атаки для класичних комп’ютерів складає:

O(20.292b)

та

O(20.265b) (15.17)

для квантових.

Під час роботи BKZ викликає “оракула” для мінімізації у ортогональному підпросторі. Найкраща відома складність оракула складає O(20.2075b).

Аналіз атаки Primal Attack

Проблема MLWEk,l,D зводиться до LWEnk,nl,D, яку, у свою чергу, можна звести до unique-SVP на решітці розмірності $d = n * l + n * k + 1$. В ній норму найменшого вектора можна оцінити, як $\lambda \approx \xi * \sqrt{d}$, де ξ є середньоквадратичним відхиленням для розподілу, що використовується. Оскільки в Dilithium використовується рівномірний розподіл, то $\xi = sqrt\left(\frac{\sum i^2}{2*\eta+1}\right)$.

Атака вважається успішною, якщо була знайдена довжина блока, для якої проєкція найменшого вектора *v* у векторний простір, “натягнутий” на останні *b* вектори Грамма – Шмідта, коротше ніж *b*d–b*. Довжину вектора *v* можна

оцінити, як $\delta^{2*b-d-1} * q^{\frac{m}{d}}$, де $\delta = \left((\pi * b)^{1/b} * \frac{b}{2*pi*e}\right)^{\frac{1}{2(b-1)}}$, а норму проєкції, як $\xi\sqrt{b} \leq \delta^{2*b-d-1} * q^{\frac{m}{d}}$. Алгоритм реалізації Primal Attack наведено нижче.

Алгоритм реалізації Primal Attack

Алгоритм Primal Attack
Вхідні дані: n, q, η, l, k
Вихідні дані: криптобезпека λ_1.

bestcost = $+\infty$
for dim $\in [5, n*k]$, $b \in [50, n*k+n*l]$ {
$d = n*l + dim + 1$
$\delta = \left((\pi * b)^{\frac{1}{b}} * \frac{b}{2*pi*e}\right)^{\frac{1}{2(b-1)}}$
$l = \delta^{2*b-d-1} * q^{\frac{m}{d}}$
$\xi = sqrt\left(\frac{\sum i^2}{2*\eta+1}\right)$
Якщо $\xi\sqrt{b} \leq l$ і 0.296*b <bestcost, тоді bestcost = 0.296*b
}
Return bestcost

Аналіз атаки Dual attack

Атака Dual attack полягає у вирішенні decision–LWE на дуальній решітці. Нехай знайдено певний вектор. Якщо рішення належить до LWE, то його коефіцієнти повинні мати нормальний розподіл, інакше буде рівномірний розподіл. Відстань між нормальним та рівномірним розподілом можна оцінити, як:

$$\epsilon = 4\exp\left(-2 * \pi^2 * \left(l * \frac{\xi}{q}\right)^2\right),$$

де l – довжина вектора, ξ – середньоквадратичне відхилення. Це значення можна трактувати, як ймовірність успіху. На жаль, ця ймовірність досить мала. Для успішної атаки потрібно атаку виконати щонайменше $R = \max\left(\frac{1{,}1}{2^{(0.2075*b)\epsilon^2}}\right)$ разів. Довжину вектора можна оцінити, як $\delta^{d-1} * q^{\frac{n*l}{d}}$, де d – розмірність підрешітки, що обрана для атаки.

Алгоритм реалізації Dual attack наведено нижче.

Результати застосування SIS та SelfTargetMSIS до ЕП «DILITHIUM»

У [172] було показано, що SelfTargetMSIS можна звести до MSIS, а відповідну проблему вирішення MSIS – до вирішення SIS. Задача SIS полягає у знаходженні вектора цілих чисел, які мають досить малу норму || ||. Автори пропонують шукати такий вектор за допомогою BKZ (у якому використовується норма ||2||). Згідно з [136,172], після редукції решітки, що асоційована з

криптосистемою ЕП «DILITHIUM», в векторах умовно можна виділити три зони. У першій зоні коефіцієнти будуть розподілені рівномірно за модулем q.

Алгоритм реалізації Dual attack

Алгоритм Dual Attack
Вхідні параметри: n, q, k, l, η
Вихідні параметри: криптостійкість λ_2

bestcost = $+\infty$
for dim $\in [5, n*k]$, $b \in [50, n*k+n*l]$ {
$d = n*l+dim+1$
$\delta = \left((\pi*b)^{\frac{1}{b}} * \frac{b}{2*pi*e}\right)^{\frac{1}{2(b-1)}}$
$l = \delta^{d-1} * q^{\frac{n-l}{d}}$
$\xi = sqrt\left(\frac{\sum i^2}{2*\eta+1}\right)$
$R = \max\left(\frac{1{,}1}{2^{(0.2075*b)\epsilon^2}}\right)$
Якщо $\log_2(R*2^{0.2075b})$<bestcost, тоді bestcost = $\log_2(R*2^{0.2075b})$
}
Return bestcost

У другій зоні коефіцієнти матимуть нормальний розподіл, а у третій зоні усі коефіцієнти дорівнюють 0. Криптостійкістю є обернена величина до добутку ймовірності того, що усі коефіцієнти менші за певне значення (що є параметром атаки) на час роботи підпроцедури, що виконує редукцію в ортогональному підпросторі із застосуванням алгоритму BKZ. Для виявлення меж зон i, j використовується ряд евристичних міркувань. До редукції перші вектори матимуть евклідову норму q, а останні 1. Після редукції з'явиться певний "схил". Довжина цього схилу і визначатиме межі другої зони (див. Рис. 15.2).

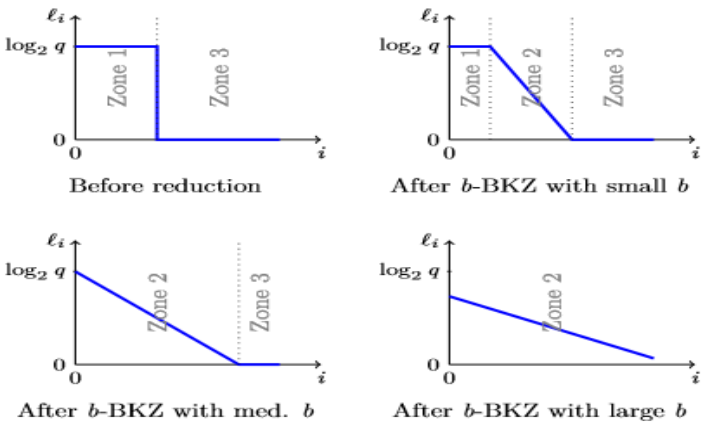


Рис. 15.2

Згідно з [172] “крутизну схилу” можна обчислити як:

$$slope(b) = -\frac{1}{b-1}\left((\pi * b)^{\frac{1}{b}} * \frac{b}{2 * pi * e}\right).$$

Алгоритм обчислення меж зон наведено нижче

Алгоритм обчислення меж зон
Вхідні дані: $b, q, q_{count}, zero_{count}$
Вихідні дані: i – кінець першої зони. j – кінець другої зони. l – довжина найбільшого вектору “на схилі”.
$slope = \frac{1}{b-1}\left((\pi * b)^{\frac{1}{b}} * \frac{b}{2 * pi * e}\right)$ slope_count = $\frac{\log_2 q}{slope}$ slope_sum=slope_count*log(q)-slope*slope_count*(slope_count+1)/2 I = floor((q_count*log(q)-slope_sum)/log(q) diff = log(q)*q_count-slope_sum – i*log(q) L = log(q) +log_slope + 1.0*diff/slope_count j = i + slope_count return (i,j,L)

У [154] сказано, що рандомізація базису дозволяє зменшити складність атаки. На практиці це означає, що ***i*=0** у алгоритмі SIS attack, який наведено нижче.

Алгоритм SIS attack
Вхідні дані: криптостійкість $\lambda_з$
Вихідні дані:
$\boldsymbol{\sigma = \frac{L}{\sqrt{J - I + 1}}}$ $\boldsymbol{p_1 = \left(2 * \frac{\xi}{q}\right)^I, p_2 = \Phi\left(\frac{\xi}{\sigma * \sqrt{2}}\right)^{J-I+1}}$ return $\log_2\left(\frac{1}{p_1 * p_2 * 2^{0.2075b}}\right)$

В підрозділі 4 наведено результати обчислення (генерації) системних параметрів щодо удосконаленого ЕП DILITHIUM для 128, 256, 384, 512 біт стійкості.

4 Генерація загальносистемних параметрів ЕП DILITHIUM для 128, 256, 384, 512 біт стійкості

Загальні положення

Для генерації системних параметрів удосконаленого ЕП DILITHIUM використаємо результати аналізу відомих атак на криптосистему, що наведені в

підрозділі 3, та встановимо умови, за яких забезпечується захист від них. Найочевиднішим підходом атаки криптосистеми є відновлення особистого ключа на основі використання значення відкритого ключа.

Атака відновлення особистого ключа на основі відкритого ключа

Як показано вище, в криптосистемі ЕП DILITHIUM вирішення цієї задачі зводиться до задачі MLWE (Module learning with errors) [172]. В літературі для аналізу MLWE використовується стратегія зведення до проблеми LWE [136,172,173]. З однієї сторони, це полегшує аналіз, але з іншої – залишає простір для оптимізацій. Проблема LWE може бути вирішена різними шляхами. До першого можна віднести комбінаторні атаки, такі, як BKW, Arora-Ge та зустріч посередині. Ці атаки іноді є ефективними, коли коефіцієнти полінома, що є особистим ключем, належать до деякої невеликої множини (наприклад, $\{-1,0,1\}$). Як правило, їх розгляд має сенс тільки у разі гомоморфних криптосистем [136,172]. Для ЕП DILITHIUM вони значно поступаються в ефективності і надалі розглядатися не будуть. До другої категорії належать атаки через редукцію решіток. Розглядають атаки на решітці та дуальній до неї [172]. У першому випадку будується решітка:

$$\Lambda = \{x \in \mathbf{Z}^{m+n+1} : (A \mid I_m \mid -c) * x = 0 \bmod q\}. \tag{15.18}$$

Далі за допомогою алгоритмів редукції базису знаходиться найменший унікальний вектор (фактично задача зводиться до USVP) [173]. Для оцінки стійкості до цієї атаки зазвичай використовується підхід, що був запропонований в роботі [172]. Фактично виконується пошук найменшого розміру блоку β для BKZ, такого, що виконується умова:

$$\sigma\sqrt{\beta} \le \delta_0^{2\beta-d-1} q^{\frac{n}{d}},$$

де δ, q – загальносистемні параметри, d – розмірність решітки, δ_0– максимальне значення фактора Ерміта, за яким решітка буде достатньо редукованою. Причому δ_0 можна виразити через β, як $\delta \approx \left((\pi\beta)^{\frac{1}{\beta}} * \beta/2\pi e\right)^{1/2(\beta-1)}$. Таким чином, маємо нелінійне діафантове рівняння. Вирішувати його можна повним перебором або більш оптимізованими шляхами. Для дуальної атаки будується решітка:

$$L = \{(x,y) \in \mathbf{Z}_q^m \times \mathbf{Z}_q^n \mid A^* x = y \bmod q\}. \tag{15.19}$$

Для неї задача криптоаналітика полягає у тому, щоб визначити, чи буде знайдений після редукції вектор на цій решітці належати до розподілу, з якого отримано ключ, або рівномірного розподілу. Згідно з [172], статистична відстань між цими двома розподілами складає $\varepsilon = 4\,exp\left(-2\pi^2\left(\frac{l\sigma}{q}\right)^2\right)$, де l – довжина найменшого вектора. Відповідно, потрібно перебрати $1/\varepsilon^2$ значень для вдалої атаки таких векторів. Нехай вирішувач працює за час T, тоді кількість спроб можна розрахувати, як $R = \max\left(1, \frac{1}{T*\varepsilon^2}\right)$. Оскільки довжина вектора l залежить

від фактора Ерміта, як $\delta_0^{d-1} q^{\frac{n}{d}}$, а фактор Ерміта може бути виражений через розмір блока, як:

$$\delta \approx ((\pi\beta)^{\frac{1}{\beta}} * \beta / 2\pi e)^{1/2(\beta-1)},$$

то задача знову зводиться до перебору значень параметра β .

Деталізовані вище дві атаки називаються Primal Attack і Dual Attack відповідно і складають основу для оцінки складності криптосистем на основі LWE (та його різновидів) [172,173]. Всі інші покращення, як правило, є евристиками.

Атаки засобом підробки ЕП

Іншим шляхом здійснення зловмисником атаки є підробка ЕП. У цьому разі атака здійснюється шляхом вирішення задачі SIS. Візьмемо до уваги, що дуальна атака фактично є вирішенням SIS [172,173].

Проте автори ЕП DILITHIUM запропонували інший підхід до вирішення цієї задачі. В SIS потрібно знайти вектор, всі елементи якого менші за певну величину. Автори DILITHIUM пропонують шукати малі вектори на решітці до тих пір, доки всі елементи такого малого вектора не будуть менші за задану величину в задачі SIS. Нехай T – час роботи вирішувача, p -ймовірність знаходження такого вектора. Тоді ймовірність атаки можна оцінити, як $\frac{1}{T*p}$. Відповідно, повторивши обчислення декілька разів, ця ймовірність зросте до потрібного рівня. Недоліком такого підходу є використання великої кількості евристичних міркувань. Проте складність оцінки криптостійкості виявляється меншою, ніж при використанні дуальної атаки. Далі, після редукції решітки в векторах умовно можна виділити, як показано вище в підрозділі 3, три зони. У першій зоні довжини норм Грамма – Шмідта будуть розподілені рівномірно за модулем q, у другій - довжини норм Грамма – Шмідта матимуть нормальний розподіл, а у третій зоні усі довжини норм Грамма – Шмідта дорівнюють 0. Якщо рандомізувати базис решітки перед редукцією, то перша зона зникає і редукція відбувається швидше.

Таким чином, потрібно, щоб перші j довжини норм Грамма – Шмідта (які розподілені за нормальним розподілом) були менші за задане значення ζ. Ймовірність цього складає:

$$p = \Phi\left(\frac{\zeta}{\sigma\sqrt{2}}\right)^{j+1}. \qquad (15.20)$$

Головна проблема полягає у знаходженні необхідного значення j. Варто окремо згадати про алгебраїчні атаки. При достатньо малих σ та достатньо великих q існують алгебраїчні атаки на LWE [193]. Для значень δ та q, що використовуються в Dilithium, ці атаки не можуть бути здійснені.

Обчислення основних параметрів для удосконаленого ЕП DILITHIUM

Основні параметри для реалізації LWE та SIS атак представлені в табл. 15.10.

Таблиця 15.10.

Основні параметри ЕП DILITHIUM

Параметри	Значення
(N, q)	Степінь та модуль коефіцієнтів поліномів.
(k, l)	Кількість поліномів в матриці.
η	Значення коефіцієнтів особистих ключів s (знаходяться в межах $[-\eta, \eta]$).
(γ_1,γ_2)	Обмеження на максимальні значення коефіцієнтів під час вироблення підпису. Сильно впливають на стійкість до атак на підробку підпису.
β	Також впливає на максимальне значення коефіцієнтів, проте впливає не стільки на стійкість, скільки на розмір підпису. Фактично дозволяє знаходити баланс між стійкістю та техніко-економічними показниками.
d	Впливає на стійкість до атак до підробки підпису.
h	Визначає об'єм простору, з якого обирається challenge.
w	Впливає тільки на розмір підпису. Знаходиться з практичних міркувань.

Тут враховано, що чим менше *N*, тим швидше працюватиме схема. Автори ЕП DLILITHIUM використовують *N* =256. Збільшуючи кількість поліномів через параметри (*k*, *l*), можливо досягти будь-якого рівня стійкості до MLWE, проте, для забезпечення стійкості до інших атак при використанні *N* =256 вже для рівня стійкості 384 біт не існує стійких параметрів, тому для $\lambda \in \{384,512\}$ потрібно збільшити *N* до 512.

Параметр q сильно впливає на стійкість. Він повинен бути простим числом. Для створення ефективних реалізацій з використанням NTT повинно виконуватися співвідношення $q \equiv 1 mod 2N$. Автори Dilithium використовували q=8389417. При збільшенні q буде знижуватися криптостійкість; немає сенсу збільшувати його значення, оскільки умова $q \equiv 1 mod 2N$ виконується для 256, 384, 512 біт безпеки.

Для обчислення параметрів (γ_1, γ_2) автори використовували формули:

$$\gamma_1 = (q-1)/16, \ \ \gamma_2 = \gamma_1/2 . \tag{15.21}$$

При такому виборі стійкість до всіх можливих атак буде приблизно однаковою. Такий вибір має сенс і при генерації параметрів для 256, 384, 512 біт безпеки.

Параметр h визначає кількість ненульових елементів в поліномі *C*. Для стійкості γ кількість можливих поліномів повинна бути не меншою за 2^{λ}. Тож має виконуватися нерівність:

$$2^h \binom{n}{h} \geq 2^{\lambda}. \tag{15.22}$$

Атаки на LWE можуть бути задані трьома параметрами: n, q, σ, де n – розмірність решітки (у випадку Dilithium $n=N*l+1+m$, $m \in \{0, N*k)$, q – найбільше значення елементів векторів. Співпадає із значенням q в загальносистемних параметрах. σ – середньоквадратичне відхилення для розподілу, з якого отримується секретний ключ. Для ЕП DILITHIUM це значення обчислюється як:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{\eta} i^2}{2\eta+1}}\,. \tag{15.23}$$

Атаки на SIS параметризовано трьома параметрами: x, y, ς. Для ЕП DILITHIUM $x = N*k, y = N*l$. Параметр ς є обмеженням на максимальне значення елементів векторів. Для ЕП DILITHIUM є дві атаки, що зводяться до SIS, тому відповідно два різних значення ζ:

$$\begin{aligned} \zeta_1 &= \max(\gamma_1 - \beta, 2\gamma_2 + 1 + 2^{d-1}h)\,, \\ \zeta_2 &= \max(2(\gamma_2 - \beta), 4\gamma_2 + 2)\,. \end{aligned} \tag{15.24}$$

Якщо встановити обмеження $2^{d-1}h + 1 \le 2\gamma_2$ і врахувати, що $\gamma_2 = \gamma_1/2$, то маємо:

$$\begin{aligned} \zeta_1 &\le 4\gamma_2\,, \\ \zeta_2 &\le 4\gamma_2 + 2. \end{aligned} \tag{15.25}$$

Параметр β визначає максимальне значення коефіцієнтів поліномів $c* s_i$. Враховуючи обмеження (15.25), максимальне значення, яке може бути отримане, це $\beta = \eta h$. При такому виборі забезпечуватиметься максимальний захист від атак SIS, проте, параметр β при всіх можливих значеннях майже не впливає на безпеку, тож його вплив можливо не враховувати. Проте, параметр β сильно впливає на ймовірність повтору циклу. Ця ймовірність становить

$$\approx \exp\left(-N\beta\left(\frac{l}{\gamma_1} + \frac{k}{\gamma_2}\right)\right)\,. \tag{15.26}$$

Тож саме ця ймовірність є критерієм вибору параметра β.

Параметр w не впливає на криптостійкість, проте дозволяє зменшити розмір підпису ціною повтору циклу. Практичні експерименти показали, що при значенні w=0.08nk отримуються гарні результати, проте можливі подальші оптимізації.

Враховуючи приведені критерії, алгоритм генерації загальносистемних параметрів виглядає наступним чином:

1. Визначити потрібний рівень безпеки $\lambda \in \{256,384,512\}$;
2. Обрати значення N. Якщо $\lambda = 256$, то N=256, інакше N=512;
3. Обрати значення q. q=8389417;

4. Обчислити γ_1 та γ_2 за формулами $\gamma_1 = (q-1)/16$, $\gamma_2 = \gamma_1/2$;

5. Обчислити значення η. За замовчуванням встановити $\eta = 2$. На наступних кроках значення буде уточнене;

6. Встановити значення (k, l)=(2, 1);

7. Обчислити γ_1 - стійкість до Primal Attack (Підрозд. 3, алгоритм Primal Attack). Якщо стійкість менша за λ, то оновити параметри (k, l) = (k +1, l+1) та повернутися до кроку 7 або збільшити η та повернутися до кроку 7;

8. Обчислити λ_2- стійкість до Dual Attack (Підрозд. 3, алгоритм Dual Attack). Якщо стійкість менша за λ, то оновити параметри (k,l) = (k +1,l+1) та повернутися до кроку 7 або збільшити η та повернутися до кроку 7;

9. Обчислити λ_3 - стійкість до SIS з ς_1 (Підрозд. 3, алгоритм обчислення SIS attack). Якщо стійкість менша за λ, то оновити параметри (k, l) = (k +1, l+1) та повернутися до кроку 7 або збільшити k=k+1 та повернутися до кроку 7;

10. Обчислити λ_4- стійкість до SIS з ς_2 (Підрозд. 3, алгоритм обчислення SIS attack). Якщо стійкість менша за λ, то оновити параметри (k, l) = (k +1, l+1) та повернутися до кроку 7 або збільшити k=k+1 та повернутися до кроку 7;

11. Обчислити h як найбільше ціле, для якого виконується нерівність:

$$2^h \binom{n}{h} \geq 2^\lambda;$$

12. Обчислити d як найбільше ціле, для якого виконується нерівність:

$$2^{d-1}h + 1 \leq 2\gamma_2;$$

13. Встановити $\beta = \eta h$ та зменшувати β, щоб ймовірність повтору циклу була достатньо малою;

14. Обчислити w=0.08nk (цей крок не впливає на криптостійкість та його можливо оптимізувати).

В таблиці 15.11 наведено значення параметрів для удосконаленого ЕП DILITHIUM.

Таблиця 15.11.

Значення параметрів для 256, 384, 512 біт стійкості

Набір	N, q	γ_1	γ_2	k, l	η	β	d	h	ω
256	(256,8380417)	523776	261888	(9,8)	2	144	14	60	184
384	(512,8380417)	523776	261888	(7,5)	5	100	13	77	286
512	(512,8380417)	523776	261888	(9,8)	2	74	13	118	368

Ймовірність повтору циклу при цьому складає для 256 біт – 0,15442678312246608, для 384 біт – 0,15609624568669475 і для 512 біт – 0,15247678668181552.

Результати оцінки криптостійкості для удосконаленого ЕП DILITHIUM (в бітах) з використанням параметрів з табл. 15.11 наведено в таблиці 15.12.

Таблиця 15.12.

Оцінки криптостійкості для удосконаленого ЕП DILITHIUM

Набір	Primal Attack (класичний)	Primal Attack (квантовий)	Dual Attack (класичний)	Dual Attack (квантовий)	SIS (класичний)	SIS (квантовий)
256	298	270	296	269	293	266
384	440	399	438	397	503	456
512	582	527	579	525	590	535

Висновки

1. Наразі спостерігається стійкий прогрес у створенні квантових комп'ютерів. Практично завершується створення математичних основ та програмного забезпечення для таких квантових комп’ютерів. Розробляються квантові комп'ютери, що призначаються для криптоаналізу існуючих стандартизованих криптосистем з відкритим ключем – електронних підписів, асиметричних шифрів та криптографічних протоколів різного призначення. Національний інститут стандартів і технологій (NIST) США закінчив та прийняв рішення у вигляді проєкту стандарту NIST 8309 щодо 2-го раунду конкурсу на перспективні стандартні алгоритми електронного (цифрового) підпису (ЕП). Його підсумком є визначення фіналістів другого етапу конкурсу у вигляді проєктів CRYSTALS-DILITHIUM, FALCON та Rainbow [135,136]. Також визначені три альтернативних кандидатів, які потребують більш детальних досліджень уже на четвертому етапі конкурсу – GeMSS, Picnic та SPHINCS+.

2. Одним із основних проблемних питань, що потрібно вирішувати, є обґрунтування необхідності та розробки удосконаленої версії ЕП Dilithium, що може забезпечувати в постквантовий період 128, 256, 384 і 512 біт безпеки проти класичного та 64, 128, 192 та 256 біт проти квантового криптоаналізу від найбільш загрозливих атак [135, 146 – 148].

3. Метод (схема) ЕП Dilithium ґрунтується на підході, що отримав назву "Fiat-Shamir з перериваннями”. Він в певній мірі схожий на схему, що запропонована з послідовним удосконаленням в [146, 147]. В перспективі необхідно вирішувати проблему обчислення (генерування) системних параметрів для удосконаленого ЕП Dilithium.

4. В якості часткових складових комплексної моделі безпеки щодо асиметричних криптоперетворень типу ЕП прийнято та визначено такі приватні моделі:

- модель порушника щодо асиметричних криптоперетворень типу ЕП;
- модель загроз щодо асиметричних криптоперетворень типу ЕП;
- модель безпеки щодо асиметричних криптоперетворень типу ЕП.

5. Отримані результати дозволяють зробити висновок, що атаки на LWE можливо розділити на два великі класи – атаки, що ґрунтуються на переборі, та атаки, що ґрунтуються на редукції решіток. До першого класу належать атаки повного перебору, зустріч посередині та Arora-Ge. Підхід, що використаний в

атаці Arora-Ge, є цікавим та перспективним, але він поки що поступається атакам на решітках.

6. Попередній аналіз дозволяє зробити висновок, що сучасні варіанти механізмів LWE ґрунтуються на поліноміальних кільцях, зокрема, на $\boldsymbol{R_q = Z_q[X]/(x^{2^n} + 1)}$. Властивості поліномів кільця дозволяють довести ряд теоретичних тверджень щодо стійкості криптосистеми і розробляти ефективні програмні реалізації. Проте, такі кільця мають нетривіальні підполя, що теоретично може використовуватися для криптоаналізу, проте, на практиці атак, що застосовують ці додаткові структури, не було знайдено або ці атаки знаходяться в незавершеному вигляді досліджень.

7. Проблеми криптоаналізу RLWE та MLWE по суті зводяться до LWE проблеми. Таке зведення можливо для $\boldsymbol{R_q = Z_q[X]/(x^{2^n} + 1)}$, оскільки доведено, що RLWE є не менш стійким, ніж LWE. Проте, при такому підході внутрішня структура кільця ігнорується.

8. Атаки на решітках полягають у зведенні проблеми LWE до достатньо вивчених теоретичних проблем в теорії решіток. Існують три основні підходи: зведення LWE до BDD, зведення LWE до SIS, зведення LWE до SVP. Кожен з цих підходів в кінцевому випадку зводиться до задачі пошуку достатньо малого вектора на решітці, для чого використовується алгоритм BKZ та його варіації.

9. Точні оцінки для BKZ та його варіацій невідомі. При практичній оцінці використовується ряд евристичних підходів та екстраполяція результатів, що отримані на решітках меншої розмірності. Це становить основну проблему при оцінці криптостійкості систем, подібних Dilithium, оскільки немає гарантії, що не з’явиться кращий спосіб редукції решіток або оцінка виявиться недопустимо неточною.

10. Для генерації системних параметрів удосконаленого ЕП DILITHIUM використаємо результати аналізу відомих атак на криптосистему, що наведені в підрозділі 3, та встановимо умови, за яких забезпечується захист від них. Найочевиднішим підходом атаки криптосистеми є відновлення особистого ключа на основі використання значення відкритого ключа.

11. Деталізовані дві атаки називаються Primal Attack і Dual Attack відповідно і складають основу для оцінки складності криптосистем на основі LWE (та його різновидів) [172,173]. Всі інші покращення, як правило, є евристиками.

12. Атака підробки ЕП здійснюється шляхом вирішення задачі SIS. Візьмемо до уваги, що дуальна атака фактично є вирішенням SIS. Проте автори ЕП DILITHIUM запропонували інший підхід до вирішення цієї задачі. В SIS потрібно знайти вектор, всі елементи якого менші за певну величину. Автори Dilithium пропонують шукати малі вектори на решітці до тих пір, доки всі елементи такого малого вектора не будуть менші за задану величину в задачі SIS.

13. Враховано, що чим менше N, тим швидше працюватиме схема. Автори ЕП DLILITHIUM використовують N=256. Збільшуючи кількість поліномів через параметри (k,l), можливо досягти будь-якого рівня стійкості до MLWE, проте, для забезпечення стійкості до інших атак при використанні N=256 вже для рівня

стійкості 384 біт не існує стійких параметрів, тому для $\lambda \in \{384,512\}$ потрібно збільшити *N* до 512.

14. В табл. 15.11, 15.12 наведені значення параметрів та оцінки криптостійкості для удосконаленого ЕП DILITHIUM з рівнями безпеки 256, 384, 512 біт.

15.2.1 Алгоритми реалізації перспективного ЦП CRYSTALS-DILITHIUM

В розділі 15.2 наведено узагальнений механізм ЦП Crystals-Dilithium. З використанням такого опису можна тільки у загальному виді подати сутність механізму. Наразі для розроблення, дослідження та удосконалення чи взагалі побудування на основі загальної моделі ЦП Crystals-Dilithium, що забезпечили б 5 -7 рівнів безпеки, необхідний опис ЦП Crystals-Dilithium типу «дорожна карта». Вказане завдання і наводиться нижче в цьому підрозділі.

1. Методи побудування параметрів механізму Crystals-Dilithium

1.1 Постійні параметри Crystals-Dilithium

В цьому пункті наводяться опис та пояснення стосовно постійних параметрів Crystals-Dilithium:

N=256 – степінь поліному;

q=2^{23}–2^{13}+1=8380417– більший модуль перетворення коефіцієнтів поліномів;

QBITS=23– значення $log_2 q$;

D=14 – параметр, який використовується при виділенні старшої і молодшої частин 32 - бітного слова;

γ_1=((q–1)/16) = 523776;

γ_2=(GAMMA1/2) = 261888;

ALPHA=(2*γ_2) – 523776 (19 бітів) - параметри, які використовуються при обчисленні норми ROOT_OF_UNITY 1753U.

1.2 Параметри, які можуть змінюватися у залежності від вимог, що висуваються до стійкості та складності криптографічних перетворень при виконанні ЕП

Параметри, які можуть змінюватися, задано в таблиці 15.13.

В таблиці 15.13 параметри *K* та *L* визначають розмір матриці *A*, η – основа алфавіту ключових даних, $\boldsymbol{\beta}$, $\boldsymbol{\omega}$ та $\boldsymbol{\zeta}$ – додаткові параметри забезпечення стійкості та оптимізації.

Таблиця 15.13.

Параметри, що можуть змінюватись

Модель	K	L	η	ζ	β	ω
0	3	2	7	4	375	64
1	4	3	6	4	325	80
2	5	4	5	4	275	96
3	6	5	3	3	175	120

2. Методи та алгоритми генерації ключів

2.1 Узагальнений алгоритм генерації ключів

В процесі генерування ключів послідовно виконуються такі кроки (*алгоритм 1*):

1. Генерація випадкових послідовностей у вигляді відкритого початкового ключа ρ (32 байти) та початкових секретних ключів (ρ_1 та Key) (32* 2=64 байти), що визначають початкову ентропію системи генерування відкритих та особистих (секретних) ключів ЦП Dilithium.

2. Генерація відкритої матриці поліномів *A* розміром [K][L][256] з використанням відкритого ключа ρ (32 байти) для значень *K* та *L*, що наведені в таблиці 15.13.

3. Генерація особистих (секретних) ключів у вигляді масиву поліномів S1[L][256] та S2[K][256] з використанням секретного початкового ключа ρ1 (32 байти).

4. Обчислення масиву *t* із *K* поліномів, використовуючи матрицю поліномів *A* та секретні ключі S_1 і S_2 згідно формули $t=A* S_1+S_2$.

5. Представлення кожного коефіцієнта поліному масиву поліномів (векторів) *t* у вигляді старших (high) та молодших (low) бітів **t[i][j]={t[i][j]$_{high}$, t[i][j]$_{low}$}** згідно формули:

$$t[i][j]=t[i][j]_{high}*2^{D}+t[i][j]_{low} \ (D=14). \quad (15.27)$$

6. Формування відкритого ключа в складі відкритих даних.

$$\{\rho, t_{high}\}. \quad (15.28)$$

7. Перетворення відкритого ключа (15.28) в масив октетів та обчислення геш – значення від даних ключа (15.28)

$$tr = H(\rho I I thigh) \quad (15.29)$$

причому в якості функції гешування в оригіналі Dilithium використовується геш - функція shake256 [134].

8. Формування особистого ключа у складі відповідних відкритих та особистих ключових даних:

$$\{\rho_1, key, tr, S1, S2, t_{low}\}. \quad (15.30)$$

Розглянемо (15.28 - 15.30) більш детально.

I. Генерація матриці поліномів A в Dilithium

При генерації (обчисленні) матриці поліномів A використовується відкритий ключ ρ (32 байти). При генерації матриці A (коефіцієнтів поліномів матриці A) попередньо необхідно сформувати з використанням функції гешування чи симетричного шифру псевдовипадкову послідовність з довжиною не менше K*L*256*32 бітів. Але, оскільки кожний коефіцієнт визначається значенням параметру QBITS=23 (значення $\log_2 q$) і займає 23 біта, то число необхідних бітів можна скоротити до K*L*256*23 бітів.

Для формування коефіцієнтів поліномів в проєкті стандарту ЦП Dilithium пропонується використовувати функцію гешування Keccak [134] чи алгоритм симетричного шифрування AES [134]. У випадку використання функції гешування послідовно гешуються спочатку відкритий ключ ρ+0, а потім для вироблення необхідного числа бітів коефіцієнтів матриці A значення ρ доповнюється кожен раз, як ρ+1, ρ+2, …, ρ+32, ρ+33, … тощо для необхідного числа бітів. Таким чином, наприклад, якщо K = 5, а L = 4, то необхідно згенерувати, як мінімум, всього 5* 4* 256*23= 117760 бітів. У загальному випадку для побудування матриці A необхідно згенерувати 23*32*K*L байтів. Процес завершується після отримання усіх елементів матриці A. Матриця A використовується в операції множення. Для використання швидкого NTT алгоритму множення поліномів попередньо потрібно перетворити елементи даних, для яких виконується множення, в NTT- форму [135]. Але, оскільки матриця A випадкова, то для скорочення обчислень можна вважати, що вона вже і є матрицею в NTT формі.

II. Генерація секретних ключів S_1 та S_2 ЦП Dilithium

Секретні ключі S1[L][256] та S2[K][256] генеруються у вигляді множини відповідно K та L поліномів з використанням ρ_1. Значення коефіцієнтів поліномів, які утворюють вектори S_1 та S_2, не можуть перевищувати значення ± η. З урахуванням усіх параметрів значення η не перевищує 7, тобто, потребує 4 або 3 байти (тобто, η=3). Така кількість байтів, які потрібно згенерувати, враховує, що коефіцієнти поліному можуть бути як позитивні, так і негативні. Для уніфікації обчислень передбачено, незалежно від кількості потрібних бітів, що завжди відводиться 4 біта. Вказану кількість бітів можна зменшити, але це буде додатковим навантаженням на ту частину програми, яка формує псевдовипадкову послідовність для генерації відповідних коефіцієнтів. В якості початкового випадкового елементу при генерації секретних векторів S_1 та S_2 використовується секретний початковий ключ ρ_1 (32 байти). При необхідності ентропія ρ_1 може бути змінена, як зменшена, так і збільшена. Безпосередньо генерування необхідного числа псевдовипадкових бітів – коефіцієнтів векторів поліномів S_1 та S_2 генерується аналогічно, як при обчисленні матриці A. Наприклад, у випадку використання функції гешування послідовно гешуються спочатку відкритий ключ ρ_1+0, а потім, для вироблення необхідного числа бітів коефіцієнтів ρ_1, доповнюється кожен раз у вигляді ρ_1+1, ρ_1+2, …, ρ_1+32, ρ_1+33, … для необхідного числа бітів.

Наприклад, якщо L = 4, то необхідно згенерувати для S1[L][256], як мінімум, всього 4* 256*4= 4096 бітів, а для S2[K][256], як мінімум 5* 256*4= 8192 бітів. Далі, якщо значення η не перевищує 2 (-1, 0, +1), то число бітів для формування поліномів секретних ключів S_1 та S_2 зменшується і залежить від значень K та L.

Відмітимо, що після генерації усіх поліномів для векторів S_1 та S_2 виконується їх перетворення в NTT форму.

III. Формування старшої та молодшої частин коефіцієнтів поліному $t_{high\ та}$ t_{low}

При заданному значені D t[i][j] коефіцієнти (15.27) записуються в формі $t=t_{high}*2^D+t_{low}$. При чому, t_{low} – це значення t% 2^D відносно середини, тобто, відхилення від відповідного значення D/2, яке може бути як позитивним, так і негативним.

IV. Склад секретного ключа ЦП Dilithium

У загальному склад секретного ключа визначено в (15.30). Але безпосередньо секретними компонентами є {ρ_1, key, S1, S2, t_{low} }. Значення ρ є відкритим ключем і використовується для генерування матриці *A*. Значення tr (48 байтів) є геш відкритого ключа – байтового рядка для відкритого ключа (використовується shake256) для формування геш заданої довжини. Значення ρ_1 використовується як секретне при генеруванні секретних ключів S_1 та S_2. Значення початкового ключа key (32 байти) використовується при генеруванні *s* секретного ключа маскування *Y*. Секретний ключ S_1 є вектор довжиною *L* поліномів. Секретний ключ S_2 є вектор довжиною *K* поліномів. Кожний коефіцієнт поліномів S_1 та S_2 ключів обмежується значеннями [–η, +η]. Якщо критичним параметром є час обчислення цифрового підпису, то замість S_1 можна записати NTT перетворення для S_1. Це збільшить довжину особистого ключа, тому що для запису одного елемента потрібно не 3, а 23 біта. S_2– вектор довжиною *K* поліномів. Кожний компонент – поліном з елементами, значення яких обмежується [–η, +η]. Секретна складова ключа t_{low} обчислюється згідно формули $t = t_{high}*2^D+t_{low}$ (D=14) , причому t_{low} – це значення t % 2^D відносно середини, тобто, для відхилення від D/2, яке може бути як позитивним, так і негативним, цей елемент потребує D бітів.

V. Склад відкритого ключа ЦП Dilithium

Склад відкритого ключа ЦП Dilithium визначається згідно формули (15.28). Значення ρ є по суті початковою ентропією, що визначає матрицю поліномів *A*. В ЦП Dilithium це початкове значення має 32 байти, кожний елемент якого є поліном з коефіцієнтами, що обмежені значенням *q*. Відкрите значення thigh обчислюється згідно формули (15.27), тобто, у вигляді $t = t_{higt}*2^D+t_{low}$, причому D =14.

VI. Константи та особливості упаковки–розпаковки ключових даних

В якості постійних величин (констант) в механізмі ЦП Dilithium використовуються (повинні бути заданими) такі константи:

- SEEDBYTES 32 байти – довжини початкових ключів ρ, ρ_1, key;
- SHAKE128_RATE 168 байтів – довжина геш - значення;
- CRYPTO_PUBLICKEYBYTES 1760 байтів – довжина відкритого ключа;
- CRYPTO_SECRETKEYBYTES 3856 байтів – довжина секретного ключа;
- CRYPTO_BYTES 3366U байтів – розмір ЦП;
- γ_1= ((Q–1U)/16U)=523776 – постійне значення;

- CRHBYTES 48 байтів - довжина геш - значення tr.

Як правило, упаковка при записі, розпаковка при читанні ключа виконується для тих компонентів ключів, елементи яких займають нецілe число байтів. При упаковці–розпаковці в авторській реалізації не використовується загально прийнята Converting Between Bit Strings та Right-Padded Octet Strings (BS2ROSP та ROS2BSP). Замість цього використовується Left-Padded Octet Strings.

3. Деталізований алгоритм вироблення ЦП Dilithium

При виконанні ЦП в якості вхідних значень використовуються (повинні бути заданими) такі дані:

- m – повідомлення, рядок (октетів);
- mlen – довжина повідомлення (октетів);
- s_k– особистий ключ, рядок октетів довжиною CRYPTO_SECRET-KEYBYTES.

В результаті виконання ЦП виробляються та є вихідними:

- sm – цифровий підпис, рядок (октетів);
- clen – довжина цифрового підпису (октетів).

При виконанні ЦП повинні бути виконаними такі кроки (*алгоритм 2*):

1. Розпаковка особистого ключа та отримання наступних компонентів:

- ρ, key, tr, S1, S2, $t_{Low;}$
- запис значення key, S1, S2 у виділений буфер довжиною 2*SEEDBYTES + CRHBYTES;
- запис значення tr в кінець криптографічної частини ЦП.

Запис окремих полів особистого ключа наступний (рис. 15.3):

ρ 1(SEEDBYTES)	key (SEEDBYTES)	μ (CRHBYTES)

Рис. 15.3

Призначення даних μ наведено в п. 3 алгоритму ЦП.

2. Запис повідомлення *m* довжиною mlen згідно рис. 15.4.

3. Обчислення геш значеня μ=hash (tr || m), тобто, значень полів, позначених (1) та (2), об’єднаних за допомогою операції конкатенації. Загальна довжина поля для обчислення гешу має величину CRHBYTES + mlen. В алгоритмі ЦП Dilithium геш - значення об числюється за допомогою алгоритму shake256 [136], воно має дов жину CRHBYTES і записується в кінець ЦП.

Криптографічна частина (CRYPTO_BYTES)		Повідомлення m (mlen)
Решта цифрового підпису (буде визначена пізніше)	tr (1) (CRHBYTES)	(2)

Рис. 15.4

4. Генерування матриці A[K][L][N] згідно *алгоритму 1*.

5. Подання складових секретного ключа в NTT формі: s1=NTT (S1), s2=NTT (S2), t_{low}=NTT (t_{low})/

6. Обчислення сеансового ключа маскування ЦП Y[L][N]. В якості випадкового компонента для формування множини (вектора) поліномів використовується 32 - бітний key та геш-повідомлення μ (див. крок 3), яке записано після key в буфері (мал. 15.3). Згідно мал. 15.3, ці поля знаходяться поряд. Таким чином, у цілому, довжина псевдовипадкового компоненту буде 32 + 48 = 80 байтів (SEEDBYTES + CRHBYTES). Для обчислення компонентів вектора (множини поліномів) Y використовується технологія, як для секретних ключів S_1, S_2 (див. Крок 3). Довжина елемента вектора визначається параметром GAMMA1 (довжина 20 бітів).

7. Обчислення множини *K* поліномів у вигляді w=A*Y.

8. Представлення w у вигляді двох послідовностей - w_{low} та w_{high} бітів. Формування послідовності бітів *w1* у вигляді w1= w_{high} на основі множини поліномів.

9. Генерація поліному *c*, який містить 60 одиниць або мінус одиниць, решту 0, в якості випадкового компоненту генерації використовується геш-повідомлення μ та *w1*, яке записується в упакованому форматі.

10. Обчислення множини поліномів ЦП z=c*S1+Y (*L* поліномів).

11. Перевірка норм множини поліномів z, які не повинні перевищувати значення GAMMA1–β. Інакше перейти на Крок 6.

12. Обчислення множини поліномів wcS2=w–c*S2 (K поліномів).

13. Представлення wcS2 у вигляді двох послідовностей - wcS2={$wcS2_{low}$, $wcS2_{high}$}, (по аналогії, як для кроку 8).

14. Перевірка норм поліномів $wcS2_{low}$, вона не повинна перевищувати значення GAMMA2–β. Інакше перейти на Крок 6.

15. Перевірка виконання умов, що $wcS2_{high}$==w_{high}. Інакше перейти на Крок 6.

16. Обчислення множини поліномів ct_{low}=t_{low}*c (K поліномів).

17. Перевірка норм множини поліномів ct_{low}, вони не повинні перевищувати значення GAMMA2–β. Інакше перейти на Крок 6.

18. Обчислення множини K поліномів tmp=ct_{low} + wcS2.

19. Перетворення множини поліномів ct_{low} =–$ct_{low.}$

20. Обчислення K векторів (h) за значеннями tmp та ct_{low}+tmp. Для кожного елемента (полінома) *i* для коефіцієнта полінома *j* обчислюється старша частина tmp та ct_{low}+tmp і виконується їх порівняння. Елемент матриці h [i][j] встановлюється в 1, якщо tmp $[i][j]_{high}$ не співпадає з (ct_{low}+tmp) $[i][j]_{high}$ і 0, якщо співпадають. Одночасно підраховується кількість 1. Якщо кількість одиниць перевищує ω, то перехід на Крок 6.

21. Упаковка цифрового підпису та запис полів у вигляді ЦП: z[L][N], h[K][N], c[N].

4. Деталізований алгоритм вироблення ЦП Dilithium (*алгоритм 3*)

При перевірці ЦП в якості вхідних значень використовуються (повинні бути заданими) такі вхідні дані:

- sm – цифровий підпис;
- smlen – довжина цифрового підпису (байтів).

В результаті перевірки ЦП виробляються та є вихідними:

- m – повідомлення;
- mlen – довжина повідомлення (байтів);
- результат перевірки: 0 – ОК, 1 – Помилка.

При виконанні перевірки ЦП повинні бути виконаними такі кроки:

1. Перевірка значення smlen. Якщо smlen < CRYPTO_BYTES – повернути результат 1.
2. Обчислення довжини повідомлення:

mlen = smlen –CRYPTO_BYTES.

3. Розпакування відкритого ключа, визначення ρ та t1 (масив векторів).
4. Розпакування цифрового підпису та визначення z, h, c, m.
5. Перевірка норм векторів z, вони не повинні перевищувати значення GAMMA1–β. Інакше повернути 1.
6. Обчислення геш – значення відкритого ключа tr.
7. Обчислення геш – значення dsl tr та m - μ = hash (tr || m).
8. Генерація матриці A[K][L][N] згідно ρ (див. I).
9. Множення матриці A на масив поліномів z. Отримання масиву поліномів tmp1 (K поліномів).
10. Для масиву K поліномів обчислити t1<<=D.
11. tmp2 = t1 * c (tmp2 - масив поліномів довжиною K).
12. tmp1 = - tmp2 (tmp1, tmp2 - масив поліномів довжиною K).
13. По молодшій і старшій частині відновлення значенням tmp1, h, отримання масиву поліномів w1.
14. Визначення по відкритим даним поліному c_p та порівняння його з поліномом c, що отриманий в ЦП. Якщо значення c та c_p не співпадають, то повернути 1.
15. Копіювання отриманого повідомлення *m* з метою його подальшого використання.

Повернути 0.

5. Порядок передобчислення для застосування NTT методу

За умови застосування при множенні поліномів швидкого методу NTT множення необхідно виконати пункти 1 –5.

Розглянемо обчислення примітивного кореня для n, Q.

Для цього треба обчислити мінімальне число root ≠1, таке що $root^N \bmod Q = 1$ або Q – 1.

Для n = 256, Q = 8380417 отримаємо значення 1753.

I. Перетворення Монтгомері

Нехай r = 2^{32}. Тоді для обчислення числа *a* в формі Монтгомері необхідно знайти α = a*r mod Q = a* r', де r' = r mod Q = 2^{32} mod Q.

Для Q = 8380417, r = 2^{32} отримаємо значення **r'** = 4193792.

II. Коефіцієнти перетворення Монтгомері для прямого NTT перетворення

Для обчислення коефіцієнтів необхідно виконати обчислення $Root^i$ mod Q для i = 1, 2, ..., 255.

Для обчислення індексів цих коефіцієнтів треба інвертувати біти в їх порядковому номері. Для подальшого використання цих коефіцієнтів для множення Монтгомері треба обчислений результат помножити на $\boldsymbol{r'}$, тобто, фактично треба виконати наступний алгоритм:

```
for (i = 1; i < N-1; i++){
        res = res * root % Q;
        ind = bits_invers(i);
        Z[ind] = res * r' % Q;
 }
```

Для Q = 8380417, root = 1753, r'= 4193792 отримуємо значення коефіцієнтів:

0, 25847, 5771523, 7861508, 237124, 7602457, 7504169, 466468, 1826347, 53451, 8021166, 6288512, 3119733, 5495562, 3111497, 2680103, 2725464, 1024112, 7300517, 3585928, 7830929, 7260833, 2619752, 6271868, 6262231, 4520680, 6980856, 5102745, 1757237, 8360995, 4010497, 280005, 2706023, 95776, 3077325, 3530437, 6718724, 4788269, 5842901, 3915439, 4519302, 5336701, 3574422, 5512770, 3539968, 8079950, 2348700, 7841118, 6681150, 6736599, 3505694, 4558682, 3507263, 6239768, 6779997, 3699596, 811944, 531354, 954230, 3881043, 3900724, 5823537, 2071892, 5582638, 4450022, 6851714, 4702672, 5339162, 6927966, 3475950, 2176455, 6795196, 7122806, 1939314, 4296819, 7380215, 5190273, 5223087, 4747489, 126922, 3412210, 7396998, 2147896, 2715295, 5412772, 4686924, 7969390, 5903370, 7709315, 7151892, 8357436, 7072248, 7998430, 1349076, 1852771, 6949987, 5037034, 264944, 508951, 3097992, 44288, 7280319, 904516, 3958618, 4656075, 8371839, 1653064, 5130689, 2389356, 8169440, 759969, 7063561, 189548, 4827145, 3159746, 6529015, 5971092, 8202977, 1315589, 1341330, 1285669, 6795489, 7567685, 6940675, 5361315, 4499357, 4751448, 3839961, 2091667, 3407706, 2316500, 3817976, 5037939, 2244091, 5933984, 4817955, 266997, 2434439, 7144689, 3513181, 4860065, 4621053, 7183191, 5187039, 900702, 1859098, 909542, 819034, 495491, 6767243, 8337157, 7857917, 7725090, 5257975, 2031748, 3207046, 4823422, 7855319, 7611795, 4784579, 342297, 286988, 5942594, 4108315, 3437287, 5038140, 1735879, 203044, 2842341, 2691481, 5790267, 1265009, 4055324, 1247620, 2486353, 1595974, 4613401, 1250494, 2635921, 4832145, 5386378, 1869119, 1903435,

7329447, 7047359, 1237275, 5062207, 6950192, 7929317, 1312455, 3306115, 6417775, 7100756, 1917081, 5834105, 7005614, 1500165, 777191, 2235880, 3406031, 7838005, 5548557, 6709241, 6533464, 5796124, 4656147, 594136, 4603424, 6366809, 2432395, 2454455, 8215696, 1957272, 3369112, 185531, 7173032, 5196991, 162844, 1616392, 3014001, 810149, 1652634, 4686184, 6581310, 5341501, 3523897, 3866901, 269760, 2213111, 7404533, 1717735, 472078, 7953734, 1723600, 6577327, 1910376, 6712985, 7276084, 8119771, 4546524, 5441381, 6144432, 7959518, 6094090, 183443, 7403526, 1612842, 4834730, 7826001, 3919660, 8332111, 7018208, 3937738, 1400424, 7534263, 1976782

Значення коефіцієнтів для зворотного NTT перетворення

Для обчислення необхідно знайти $root^{-1}$ таке, що root * $root^{-1}$ =1 mod Q. Для Q = 8380417, root = 1753, $root^{-1}$ =731434.

Для обчислення коефіцієнтів необхідно виконати обчислення $root^{-i}$ mod Q для i = 1…255. Для обчислення індексів цих коефіцієнтів треба інвертувати біти в їх порядковому номері. Для подальшого використання цих коефіцієнтів для множення Монтгомері треба обчислений результат помножити на r'.

Алгоритм обчислення співпадає з алгоритмом для прямих коефіцієнтів, але замість root використовується $root^{-1}$.

Для Q = 8380417, $root^{-1}$ =.731434, r' = 4193792 отримуємо значення коефіцієнтів:

6403635, 846154, 6979993, 4442679, 1362209, 48306, 4460757, 554416, 3545687, 6767575, 976891, 8196974, 2286327, 420899, 2235985, 2939036, 3833893, 260646, 1104333, 1667432, 6470041, 1803090, 6656817, 426683, 7908339, 6662682, 975884, 6167306, 8110657, 4513516, 4856520, 3038916, 1799107, 3694233, 6727783, 7570268, 5366416, 6764025, 8217573, 3183426, 1207385, 8194886, 5011305, 6423145, 164721, 5925962, 5948022, 2013608, 3776993, 7786281, 3724270, 2584293, 1846953, 1671176, 2831860, 542412, 4974386, 6144537, 7603226, 6880252, 1374803, 2546312, 6463336, 1279661, 1962642, 5074302, 7067962, 451100, 1430225, 3318210, 7143142, 1333058, 1050970, 6476982, 6511298, 2994039, 3548272, 5744496, 7129923, 3767016, 6784443, 5894064, 7132797, 4325093, 7115408, 2590150, 5688936, 5538076, 8177373, 6644538, 3342277, 4943130, 4272102, 2437823, 8093429, 8038120, 3595838, 768622, 525098, 3556995, 5173371, 6348669, 3122442, 655327, 522500, 43260, 1613174, 7884926, 7561383, 7470875, 6521319, 7479715, 3193378, 1197226, 3759364, 3520352, 4867236, 1235728, 5945978, 8113420, 3562462, 2446433, 6136326, 3342478, 4562441, 6063917, 4972711, 6288750, 4540456, 3628969, 3881060, 3019102, 1439742, 812732, 1584928, 7094748, 7039087, 7064828, 177440, 2409325, 1851402, 5220671, 3553272, 8190869, 1316856, 7620448, 210977, 5991061, 3249728, 6727353, 8578, 3724342, 4421799, 7475901,

1100098, 8336129, 5282425, 7871466, 8115473, 3343383, 1430430,
6527646, 7031341, 381987, 1308169, 22981, 1228525, 671102,
2477047, 411027, 3693493, 2967645, 5665122, 6232521, 983419,
4968207, 8253495, 3632928, 3157330, 3190144, 1000202, 4083598,
6441103, 1257611, 1585221, 6203962, 4904467, 1452451, 3041255,
3677745, 1528703, 3930395, 2797779, 6308525, 2556880, 4479693,
4499374, 7426187, 7849063, 7568473, 4680821, 1600420, 2140649,
4873154, 3821735, 4874723, 1643818, 1699267, 539299, 6031717,
300467, 4840449, 2867647, 4805995, 3043716, 3861115, 4464978,
2537516, 3592148, 1661693, 4849980, 5303092, 8284641, 5674394,
8100412, 4369920, 19422, 6623180, 3277672, 1399561, 3859737,
2118186, 2108549, 5760665, 1119584, 549488, 4794489, 1079900,
7356305, 5654953, 5700314, 5268920, 2884855, 5260684, 2091905,
359251, 6026966, 6554070, 7913949, 876248, 777960, 8143293,
518909, 2608894, 8354570

Особливості використання останнього коефіцієнту зворотного NTT перетворення

Останній коефіцієнт NTT перетворення і тільки він використовується для обчислення коефіцієнтів поліному – результату $\boldsymbol{c}$ номерами. Зазвичай для обчислення значення коефіцієнту необхідно:

montgomery_reduce (montgomery_reduce (a * z), f_),

де f_ = (((uint64_t) $r' * r'$ % Q) * (Q - 1) % Q) * ((Q - 1) >> 8) % Q.

Тут використовується z загальне, тому можна два перетворення переробити в одне за рахунок передобчислення z:

```
z:= 8354570;
f:= 41978;
q:= 8380417;
zf:= z * f;
QINV:=4236238847;
t:= zf * QINV; // zf(-1/Q)
t:= t & ((1 << 32) - 1) = 2037512764;
t:= t * q;
t:= zf + t;
print "t + zf =", t:Hex;
t := t /  4294967296.
```

Висновки до розділу 15.2.1

1. Для розроблення, дослідження та удосконалення чи взагалі побудування на основі загальної моделі ЦП Crystals-Dilithium, що забезпечили б 5 -7 рівнів безпеки, необхідний опис ЦП Crystals-Dilithium типу «дорожна карта». Вказане завдання і наводиться в цьому розділі.

2. В пункті 2 наведені та дається опис стосовно постійних параметрів Crystals-Dilithium, що повинні бути заданими.

3. В таблиці 15.13 параметри *K* та *L* визначають розмір матриці *A*, η – основа алфавіту ключових даних, β, ω та ζ – додаткові параметри забезпечення стійкості та оптимізації.

4. Перед застосуванням ЦП Dilithium необхідно здійснити формування особистого ключа у складі відповідних відкритих та особистих ключових даних.

5. {ρ1, key, tr, S1, S2, tlow}

6. При генерації (обчисленні) матриці поліномів *A* використовується відкритий ключ ρ (32 байти). При генерації матриці *A* (коефіцієнтів поліномів матриці *A*) попередньо необхідно сформувати з використанням функції гешування чи симетричного шифру псевдовипадкову послідовність з довжиною не менше K*L*256*32 бітів.

7. Матриця *A* використовується в операції множення, причому для використання швидкого NTT алгоритму множення поліномів попередньо потрібно перетворити елементи даних, для яких виконується множення, в NTT форму [136]. Але, оскільки матриця *A* випадкова, то для скорочення обчислень можна вважати, що вона вже і є матрицею в NTT формі.

8. Секретні ключі S1[L][256] та S2[K][256] генеруються у вигляді множини відповідно *K* та *L* поліномів з використанням ρ1. Значення коефіцієнтів поліномів, які утворюють вектори S_1 та S_2, не можуть перевищувати значення ± η.

9. Безпосередньо секретними компонентами ЦП є {ρ1, key, S1, S2, tlow }. Значення ρ є відкритим ключем і використовується для генерування матриці *A*.

10. При виробленні ЦП необхідно виконати пункти 1 - 21 підрозділу 4. Якщо підпис успішний, то в якості підпису формуються такі дані ЦП: z[L][N], h[K][N], c[N].

11. При перевірці ЦП необхідно виконати пункти 1 - 15 підрозділу 4. Якщо перевірка ЦП успішна, то виконується копіювання отриманого повідомлення *m* та його видача в якості вихідного значення.

12. За умови застосування при множенні поліномів швидкого методу множення необхідно виконати підрозділи 1 – 5.

15.2.2 Вибір параметрів схеми цифрового підпису DILITHIUM

Вступ

На сьогодні питання знаходження стійкого алгоритму направленого шифрування для постквантового періоду залишається актуальним.

Це пов’язано в першу чергу з розвитком технологій, а саме, з побудовою квантового комп’ютера. Як відомо, декілька років тому NIST оголосив про конкурс стандартизації одного чи декількох квантово-стійких алгоритмів шифрування з відкритим ключем.

Розглянемо схему цифрового підпису DILITHIUM, безпека якого базується на складності пошуку коротких векторів на решітці, а саме, вибір необхідних параметрів для забезпечення необхідної стійкості алгоритму.

Постійні параметри та параметри, що можуть змінюватися.

1. Постійні параметри:

N – розмір поліному. Приймаємо $N = 256$.

Q – модуль, за яким виконуються усі обчислення. $Q = 8380417$.

D – параметр, який використовується при виділені старшої і молодшої частини $D = 14$:

$QBITS = \log_2 q = 23$, $\gamma_1 = (q-1)/16 = 523776$, $\gamma_2 = \gamma_1/2 = 261888$, $\alpha = (2 \cdot \gamma_2) = 523776$ – параметри, які використовуються при обчисленні норми. Вага $c = 60$.

Параметри, які можуть змінюватися, наведені в таблиці 15.14.

Дані параметри наведені в [134]. Але значення параметрів, що наведені в табл. 15.14, не забезпечують стійкості схеми цифрового підпису на рівні 256 бітів. Фактично нижні оцінки стійкості, на складності яких базується стійкість, наведено в таблицях 15.15 та 15.16 відповідно. Це задачі SIS та LWE. Для кожної з них зазначені параметри обчислюються за формулами:

Таблиця 15.14.

Параметри, що можуть змінюватися

	I	II	III	IV
k	3	4	5	6
l	2	3	4	5
η	7	6	5	3
ξ	4	4	4	3
β	375	325	275	175
ω	64	80	96	120

$$\text{Найвідоміший класичний біт } = 0{,}292b, \quad (15.31)$$

$$\text{Найвідоміший квантовий біт } = 0{,}265b, \quad (15.32)$$

де b є довжиною блоку (BKZ розмір блоку b для зламування SIS або LWE) [134]. В ролі кінцевої оцінки стійкості використовується найменше з двох значень (15.31), обчислених для $\boldsymbol{b}$, що є довжиною блоку для задачі SIS та задачі LWE відповідно.

Таблиця 15.15.

Нижні оцінки стійкості для задачі SIS

	I	II	III	IV
BKZ розмір блоку b для зламування SIS	235	355	475	605
Найвідоміший класичний біт	68	103	138	176
Найвідоміший квантовий біт	62	94	125	160

Таблиця 15.16.

Нижні оцінки стійкості для задачі LWE

	I	II	III	IV
BKZ розмір блоку b для зламування LWE	200	340	485	595
Найвідоміший класичний біт	58	100	141	174
Найвідоміший квантовий біт	53	91	128	158

Тоді можна порахувати кінцеву нижню оцінку стійкості схеми цифрового підпису. За вимогами NIST рівень безпеки повинен бути не менший за III. Тому для останнього стовпця таблиць 15.15 та 15.16 відповідно маємо: BKZ розмір блоку b для зламування SIS $= 605$. Отже, значення (15.31) та (15.32) дорівнюють відповідно $0{,}292b = 176$, $0{,}265b = 160$; BKZ розмір блоку b для зламування LWE $= 595$, тому значення (15.31) і (15.32) дорівнюють відповідно $0{,}292b = 174$ та $0{,}265b = 158$. Таким чином, кінцева нижня оцінка стійкості схеми цифрового підпису для випадку, що розглядається, є 174, тобто, схема забезпечує стійкість відносно класичних атак на рівні не менше, ніж 2^{174} (а відносно квантових атак на рівні не менше, ніж 2^{158}).

Довжина блоку b обчислюється за параметрами k, l, η з табл. 15.14 за допомогою окремих алгоритмів для кожної задачі SIS та LWE. Яких саме, потребує додаткового вивчення, але автори [134] використовують для цього методику, наведену в [135,136].

2. Загальні обмеження, які треба врахувати при виборі параметрів.

Відзначимо загальні обмеження, які треба врахувати при виборі параметрів для забезпечення стійкості на рівні λ бітів, де $\lambda \in \{256,\ 384,\ 512\}$.

1). Геш-функція CRH, що використовується у схемі, повинна бути стійкою до колізій. Отже, довжина її вектора значень повинна бути, як мінімум, 2λ бітів (зокрема, при $\lambda = 256$ функція CRH повинна приймати значення довжини 512, а не 384, як в оригінальній схемі).

2). Криптографічна функція H повинна бути стійкою відносно знаходження другого прообразу [137] і, отже, приймати, принаймні, 2^{λ} різних значень, що є поліномами з кільця R_q, які мають коефіцієнти 0, 1, -1 та містять точно h ненульових коефіцієнтів.

При $\lambda = 256$ в [134] рекомендується використосувати параметри $n = 256$, $h = 60$, тоді умова:

$$2^h \binom{n}{h} \geq 2^{\lambda}$$

виконується.

При $\boldsymbol{\lambda} \in \{384,\ 512\}$ та $n = 256$ забезпечити виконання умови (2) не можливо, якщо $h \leq \frac{n}{2} = 128$. Отже, треба збільшити n до 512; при цьому числа q, γ_1, γ_2 можна залишити такими самими, як в [134]:

$$q = 2^{23} - 2^{13} + 1,\ \gamma_1 = (q-1)/16\ ,\ \gamma_2 = \gamma_1/2. \qquad (15.33)$$

3). Довжини векторів ρ та K повинні бути не менше, ніж $\boldsymbol{\lambda}$.

Зауважимо, що при $n = 512,\ q = 2^{23} - 2^{13} + 1$ час формування підпису може виявитися занадто великим; в цьому випадку треба збільшити q (приблизно в 2 рази).

4). Знаходження довжини блоку b за параметрами k, l, η.

Це – найбільш складна частина алгоритму вибору параметрів. Довжина блоку залежить тільки від значень $k, l, \eta, n,\ q$ і розраховується окремо для кожної з двох задач (SIS та LWE) та двох атак (прямої та дуальної) на кожну з них, згідно з методикою, наведеною в [135,136] (таким чином, для обчислення b слід застосувати 4 різні алгоритми та взяти найменше з отриманих значень).

На сьогодні вдалося "відновити" лише один з них, проте його застосування до параметрів в [134] призводить до значень b, які є приблизно на 10 % більше наведених в таблиці 15.17.

Алгоритм 1(оцінювання довжини блоку b для задачі LWE; пряма атака).

Вхідні дані: натуральні числа. $k,\ l,\ \eta,\ n,\ q$

Алгоритм обчислень: для кожного $m = 0,\ 1,\ \ldots,\ nk$ виконати такі дії:

1) покласти $d = nl + m + 1$;

2) знайти найбільше натуральне $\tilde{b} = \tilde{b}(m) \le d$ таке, що

$$\eta\sqrt{\tilde{b}} \le \delta^{2\tilde{b}-d-1} q^{\frac{m}{d}},$$

де

$$\delta = \left((\pi\tilde{b})^{\frac{1}{\tilde{b}}} \frac{\tilde{b}}{2\pi e}\right)^{\frac{1}{2(\tilde{b}-1)}}.$$

Результатом виконання алгоритму є число

$$b = \min_{0 \le m \le nk} \{\tilde{b}(m)\} \qquad (15.34)$$

В табл. 15.17 наведені значення довжини блоку, отримані за допомогою алгоритму 1, та відповідні значення цього параметра з табл. 15.16.

Таблиця 15.17.

Значення довжини блоку

$(k,\ l)$	(3, 2)	(4, 3)	(5, 4)	(6, 5)
η	7	6	5	3
Розмір блоку BKZ b, щоб зламати LWE [134]	200	340	485	595
Формула (15.34)	220	373	525	639

Як видно з таблиці 15.17, формула (15.34) приводить до більших значень в порівнянні з обчисленнями в [134]. Поряд з тим, значення довжини блоку, отримані за цією формулою, базуються тільки на розгляді однієї з двох атак на задачу LWE.

Висновки

Таким чином, для забезпечення стійкості схеми цифрового підпису на рівні $\lambda \in \{256,\ 384,\ 512\}$ необхідно:

1) використовувати геш-функцію CRH, значеннями якої є двійкові вектори довжини 2λ;

2) використовувати двійкові вектори ρ та K, що мають довжину λ;

3) покласти $n = 256$, якщо $\lambda = 256$; $n = 512$, якщо $\lambda \in \{384,\ 512\}$;

4) вибрати просте число $q \equiv 1\, mod\, 2\, n$ та обчислити γ_1, γ_2 за формулами (15.33);

5) обчислити вагу c як найменше натуральне h, що задовольняє умові (3).

15.3 Несиметричні криптосистеми на алгебраїчних кодах для постквантового періоду

Вступ

Сучасні несиметричні криптосистеми (та їх математична основа) є такими двоключовими конструкціями, у яких завдання пошуку секретного ключа (приватного ключа) за відомим відкритим ключем пов’язано з рішенням відомих та дуже складних математичних задач (факторизації, дискретного логарифмування та ін.) [176-177]. Однак з появою квантових обчислень, заснованих на принципах квантової механіки, зокрема, на принципі суперпозиції та квантової переплутаності, швидкість вирішення деяких математичних задач значно зростає [179]. Існує ряд квантових алгоритмів, наприклад, алгоритми Дойча та Йожі, Саймона, Гровера, Шора та інші, виконання яких займає набагато менше часу, ніж виконання будь-якого класичного алгоритму [52,81, 180-184]. Алгоритм Шора дозволяє знайти за кінцевий час усі прості множники великих чисел або вирішити задачу дискретного логарифмування і, як наслідок, знайти секретний ключ відповідного асиметричного криптоалгоритму (наприклад, у криптосистемі RSA) [184]. Отже, розробка та теоретичне обґрунтування нових криптографічних алгоритмів, у тому числі алгоритмів несиметричного шифрування та електронного цифрового підпису (ЕЦП), у яких складність пошуку секретного параметра за відомим відкритим ключем залишається високою навіть з урахуванням можливого застосування квантових обчислень (тобто, постквантового періоду), є надзвичайно важливим науковим завданням [120,185,186].

Перспективним напрямком у розвитку постквантової криптографії (Post-Quantum Cryptography) є кодові криптосистеми (Code-Based Cryptography), побудова яких заснована на використанні алгебраїчних кодів, замаскованих під код загального положення (випадковий код, повний код) [186,198]. У

англомовній літературі подібні алгоритми отримали назву теоретико-кодових схем [191] або криптокодових перетворень [192]. Кодові криптосистеми дозволяють реалізувати відносно швидке (порівняно, наприклад, з криптосистемами RSA, ECC тощо) криптографічне перетворення даних, а також реалізувати додатковий контроль помилок, що виникають [192]. У роботах [186,198–200] показано, що криптокодові перетворення залишаються стійкими навіть у разі використання квантових обчислень.

На сьогоднішній день відомі різні криптографічні примітиви, засновані на використанні алгебраїчних кодів: алгоритми несиметричного [118,187,189, 190] та симетричного [188] шифрування, генератори псевдовипадкових послідовностей [193], протоколи доказу з нульовим розголошенням [196], схеми ЕЦП [198,199], ідентифікації [200,201] та ін. Це дозволяє на основі єдиного математичного та програмного забезпечення реалізувати широкий спектр ефективних механізмів криптографічного захисту інформації. І хоча відомі також і обчислювально ефективні атаки на окремі варіанти теоретико-кодових схем [190,202-205], базова конструкція [118], запропонована близько 40 років тому, залишається стійкою до всіх відомих методів криптоаналізу, що з історичної ретроспективи підтверджує надійність і перспективність криптокодових перетворень, особливо у контексті побудови ефективних постквантових алгоритмів криптографічного захисту [206].

Метою даного розділу є огляд відомих несиметричних криптосистем на алгебраїчних кодах, у тому числі, схем формування та перевірки ЕЦП (Code-Based Signatures), дослідження сучасного стану існуючих протиріч та перспектив практичного застосування на постквантовий період.

15.3.1 Алгебраїчні коди для несиметричних криптосистем

Розглянемо векторний простір V_n як безліч n-послідовностей з елементами з кінцевого поля $GF(q)$ з покомпонентним додаванням і множенням на скаляр. Лінійний (n,k,d) код V_k над $GF(q)$ є підпростір в V_n , тобто, непорожня безліч n-послідовностей (кодових слів) з елементами з $GF(q)$, де k –розмірність лінійного підпростору, d – мінімальна вага Хеммінга (число ненульових елементів) $w_h(c)$ довільного ненульового кодового слова c коду V_k: $d = \min\limits_{\forall c \in V_k, c \neq 0} w_h(c)$. З огляду на лінійність підпростору V_k , набір ваг різних ненульових кодових слів збігається з набором відстаней по Хеммінгу між різними кодовими словами, тобто, d називають також мінімальною кодовою відстанню по Хеммінгу коду V_k. Величину $R= \frac{k}{n}$ називають відносною швидкістю коду, а $\delta = \frac{d}{n}$ називають відносною мінімальною кодовою відстанню. Лінійний код V_k (як лінійний підпростір в V_n) задається набором базисних (лінійно незалежних) векторів

$$(g_{0,0}, g_{0,1}, \ldots, g_{0,n-1}),$$

$$(g_{1,0}, g_{1,1}, \ldots, g_{1,n-1}),$$

$$\ldots$$

$$(g_{k-1,0}, g_{k-1,1}, \ldots, g_{k-1,n-1}),$$

які, зазвичай, представляються в матричному вигляді через породжуючу матрицю

$$G = \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \cdots & \cdots & \cdots & \cdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

рангу $rank(G) = k$ та розмірності $k \times n$.

Довільне кодове слово $c = (c_0, c_1, \cdots, c_{n-1})$ коду V_k є лінійна комбінація рядків із матриці G. Кодування полягає у зіставленні кожного інформаційного слова $i = (i_0, i_1, \cdots, i_{k-1})$ із символами з $GF(q)$ деякого кодового слова $(c_0, c_1, \cdots, c_{n-1})$. Найбільш простий спосіб кодування задається виразом $c=iG$. Лінійний підпростір, що ототожнює код V_k , має ортогональне доповнення (позначимо його V_{n-k}). Базис підпростору V_{n-k} задається векторами

$$(h_{0,0}, h_{0,1}, \ldots, h_{0,n-1}),$$
$$(h_{1,0}, h_{1,1}, \ldots, h_{1,n-1}),$$
$$\ldots$$
$$(h_{n-k-1,0}, h_{n-k-1,1}, \ldots, h_{n-k-1,n-1})$$

і, зазвичай, подається в матричному вигляді через перевірочну матрицю:

$$H = \begin{pmatrix} h_{0,0} & h_{0,1} & \ldots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \ldots & h_{1,n-1} \\ \ldots & \ldots & \ldots & \ldots \\ h_{n-k-1,0} & h_{n-k-1,1} & \ldots & h_{n-k-1,n-1} \end{pmatrix}$$

рангу $rank\ (H) = n\text{-}k$ та розмірності $(n\text{-}k) \times n$.

Умова ортогональності векторів із V_k і V_{n-k} в матричному вигляді записується як $GH^T = 0$, де під нулем розуміється нульова матриця розмірності $k \times (n-k)$.

Основною метою надлишкового кодування інформації є контроль (виявлення та виправлення) помилок, що відбулися при передачі повідомлення каналом з шумами [207-209]. Для контролю помилок кодуючий пристрій вносить надмірність (перевірочну частину довжини $r = n - k$) в дані, що передаються. На приймальній стороні, аналізуючи властивості перевірочної частини та її відповідність переданим даним, декодер зменшує вплив помилок, що виникли під час передачі.

Позначимо вектор помилок, що впливає на кодове слово c, що передається, як n-послідовність з елементами з $GF(q)$. Спотворене кодове слово позначимо вектором $c^* = c + e = (c_0 + e_0, c_1 + e_1, \cdots, c_{n-1} + e_{n-1})$.

Синдромом в теорії кодування називають вектор s=($s_0, s_1, \cdots, s_{n-k-1}$) з елементами з $GF(q)$, який характеризує вплив вектора помилок на довільне кодове слово:

$$s = c * H^T = cH^T + eH^T = eH^T,$$

тобто, значення вектора s залежить лише від вектора помилок $e = (e_0, e_1, \cdots, e_{n-1})$ і не залежить від вибраного кодового слова $c = (c_0, c_1, \cdots, c_{n-1})$.

Слід зазначити, що при великих n і k задача пошуку вектора e по ненульовому синдрому s для випадково обраного в просторі V_n лінійного коду V_k є надзвичайно складною математичною задачею. У загальному випадку ця задача належить до класу NP-складних [206]. Проте, для алгебраїчних кодів зі специфічною структурою матриць G і H декодування (задача пошуку вектора помилок e і/або відновлення безпомилкового кодового слова c) є поліноміально вирішуваною задачею. До таких кодів відносять: коди Ріда-Соломона та їх узагальнення [207-209] і навіть різні обмеження зазначених конструкцій на довільне підполе [209].

Визначення 1 [207-209]. Нехай $X = (X_0, X_1, \ldots, X_{n-1})$ вектор над $GF(qm)$, причому всі X_i– різні елементи $GF(qm)$. Нехай також $B = (B_0, B_1, \ldots, B_{n-1})$– вектор над $GF(qm)$ з необов'язково різними Bi елементами $GF(qm)$. Тоді (n, k, d) узагальнений код Ріда-Соломона OPC_k (X, h) складається з усіх векторів виду

$$(B_0 \cdot \mathrm{F}(X_0), B_1 \cdot \mathrm{F}(X_1), \ldots, B_{n-1} \cdot \mathrm{F}(X_{n-1})),$$

де F(x) – будь-який многочлен з коефіцієнтами $GF(qm)$, степінь якого не перевищує k. Код ОРС є МДР кодом, його перевірна матриця OPC_k (X, h) дорівнює:

$$H = \begin{pmatrix} Y_0 & Y_1 & \ldots & Y_{n-1} \\ X_1 \cdot Y_0 & X_2 \cdot Y_1 & \ldots & X_{n-1} \cdot Y_{n-1} \\ X_1^2 \cdot Y_0 & X_2^2 \cdot Y_1 & \ldots & X_{n-1}^2 \cdot Y_{n-1} \\ \ldots & \ldots & \ldots & \ldots \\ X_1^{n-k-1} \cdot Y_0 & X_2^{n-k-1} \cdot Y_1 & \ldots & X_{n-1}^{n-k-1} \cdot Y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ X_1 & X_2 & \ldots & X_{n-1} \\ X_1^2 & X_2^2 & \ldots & X_{n-1}^2 \\ \ldots & \ldots & \ldots & \ldots \\ X_1^{n-k-1} & X_2^{n-k-1} & \ldots & X_{n-1}^{n-k-1} \end{pmatrix} \cdot \begin{pmatrix} Y_0 & 0 & \ldots & 0 \\ 0 & Y_1 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & Y_{n-1} \end{pmatrix},$$

де вектор Y = ($Y_0, Y_1, \ldots, Y_{n-1}$) такий, що $\forall Y_i \in$GF(qm), $Y_i \neq 0$ та дуальним до OPC_k (X, B) є OPC_{n-k} (X, Y). Через обмеження ОРС на підполі вводиться клас так званих (т.зв.) альтернантних кодів [207-209].

Визначення 2 [207-209]. Альтернантний (n, k, d) код $A(X, B)$ складається з усіх слів коду ОРСk(X, B) таких, що їх компоненти лежать у полі $GF(q)$. Інакше кажучи, $A(X, B)$ дорівнює обмеженню коду РСk (X, B) на підполі $GF(q)$, тобто, він складається з усіх векторів c над $GF(q)$, для яких виконується рівність $GH^T = 0$, де H – перевірна матриця OPC_k (X, B). Породжувальна матриця $A(X, B)$ може бути отримана заміною кожного елемента матриці H відповідним вектор - стовпцем довжини m над $GF(q)$.

Параметри коду *A(X, B)* пов'язані співвідношеннями: $n - mr \le k \le n - r$; $d \ge r + 1$, причому доведено [207-209], що серед великої кількості всіх можливих альтернантних кодів при фіксованому *n* і *k* знайдуться такі коди, параметри яких лежать вище за кодові межі (15.33) і (15.34). Одним із окремих випадків *A(X, B)* є коди Гоппи [211,212].

Визначення 3 [209]. Альтернантний *(n, k, d)* код Гоппи Г(L, G) над *GF(q)* складається з усіх векторів $c = (c_1, c_2, \ldots, c_n)$ таких, що

$$Rc(x) \equiv 0 \bmod G(x),$$

$$Rc(x) = \sum_{i-1}^{n} \frac{c_i}{x - a_i},\ cH^T = 0.$$

де G(x) – багаточлен з коефіцієнтами з *GF*(*qm*) (багаточлен Гоппи) $L = (a_1, a_2, \ldots, a_n)$ – підмножина елементів із *GF*(*qm*) таких, що $G(a_i) \neq 0\ \forall a_i \in L$. Багаточлен $x - a_i$ у кільці багаточленів за модулем G(x) має зворотний багаточлен:

$$(x - a_i)^{-1} = -\frac{G(x) - G(a_i)}{x - a_i}\, G^{-1}(a_i).$$

Отже, вектор $c = (c_1, c_2, \ldots, c_n)$ належить коду Гоппи Г(L, G), тільки якщо

$$\sum_{i=1}^{n} \frac{G(x) - G(a_i)}{x - a_i}\, G^{-1}(a_i) = 0.$$

Якщо $G(x) = \sum_{i=0}^{r} g_i x^i$, де $g_i \in$ GF(qm) і $g_r \neq 0$, то

$$\frac{G(x) - G(\alpha_i)}{x - \alpha_i} = g_r\left(x^{r-1} + x^{r-2}\alpha_i + \ldots\right.$$

$$\left. + \alpha_i^{r-1}\right) + g_{r-1}\left(x^{r-2}\alpha_i + \ldots + \alpha_i^{r-1}\right) + \ldots + g_2\left(x + \alpha_i\right) + g_1.$$

Прирівнюючи нулю всі коефіцієнти при x_r-1, x_r-2, …, 1, отримаємо, що умова cHT = 0 виконається, лише якщо перевірна матриця

$$H = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \ldots & \alpha_n^{r-1} \end{pmatrix} \begin{pmatrix} G^{-1}(\alpha_1) & 0 & \ldots & 0 \\ 0 & G^{-1}(\alpha_2) & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & G^{-1}(\alpha_n) \end{pmatrix} = \begin{pmatrix} G^{-1}(\alpha_1) & G^{-1}(\alpha_2) & \ldots & G^{-1}(\alpha_n) \\ \alpha_1 G^{-1}(\alpha_1) & \alpha_2 G^{-1}(\alpha_2) & \ldots & \alpha_n G^{-1}(\alpha_n) \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_1^{r-1} G^{-1}(\alpha_1) & \alpha_2^{r-1} G^{-1}(\alpha_2) & \ldots & \alpha_n^{r-1} G^{-1}(\alpha_n) \end{pmatrix}$$

також задає (*n, k, d*) код Гоппи Г(L, G) над *GF(q)*.

Останній вираз при Y = (Y_1, Y_2, …,Y_n), Y_1= *G*-1(a_1), Y_2= G-1(a_2), …, Y_n = *G*-1(a_n) еквівалентний матриці *H* для ОРС . Перевірочну матрицю Г(L, G) над *GF(q)* з елементами *GF(q)* можна отримати шляхом представлення кожного елемента з GF(qm) вектор-стовпцем довжини *m* символів з *GF(q)*. Справедлива наступна оцінка.

Теорема [209,211,212]. Параметри (*n, k, d*) коду Гоппи Г(L, G) пов'язані співвідношеннями: $n = |L|$, $k \ge n - mr$, $r = \deg G(x)$, $d \ge r + 1$.

Для сепарабельних (коли многочлен $G(x)$ не має кратних коренів в жодному розширенні поля) двійкових кодів Гоппи ці співвідношення набудуть вигляду:

$$n = 2m,\ k \geq n - mr,\ r = \deg G(x),\ d \geq 2r + 1. \qquad (15.35)$$

Таким чином, при відповідному виборі вектора-шаблону $Y = (Y_1, Y_2, \ldots, Y_n)$ вдається побудувати блокові коди з високими конструктивними характеристиками, причому кількість таких кодів визначається кількістю відповідних шаблонів (або багаточленів $G(x)$ у разі кодів Гоппи) [209,211,212]. Ця властивість розглянутих кодових конструкцій широко використовується при розв'язанні різних інженерних завдань як в області підвищення стійкості перешкод інформації, так і для криптографічного захисту. Зокрема, несиметричні криптосистеми стійкості, що доводиться (provable security), побудовані на кодах Гоппи, крім високої швидкості та можливості поєднувати контроль помилок із захистом від несанкціонованого ознайомлення [118,188-191], залишаються стійкими навіть на постквантовий період [197].

15.3.2 Несиметричне шифрування на основі кодів (теоретико-кодові схеми)

Першою і найбільш вивченою схемою несиметричного шифрування, заснованою на використанні блокових алгебраїчних кодів, є запропонована в 1978 році криптосистема Мак-Еліса (McEliece) [118]. Вона має незаперечні переваги: високу швидкість криптографічного перетворення, а також можливість поєднувати контроль помилок із захистом від несанкціонованого доступу [118,187-191]. Крім того, подібні (криптокодові) перетворення залишаються стійкими навіть у разі використання квантових обчислень [178].

Криптосистема Мак-Еліса. Відкритим ключем у схемі Мак-Еліса є матриця

$$G_x = X \cdot G \cdot P \cdot D\ , \qquad (15.36)$$

де – G породжуюча матриця алгебраїчного (n, k, d) коду над $GF(q)$ (в оригінальній статті [118] пропонувалося використовувати розглянутий вище двійковий код Гоппи), X - невироджена $k \times k$ матриця з елементами $GF(q)$ з P і D - перестановна і діагональна $n \times n$ матриці (для двійкових кодів використовується тільки матриця P). Матриці X, P і D є секретним ключем, який маскує використовуваний блоковий алгебраїчний код під випадковий код (код загального положення), тобто, відкритий ключ G_X представляється зловмиснику, як випадково сформована g породжуюча матриця деякого лінійного коду, для якого невідомий алгоритм швидкого декодування. Навпаки, уповноважений користувач, який знає секретний ключ (матриці X, P і D), може зняти дію маскуючих матриць і скористатися швидким алгоритмом декодування алгебраїчного коду з породжуючою матрицею G.

Криптограма являє собою вектор довжини n, який обчислюється за правилом:

$$c_X^* = I \cdot G_X + e, \qquad (15.37)$$

де вектор $c_x = IG_x$ є кодовим словом замаскованого коду, тобто, c_x належить (n, k, d) коду з породжуючою матрицею G_x, $(I\text{-}\ k)$ - розрядний інформаційний вектор над *GF(q)*, вектор e – секретний вектор помилок ваги $w_k(e) \le t = \left\lfloor \frac{d-1}{2} \right\rfloor$. Вектор e слід розглядати як одноразовий секретний сеансовий ключ, його вага визначає складність декодування спотвореного кодового слова (криптограми) c_X^*. Зловмиснику необхідно декодувати кодограму c_X^*, використовуючи відому йому породжуючу матрицю G_x. Однак, декодування випадкового коду (при відповідних параметрах (n, k, d) і $w_h(e)$) обчислювально недосяжно. Не знаючи матриці *X, P* і *D* , зловмисник не може відновити матрицю *G* і скористатися алгоритмом декодування поліноміальної складності. З цих міркувань величину $w_h(e)$ слід максимізувати. Наприклад, при $w_h(e) = t$ складність декодування буде максимальною, що забезпечить найвищий рівень стійкості кодової криптосистеми для заданих параметрів (n, k, d). Для уповноваженого користувача (який знає секретний ключ) декодування – поліноміально вирішувана задача. Дійсно, легітимний користувач, отримавши вектор c_X^*, будує вектор $\overline{c}^* == c_X^* \cdot D^{-1} \cdot P^{-1}$. Матриця $\Lambda = PD$ зберігає вагу та відстань за Хеммінгом, тобто, для будь-яких кодових слів c і c' виконуються рівності:

$$w_h(c) = w_h(c \cdot \Lambda),\ w_h(c, c') = w_h(c \cdot \Lambda, c' \cdot \Lambda),\quad \overline{c}^* = I' \cdot G + e'.$$

Це означає, що вектор $\overline{c}^*$ є спотвореним не більше, ніж у *t* розрядах кодовим словом алгебраїчного коду з породжуючою матрицею *G* і його можна декодувати швидким алгоритмом поліноміальної складності [190].

Уповноважений користувач, використовуючи алгоритм поліноміальної складності, декодує вектор $\overline{c}^* = I' \cdot G + e'$, тобто, знаходить I'. Потім він обчислює *k*-розрядний інформаційний вектор $I = I'^{X^{-1}}$.

Таким чином, у криптосистемі Мак-Еліса основним засобом маскування лінійного блокового *(n, k, d)* коду підлінійного випадкового коду (код загального положення) є матриці *X, P, D*. Додатковим секретним параметром, який можна використовувати у разі кодів Гоппи, є багаточлен Гоппи *G(x)*, або, у ширшому сенсі, вектор Y = (Y_0, Y_1, … …,Y_{n-1}) у разі альтернантних кодів. Зміна шаблону не знижує конструктивних кодових параметрів, тобто, з погляду криптографічного перетворення не призведе до зниження безпеки. Проте знання вектора-шаблона Y = (Y_0, Y_1, …,Y_{n-1}) (або многочлена *G(x)*) є необхідним для правильного декодування інформаційного повідомлення, тобто, для коректного розшифрування на приймальній стороні. На сьогоднішній день опубліковано велику кількість різних атак на крипто-кодові схеми захисту інформації [190,202-205], деякі з яких виявилися досить ефективними щодо окремих варіантів кодових криптосистем. Однак базова конструкція [118], запропонована близько 40 років тому, залишається стійкою до всіх відомих методів криптоаналізу, у тому числі, і у разі використання квантових обчислювальних систем. Найбільш природним напрямком у розвитку методів криптоаналізу кодової схеми Мак-Еліса є використання неалгебраїчних методів декодування. Дійсно, якщо існує обчислювально ефективний спосіб декодування кодового слова (15.37) тільки по

відомій матриці (15.36), тоді інформаційне повідомлення *I* може бути ефективно відновлено і без знання секретного ключа (матриць *X, P* і *D*).

Серед універсальних методів декодування лінійних блокових кодів, заданих довільною матрицею, що породжує, особливе місце займають перестановочні алгоритми [207-209]. Основна ідея такого декодування полягає у використанні різних наборів інформаційних множин. Представимо матрицю, що породжує (15.36), в канонічному вигляді. При цьому поодинокі вектори-стовпці можуть бути обрані довільно з відповідним формуванням одиничних підматриць і систематичним розміщенням *k* символів інформаційної множини. Решта (*n*-*k*) символів однозначно обчислюються за елементами інформаційної множини. Позиції цих (*n*-*k*) символів задають розміщення одиничних вектор-стовпців відповідної перевірочної матриці. Якщо вибрати розміщення (*n*-*k*) одиничних вектор-стовпців таким чином, щоб вони покрили всі *t* позицій ненульових елементів вектора помилок *e*, тоді кодове слово по *k* символам інформаційної множини не міститиме помилок, тобто, слово (15.37) можна декодувати навіть без знання спеціальної структури алгебри породжуючої (перевірочної) матриці використовуваного алгебраїчного коду.

Таким чином, при реалізації перестановного декодування конкретна комбінація помилок буде виправлена, тільки якщо вдасться знайти таку інформаційну множину, яка повністю містить цю комбінацію. Таку множину, що є покрівельною комбінацією помилок, і набір перевірочних множин, які покривають всі набори помилок цього типу, називають покриттям [207]. Задача декодера полягає в тому, щоб знайти перевірочну множину, яка покриває невідому комбінацію помилок. Розглянемо межі кількості покрівельних множин. Припустимо, що з допомогою *(n, k, d)* коду виправляються всі комбінації з *t* чи менше помилок. Розглянемо комбінацію лише з *t* кратних помилок, оскільки всі помилки меншої кратності будуть покриті. Загальна кількість комбінацій помилок у всіх *n* позиціях дорівнює C_n^t. Оскільки об'єм покрівельної множини дорівнює $n - k$, максимальна кількість комбінацій помилок, які можуть бути покриті цією множиною, дорівнює C_{n-k}^t. Найменша кількість множин, які можуть виправити всі комбінації з *t* помилок, обмежується виразом [207]:

$$N \geq \frac{C_n^t}{C_{n-k}^t} = \frac{n!(n-k-t)!}{(n-t)!(n-k)!}. \qquad (15.38)$$

На рис. 15.5 наведено залежності найменшої кількості покрівельних множин, які потрібні для виправлення всіх комбінацій з *t* помилок довільного лінійного блокового коду. Ці залежності не враховують обчислювальну складність формування слів-кандидатів, що обчислюються за обраною конфігурацією інформаційної множини (реальна стійкість буде ще вище). Оцінки *N* наведено у логарифмічному масштабі залежно від відносної швидкості кодування $R = k/n$ та розраховані для параметрів (15.35).

Як випливає з наведених на рис. 15.5 залежностей, найбільшу стійкість схема Мак-Еліса забезпечує при використанні кодів із відносною швидкістю $R \approx 2/3$, що узгоджується з висновками більшості досліджень [197]. У таблиці 15.18 наведено параметри деяких схем з кодами Гоппи та $R \approx 2/3$: оцінки

стійкості до атаки перестановочного декодування, оцінки обчислювальної складності кодування (зашифрування) та декодування (розшифрування), а також аналогічні оцінки для несиметричного шифрування RSA.

Для криптосистеми Мак-Еліса значення в таблиці 15.18 оцінювалися наступним чином. Розмір відкритого ключа оцінювався, як число двійкових елементів матриці C_X (kn біт). Розмір закритого ключа оцінювався, як число елементів матриці X (k^2 біт) плюс число елементів, необхідних для зберігання правила перестановки ($n \log_2 n$ біт). Складність шифрування оцінювалася як максимальна кількість операцій, які необхідно виконати для формування кодового слова за допомогою матричного обчислення виразу (15.37). Для двійкового (n, k, d) коду це відповідає $k \cdot n$ двійковим операціям. Складність розшифрування оцінювалася як число двійкових операцій, які потрібно виконати для декодування кодового слова з помилками. Складність алгебраїчного декодування двійкових кодів Гоппи оцінювалася як $t^2 \cdot m^3$ двійкових операцій [198-201].

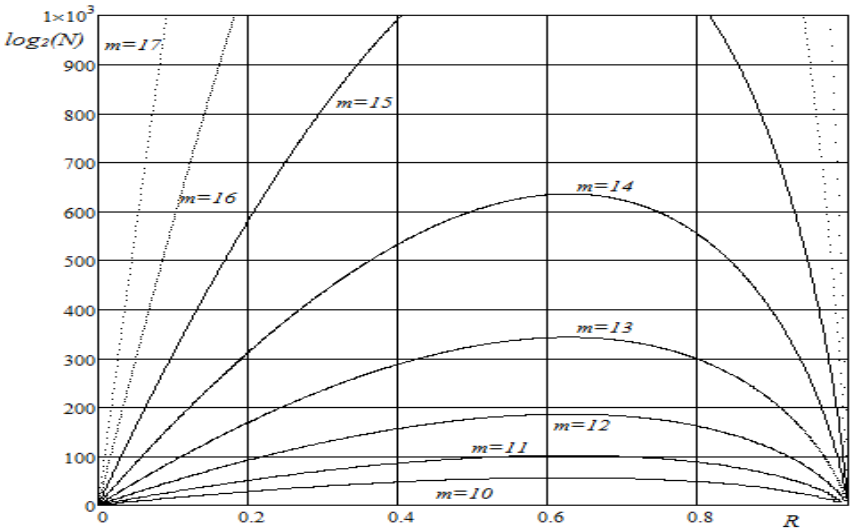


Рис. 15.5. Залежність найменшої кількості покрівельних множин

Оцінка стійкості криптосистеми Мак-Еліса до квантового криптоаналізу проведена в [213,214]. Зокрема, у [214] наводиться оцінка кількості ітерацій для декодування квантовим алгоритмом Гровера (Grover's algorithm). Ця оцінка має вигляд:

Таблиця 15.18.

	Рівень стійкості (без урахування квантового криптоаналізу)		
	Достатній (280 ... 2128)	Високий (2192 ... 2256)	Зверх високий (> 2512)
Криптосистема Мак-Еліса			
Параметри (n, k, d)	(2048, 1300, 137)	(4096, 2584, 253)	16384, 10322, 867)
Розмір секретного ключа, біт	1 712 528	6 726 208	106 773 060

Продовження табл. 15.18.

Розмір відкритого ключа, біт	2 662 400	10 584 064	169 115 648
Складність шифрування бітових операцій	2,6·106	1,1·107	1,7·108
Складність розшифрування бітових операцій	6,2·106	2,7·107	5,1·108
Оцінка стійкості (еквівалентна довжина ключа симетричного шифру), $\log_2 N$	102	186	636
Оцінка стійкості до квантового криптоаналізу, біт	49	91	310
Криптосистема RSA			
Розмір модуля та відкритого (закритого) ключа, біт	2 048	7 680	15360
Складність шифрування (розшифрування), бітових операцій	3,2·109	1,7·1011	1,4·1012
Оцінка стійкості (еквівалентна довжина ключа симетричного шифру), біт	112	192	256
Оцінка стійкості до квантового криптоаналізу, біт	40	41	44

$$C^{\frac{n}{2\log n}},\ C = \frac{1}{(1-R)^{1-R}}, \qquad (15.39)$$

де $R = k/n$ - відносна швидкість використовуваного коду.

Значення таблиці 15.18 розраховані за співвідношенням (15.39). На практиці оцінка (15.39) знижує стійкість криптосистеми – приблизно вдвічі зменшується еквівалентна довжина ключа, що цілком очікувано для надійних постквантових алгоритмів (як і для більшості симетричних шифрів). Для криптосистеми RSA значення таблиці 15.18 оцінювалися наступним чином. Швидкість криптоперетворення (шифрування та розшифрування) оцінювалася як складність модульного піднесення у степінь. У роботі [215, с. 613] показано, що у загальному випадку для l-бітних чисел операція модульного піднесення в степінь вимагає порядку $(3/2)\ l^3$ двійкових операцій. Нехай p і q – два l-бітних простих числа, модуль перетворення RSA (загальносистемний параметр) дорівнює $n = pq$ ($2l$-бітне число), а відкритим (секретним) ключем є $2l$-бітові числа e та d. Тоді складність модульного піднесення у степінь при шифруванні (розшифруванні) вимагатиме $\frac{3}{2}(2l)^3 = 12l^3$ операцій. Більш ефективним є послідовне обчислення піднесення в степінь за модулями $(p - 1)$ та $(q - 1)$ відповідно. Такий алгоритм вимагатиме вдвічі більшої кількості операцій, однак, у зв'язку зі зменшенням розмірності модулів, загальна кількість операцій скоротиться. Складність перетворення складе $2\frac{3}{2}l^3 = 3l^3$ і значення, наведені в

таблиці 15.18, відповідають цій оцінці. Розмір модуля та відповідна оцінка стійкості (як еквівалентна довжина ключа симетричного шифру) зазначена у роботі [216]. Оцінки обсягу квантових ресурсів, необхідних для вирішення деяких асиметричних криптографічних задач за допомогою алгоритму Шора, при різних параметрах цих задач, та порівняння їх зі складністю розв'язання перебірної задачі при пошуку ключа симетричного шифру наведено в [217]. Зокрема, для m – бітного числа дається оцінка $4m^3$ часової складності квантового алгоритму факторизації Шора і значення, наведені в таблиці 15.18, відповідають цій оцінці.

Слід звернути увагу на високу швидкість криптографічного перетворення в схемі Мак-Еліса, що у 3-5 порядків перевищує швидкість шифрування у системі RSA (при порівнянних показниках стійкості).

Другою важливою перевагою схеми Мак-Еліса є можливість поєднувати криптографічне перетворення з контролем помилок, що виникають. Дійсно, якщо при формуванні криптограми (15.36) використовувати випадковий вектор помилок e, ваги $w(e) < t$, тоді з'являється можливість одночасно з криптографічним перетворенням даних контролювати помилки в межах здатності, що виправляє. Зменшення ваги вектора e зменшить криптографічну стійкість схеми Мак-Еліса, але підвищить завадостійкість передачі даних, тобто, у такій «гібридній» схемі, змінюючи $w(e)$, можна адаптивно реагувати на потребу у відповідних послугах безпеки. Позначимо частку ваги вектора помилок вектора e, що припадає на штучне внесення для формування криптограми (див. вираз (15.36)) символом $\rho = w(e) / t$. Тоді стійкість криптосистеми, яка побудована на алгебраїчних кодах, визначатиметься функцією від величини $\rho \cdot t$, а забезпечувана завадостійкість переданих криптограм визначатиметься функцією від величини $(1 - \rho) \cdot t$.

Третя і, очевидно, одна з найважливіших позитивних властивостей криптосистеми Мак-Еліса полягає у високій стійкості до квантового криптоаналізу. У порівнянні з іншими несиметричними криптосистемами, наприклад RSA, складність квантового криптоаналізу кодової криптосистеми зі збільшенням її параметрів зростає дуже швидко. Фактично складність криптоаналізу при використанні квантових алгоритмів можна порівняти з вирішенням перебірних задач пошуку еквівалентних ключів симетричних шифрів. Дані таблиці 15.18 наочно підтверджують цю тенденцію.

Основними недоліками розглянутої кодової криптосистеми є величезні обсяги ключових даних (десятки мегабіт), а також зниження відносної швидкості передачі інформації (найбільша стійкість криптосистеми досягається при відносній швидкості кодування $R = k/n \approx 2/3$).

Криптосистема Нідеррайтера. Альтернативним прикладом криптосистем на кодах є схема Нідеррайтера, вперше запропонована в [187]. Відкритим ключем у цій криптосистемі є матриця

$$H_X = X \cdot H \cdot P \cdot D, \qquad (15.40)$$

де H – перевірна матриця алгебраїчного (n,k,d) коду над $GF(q)$ (в оригінальній статті [187] пропонувалося використовувати узагальнені коди Ріда-Соломона),

X - невироджена $(n-k) \times (n-k)$ матриця з елементами з $GF(q)$, P і D - перестановна та діагональна $(n) \times (n)$ матриці (для двійкових кодів використовується тільки матриця P). Матриці X, P і D (як і для криптосистеми Мак-Еліса) є секретним ключем, який маскує використовуваний алгебраїчний блоковий код під випадковий код (код загального стану), тобто, відкритий ключ (15.40) представляється зловмиснику як випадково сформована перевірна матриця деякого лінійного коду, для якого невідомий алгоритм швидкого декодування. Навпаки, уповноважений користувач, який знає секретний ключ (матриці X, P і D), може зняти дію маскуючих матриць і скористатися швидким алгоритмом декодування коду з перевірочною матрицею H.

Криптограма s_X являє собою вектор довжини $(n-k)$:

$$s_X = e \cdot H_X^T, \tag{15.41}$$

де вектор e – вектор довжини n та ваги $w_h(e) \le t$, який несе конфіденційну інформацію (інформаційне повідомлення, яке підлягає зашифруванню). Найбільша стійкість забезпечується при $w_h(e) = t$. Для розшифрування криптограми s_x виконуються наступні дії [187].

Уповноважений користувач (який має секретний ключ) знаходить одне з q^k рішень виразу $s_x = c_x^* \cdot H_x^T$. Знайдене рішення - кодове слово з помилками $c_x^* = I \cdot G_x + e$. Далі, як і у схемі Мак-Еліса, уповноважений користувач будує вектор $\overline{c}^* = c_X^* \cdot D^{-1} \cdot p^{-1}$ та декодує отримане слово. Однак, замість відновлення інформаційного слова I', він обчислює кодове слово $c' = I' \cdot G$, а потім і вектор помилок $e' = \overline{c}^* - c'$. На останньому етапі здійснюється обчислення вектора $e = e' \cdot P \cdot D$, який несе конфіденційну інформацію. Розшифрування s_x може бути виконано за наступною схемою [186]. Спочатку зауважимо, що вираз (15.41) можна переписати у вигляді $s_x^T = H_x \cdot e^T$. У цьому випадку уповноважений користувач (що має матриці X, P і D) для розшифрування криптограми обчислює вектор $s_x^{*T} = X^{-1} \cdot s_x^T = X^{-1} \cdot H_x \cdot e^T = H \cdot P \cdot D \cdot e^T = H \cdot \overline{e}^T$, значення якого залежить від вектора $\overline{e}^T = P \cdot D \cdot e^T$. Вектор $s_x^{*T} = H \cdot \overline{e}^T$ являє собою синдром, обчислений за перевірочною матрицею H алгебраїчного (n,k,d) коду, тобто, алгоритм швидкого (алгебраїчного) декодування дозволяє знайти вектор $\overline{e}^T$, після чого уповноважений користувач знімає дію матриць маскування P, D і знаходить вектор

$$e^T = D^{-1} \cdot P^{-1} \cdot \overline{e}^T = D^{-1} \cdot P^{-1} \cdot P \cdot D \cdot e^T . \; R = k/n \approx 2/3 .$$

Таким чином, у криптосистемі Нідеррайтера основним засобом маскування лінійного коду під випадковий код є (як і в криптосистемі Мак-Еліса) матриці X, P, D. Якщо використовувати коди Гоппи, то багаточлен $G(x)$ може бути додатковим секретним параметром.

У роботі [187] показано, що стійкості криптосистем Мак-Еліса і Ніддерайтера еквівалентні та ефективну атаку на одну із схем можна легко трансформувати в атаку на іншу схему. У цьому сенсі оцінки стійкості таблиці 15.18 справедливі і стосовно криптосистеми Ніддерайтера. Інші характеристики

цих криптосистем (швидкість криптоперетворення, обсяги ключів) також можна порівняти.

Очевидною перевагою теоретико-кодової схеми Нідеррайтера в порівнянні з криптосистемою Мак-Еліса є потенційно більша відносна швидкість передачі даних. Дійсно, відносна швидкість в криптосистемі Мак-Еліса визначається відносною швидкістю використовуваного (n,k,d) коду, тобто, дорівнює $R=k/n$, причому найбільша стійкість досягається при $R= k/n \approx 2/3$ (див. рис. 15.5). Інформаційне повідомлення в системі Нідеррайтера спочатку перетворюється на рівноважну послідовність e довжини n і ваги $w_h(e) \le t$, а потім множиться на перевірочну матрицю, як у (15.41). Покладемо $w_h(e) = t$ (у цьому випадку буде забезпечено максимальну стійкість криптосистеми для заданих параметрів (n,k,d) коду). Тоді максимальне число біт інформаційних даних, які можна зашифрувати в системі Нідеррайтера при використанні двійкового (n,k,d) коду, визначатиметься виразом: $l_{inf} = \lfloor \log_2 C_n^t \rfloor$, де $\lfloor x \rfloor$- найбільше ціле число, менше x. Криптограма (15.41) є синдромний вектор довжини n - k, тобто, відносна швидкість передачі у криптосистемі Нідеррайтера (для двійкових кодів) визначатиметься виразом: $R^* = l_{inf}/(n-k)$. Ця формула легко узагальнюється на випадок недвійкових кодів із основою q:

$$R^* = \frac{\left\lfloor \log_q \left((q-1)^t \frac{n!}{t!(n-t)!} \right) \right\rfloor}{n-k}. \qquad (15.42)$$

Якщо припустити, що інформаційна послідовність буде перетворюватися на всі можливі вектори e довжини n і ваги $0 \le w_h(e) \le t$, тоді:

$$R^* = \frac{\left\lfloor \log_q \left(\sum_{i=0}^{t} (q-1)^i \frac{n!}{i!(n-i)!} \right) \right\rfloor}{n-k}. \qquad (15.43)$$

Вираз (15.43) досягає максимуму для т. зв. досконалих кодів (perfect codes), кодові (n,k,d) параметри яких задовольняють верхній межі Хеммінга для потужності (числа кодових слів) $A_q(n,d)$ лінійного q –ого коду [207–209]:

$$A_q(n,d) \le \frac{q^n}{\sum_{i=0}^{t} (q-1)^i \frac{n!}{i!(n-i)!}}. \qquad (15.44)$$

Потужність лінійного (n,k,d) коду над $GF(q)$ дорівнює q^k, отже, із (15.44) слідує обмеження на кількість інформаційних символів коду

$$k \le n - \log_q \left(\sum_{i=0}^{t} (q-1)^i \frac{n!}{i!(n-i)!} \right).$$

Якщо параметри (n,k,d) коду задовольняють верхній межі Хеммінга, тоді

$$\log_q \left(\sum_{i=0}^{t} (q-1)^i \frac{n!}{i!(n-i)!} \right) = n-k, \qquad (15.45)$$

що після підстановки в (15.43) дає $R^*=1$, тобто, відносна швидкість максимальна та криптограма у схемі Нідеррайтера не міститиме надмірних символів.

Як приклад, наведемо досконалий двійковий (q = 2) код Хеммінга, що виправляє одну помилку (t = 1). Він визначений для будь-якого позитивного цілого $m>2$ та має кодові параметри n - k = m (n=2^m-1, $k=2^m-m-1, d=3$) [207–209].

Очевидно, що для цих значень

$$\sum_{i=0}^{t}\frac{n!}{i!(n-i)!}=2^m,\ n-k=m$$

та відносна швидкість (15.43) дорівнює 1.

Іншим прикладом є досконалий двійковий код Голея (perfect binary Golay code) з параметрами (n=23, k=12, d=7) [207–209]. Для цих значень маємо:

$$\sum_{i=0}^{t}\frac{n!}{i!(n-i)!}=2048,\ n-k=12,$$

тобто, відносна швидкість (15.43) також дорівнює 1.

В [220,221] показано, що будь-який нетривіальний досконалий код має параметри коду Хеммінга чи коду Голея, тобто, досягнення R^*=1 обмежується лише цими конструкціями. Більшість інших кодів, у тому числі і коди Гоппи, володіють конструктивними параметрами *(n,k,d)*, що лежать істотно нижче верхньої межі (15.38) (реальні дистанційні характеристики кодів Гоппи вище).

Наприклад, для коду Гоппи з параметрами n = 1024, k = 524, t = 50 (використаний в авторському варіанті [118] схеми Мак-Еліса) відносна швидкість шифрування (15.42) у схемі Нідеррайтера дорівнює $R^* \approx 0{,}57$, що трохи вище в порівнянні зі швидкістю R≈0,51 в схемі Мак-Еліса. Зі збільшенням довжини коду конструктивні параметри кодів Гоппи погіршуються, що призводить до зниження швидкості. Цю тенденцію наочно демонструють результати розрахунків у таблиці 15.19, у якій наводяться оцінки відносної швидкості під час використання кодів з параметрами таблиці 15.18.

Очевидно, що зі збільшенням довжини коду Гоппи швидкість (15.42), (15.43) для схеми Нідеррайтера знижується і не перевищує відносної швидкості кодування $R=k/n$. У авторській статті [187] у схемі Нідеррайтера пропонувалося використовувати узагальнені коди Ріда-Соломона, їх (n,k,d) параметри пов'язані співвідношенням $d=n-k+1$, тобто, задовольняють верхній межі Сінлгтона [207-209]. Тоді, наприклад, для q=1024 розширений код Ріда-Соломона буде мати параметри (1024, 524, 501) і оцінка (15.42) дає відносну швидкість для схеми Нідеррайтера $R^* \approx 0{,}66$, що на 30% вище в порівнянні з R≈0,51 для схеми Мак-Еліса. Проте у роботі [211] запропоновано ефективну атаку на криптосистеми з узагальненими кодами Ріда-Соломона, тобто, застосування цього класу кодів неспроможне. Таким чином, стійкі до всіх відомих на сьогоднішній день атак

криптосистеми Мак-Еліса та Нідеррайтера на двійкових кодах Гоппи можна порівняти щодо відносної швидкості передачі даних.

Таблиця 15.19.

Кодові *(n,k,d)* параметри	Конструктивні кодові характеристики			
	(1024, 524, 101)	(2048, 1300, 137)	(4096, 2584, 253)	(16384, 10322, 867)
Схема Мак-Еліса	≈ 0,51	≈ 0,63	≈ 0,63	≈ 0,63
Схема Нідеррайтера	≈ 0,57	≈ 0,57	≈ 0,53	≈ 0,48

Електронний цифровий підпис на основі кодів

Перший відомий алгоритм формування та перевірки ЕЦП з використанням алгебраїчних кодів заснований на криптосистемі Нідеррайтера і був представлений Courtois, Finiasz та Sendrier у [198]. Оцінка стійкості цієї схеми (названої за ініціалами її винахідників – CFS) проти підробки підпису зводиться до оцінки складності розв'язання задач синдромного декодування. Знання секретного ключа дозволяє декодеру вирішити це завдання для деякої частки випадкових кодових слів. В алгоритмі CFS реалізований принцип, який полягає у багаторазовому хешуванні документа, рандомізованого лічильником бітової довжини *r,* доки не буде отримано правильно виділений синдром. Той, хто підписався, використовує свій секретний ключ для визначення відповідного вектора помилок. Разом з поточним значенням лічильника цей вектор помилок використовується як підпис.

Реалізація схеми CFS для формування та перевірки ЕЦП здійснюється відповідно до наступних алгоритмів.

1. *Системні параметри:* $m, t \in N$;

2. *Генерація ключа:* генерація пари ключів, як у криптосистемі Нідеррайтера, на основі використання алгебраїчного коду з класу $(n = 2m,\ k = n - mt,\ 2t + 1)$ двійкових непривідних кодів Гоппи (див. підрозд. 15.3.1), для якого породжуються наступні матриці:

– матриця H: $(n - k) \times n$ - перевірна матриця алгебраїчного коду із здатністю t помилок, що виправляються,

– матриця X: $(n - k) \times (n - k)$ - випадкова оборотна матриця,

– матриця P: $n \times n$ – випадкова матриця перестановок.

Тут слід зазначити, що діагональна матриця D використовується лише для недвійкових кодів і в даному (двійковому) випадку матриця D для генерації пари ключів не потрібна.

Відкритий ключ: матриця $H_x = X \cdot H \cdot P$ та число t (що виправляє здатність коду).

Секретний ключ: матриці X, P і швидкий (поліноміальної складності) алгоритм декодування алгебраїчного коду.

Алгоритм декодування дозволяє за введеною синдромною послідовністю s=$(s_0, s_1, \cdots, s_{n-k-1})$, у разі успіху декодування, знайти вектор помилок e =$(e_0, e_1, \cdots, e_{n-1})$ та кодове слово c =$(c_0, c_1, \cdots, c_{n-1})$. Інакше (якщо декодування не вдалося), алгоритм видає відмову у обробці синдрому s=$(s_0, s_1, \cdots, s_{n-k-1})$, тобто, за такою послідовністю алгоритм не може знайти вектор помилок e =$(e_0, e_1, \cdots, e_{n-1})$ та кодове слово c =$(c_0, c_1, \cdots, c_{n-1})$.

3. *Формування підпису*

Вхід:

1. h – функція хешування, яка застосовується до вхідних даних (аргументу функції) довільної довжини. Результатом хешування є хеш-код $h(x)$ довжини $(n-k)$ біт;

2. Швидкий (поліноміальної складності) алгоритм декодування алгебраїчного коду, який застосовується до синдромної послідовності s=$(s_0, s_1, \cdots, s_{n-k-1})$. Передбачається, що в результаті виконання алгоритму декодування можливі дві ситуації:

– якщо декодування успішне – виводиться знайдений вектор помилок e =$(e_0, e_1, \cdots, e_{n-1})$, який відповідає вектору s=$(s_0, s_1, \cdots, s_{n-k-1})$;

- якщо декодування не успішне – видається повідомлення про неможливість знайти вектор помилок e =$(e_0, e_1, \cdots, e_{n-1})$ для введеного вектора s=$(s_0, s_1, \cdots, s_{n-k-1})$;

3. Відкритий текст *M*, для якого необхідно сформувати ЕЦП за схемою CFS.

Вихід:

ЕЦП за схемою CFS Y для відкритого тексту M.

Алгоритм формування ЕЦП за схемою CFS

Крок 1. Хешування відкритого тексту *M*, тобто, обчислення хеш-коду *h(M)*. Присвоєння змінній *i* значення ***i*** = 1;

Крок 2. Обчислення хеш-коду *h(h(M)‖ i)*, де *h(M)‖ i* – конкатенація (об'єднання) значень *h(M*) і *i*, представлених у вигляді бітових послідовностей;

Крок 3. Значення *h*(*h(M)‖ i*) інтерпретується як синдромна послідовність s=$(s_0, s_1, \cdots, s_{n-k-1})$, яка обчислена для деякого (довільного) кодового слова та вектора помилок e =$(e_0, e_1, \cdots, e_{n-1})$, тобто, передбачається виконання рівності (15.41) для відповідного відкритого ключа $H_x = X \cdot H \cdot P$;

Крок 4. Обчислення значення вектора

$$s_X^{*T} = X^{-1} \cdot s_X^{\ T},$$

який (як передбачається) є синдром, обчислений по перевірочній матриці алгебраїчного (*n*,*k*,*d*) коду, тобто, передбачається, що

$$s_X^{*T} = X^{-1} \cdot s_X^{\ T} = X^{-1} \cdot H_X \cdot e^T = H \cdot P \cdot e^T = H \cdot \overline{e}^T$$

і алгоритм швидкого декодування дозволить знайти вектор $\overline{e}^T = P \cdot e^T$;

Крок 5. Для синдромної послідовності $\boldsymbol{S_x^*}$ реалізується виконання швидкого алгоритму декодування:

– якщо декодування успішне – виводиться знайдений вектор помилок $\overline{e}^T = P \cdot e^T$, який відповідає вектору S_x^*;

– якщо декодування не успішне – видається повідомлення про неможливість знайти вектор помилок $\overline{e}^T = P \cdot e^T$ для введеного вектора S_x^*. Присвоювання змінній *i* значення *i*=*i*+1 та перехід на **Крок 2**;

Крок 6. Обчислення вектора $e^T = P^{-1} \cdot \bar{e}^T = P^{-1} \cdot P \cdot e^T$;

Крок 7. Формування ЕЦП за схемою CFS *Y*=(*e*,*t*) для відкритого тексту *M*.

Таким чином, в результаті виконання розглянутого алгоритму формування ЕЦП за схемою CFS обчислюється таке найменше позитивне ціле число *i*, для якого значення $h(h(M)\| i)$ інтерпретується як синдромна послідовність s=$(s_0, s_1, \cdots, s_{n-k-1})$, що відповідає вектору помилок e =$(e_0, e_1, \cdots, e_{n-1})$, тобто, формально запишемо:

$$Y = (e, i) : H_X \cdot e^T = \left(h(h(M)\|i)\right)^T. \quad (15.46)$$

Задача обчислення вектора e =$(e_0, e_1, \cdots, e_{n-1})$ за відомим вектором $h(h(M)\| i)$ пов'язана з розв'язанням задачі декодування (*n*,*k*,*d*) коду:

– для уповноваженого користувача (який знає секретний ключ) - це обчислювально проста задача (поліноміальної складності);

– для зловмисника (який знає лише відкритий ключ) - це обчислювально складна задача декодування випадкового коду (що відноситься до класу складності *NP*-повних задач).

Для верифікації (перевірки правильності ЕЦП $Y = (e, i)$ повідомленню M) необхідно переконатися, чи є результат хешування $h(h(M)\| i)$ синдромною послідовністю, що обчислена за вектором $e = (e_0, e_1, \cdots, e_{n-1})$ (який інтерпретується як вектор помилок).

4. *Верифікація*

Вхід:

1. Відкритий ключ (матриця $H_x = X \cdot H \cdot P$ та число $\boldsymbol{t}$);
2. Функція хешування $\boldsymbol{h}$;
3. ЕЦП $\boldsymbol{Y}$ = **(*e*,*t*)**;
4. Відкритий текст *M*.

Вихід:

рішення про правильність чи неправильність ЕЦП.

Алгоритм верифікації (алгоритм перевірки ЕЦП за схемою CFS):

Крок 1. Обчислення вектора

$$(s'_X)^T = H_X \cdot e^T;$$

Крок 2. Обчислення вектора

$$(s''_X)^T = h(h(M)\|i);$$

Крок 3. Прийняття рішення про правильність або неправильність ЕЦП:

– якщо $\boldsymbol{S'_x = S''_x}$, тоді приймається рішення про правильність ЕЦП;

– якщо $\boldsymbol{S'_x \neq S''_x}$, тоді ухвалюється рішення про неправильність ЕЦП.

Оцінимо конструктивні характеристики схеми CFS для формування ЕЦП з використанням двійкових блокових (*n,k,d*) кодів.

Складність формування та перевірки ЕЦП

Коректне використання розглянутої схеми підпису CFS передбачає успішне декодування на **кроці 5** алгоритму формування ЕЦП. Оцінимо середнє число спроб декодування (середнє число виконання **кроку 5** алгоритму формування ЕЦП).

Для двійкового блокового (*n,k,d*) коду будь-яка конфігурація із $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ і менше помилок може бути гарантовано виправлена і відповідає, таким чином, унікальній синдромній послідовності. Отже, кількість синдромних послідовностей, які гарантовано будуть коректно оброблені алгоритмом декодування і призведуть до успіху на кроці 5 алгоритму формування ЕЦП, визначається виразом: $N=\sum_{i=0}^{t} C_n^i$. Синдромна послідовність має довжину $n-k$ і загальна кількість всіх можливих таких послідовностей дорівнює 2^{n-k}. Тоді, якщо прийняти припущення про рівноймовірність формування хеш-кодів $h(h(M)\| i)$ на другому кроці алгоритму формування ЕЦП, ймовірність успіху декодування для одиничної спроби на **кроці 5** визначатиметься виразом:

$$P_{у.д.} = \frac{\sum_{i=0}^{t} C_n^i}{2^{n-k}}. \qquad (15.47)$$

Якщо скористатися апроксимацією $\sum_{i=0}^{t} C_n^i = \frac{n^t}{t!}$ [199], то для розглянутих вище двійкових сепарабельних кодів Гоппи з параметрами (див. підрозд. 15.3.1) вираз (15.47) набуде вигляду:

$$P_{у.д.} = \frac{\sum_{i=0}^{t} C_n^i}{2^{n-k}} = \frac{\frac{n^t}{t!}}{n^t} = \frac{1}{t!}. \qquad (15.48)$$

Таким чином, успіх у декодуванні (на **кроці 5** алгоритму формування ЕЦП) буде досягнутий при реалізації в середньому після ***t***! спроб. Кожна спроба потребує $t^2 \cdot m^3$ двійкових операцій [199], тобто, середня кількість бітових операцій, які необхідно витратити на формування ЕЦП за схемою CFS, визначається як:

$$N_{ф.п.} = t^2 \cdot m^3 \cdot t!. \qquad (15.49)$$

Зауважимо, що у формулі (15.49) не враховані витрати на формування хеш-кодів (**кроки** 2 та 3 алгоритму формування ЕЦП), а також не враховані витрати на зняття дії матриці маскування $s_x^{*T} = X^{-1} \cdot s_x^T$ (на **кроці** 4 алгоритму).

Для перевірки (верифікації) ЕЦП, сформованої для повідомлення M, необхідно обчислити хеш-код $h(h(M)\| i)$ і порівняти його з результатом добутку $H_x \cdot e^T$. Якщо не враховувати складність хешування, тоді складність перевірки ЕЦП визначатиметься виразом:

$$N_{п.п.} = (n-k) \cdot n = m \cdot t \cdot 2^m \qquad (15.50)$$

(потрібно (n-k) $\cdot n$ бітових операцій множення і додавання для обчислення $H_x \cdot e^T$).

Стійкість ЕЦП за схемою CFS. В основі побудови схеми ЕЦП CFS лежить використання несиметричної криптосистеми Нідеррайтера, стійкість якої (як кількість покрівельних множин, при яких перестановочне декодування дозволить виправити всі комбінації з t помилок без знання секретного ключа) визначається виразом (15.38). У той самий час, на формування помилкового підпису $Y' = (e', i')$ підробленого повідомлення M' зловмиснику необхідно не тільки реалізувати декодування випадкового коду, а виконати це декодування у середньому $t!$ раз (як і уповноваженому користувачеві), тобто, знайти таке найменше позитивне ціле число i, для якого здійсниться рівність (15.46). З урахуванням останнього зауваження вираз оцінки стійкості ЕЦП за схемою CFS запишемо як:

$$N_c \geq t! \frac{C_n^t}{C_{n-k}^t} = t! \frac{n!(n-k-t)!}{(n-t)!(n-k)!} = t! \frac{2^m!(mt-t)!}{(2^m-t)!(mt)!}. \qquad (15.51)$$

Для оцінки стійкості ЕЦП до квантового криптоаналізу скористаємося формулою (15.39), яка встановлює кількість ітерацій для декодування випадкового коду квантовим алгоритмом Гровера. Таке декодування необхідно виконати в середньому $\boldsymbol{t!}$ разів. Якщо припустити, що квантовий алгоритм, призначений для вирішення перебірних задач, може бути використаний для перебірного пошуку числа $\boldsymbol{i}$ і вимагатиме в середньому $\frac{\pi}{4}\sqrt{\boldsymbol{t!}}$ спроб, тоді вираз для оцінки стійкості ЕЦП за схемою CFS до атак квантового криптоаналізу набуде вигляду:

$$N_{c.к.} \geq \frac{\pi}{4}\sqrt{t!}\left(\frac{1}{(1-R)^{1-R}}\right)^{\frac{n}{2\log n}} =$$
$$= \frac{\pi}{4}\sqrt{t!}\left(\left(1-\frac{k}{n}\right)^{\frac{k}{n}-1}\right)^{\frac{n}{2\log n}} = \frac{\pi}{4}\sqrt{t!}\left(1-\frac{k}{n}\right)^{\frac{k-n}{2\log n}} = \frac{\pi}{4}\sqrt{t!}\left(\frac{m\cdot t}{2^m}\right)^{-t/2}. \qquad (15.52)$$

Довжина ЕЦП за схемою CFS. Цифровий підпис $Y=(e,t)$ складається з двох частин: двійкового вектора e довжиною n біт і цілого числа i, яке може набувати значення в діапазоні $0,1,\cdots,2^{n-k}-1$. Таким чином, бітова довжина ЕЦП (яка записана у вигляді послідовності (e, i)) буде визначатися виразом:

$$l_{ЕЦП} = 2 \cdot n - k = 2^m + m \cdot t. \qquad (15.53)$$

Слід зазначити, що n-бітний вектор e не може набувати всіх 2^n значень. Цей вектор інтерпретується у схемі CFS як бітовий вектор помилок, який можна виправити з допомогою двійкового блокового (n,k,d) коду, тобто, вага Хеммінга вектора e не може перевищувати здатності коду, що виправляється:

$$w(e) \leq t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Отже, кількість всіх допустимих векторів e (як елемента ЕЦП за схемою CFS) визначається виразом $N_{w(e)\le t} = \sum_{i=0}^{t} C_n^i$, і вектор e може бути перетворений на беззбиткову послідовність e^* довжини $\lceil \log_2(N_{w(e)\le t}) \rceil$ біт. З урахуванням беззбиткового кодування вектора e^* вираз (15.53) перепишемо у вигляді:

$$l^*_{ЕЦП} = \lceil log_2(\sum_{i=0}^{t} C_n^i) \rceil + n - k = \lceil log_2(\sum_{i=0}^{t} C_{2^m}^i) \rceil + m \cdot t.$$

Використовуючи вираз (15.44) для верхньої межі Хеммінга, останню формулу можемо переписати у вигляді:

$$l^*_{ЕЦП} \le \lceil log_2(2^{n-k}) \rceil + n - k = 2 \cdot m \cdot t. \qquad (15{,}53^*)$$

Для формування ЕЦП у стислому вигляді $Y=(e^*,t)$ слід використовувати методи рівноважного кодування, викладені, наприклад, в [186, 219].

Об'єм ключових даних, що використовуються у схемі CFS, визначається обсягом ключових даних несиметричної теоретико-кодової схеми Нідеррайтера. Запишемо у явному вигляді ці вирази:

- бітова довжина відкритого ключа (число двійкових осередків матриці $H_x = X \cdot H \cdot P$):

$$l_{\text{в.к.}} = (n-k) \cdot n = n^2 - kn = m \cdot t \cdot 2^m; \qquad (15.54)$$

- бітова довжина закритого ключа (число двійкових осередків матриці X плюс бітова довжина n цілих чисел у діапазоні $0,1,\cdots, n-1$ для вказівки правила заповнення матриці P):

$$l_{з.к.} = (n-k)^2 + n \cdot \lceil \log_2 n \rceil = (m \cdot t)^2 + 2^m \cdot m. \qquad (15.55)$$

В авторській статті [198] розглянуто приклади ЕЦП з $t \in \{8,9,10\}$ і $m \in \{11,12,\cdots,17\}$, оцінки характеристик (15.49) - (15.55) для цих параметрів наведені в таблиці 15.20. В останніх рядках таблиці наведено параметри схеми CFS, стійкої до квантових атак. Аналізуючи дані таблиці 15.20, слід відзначити високу стійкість ЕЦП за схемою CFS, яка, як і очікувалося, являється високою і до атак квантового криптоаналізу. Використання квантових обчислень призводить до квадратичного зниження стійкості схеми CFS, тобто, складність криптоаналізу можна порівняти з вирішенням перебірних задач пошуку еквівалентних ключів симетричних шифрів. Порівняно, наприклад, із криптосистемою RSA (див. таблицю 15.18), спостерігається дуже швидке зростання стійкості.

В той же час очевидним недоліком схеми CFS є висока складність формування ЕЦП. Навіть для невисоких (за сучасними уявленнями) рівнів стійкості 280 .. 290 формування ЕЦП вимагатиме порядку 109 .. 1010 бітових операцій. При цьому, як випливає з даних таблиці 15.20, складність формування ЕЦП з рекомендованими авторами [198] параметрами двійкових кодів Гоппи, від 214 до 216 разів перевищує складність перевірки (верифікації) ЕЦП. Це пояснюється великою кількістю спроб декодування на **кроці** 5 алгоритму формування ЕЦП, які необхідно виконати для знаходження такого позитивного

цілого числа *i*, для якого виконувалася б рівність (15.46). Зазначимо, що складність формування ЕЦП знижується в міру поліпшення кодових співвідношень використовуваного (*n,k,d*) коду і для досконалого коду, що задовольняє верхній кодовій межі Хеммінга (15.44) оцінка (15.47) набуде вигляду: $P_{\text{у.д.}} = 1$. Тоді кожна спроба декодування (для довільного значення ***i***) на **кроці** 5 алгоритму формування ЕЦП призводила б до успіху і не потрібно було б зовсім використовувати параметр ***i.*** Однак побудова довгих двійкових кодів (у тому числі, й сепарабельних кодів Гоппи), близьких за своїми кодовими співвідношеннями до верхньої межі Хеммінга, є відкритою проблемою теорії завадостійкого кодування, яка в контексті розвитку схеми CFS набуває особливої актуальності.

Ще один недолік схеми CFS, не зазначений раніше в інших роботах, полягає в можливості швидкого підроблення підпису *Y=(e, i)*, використовуючи кодові слова *(n,k,d)* коду, що застосовується.

Таблиця 15.20.

m	t	$N_{ф.п.}$	$N_{п.п.}$	$N_{с}$	$N_{с.к.}$	$l^{*}_{\text{ЕЦП}}$	$l_{о.к.}$	$l_{з.к.}$
11	8	231.7	217.5	252,1	225,5	176	217.5	214.9
	9	235.2	217.6	258,3	226.4	198	217.6	215.0
	10	238.8	217.8	264,6	231.6	220	217.8	215.1
12	8	232.1	218.6	259,0	229.0	192	218.6	215.8
	9	235.6	218.8	266,2	229.9	216	218.8	215.9
	10	239.2	218.9	273,3	236.0	240	218.9	216.0
13	8	232.4	219.7	266,0	232.5	208	219.7	216.8
	9	235.9	219.9	274	233.4	234	219.9	216.9
	10	239.5	220.0	282,1	240.4	260	220.0	216.9
14	8	232.7	220.8	273,2	236.1	224	220.8	217.9
	9	236.2	221.0	282,1	237.0	252	221.0	217.9
	10	239.9	221.1	290,9	244.9	280	221.1	217.9
15	8	233.0	221.9	280,4	239.7	240	221.9	218.9
	9	236.5	222.1	290,2	240.6	270	222.1	219.0
	10	240.2	222.2	299,9	249.4	300	222.2	219.0
16	8	233.3	223.0	287,6	243.3	256	223.0	220.0
	9	236.8	223.2	298,3	244.2	288	223.2	220.0
	10	240.4	223.3	2108,9	253.9	320	223.3	220.0
17	8	233.6	224.1	294,9	247.0	272	224.1	221.1
	9	237.1	224.3	2106,5	247.9	306	224.3	221.1
	10	240.7	224.4	2118,1	258,5	340	224.4	221.1
24	9	238.6	231.8	2165,0	282,0	432	231.8	228.6
	10	242.2	231.9	2183,0	291,0	480	231.9	228.6
	11	246	232.0	2201,1	2100,0	528	232.0	228.6
28	10	242.9	236.1	2220,7	2109.9	560	236.1	232.8
	11	246.6	236.3	2242,5	2110.9	616	236.3	232.8
	12	250.4	236.4	2264,14	2131.7	672	236.4	232.8

Дійсно, якщо вибрати довільне кодове слово c=$(c_1, c_2, \cdots . c_{n-1})$ використовуваного *(n,k,d)* коду з перевірочною матрицею H_x, тоді очевидна рівність $H_x \cdot c^T = 0$ приведе до можливості гарантовано підробити підпис $Y_n = (e + c, i)$, причому рівність (15.46) також виконуватиметься:

$$Y_n = (e + c, i) : H_X \cdot (e + c)^T =$$
$$= H_X \cdot e^T + H_X \cdot c^T = H_X \cdot c^T = \left(h(h(M)\|i)\right)^T.$$

Для захисту від такої примітивної підробки в алгоритм верифікації ЕЦП необхідно додати обов'язкову перевірку ваги Хеммінга вектора ***e,*** тобто, **крок** 3 необхідно переписати у вигляді:

Крок 3. Прийняття рішення про правильність або неправильність ЕЦП:

– якщо $s'_x = s''_x$ і w(e)≤ t, тоді приймається рішення щодо правильності ЕЦП;

– якщо $s'_x \neq s''_x$ та (або) w>$(e)t$, тоді приймається рішення про неправильність ЕЦП.

Зазначене зауваження може бути опосередковано реалізовано через застосування алгоритму рівноважного кодування [186]. Так, при формуванні ЕЦП в стислому вигляді $Y=(e^*, t)$ число всіх можливих значень вектора e^* не перевищує 2^{m-t}, частина підробленого ЕЦП, тобто, вектор $e+c$ не буде коректно перетворений на беззбиткову послідовність і навпаки. Однак, при такій організації схеми CFS для захисту від підробки підпису в алгоритм рівноважного кодування, наприклад, в алгоритм 2.2, стор. 99 в [186] необхідно внести можливість відмови в обробці з видачею повідомлення про некоректність введених даних.

Висновки

Несиметричні криптосистеми на основі алгебраїчних блокових кодів були запропоновані близько 40 років тому і сприймалися тоді більшістю дослідників як якийсь екзотичний та малозастосовний напрямок у криптографії. Очевидні недоліки (величезні обсяги ключових даних та зниження відносної швидкості передачі) протягом тривалого часу стримували їх подальший розвиток та практичне використання. І тільки останніми роками, коли стало зрозуміло, що багато існуючих, стандартизованих і широко використовуваних на практиці криптоалгоритмів можуть виявитися беззахисними проти атак квантового криптоаналізу, кодові криптосистеми отримали заслужену увагу дослідників. Декодування випадкового коду – надзвичайно складне обчислювальне завдання та перебірний пошук при його вирішенні, ймовірно, є найкращим із відомих на сьогоднішній день варіантів. Квантові алгоритми прискорюють цей процес, що знижує часові витрати криптоаналізу, але це зниження не є критичним (приблизно вдвічі зменшується еквівалентна довжина ключа). Фактично слід визнати, що кодові криптосистеми є реальною альтернативою сучасних несиметричних криптосистем (RSA, ECC або інших) щодо побудови надійних постквантових алгоритмів. Наведені у розділі розрахунки наочно підтверджують висновок. Крім того, особливості побудови кодових схем дозволяють одночасно

з криптозахистом реалізувати додаткову послугу контролю помилок, які виникають, що, безумовно, становить інтерес для їх застосування в телекомунікаційних системах спеціального призначення.

Для практичного застосування кодових криптосистем необхідно вирішити (або змиритися з їх існуванням) кілька конструктивних проблем і найбільш очевидна з них – величезні обсяги ключових даних. У зв'язку з можливістю використання квантових обчислювальних систем ці обсяги доведеться значно збільшити (приблизно вчетверо). Наприклад, для розглянутих у розділі варіантів обсяги ключів досягають сотень мегабіт і поки неможливо їх зменшити без зниження стійкості криптосистеми. Ключі в кодових схемах – це генераторні, що породжують, (та/або перевірочні) матриці лінійного коду, які мають виглядати для зловмисника, як випадковий набір лінійно незалежних векторів. Стиснути або якимось чином зменшити цей набір без зниження стійкості неможливо.

Очевидним недоліком схеми ЕЦП CFS є висока складність формування підпису. Це пояснюється великою кількістю спроб декодування для формування ЕЦП. У міру поліпшення кодових співвідношень складність формування ЕЦП знижується і для досконалого коду, що задовольняє верхній кодовий кордон Хеммінга, досягається оптимальне значення. Таким чином, побудова довгих двійкових кодів (у тому числі і сепарабельних кодів Гоппи), близьких за своїми кодовими співвідношеннями до верхньої межі Хеммінга в контексті розвитку схеми CFS, набуває особливої актуальності. Питання підвищення обчислювальної ефективності схем формування та перевірки ЕЦП на основі кодів, оцінка стійкості до сучасних методів криптоаналізу є перспективним напрямом подальших досліджень.

15.4 Реалізація постквантового алгоритму електронно-цифрового підпису

Розділ присвячений реалізації та аналізу постквантового алгоритму електронного цифрового підпису (ЕЦП) з використанням хеш-функцій SHA-2 та ДСТУ 7564:2014. Демонструються часові показники роботи алгоритму.

Вступ

На сьогоднішній день стійкість криптографічних алгоритмів з відкритим ключем базується на обчислювальній складності теоретико-числових задач факторизації великих чисел, дискретного логарифмування, перетворень у полі точок на еліптичній кривій. Відомими є алгоритми RSA, DSA, ECDSA.

Проте дослідження у сфері квантових обчислень формують нові виклики у цій галузі криптографії. За твердженнями фахівців NIST та ETSI, дані алгоритми стануть небезпечними після того, як з'явиться прикладний квантовий комп'ютер [100,194,222]. З використанням квантового комп'ютера та алгоритму Шора криптоалгоритми з відкритим ключем, які використовуються на сьогоднішній день, будуть скомпрометовані (табл. 15.21) [194,223,224]. У зв'язку з цим особливо актуальним стало завдання пошуку альтернативних криптографічних примітивів, які в перспективі будуть стійкими до атак, що реалізуються на

квантових комп'ютерах. Такі алгоритми отримали назву квантово-безпечні (quantum-are safe) (ETSI, EU) або квантово-захищені (quantum-protected) (NIST, USA) алгоритми, а відповідний розділ криптографії, що вивчає проблеми проєктування таких алгоритмів, отримав назву постквантової криптографії (post-quantum cryptography) [100,194, 222-224].

Робочі групи ETSI і NIST визначили перспективні напрямки, в рамках яких можна отримати прийнятні рішення, а саме: суперсингулярні еліптичні криві, мультиваріативна криптографія, криптографія на основі завадостійкого кодування, криптографія на основі хеш функцій.

Сьогодні актуальним завданням є пошук алгоритму цифрового підпису як найбільш масового у застосуванні криптопримітива. В Україні прийнято національний стандарт функції хешування Купина ДСТУ 7564:2014 [78]. За деякими оцінками, дана хеш-функція цілком може бути віднесена до постквантових примітивів. У зв'язку з цим вона представляє інтерес для побудови на її основі постквантової ЕЦП класу Hash-Based digital signature (HB-підпис). У цьому розділі розглядається варіант побудови ЕЦП на основі схеми Lamport-Diffie-Winter-nitz-Merkle (LDWM) з використанням хеш-функції "Купина". На нашу думку, це перспективний напрямок для розробки національної постквантової ЕЦП.

Таблиця 15.21.

Рівень захищеності алгоритмів,
що використовуються на сьогодні

Алгоритм	Довжина ключа	Безпечний рівень	
		Класичний комп’ютер	Квантовий комп’ютер
RSA-1024	1024 b	80 b	0 b
RSA-2048	2048 b	112 b	0 b
ECC-256	256 b	128 b	0 b
ECC-384	384 b	256 b	0 b

Схема одноразового підпису та схема дерева сигнатур Мерклі Lamport-Diffie-Winternitz-Merkle (LDWM) забезпечує контроль цілісності повідомлень на принципах несиметричної криптографії. Повідомлення підписується секретним ключем, а перевірка здійснюється відкритим ключем. Схема одноразового підпису може бути використана тільки для одного повідомлення, тому є необхідність використовувати загальну схему дерева сигнатур Мерклі, де одним із компонентів є схема одноразового підпису.

1. Аспекти реалізації схеми ЕЦП

Реалізація схеми ЕЦП здійснюється у два етапи: реалізація схеми одноразового підпису та реалізація загальної схеми дерева сигнатур.

Як базовий елемент ЕЦП у даній схемі ми будемо використовувати дві хеш-функції SHA-2 та ДСТУ 7564:2014 «Купина». Зауважимо, що стійкість HB-підпису залежить від стійкості хеш-функції.

1.1 Перший етап

Для того, щоб безпосередньо почати описувати структурні особливості схеми одноразового підпису, необхідно визначити ряд параметрів:

– *m* - довжина в байтах кожного елемента LDWM;

– *n* - довжина в байтах на виході хеш-функції;

– *w* - параметр Winternitz, який набуває значень {1,2,,4,8};

– *p* - число байтових рядків, що складають LDWM схему;

– *ls* - число, яке використовується у функції контрольної суми C.

1.1.1 Формування закритого ключа

Закритий ключ *x* є масив розміром *p* , що містить *n*-байтні рядки, які генеруються шляхом використання генератора випадкових послідовностей.

Програма формування закритого ключа

```
for i=0, i<p, i++
        set x[i] to a uniformly random value
end for
        return x
```

1.1.2 Формування відкритого ключа

У LDWM відкритий ключ *y* генерується із закритого ключа *x* з використанням операції конкатенації та хеш-функції *H*, яка в ході виконання повертає 256 значення.

Програма формування відкритого ключа

```
e = 2^w - 1
     for i=0,i<p,i++
    y[i] = F^e(x[i])
    end for
return H(y[0] || y[1] || ... || y[p-1])
```

Робота функції *F* описується виразом [238-240]:

$$F^i(x) = \begin{cases} F(F^{(i-1)}(x)) & \text{for } i > 0 \\ x & \text{for } i = 0. \end{cases}$$

Ця функція виконує операцію хешування *i* раз і повертає 20 – байтне значення.

1.1.3 Функція Checksum

Функція Checksum повертає позитивне число sum, яке потрібне для запобігання зміни ЕЦП.

Програма запобігання зміни ЕЦП

```
      sum = 0
         for i=0,i<u,i++
           sum = sum + (2^w - 1) - coef(S, i, w)
          end for
       return (sum << ls)
```

1.1.4 Генерація сигнатури

Сигнатура обчислюється шляхом конкатенації значення хеш-функції повідомлення та контрольної суми з використанням закритого ключа.

Програма обчислення сигнатури

```
V = (H(message) || C(H(message)))
    for i=0, i<p, i++
     a = coef(V, i, w)
     y[i] = F^a(x[i])
    end for
return ( y[0] || y[1] || ... || y[p-1] )
```

1.1.5 Перевірка підпису

Для того, щоб перевірити повідомлення, одержувачу необхідно завершити серію *F*, використовуючи значення хеш-функції від повідомлення та її контрольної суми. Ці дії призводять до отримання відкритого ключа.

Програма перевірки підпису

```
V = ( H(message) || C(H(message)) )
    for i=0,i<p,i++
     a = (2^w - 1) - coef(V, i, w)
     z[i] = F^a(x[i])
    end for
  if public key is equal to H(z[0] || z[1] || ... || z[p-1])
    return 1 (message signature is valid)
  else
  return 0 (message signature is invalid)
```

1.2 Другий етап

Дерево сигнатур Мерклі використовує два компоненти: схему одноразового підпису та хеш-функцію. Кожна пара відкритого/закритого ключів асоційована із *k*-way деревом. Кожен вузол дерева містить значення *n*-byte. Кожен аркуш дерева містить пару відкритий/закритий ключ LDWM. У свою чергу, корінь дерева називається «Відкритим ключем дерева сигнатур Мерклі». Як і в першому етапі реалізації схеми, необхідно визначити такі параметри:

- *k* – число дочірніх вузлів у внутрішньому вузлі.
- *h* – висота дерева.
- *n* – число, асоційоване з кожним вузлом.

1.2.1 Закритий ключ

Закритий ключ складається з k^h LDWM закритих ключів та закритого ключа LDWM наступного, не використаного листа дерева. Генерація проводиться з використанням псевдовипадкового генератора з використанням секретного значення.

1.2.2 Відкритий ключ

Для обчислення відкритого ключа необхідно помістити в листя дерева значення LDWM відкритих ключів, отриманих у пункті 1.1.2, далі необхідно

заповнити вище вузли шляхом застосування операції конкатенації та хеш-функції до дочірніх вузлів. У дереві Мерклі відкритим ключем є кореневий вузол. На рисунку 15.6 показаний приклад генерації відкритого ключа: для цього буде використаний перший вузол (1) зі значенням Y_0, вузол номер 3 матиме хеш-значення конкатенації першого і другого вузлів. Пофарбовані вузли (2,6,14) представляють перший шлях автентифікації. Він утворений двійниками правих вузлів, що з'єднують лист та корінь дерева, які використовуються для перевірки відкритого ключа. Алгоритм пошуку відкритого ключа описано в [238].

Генерацію відкритого ключа можна представити у вигляді наступного псевдокода:

for i = 0 to num_ldwm_keys by steps of k
level = 0
for j = 0 to k-1
push ldwm_public_key(i+j) on the data stack
push ldwm_public_key(i+j) on the data stack
push level on the integer stack
end for
while the height of the stack is at least k
if the top k elements on the integer stack are equal
pop the top k elements of the data stack
pop the top k elements of the integer stack,
hash the top k elements of the data stack
push the hash result on the data stack
push level on to the integer stack
endif endwhile end for

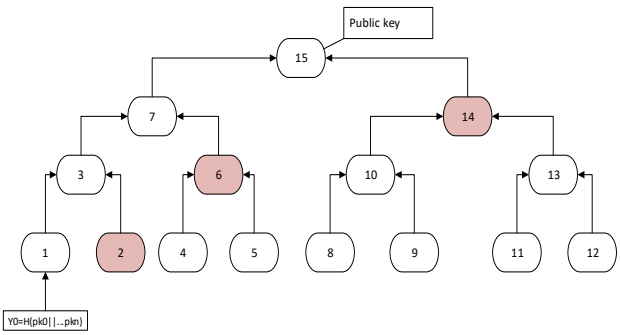


Рис. 15.6. Генерація відкритого ключа

1.2.3 Генерація підпису

Підпис складається із трьох елементів:

– підпис LDWM;.

– індекс використаного листа;

– шлях аутентифкації.

1.2.4 Перевірка підпису

Для того, щоб перевірити підпис, необхідно спочатку переконатися, що підпис LDWM правильний. Інакше перевірка далі не здійснюється і вважається,

що підпис невірний. Далі необхідно «прокрокувати» по дереву, використавши вираз [238-240]:

$$p_i = \begin{cases} g(Aut_{i-1} \| p_{i-1}) \text{ if } \lfloor s/(2^{i-1}) \rfloor \equiv 1 \bmod 2; \\ g(p_{i-1} \| Aut_{i-1}) \text{ otherwise.} \end{cases}$$

Якщо значення кореня дерева та обчислене значення співпали, то приймається рішення, що підпис правильний, у всіх інших випадках підпис недійсний.

2. Порівняльний аналіз

Перед авторами стояло завдання реалізувати НВ-підпис з використанням національного стандарту хеш-функції та порівняти з реалізацією на основі хеш-функції SHA-2.

Український національний стандарт функції хешування здатний працювати із довжиною хеш-значення 256/384/512. Ця хеш-функція базується на структурі Rijndael. Стійкість "Купини" представлена в табл. 15.22.

Таблиця 15.22.

Стійкість до криптографічних атак

Вид атаки	Купина – 256
Колізія	2^{128}
Прообраз	2^{256}
Другий прообраз	2^{256}
Фіксована точка	2^{256}
Збільшення довжини	2^{256}

Хеш-функція SHA-256 розроблена Агентством національної безпеки США та побудована на основі структури Мерклі-Дамгарда. Ця функція має при своїх 64 раундах достатню стійкість до 112 біт безпеки.

У таблиці 15.23 наведено результати порівняння двох реалізацій НВ-підпису.

Таблиця 15.23.

Часові співвідношення

Алгоритм	Час генерації ключів (мс)	Час підпису (мс)	Час перевірки підпису (мс)
SHA-2	50	4	4
«Купина»	47	3	3

У ході експерименту було використано схему дерева Мерклі з глибиною дерева, що дорівнює п'яти. Дане програмне забезпечення було запущено на процесорі Intel Pentium (R) CPU G850 @ 2.90 GHz.

Висновок

У розділі були описані прикладні аспекти реалізації постквантового алгоритму ЕЦП з ілюстрацією алгоритму, описаного на псевдокоді. Показана принципова можливість реалізації постквантового цифрового підпису з використанням національного стандарту хеш-функції. У результаті аналізу алгоритму, з використанням різних хеш-функцій, було виявлено невелику перевагу алгоритму «Купина-256» над функцією хешування SHA-256. Це дозволить нам зробити висновок, що можна спрямувати зусилля дослідників у цьому напрямку та продовжити аналіз різних характеристик НВ-підпису на основі національного стандарту хеш-функції. Крім того, цей напрямок бачиться як найбільш перспективний для розробки нового національного стандарту цифрового підпису.

15.5 Методи побудови та властивості логарифмічних підписів

Вступ

У розділі запропонований огляд властивостей перспективного напряму розвитку криптографічних систем на основі логарифмічних підписів і покриттів кінцевих груп, який належить до постквантової криптографії. Актуальний стан цього напряму й праці останніх років дають підстави припускати, що завдання факторизації елемента кінцевої групи в теорії побудови криптосистем на основі неабелевих груп з використанням логарифмічних підписів є обчислювально складні, що потенційно забезпечує необхідний рівень криптографічного захисту перед атаками, що використовують можливості квантових обчислень. Наведено основні визначення логарифмічних підписів і покриттів кінцевих груп, їх класифікацію, властивості. У рамках аналізу підходів розглянуто практичні випадки використання властивостей логарифмічних підписів для забезпечення криптостійкості базових платформ для побудови криптосистем MST3.

Разом зі зростанням практичних можливостей використання квантових обчислень зростає загроза класичним схемам шифрування та електронного підпису, які використовують, як основу, класичні математичні проблеми, що долаються обчислювальними можливостями квантових комп’ютерів. Цей факт мотивує дослідження фундаментальних теорем, що стосуються математичних та обчислювальних аспектів постквантових криптосистем-кандидатів. У розділі представлено логарифмічні підписи, як особливий тип факторизації в кінцевих групах, розглянуто їх властивості та методи побудови.

Логарифмічний підпис

Нехай G – кінцева група. Логарифмічний підпис α для групи G є послідовністю підмножин $A_i \subseteq G$ виду $\alpha = [A_1, \dots, A_s]$, таких, що для кожного елемента g групи G є лише одна факторизація $(*)$ $g == a_q \cdot a_2 \cdot \ldots \cdot a_s$, де $a_i \in A_i$ для $i = 1, \dots, s$. Множини A_i називають блоками. Розмір списку блоків позначають через $r_i := |A_i|$. Для спрощення ми називаємо елементи $A_i \cup \dots \cup A_s$ елементами логарифмічного підпису α . За певних умов ми розглядаємо впорядкування елементів блоку, тоді для $k_i = 0, \dots, r_i - 1$ позначаємо через a_{ik_i}

кожний $(k_i + 1)$-й елемент блоку A_i. Вектор $(r_1, \dots, r_s)$ називають типом α, a - довжиною логарифмічного підпису.

$$\ell(\alpha) = \sum_{i=1}^{s} r_i .$$

Множину логарифмічних підписів групи позначають через $\Lambda(G)$. Із визначення отримуємо певні властивості логарифмічних підписів. Через одноманітність факторизації (*) маємо $\prod_{i=1}^{s} r_i = |G|$ і тоді r_i ділить $|G|$ для всіх i. Це також показує, чому зазначені послідовності називають логарифмічними підписами. Логарифмічна функція перетворює добуток на суму, логарифмічні підписи скорочено відображають усі елементи групи довжини $r_1 + \dots + r_s$, де група містить у собі $r_1 \cdot \dots \cdot r_s$ елементів.

Якщо $e_G \in A_1 \cap \dots \cap A_s$, тоді ми можемо стверджувати, що α є нормалізованою величиною. З огляду на одиничність факторизації, ми навіть маємо $\bigcap_{i=1}^{s} A_i = \{e_G\}$ для нормалізованих логарифмічних підписів із більше ніж одним блоком. Легко помітити, що в абелевих групах $|A_i \cap A_j| \leq 1$ для $i \neq j$. У [225] доведено, що для $|G| = \prod_{j=1}^{t} p_j \cdot b_j$ (p_j – просте число) є нижня границя довжини для будь-якого логарифмічного підпису групи G, одержаного таким чином:

$$\ell(\alpha) \geq \sum_{j=1}^{t} b_j \cdot p_j .$$

Логарифмічний підпис α називають мінімальним, якщо $\ell(\alpha)$ досягає нижньої границі, тобто, кожен блок має простий степінь або степінь 4. У [225,226] порушене питання існування мінімальних логарифмічних підписів у кінцевих полях. Було запропоновано мінімальні логарифмічні підписи для всіх кінцевих груп.

Приклад 15.1. Нехай $n \in \mathbb{N}$. Для циклічної групи $(\mathbb{Z}_{2^n}, +)$ послідовність виду $\alpha = \left[[0, 2^{n-1}], [0, 2^{n-2}], \dots, [0,2], [0,1]\right]$ є нормалізованим логарифмічним підписом типу $(2, \dots, 2)$. Обчислення факторизації елемента еквівалентне обчисленню його двійкового відображення, зокрема, якщо $n = 4$, $9 = 1001$ має факторизацію $2^3 + 0 + +0 + 2^0$.

Розглянемо можливість обчислення факторизації елемента групи для зазначеного логарифмічного підпису і певного елемента групи. Для прикладу, атака повним перебором за допомогою пошуку всіх можливих факторизацій, репрезентованих логарифмічним підписом $\alpha = [A_1, \dots, A_s]$ групи G, становить $|G| \times (s - 1)$ групових операцій у найгіршому випадку. Такий перебір знаходить правильну факторизацію для будь-якого логарифмічного підпису, але загалом неможливий. *Приклад 15.1*показує, що для певних логарифмічних підписів легко обчислити факторизації. Для практичного використання в криптосистемах MST необхідно визначити логарифмічні підписи, для яких факторизація є обчислювально нездійсненною, а також підписи, для яких є ефективні алгоритми розкладання. Здебільшого, терміни «прості» і «складні» логарифмічні підписи

вживають для позначення різниці між логарифмічними підписами, для яких відповідно обчислювально легко та складно отримати факторизації [227,228]. Факторизація одного логарифмічного підпису становить постійний час, але коли ми вивчаємо питання ефективності обчислень для логарифмічних підписів, то розглядаємо сімейство (G_n, α_n) логарифмічних підписів $\alpha_n = \left[A_1^n, \dots, A_{s_n}^n\right]$ груп G_n для $n \in \mathbb{N}$. Далі ми припускаємо, що $|G_n| \leq |G_{n+1}|$ і $\alpha_n \neq \alpha_m$ для $n \neq m$. Крім того, нехай одноманітний опис елементів G_n є таким, що $|g|_2 = \mu$ для всіх $g \in G_n$. Зауважуємо, що $\mu_n \geq log|G_n|$ для всіх $n \in \mathbb{N}$.

Робимо одне базове припущення: для зазначеного сімейства (G_n, α_n) є детермінований алгоритм поліноміального часу A такий, що, маючи вхідні значення $\left(a_1, \dots, a_{s_n}\right)$ із $A_1^n \times \dots \times A_{s_n}^n$, обчислює добуток $a_1 \cdot \dots \cdot a_{s_n}$. Ми ідентифікуємо A з ін'єктованою функцією, що його обчислює.

Визначення 1. Для $n \in \mathbb{N}$ нехай α_n буде як у наведеному прикладі. Тоді сімейство $(G_n, \alpha_n)_{n \in \mathbb{N}}$ називають складним, якщо для кожного ймовірнісного алгоритму поліноміального часу A' для кожного позитивного полінома p і всіх істотно великих n маємо:

$$pr\left(A'(g_n, \alpha_n) = A^{-1}(g_n)\right) < \frac{1}{p(\lceil log|G_n| \rceil)},$$

де g_n позначає випадковий елемент, одноманітно вибраний із G_n.

Для сімейства складних логарифмічних підписів $(G_n, \alpha_n)_{n \in \mathbb{N}}$ функція A визначає односторонню функцію, як зазначено в [229]. Серед логарифмічних підписів, що не є складними, ми визначаємо ті, які мають ефективні факторизації для всіх елементів групи.

Визначення 2. Для $n \in \mathbb{N}$ нехай a_n буде логарифмічним підписом групи G_n. Сімейство $(G_n, \alpha_n)_{n \in \mathbb{N}}$ називають простим, якщо є алгоритм A'', що, одержуючи на вході елементи $g_n \in G_n$ і α_n, обчислює факторизацію g_n щодо α_n за поліноміальний час. Різницю між «нескладний» і «простий» уперше розглянуто в [230], і, на відміну від попередніх визначень (у яких ці два поняття еквівалентні), визначення є більш точним. Логарифмічний підпис, що не є ні простим, ні складним, може бути таким, для якого ми можемо ефективно знайти факторизацію для рівно половини групових елементів. Є алгоритм із заданими g і α_n, що випадково вгадує факторизацію з імовірністю $\frac{1}{|G_n|}$. Отже, алгоритм A' у другому визначенні дає значно більший шанс знайти факторизацію, як і раніше менший, ніж $\frac{1}{p(\lceil log|G_n| \rceil)}$ (для будь-якого p). Алгоритм A' додатково одержує n на вхід для кодування групи G_n, такий, що A' обчислює G_n. Але твердження, що, маючи α_n на вході, гарантовано A' обчислює групу G_n, є непідтвердженими. Також припустимо, що для вхідних даних $\left(a_1, \dots, a_{s_n}\right)$ алгоритм A "знає", яку групову операцію застосовувати. Обчислення добутку елементів s_n, кожний із яких був випадково одноманітно вибраним із різних блоків a_n (щодо A), еквівалентне вибору випадкового елемента з G_n.

Відзначимо, що питання наявності варіантів для складних логарифмічних підписів залишається відкритим. Усі логарифмічні підписи, описані в літературі,

репрезентовані простими [225–228,231]. Криптосистеми MST використовують як складні, так і прості логарифмічні підписи для побудови криптосистем відкритого ключа. Криптосистема MST_1 ґрунтується на складних логарифмічних підписах; MST_3, у свою чергу, використовує прості логарифмічні підписи в елементарних абелевих 2-групах. Отже, нас цікавить структура простих логарифмічних підписів, а також груп, для яких можуть існувати логарифмічні підписи. Для подальшого аналізу груп важливими є три параметри. Для $n \in \mathbb{N}$ логарифмічний підпис $\alpha_n = \left[A_1^n, \dots, A_{s_n}^n\right]$ повністю описаний, як

$$|\alpha_n| = \sum_{i=1}^{s} |A_i^n|$$

елементів із групи G_n. Можливо, є коротше репрезентування певних логарифмічних підписів, але загалом нам необхідно

$$\sum_{i=1}^{s} |A_i^n| \cdot \mu_n$$

біт, щоб виразити логарифмічний підпис, у якому елемент G_n репрезентований за допомогою μ_n біт. Нехай:

$$r_n = max\{|A_i^n|, \dots, |A_{s_n}^n|\}.$$

Алгоритм факторизації A' у другому визначенні бере як вхідні дані елемент G_n і логарифмічний підпис α_n для певних $n \in \mathbb{N}$. Довжина цих вхідних даних дорівнює не більше за $(s_n \cdot r_n + 1) \cdot \mu_n$ біт.

Отже, ми пропонуємо використовувати три таких значення (s_n, r_n, μ_n), як параметри вимірювання ефективності алгоритму факторизації для α_n. Зазначимо, що ми одержуємо перші два параметри зі структури α_n, а третій параметр є незалежним від α_n та обумовлений лише репрезентуванням елементів G_n.

Зауваження. Для $n \in \mathbb{N}$ нехай (α_n) буде логарифмічним підписом у G_n. Якщо для всіх $g \in G_n$ факторизацію щодо α_n можна отримати за поліноміальний час із трьома параметрами s_n, r_n, μ_n, тоді сімейство $(\alpha_n)_n$ є простим.

Приклад 15.2. Візьмемо сімейство логарифмічних підписів у групі $G_n = \mathbb{Z}_{2^n}$ із прикладу 1. Три параметри ефективності дорівнюватимуть $s_n = n, r_n = 2, \mu_n = n$. Для заданого елемента $g \in \mathbb{Z}_{2^n}$ за допомогою його двійкового представлення $(g_1, \dots, g_n)$, де $g_i \in \{0,1\}$ – його факторизація, яка щодо α дорівнює $(g_1 \cdot 2^{n-1}, \dots, g_n \cdot 1)$, що одержано за допомогою не більше ніж n множень. Отже, маємо лінійний час у n. Тоді $(\alpha_n)_{n \in \mathbb{N}}$ є простим.

Перетворення логарифмічних підписів

Розглянемо певні перетворення логарифмічних підписів, зокрема, такі, щоб факторизація щодо вихідного логарифмічного підпису була однаково ефективною щодо перетвореного логарифмічного підпису. Ідея полягає в тому, що алгоритм факторизації одного логарифмічного підпису в певному наборі дає алгоритм факторизації для всіх логарифмічних підписів цього набору. Наведемо

стандартний підхід до класифікації логарифмічних підписів відповідно до структури блоків. Ми розглядаємо п’ять перетворень логарифмічних підписів, що не змінюють властивостей належності до простих або складних логарифмічних підписів. Нехай $\alpha = [A_1, \dots, A_s]$ буде логарифмічним підписом групи G. Ми маємо справу з перетвореннями блоків $A_1, \dots, A_s$ логарифмічного підпису α на блоки $B_1, \dots, B_s$, які є результуючою послідовністю $\beta = [B_1, \dots, B_s]$ і також репрезентують логарифмічний підпис G.

Перетворення 1. Нехай φ буде автоморфізмом G і $B_i = \varphi(A_i)$ для $i = 1, \dots, s$. Тоді β також є логарифмічним підписом. І якщо $a_1 \cdot a_2 \cdot \dots \cdot a_3$ є факторизацією елементів $g \in G$, $\varphi(a_1) \cdot \varphi(a_2) \cdot \dots \cdot \varphi(a_s)$, то є і факторизацією $\varphi(g)$ щодо β.

Перетворення 2. Нехай $g_0, \dots, g_s$ будуть елементами групи G і $B_i = g_{i-1}^{-1} A_i g_i$ для $i = 1, \dots, s$. Тоді послідовність β також є логарифмічним підписом G, що називають трансляцією α. Якщо $g_0 = g_s = e_G$, то β – сендвіч α. Зазначимо, що, якщо $a_1 \cdot a_2 \cdot \dots \cdot a_s$ є факторизацією елементів $g \in G$, $g_0^{-1} a_1 g_1 \cdot \dots \cdot g_{s-1}^{-1} a_s g_s$, то є факторизацією $g_0^{-1} g g_s$ і щодо β. Необхідно пам’ятати, що в абелевих групах блоки трансляції β від α мають вигляд $B_i = A_i + h_i$ для елементів $h_1, \dots, h_s$ групи G, а, отже, будь-яка факторизація елемента $g \in G$ щодо α миттєво дає факторизацію $g + \sum_{i=1}^{s} h_i$ щодо β.

Перетворення 3. Для $i = 1, \dots, s$ нехай π_i буде перестановкою в S_{r_i} і $B_i = \left[a_{i\pi_i(1)}, \dots, a_{i\pi_i(r_i)}\right]$ для $j = 1, \dots, r_i$, тобто, елементи блоку B_i – перестановка елементів блоку A_i. Тоді β також є логарифмічним підписом. І, якщо $a_{1k_1} \cdot a_{2k_2} \cdot \dots \cdot a_{sk_s}$ – факторизація елемента $g \in G$, то $a_{1\pi_1(k_1)} \cdot a_{2\pi_2(k_2)} \cdot \dots \cdot a_{s\pi_s(k_s)}$ – факторизація g щодо β.

Перетворення 4. Тепер нехай G буде абелевою групою, π – перестановкою в S_s і $B_i = A_{\pi(i)}$, тобто, $b_{ij} = a_{\pi(i)_j}$. Тоді послідовність β є логарифмічним підписом групи G. Її також називають перетворенням блочної перестановки α. Зазначимо, що якщо $a_{1i_1} + a_{2i_2} + \dots + a_{si_s}$ є факторизацією елемента $g \in G$ щодо α, то $a_{\pi(1)(i_1)} \cdot a_{\pi(2)(i_2)} \cdot \dots \cdot a_{\pi(s)(i_s)}$ є факторизацією g щодо β. Крім того, в неабелевих групах β може не бути логарифмічним підписом.

Перетворення 5 . Для певних $j \in \{1, \dots, s-1\}$ нехай $B_j = A_j \cdot A_{j+1} = \left[x \cdot y | x \in A_j, y \in A_{j+1}\right]$ та $B_i = A_i$ для $i = 1, \dots, s-1$ і $i \neq j, j+1$. Послідовність $\beta = [B_1, \dots, B_{s-1}]$ є логарифмічним підписом, одержаним з α в результаті перетворення – злиття двох блоків. І, якщо $a_1 \cdot a_2 \cdot \dots \cdot a_s$ є факторизацією $g \in G$ щодо α, $a_1 \cdot a_{j-1} \cdot a \cdot a_{j+2} \dots \cdot a_s$, де $a = a_j \cdot a_{j+1}$, є факторизацією g щодо β. Зворотну операцію називають перетворенням розподілу. Для кожного з п'яти наведених перетворень ми описали, як факторизація елемента щодо логарифмічного підпису негайно приводить до факторизації щодо перетвореного логарифмічного підпису. Якщо ми розглядаємо ці перетворення для родин логарифмічних підписів (α_n), то легко помітити, що перемикання між алгоритмами факторизації (α_n) і перетвореннями (α_n) виконується за поліноміальний час, якщо перетворення є відомим або ефективно обчислюваним. Це справедливо й для нормалізації.

Визначення 3. Нехай (α_n) і (β_n) є родинами логарифмічних підписів для груп G_n з параметрами r_n, s_n, μ_n для (α_n). Тоді ми стверджуємо, що (α_n) перетворюється на (β_n), якщо (β_n) можна обчислити з (α_n) за допомогою перетворень 1 – 5 за поліноміальний час для r_n, s_n, μ_n. Зазначимо, що в такому разі кількість перетворень між (α_n) і (β_n) є кінцевою. $T-$ множина (α_n), визначена як $T(\alpha_n) = \{(\beta_n)|\beta_n\ \Lambda\ (G_n)$ і α_n перетвориться для всіх $n\}$.

Приклад 15.3. Візьмемо логарифмічний підпис $\alpha_n = \left[[0, 2^{n-1}], \dots, [0,2], [0,1]\right]$ групи $\mathbb{Z}_{2^n}$ із прикладу 1. Для $n > 4$ нехай $\beta_n = \left[[1,5], [0,1,2,3], [0,8], [0,16], \dots, [0, 2^{n-1}]\right]$ і $\gamma_n = \left[[0,1,2,3, \dots, 2^n - 1]\right]$. Тоді $(\beta_n) \in T(\alpha_n)$ і $(\gamma_n) \notin T(\alpha_n)$.

Визначення 4. Нехай $g_0 = 1$ і $g_i = \left(\prod_{j=1}^{i} a_{j1}\right)^{-1}$ для $i = 1, \dots, s$. За допомогою трансляції α в $g_0, \dots, g_s$ ми одержуємо логарифмічний підпис β, у якому $b_{i1} = g_{i-1}^{-1} a_{i1} g_i = \left(\left(\prod_{j=1}^{i-1} a_{j1}\right)^{-1}\right)^{-1} a_{i1} \left(\prod_{j=1}^{i} a_{j1}\right)^{-1} = \left(\prod_{j=1}^{i} a_{j1}\right) \cdot \cdot \left(\prod_{j=1}^{i} a_{j1}\right)^{-1} = 1$, тобто, перший елемент у кожному блоці є нейтральним. Ми вважаємо β нормалізацією α. В абелевих групах ми можемо нормалізувати логарифмічний підпис, застосовуючи трансляцію $B_i = A_i - a_{i1}$. Тоді перший елемент кожного блоку дорівнюватиме $a_{i1} - a_{i1} = 0$. Інші класи мають стандартне позначення. Нехай G буде кінцевою групою. Ми називаємо точно поперечним логарифмічний підпис $\alpha \in \Lambda(G)$, якщо такий ланцюг підгруп виду $e_G = G_0 < G_1 < \dots < G_{s-1} < G_s = G$, якщо A_i є поперечною групи G_{i-1} у G_i, тобто, $G_{i-1} A_i = G_i$ та $|G_{i-1}||A_i| = |G_i|$.

Зазначимо, що блок $A_1 = G_1$ є підгрупою G. Відповідний клас позначають через ε. Якщо α є сендвічем точно поперечної логарифмічної групи, α називають поперечною. Клас поперечних логарифмічних підписів позначимо через T_{LS}. Усі інші логарифмічні підписи належать до NT_{LS} – класу непоперечних логарифмічних підписів. Ми описуємо два підкласи NT_{LS}. Якщо жодний із блоків не є підмножиною (нетривіальною) підгрупи G, логарифмічний підпис є елементом TNT_{LS} – класу абсолютно непоперечних логарифмічних підписів. Клас TA_{LS} повністю аперіодичних логарифмічних підписів містить усі логарифмічні підписи, що не мають навіть періодичного блоку, тобто, об'єднані підмножиною G. Отже, маємо $TA_{LS} \subseteq TNT_{LS} \subseteq NT_{LS}$ і $\varepsilon \subseteq T_{LS}$. Необхідно пам'ятати, що властивість простоти для точно поперечних логарифмічних підписів породжується в певних групах.

Визначення 5. Нехай (α_n) – сімейство точних логарифмічних підписів, а тестування на належність підгрупи до групи G_n проведено за поліноміальний час μ_n. Тоді (α_n) є простим. Для простішого читання опустимо індекс n. Нехай

$$e_G = G_0 < G_1 < \dots < G_{s-1} < G_s = G$$

буде ланцюгом підгруп відповідних логарифмічних підписів $\alpha = [A_1, \dots, A_s]$. Нехай $g \in G$. Є саме одна факторизація g і вона містить $a_1, \dots, a_s$, де

$$g \cdot a_s^{-1} \cdot \dots \cdot a_{k+1}^{-1} = a_1 \cdot \dots \cdot a_k \in G_k$$

для $k = 1, \ldots, s-1$ та $a_1 \cdot \ldots \cdot a_s = g$. Наступний простий алгоритм знаходить факторизацію g щодо α. Алгоритм є детермінованим. Буде потрібно $O(r_n \cdot s_n)$ раундів, у кожному з яких необхідні одне множення, одне звернення й одне тестування на належність до підгрупи, виконувані за поліноміальний час $\mu_n,\ s_n,\ r_n$. Крім того, зрозуміло, якщо $\mu_n,\ s_n,\ r_n$ є поліноміальними в $\lceil log|G_n| \rceil$ та (α_n) є простим сімейством нормалізованих точно поперечних логарифмічних підписів, тоді існує ефективний тест на належність до підгрупи для всіх підгруп G_i групи G_n: $g \in G_i$, лише якщо у факторизації $g = a_1 \cdot \ldots \cdot a_s$ отримуємо результат $a_{i+1} = \ldots = a_s = e_G$. Можливо, непоперечні й навіть абсолютно непоперечні логарифмічні підписи є гарними варіантами, щоб бути складними. Проте, в [232] було доведено, що досить легко побудувати прості сімейства логарифмічних підписів для симетричних груп, що чергуються, які належать до класу TNT_{LS}. Отже, класи не мають критеріїв, щоб розрізняти прості й складні логарифмічні підписи. Якщо немає ефективного тесту на належність до підгрупи для груп G_n, у цих групах також будуть складні логарифмічні підписи, що є поперечними. Очевидно, що для груп, для яких є ефективний тест на належність до підгрупи, тоді точно поперечні логарифмічні підписи є простими. Для формалізації цього зауваження використовуємо наступне визначення.

Визначення 6. Для $n \in \mathbb{N}$ нехай P_n буде булевим предикатом, що можна застосовувати до будь-якого логарифмічного підпису групи G_n. Ми стверджуємо, що P є властивістю, яка породжує простоту для G_n, якщо всі сімейства $(\alpha_n)_{n\in\mathbb{N}}$ логарифмічних підписів (G_n), для яких правильне твердження: $P(\alpha_n)$ правдиве для всіх $n \in \mathbb{N}$, де n є простим.

Приклад 15.4. Переглянемо знову приклад 1. Для $n \in \mathbb{N}$ нехай P_n є «кожним блоком форми $[0, 2^i]$ для $0 \leq i \leq n''$». Тоді ми стверджуємо в прикладі 4, що P є властивістю породження простоти для $\mathbb{Z}_{2^n}$. У групах, для яких є ефективний тест на належність до підгрупи, належність до поперечних логарифмічних підписів також є властивістю породження простоти [233-237].

Висновки

Криптосистеми MST використовують як складні, так і прості логарифмічні підписи для побудови криптосистем відкритого ключа. Криптосистема MST_3 використовує прості логарифмічні підписи в елементарних абелевих 2-групах. Цікавою задачею є дослідження всіх неабелевих груп з великим порядком в їх функціональних полях та структури простих логарифмічних підписів, які можуть бути використані в таких групах. З огляду на результати конкурсів з побудови постквантових криптосистем, важливою задачею є квантовий криптоаналіз реалізацій криптосистем MST_3 та пошук вразливостей, пов'язаних з побудовою логарифмічних підписів та груп, що використовуються у якості платформ для криптосистем.

15.6 Властивості багатовимірного алгоритму RAINBOW та його здатність протистояти різноманітним методам криптоаналізу і атаці сторонніми каналами

Вступ

Багатовимірні квадратичні схеми є перспективним рішенням для потреби квантових систем, стійких до атак від квантового комп'ютера. Однак, оскільки цей клас відносно молодий і багато схем цього класу були порушені в минулому, існує дуже мало їх реалізацій, особливо на вбудованих мікроконтролерах. Щоб оцінити, чи можуть ці схеми колись замінити чинні стандарти, необхідно знати, наскільки ефективно їх можна впровадити на різних платформах. У розділі дано теоретичне введення до багатовимірних квадратичних схем. Потім розглядаються схеми, які певний час витримували атаки: Unbalanced Oil and Vinegar (UOV), Rainbow та enTTS. Особлива увага приділяється виявленню усіх загальних моментів схеми Rainbow.

Атаки на RAINBOW авторами розглянуті вище, див. розділ 13.

15.6.1 Загальні положення щодо схеми ЕП RAINBOW

Наразі криптосистеми, що засновані на квадратичних поліномах, пройшли за останні 10 років суттєвий розвиток та визнання. Теоретичною основою конструкцій Oil-Vinegar є доведена теорема, згідно з якою вирішення (визначення) набору багатоваріантних поліноміальних рівнянь над кінцевим полем є експоненційно складною проблемою, хоча це є у загальному випадку як необхідною, так і достатньою умовами [242]. Цей напрямок досліджень пов'язаний з появою конструкції Мацумото та Імаї, в тому числі, з використанням рівняння лінеаризації [77]. Далі Патарін та його співробітники доклали великих зусиль для розробки безпечних багатоваріантних криптосистем. Один з конкретних напрямків, яким займались Патарін та його співробітники, пов'язаний з рівняннями лінеаризації Dragon, Oil and Vinegar, Unbalanced Oil-Vinegar [77]. Побудова механізму ЕП Rainbow на основі Oil and Vinegar, Unbalanced Oil-Vinegar грунтується на тому, що певні квадратичні рівняння можна легко розв'язати, якщо є можливість вгадувати декілька варіантів [77].

Нехай k буде кінцевим полем. Ключовою конструкцією є відображення (карта) F від k^{o+v} до k^{o}:

$$F(x_1,...,x_0,x_1^{'},...,x_v^{'}) = F(x_1,...,x_0,x_1^{'},...,x_v^{'}),...,F_0(x_1,...,x_0,x_1^{'},...,x_v^{'})$$

і кожна F_l у формі:

$$F(x_1,...,x_0,x_1^{'},...,x_v^{'}) = \sum a_{l,i,j}x_i x_j + \sum b_{l,i,j}x_i^{'}x_j^{'} + \sum c_{l,i}x_i + \sum d_{l,i}x_j^{'} + c_l,$$

де x_i, $i = 1,...,o$ - значення **Oil** та $x_j^{'}$, $j = 1,...,v$ - значення Vinegar у кінцевому полі ***k***.

Потрібно звернути увагу на схожість наведеної вище формули з рівняннями лінеаризації. Такий тип поліномів називається "поліномом Oil-Vinegar". Причина, по якій вона називається схема "Oil-Vinegar", пов'язана з тим, що в квадратичному вимірі змінні Oil та Vinegar не змішуються повністю. Це дозволяє легко знайти одне рішення для будь-якого рівняння виду:

$$F(x_1,...,x_0,x_1^{'},...,x_v^{'}) = (y_1,...,y_o),$$

коли $(y_1,...,y_0)$ дано. Щоб знайти одне рішення, потрібно лише випадковим чином вибрати значення для Vinegar змінних та підключити їх до рівнянь вище, що дасть набір ***o*** лінійних рівнянь з ***o*** змінними. Це має, з імовірністю, близькою до 1, дати рішення. Якщо цього не сталося, можна спробувати ще раз, вибравши різні значення для Vinegar змінних, поки не вдасться знайти рішення [78].

Це сімейство криптосистем розроблено спеціально для схем підписів, де потрібно лише знайти одне рішення для даного набору рівнянь, а не унікальне рішення. Застосовуючи відображення (карту F), ми «приховуємо» її, складаючи її з лівої та правої сторін за двома оборотними афінними лінійними відображеннями L_1 та L_2. Оскільки L_1 знаходиться на k^o, а L_2 на k^{o+v}, це генерує квадратичне відображення (карту):

$$F^{-} = L_1 \circ F \circ L_2$$

від k^{o+v} до k^o.

Збалансована схема Oil-Vinegar характеризується тим, що $o=v$, але її удосконалили Кіпніс та Шамір, використовуючи матриці, що відносяться до білінійних форм, визначених квадратичними поліномами [243]. Для незбалансованої схеми Oil-Vinegar, $v>0$, показано, що конкретна атака має складність приблизно $q^{v-o-1}o^4$, коли $v \approx 0$. Це означає, що якщо ***o*** не надто велике (менше ніж 100) і дане фіксоване поле розміром ***q,*** тоді $v-o$ має бути досить великим, але також не надто великим, щоб забезпечити безпеку схеми. Однак слід зауважити, що в цій схемі документ, що підписується, є вектором у k^o, а підпис – вектором у k^{o+v}. Це означає, що підпис має принаймні вдвічі більший розмір документа, і при великому $v+o$ система стає менш ефективною.

В розділі пропонується конструкція, яка використовує конструкцію Oil-Vinegar кілька разів, так що в підсумку підпис буде лише трохи довшим за документ. Отже, ця схема набагато ефективніша. Її називають схемою Rainbow.

15.6.2 Застосування різноманітних методів криптоаналізу проти алгоритму RAINBOW

Представляється короткий криптоаналіз схеми підпису Rainbow, розглянувши його для наведеного вище прикладу. Є кілька способів атак, з якими будуть мати справу користувачі алгоритму. Для тих методів, де використовуються квадратні форми, слід пам'ятати, що теорія квадратних форм

над скінченними полями відрізняється, коли характеристика дорівнює 2, у порівнянні з випадком, коли характеристика є непарною [245]. Розглянемо наступні методи:

1. Метод зниження рангу

Метод зниження рангу використовується для розбиття схеми підпису біраціональної перестановки Шаміра. Причина, по якій ця атака може спрацювати, полягає в тому, що простір, що охоплюється поліноміальними компонентами шифру схеми Шаміра, складається з прапора пробілів:

$$V1 \subset V2 \subset \cdots \subset Vt,$$

де V_t – простір, охоплений поліноміальними компонентами шифру, кожна V_i є власною підмножиною V_{t+1}, а ранг відповідної білінійної форми, що відповідає елементам у $V_{t+1} - V_i$, занадто більший, ніж у V_i, а різниця розмірів між V_i та V_{t+1} рівно 1. Завдяки цим властивостям, зокрема, останньому, це дозволяє легко знайти цей прапор просторів, а саме всі V_i, спочатку знайшовши V_{n-1}, потім V_{n-2} і так далі шляхом зменшення рангу [241]. Але цей метод атаки вже не може працювати проти цієї схеми. Причиною цього є те, що в нашому випадку існує також такий прапор просторів, коли кількість компонентів – це точно кількість рівнів, розмірність кожного компонента прапора точно відповідає розміру V_{i+1}, $i = 1,...,u-1$, але різниця в розмірах останніх двох великих просторів – це точно $O_u - 1$, яка була обрана спеціально для досить великого числа 11, на відміну від випадку Шаміра, коли вона дорівнює 1. Властивість, наведена вище, якраз і є причиною того, що атака більше не може працювати. Тут не можна використовувати метод зниження рангу через те, що $O_u - 1 = 11$ і більше не 1. „Останній товстий рівень Oil" дозволяє схемі протистояти атаці зниження рангу [238].

2. Метод атаки на Oil-Vinegar схеми

Аналіз показав, що дія L_1 полягає у змішуванні всіх поліноміальних компонентів F. Отже, кожен компонент шифру *F* тепер належить до верхнього рівня поліномів Oil-Vinegar, а саме, всі вони є елементами P_4. Це багаточлени Oil-Vinegar з 22 змінними Vinegar та 11 змінними Oil [241]. Для цього випадку можна застосувати метод для незбалансованої схеми підпису Oil-Vinegar, щоб спробувати атакувати систему, що дозволить відокремити змінні верхнього шару Oil-Vinegar. Для цього нам потрібно розділити верхній (або кінцевий) рівень з 11 змінних Oil та 22 змінних Vinegar. Відповідно до криптоаналізу, складність атаки цього першого кроку становить $q^{22-11-1} \times 11^4 > 2^{90}$.

3. Метод Мінранка

Існує два абсолютно різних способи використання методу Мінранка.

Перший – пошук полінома, асоційована матриця якого має найнижчий ранг серед усіх можливих варіантів. Цей набір поліномів із 6 змінними Vinegar та 6 Oil належить до першого рівня, тобто, P_1 і позначався $\tilde{F}_1$. Для цього спочатку ми прив'язуємо до кожного полінома білінійну форму, яка має матрицю розміром 33×33. Потім ми можемо використовувати лінійні комбінації матриць, пов'язаних із компонентами F для виведення полінома, пов'язана з яким матриця має ранг 12 [243]. В цьому випадку, щоб атакувати систему, проблемою стає пошук матриці рангу 12 серед групи з 27 матриць розміром 33×33. З методу Мінранка ми знаємо, що складність пошуку такої матриці становить $q^{12} \times 27^3$, що набагато більше, ніж 2100. Інша можливість – це пошук поліномів, що відповідають поліномам у другому останньому рівні, а саме той, який належить P_3 і походить від лінійних комбінацій $\tilde{F}_i$, $i < 4$. У цьому випадку метод Мінранка однозначно не може бути використаний, оскільки він взагалі має ранг 22. Одним із шляхів, безсумнівно, є випадковий пошук. Оскільки розмірність P_3 дорівнює 16, це стає проблемою пошуку елемента в підпросторі розмірності 16 в загальному просторі розмірності 27. Отже, такий випадковий пошук потребує щонайменше q^{11} пошуків, щоб знайти його, але нам також потрібно визначити, чи дійсно рейтинг нижче 22 для кожного пошуку. У цьому випадку загальна складність повинна бути не менше $q^{11} \times (22 \times 33^2 / 3) > 2^{100}$. Ця ідея атаки насправді пов'язана з іншим методом атаки і наведений вище аргумент пояснює, чому цей метод більше не може працювати [77].

З останніх результатів електронного друку в цьому напрямку, де вивчається дуже загальна система, яка називається STS, ми знаємо, що їх метод може бути застосований і до нашого випадку. Відповідно до їх оцінки, безпека нашої системи становить принаймні $27 \times 33^3 \times (2^8)^{12} \times 5 > 2^{100}$.

4. Атака за допомогою структури багатошаровості

Для випадку криптосистеми Мацумото – Імаї Патарін показав, що, якщо шифр складається з декількох незалежних паралельних "гілок", можна виконати поділ змінних таким чином, що всі поліноми в шифрі виведені як лінійні комбінації поліномів над кожною групою змінних. Ця властивість насправді може бути використана для атаки на систему. На перший погляд, можна подумати, що рівні виглядають як різні «гілки». Тим не менше, слід усвідомити, що рівні жодним чином не є «незалежними», оскільки кожен з них будується на попередньому. Простіше кажучи, можна сказати, що всі рівні "злипаються" і ми ніяк не можемо зробити будь-якого розділення змінних. Це зрозуміло, коли розглядаються поліноми останнього рівня P_4. Тому атака з використанням властивості паралельних незалежних гілок ***y*** тут не може працювати. Подібним чином можна стверджувати, що атака з використанням систем також не може працювати тут, оскільки немає гілок і все насправді «склеєно» [242].

5. Загальні методи

Іншими методами, які можуть бути використані для атаки на нашу схему

підписів, є ті, які безпосередньо вирішують поліноміальні рівняння, наприклад, метод *XL* та різні його узагальнення, або такі, що використовують основи Гробнера. Безумовно, дуже складно вирішити набір з 27 рівнянь із 33 змінними, оскільки для цього набору рівнянь існує надто багато рішень. Загалом, набагато краще розв'язувати рівняння лише з однією змінною. Через характер проєктування системи можна здогадатися про значення для будь-якого набору змінних $v_1 = 6$, і ми маємо ймовірність $1 / e < 1 / 2.71828 < 0.37$ отримати унікальне рішення. Тепер задача стає проблемою вирішення набору з 27 квадратних рівнянь із 33 змінними. Ми повинні думати про це так, ніби це сукупність випадково вибраних квадратних рівнянь. Відповідно до того, що прийнято вважати для вирішення цього набору рівнянь, складність становить щонайменше $23 \times 27 > 281$.

З цього ми робимо висновок, що загальна складність атаки на нашу систему становить принаймні 280 [243].

6. Загальний аналіз безпеки

На основі цього можна побачити, що для атаки на систему можна підійти до неї або з верхнього рівня, або сформувати нижній рівень. Безпека нижнього рівня залежить від того, наскільки ефективно можна використовувати метод Minrank. Загалом складність атаки дорівнює $q^{(V_2-1)}O_u^3 - 1 \; if \; V_1 > O_1$, якщо $v_1 > o_1$, або $q^{2V_1} \; O_u^3 - 1$, якщо $v_1 \leq o_1$. З цього можна отримати, що не може бути $v_2 = o_1 + v_1$ занадто малим. З останніх результатів електронного друку [WBP] безпека системи становить, принаймні, $(n - V_1) \times n^3 \times q^{(O_1+V_1)} \times u$, що, безсумнівно, вимагає, щоб $o_1 + v_1$ не був малим. Що стосується випадку атаки зверху, метод атаки для незбалансованого методу Oil-Vinegar показує, що $v_u - 1 - o_u - 1$ не може бути занадто малим. Також, щоб уникнути випадкових атак пошуку, $o_u - 1$ не повинен бути занадто малим [78].

7. Здатність алгоритму RAINBOW протидіяти атаці сторонніми каналами

Криптографічні системи повинні бути захищені від широкого кола атак, включаючи атаки сторонніми каналами. Атака сторонніми каналами належить до фізичної атаки, яка являє собою будь-яку атаку, засновану на інформації, отриманій в результаті фізичної реалізації криптографічних систем, а не на грубій силі чи теоретичних недоліках криптографічних алгоритмів. Основним принципом атаки бічного каналу є те, що інформація бічного каналу, така, як споживання енергії, електромагнітні витоки, інформація про синхронізацію або навіть звук може забезпечити додаткові джерела інформації про секрети в криптографічних системах, наприклад, криптографічні ключі, часткова інформація про стан, повні або часткові звичайні тексти, які можна використовувати для розбиття криптографічних систем. Загальні класи атаки

бічних каналів включають аналіз синхронізації, аналіз потужності, електромагнітний аналіз, аналіз несправностей, акустичний криптоаналіз, аналіз залишків даних та атаки аналізу молоткових рядів.

Атаки аналізу несправностей мають на меті маніпулювати екологічними умовами криптографічних систем, таких, як напруга, годинник, температура, випромінювання, світло і вихровий струм, щоб генерувати несправності під час секретних обчислень, наприклад, множення та інверсії в кінцевому полі і спостерігати за пов'язаною поведінкою, яка може допомогти криптоаналітику зламати криптографічні системи. Атаки аналізу несправностей можна спроєктувати, просто підсвітивши транзистор лазерним променем, що змушує деякі біти приймати неправильні значення. Ідея використання несправності, індукованої під час секретного обчислення, для вгадування секретного ключа практично спостерігалася в реалізаціях RSA, що використовують китайську теорему про залишки.

Атака аналізу потужності може надати детальну інформацію, спостерігаючи за енергоспоживанням криптографічних систем, що приблизно поділяється на простий аналіз потужності (SPA) та аналіз диференціальної потужності (DPA). У сімействі атак аналізу потужності DPA представляє особливий інтерес і є статистичним тестом, який вивчає велику кількість сигналів енергоспоживання для отримання секретних ключів.

Можна виділити наступні атаки:

- атака диференціального аналізу потужності на SFLASH;
- атака на секретні ключі від модуля SHA-1 схем SFLASH.
- атака стороннього каналу на enTTS, яка використовує диференціальний аналіз потужності та аналіз несправностей для атаки двох афінних перетворень та центральної трансформації карти. Цей метод показує, що можна отримати всі секретні ключі enTTS.

Оскільки конструкція Rainbow включає два афінні перетворення та перетворення центральної карти, такі методи мають потенціал для отримання її секретних ключів. Таким чином, обговорюється захист від можливої атаки бічного каналу для Rainbow, а контрзаходи описані нижче:

- Нехай це повідомлення і кожен елемент $\boldsymbol{y} \in GF((2^4)^2)$;
- Береться випадковий вектор $y'(y_0', y_1', ..., y_{25}')$, кожен елемент якого належить $GF((2^4)^2)$;
- Обчислюється $y'' = y' + y$;
- Обчислюється $\bar{y}' = Ay' + b$ та $\bar{y}'' = Ay''$, де A – матриця 26x26, b – вектор розміру 26;
- Обчислюється $\bar{y} = \bar{y}' + \bar{y}''$, що еквівалентно $\bar{y} = Ay + b$;
- Розраховується перше афінне перетворення; тоді ми беремо випадкові байти для Vinegar-змінних;
- Двічі перевіряються випадкові байти для захисту від атак аналізу несправностей;

- Обчислюються багатовимірні поліноміальні оцінки та розв'язування систем лінійних рівнянь до завершення перетворення центральної карти;
- $\bar{x}(\bar{x}_0, \bar{x}_1, ..., \bar{x}_{42})$ – це результат трансформації центральної карти; після цього береться два випадкових вектори $\bar{x}'$ та $\bar{x}''$, де $\bar{x} = \bar{x}' + \bar{x}''$ та елементи належать $GF((2^4)^2)$;
- Обчислюється $\bar{x}' = C\bar{x}'$ та $\bar{x}'' = C\bar{x}'' + d$, де C – матриця 43x43, b – вектор розміру 43;
- Обчислюється $\bar{x} = \bar{x}' + \bar{x}''$, що еквівалентно $x = Cx + d$;
- $x(x_0, x_1, ..., x_{42})$ - це схема підпису Rainbow для $y(y_0, y_1, ..., y_{25})$.

Використовується аналіз несправностей для атаки випадкових байтів у центральних перетвореннях карти; таким чином, ми двічі перевіряємо випадкові байти для захисту від атак аналізу несправностей. Також використовується аналіз диференціальної потужності для атаки модуля SHA-1; таким чином, ми беремо метод захисту афінних перетворень. Однак зазначений вище контрзахід є теоретичним; потрібна можливість впровадити та перевірити це на апаратному забезпеченні.

Висновки

1. Постквантова криптографія – частина криптографії, яка залишається актуальною і при появі квантових комп'ютерів і квантових атак. Так як за швидкістю обчислення традиційних криптографічних алгоритмів квантові комп'ютери значно перевершують класичні комп'ютерні архітектури, сучасні криптографічні системи стають потенційно вразливими до криптографічних атак. Більшість традиційних криптосистем спирається на проблеми факторизації цілих чисел або на задачі дискретного логарифмування, які будуть легко розв’язані на досить великих квантових комп’ютерах, що використовують алгоритм Шора.

2. Багато криптографів ведуть розробку алгоритмів, незалежних від квантових обчислень, тобто, стійких до квантових атак. Ці задачі розглянуті на другому етапі конкурсу NIST США.

3. Схема підпису Rainbow виглядає надійною проти великої кількості методів криптоаналізу та проти атак сторонніми каналами.

4. У зв’язку з можливістю появи потужного квантового комп’ютера актуальними є завдання створення постквантових алгоритмів ЕП. В цьому напрямі уже розпочато дослідження, в певній мірі визначено математичні основи, на яких можуть бути побудовані постквантові алгоритми ЕП. Для цього можна застосувати схему Rainbow.

5. Реалізація квантово-захищених алгоритмів вимагає великих матеріально-технічних ресурсів. Вказане пов’язано з великими довжинами ключів та загальних параметрів. Сучасний рівень розвитку техніки дозволяє оптимістично ставитися до можливості ефективної реалізації квантово-захищених алгоритмів.

6. Мультиваріативні квадратичні перетворення можуть бути застосованими для розроблення постквантового стандарту ЕП. Вони вже були використані для побудови схем підпису, але всі спроби побудувати надійну схему поки не були успішними. Попередній аналіз показав, що мультиваріативні квадратичні перетворення можуть вирішити проблему захищеності від атак на основі квантових комп'ютерів, але для цього ще потрібно провести величезний обсяг досліджень та робіт, а також вкласти значні ресурси.

7. Попередній аналіз показує, що розміри загальних параметрів та ключів не викликають сумнівів відносно криптографічної стійкості стандарту, розробленого на основі мультиваріативного квадратичного перетворення. Але залишається проблема просторової складності, яка пов'язана зі значними довжинами загальних параметрів та відкритих ключів.

15.7 Постквантовий алгоритм інкапсуляції ключів CLASSIC MCELIECE

Вступ

У 2016 р. Національний інститут стандартів і технологій США (The National Institute of Standards and Technology, NIST) оголосив конкурс, метою якого було вибрати алгоритми постквантової криптографії (англ. Post-Quantum Cryptography, PQC). Серед алгоритмів шифрування з відкритим ключем і алгоритмів інкапсуляції ключів були представлені як фіналісти третього раунду: Classic McEliece [241], CRYSTALS-KYBER, NTRU, SABER. Як альтернативні фіналісти в цій же групі алгоритмів були обрані: BIKE [253], FrodoKEM, HQC [254], NTRU Prime, SIKE.

Ця версія алгоритму Classic McEliece була представлена на третьому раунді конкурсу постквантової криптографії 10 жовтня 2020 р. міжнародною групою розробників з Великобританії, США, Швейцарії, Нідерландів, Данії, Німеччини [67] .

Алгоритм Classic McEliece є варіацією алгоритму на основі кодів, запропонованого Робертом Мак-Елісом (англ. Robert J. McEliece). Перша криптосистема з відкритим ключем на основі коду була представлена в 1978 математиком і інженером в Каліфорнійському технологічному інституті Робертом Мак-Елісом [118]. У цій системі відкритий ключ визначається випадковим двійковим кодом Гоппи [159,246-248] (англ. Binary Goppa code). Шифротекст є кодовим словом, об'єднаним із деяким вектором помилок. Особистий ключ забезпечує ефективне декодування: вилучення кодового слова із зашифрованого тексту, виявлення та видалення помилок. Система Мак-Еліса була розроблена як одностороння, що означає, що зловмисник не може ефективно знайти кодове слово із зашифрованого тексту та відкритого ключа, коли кодове слово вибирається випадково. Рівень безпеки системи Мак-Еліса залишався напрочуд стабільним, незважаючи на десятки документів про атаки за 40 років. Первинні параметри криптосистеми Мак-Еліса були розроблені тільки для безпеки (2^{64}), але система легко масштабується до «надлишкових»

параметрів, які забезпечують достатній запас безпеки в порівнянні з досягненнями в комп'ютерних технологіях, включаючи квантові комп'ютери.

Систему Мак-Еліса за часом існування можна порівняти з алгоритмом RSA [67,140,246–251]. Як відомо, алгоритм RSA не має необхідної стійкості проти постквантового криптоаналізу, тоді як криптосистема Мак-Еліса повинна забезпечувати таку стійкість.

Система Мак-Еліса вимагала величезної кількості додаткових доробок через роки. Деякі з них підвищують ефективність при збереженні безпеки: це включає «подвійне» шифрування з відкритим ключем (англ. public-key encryption, PKE), запропоноване Нідеррайтером (англ. Harald G. Niederreiter), прискорення програмного забезпечення та апаратних складових системи.

Крім того, тепер відомо [67,159,246–251], як ефективно перетворити OW-CPA алгоритми шифрування з відкритим ключем в алгоритми інкапсуляції ключів (англ. key encapsulation mechanisms, KEM), що захищені від атак типу IND-CCA2 на основі моделі случайного оракула (англ. Random oracle model, ROM). Це перетворення є жорстким, зберігаючи рівень безпеки, при двох припущеннях, яким задовольняє криптосистема Мак-Еліса: по-перше, шифрування з відкритим ключем є детермінованим (тобто, дешифрування відновлює всю використану випадковість); по-друге, алгоритм немає помилок дешифрування дійсних зашифрованих текстів. Більш того, внесені авторами модифікації класичного алгоритму забезпечують аналогічний степінь захисту для ширшого класу атак, а саме – атак на основі моделі квантового випадкового оракула (англ. quantum random oracle model, QROM). Ризик того, що атака, специфічна для хеш-функції, може бути швидше, ніж атака ROM або QROM, усувається стандартною практикою вибору добре вивченої "неструктурованої" хеш-функції з високим рівнем безпеки.

Classic McEliece [241] - це варіація алгоритму Мак-Еліса, яка об'єднує всі найкращі напрацювання попередніх років воєдино. Вона є алгоритмом інкапсуляції ключів із забезпеченням безпеки проти IND-CCA2 на дуже високому рівні, навіть проти квантових обчислень. Алгоритм інкапсуляції ключів побудований консервативно на основі шифрування на відкритому ключі, що дозволяє забезпечувати безпеку OW-CPA. А саме, заснований на подвійній версії алгоритму шифрування на відкритому ключі Мак-Еліса, створеній Нідеррайтером із використанням двійкових кодів Гоппи. Кожен рівень конструкції розроблений так, щоб системи, в яких було впроваджено даний алгоритм, з високою ймовірністю забезпечували довгострокову безпеку в постквантовий період.

Інша робота включає варіації алгоритму Мак-Еліса, безпека яких ще не була вивчена досить ретельно. Наприклад, у багатьох запропонованих варіаціях двійкові коди Гоппи замінюються іншими сімействами кодів, а криптографія на основі решіток замінює поняття «кодове слово плюс випадкові помилки» на «точку решітки плюс випадкові помилки».

Криптографія на основі кодів і криптографія на основі решіток є двома основними типами кандидатів, зазначених у вимогах NIST [248, 249] щодо стандартизації постквантових алгоритмів. У цьому документі основна увага

приділяється точності класичної системи Мак-Еліса, оскільки вона була ретельно вивчена.

1. Базові визначення

1.1 Криптографія з відкритим ключем

Криптосистема з відкритим ключем [67,159,246–251] використовує односторонню функцію з таємним входом *E*, яка буде служити функцією шифрування. Обчислення зворотної E^{-1} , що називається функцією дешифрування, можливо тільки, якщо відомий секрет *K*. Ця концепція односторонньої функції формує основу криптографії з відкритим ключем, у якій секретним ключем є *K*, а відкритим ключем – *E*.

Точніше, криптосистема шифрування з відкритим ключем має забезпечувати три алгоритми: *KeyGen*, *Encrypt* і *Decrypt*. *KeyGen* – це ймовірнісний алгоритм з поліноміальним часом, який на вході отримує $\{0,1\}^k$, де $k \geq 0$ – параметр безпеки, а на виході формує пару відкритого/особистого ключа (*pk*,*sk*). *KeyGen* також визначає кінцевий простір повідомлень M_{pk}. *Алгоритм Encrypt* – це ймовірнісний алгоритм з поліноміальним часом, який із вхідних параметрів $\{0,1\}^k$, відкритого ключа *pk* та повідомлення $x \in M_{pk}$ формує на виході шифротекст *c*. *Алгоритм Decrypt* - це детермінований алгоритм з поліноміальним часом, який на основі вхідних даних $\{0,1\}^k$, *sk* і шифротексту *c* відновлює повідомлення *x*. Криптосистема повинна задовольняти властивості правильності, що означає, що дешифрування має однозначно відновлювати зашифроване повідомлення.

1.2 Двійковий код Гоппи

Двійковий код Гоппи [67,159,246–251], що позначається $\Gamma(g(z), L)$, де $g(z)$– породжуючий поліном Гоппи степеня *t* над розширенням поля $GF(2^m)$, *m* - порядок розширення, а *L* – діапазон значень коду, де $L \subseteq GF\{2^m\}$.

Представимо двійковий код Гоппи у наступному вигляді:

$$g(z) = \sum_{i=0}^{t} g_i z^i = g_0 + g_1 z + \ldots + g_t z^t \ ,$$

$$L = \left\{ \forall \alpha_i \in GF\left(2^m\right) \setminus g(\alpha_i) = 0 \right\}$$

з вектором c над $GF(2)$ таким, що $c = (c_1, c_2, \ldots, c_n)$ і

$$R_e(z) = \sum_{i=1}^{t} \frac{c_i}{z - \alpha_i} \ .$$

Якщо $n = 2^m$– довжина кодового слова, що обмежена діапазоном *L*, *k* – розмірність, що обмежена $k > (n - mt)$, та мінімальна відстань $d \geq (2t + 1)$, тоді $[n, k, d]$ представляє параметри коду Гоппи $\Gamma(g(z), L)$.

Для двійкового коду Гоппи будь-який згенерований багаточлен $g(z)$називається розділеним, якщо багаточлен не має коренів з кратністю, більше одиниці (тобто, не має коренів, що повторюються). Коди Гоппи, що складаються

з таких поліномів, називаються роздільними двійковими кодами Гоппи. Для таких кодів існує хоча б одне значення a, при якому $g(\alpha_i) = 0$.

Для двійкового коду Гоппи будь-який згенерований багаточлен $g(z)$ називається неприведеним, якщо для $\forall\alpha_i$ справедливо $g(\alpha_i) \neq 0$. Коди Гоппи, що складаються з таких поліномів, називаються двійковими кодами Гоппи, що не приводяться.

Обидва типи мають можливість виправляти максимальну кількість помилок у порівнянні з іншими сімействами кодів.

Мінімальна відстань d коду Гоппи повинна бути більшою за $d \geq (2t + 1)$. Виправляюча здатність коду оцінюється параметром t.

Породжувальна матриця G двійкового коду Гоппи використовується для кодування та декодування повідомлення, а контрольна сума c з перевіркою на парність важлива для виявлення і виправлення помилок.

Породжувальна матриця G отримується з матриці перевірки на парність H, простір рядків G – це вектори нульового простору H за модулем 2, такі, що $GH^T = 0$. Матриця перевірки на парність H визначається як:

$$H_{di} = \sum_{\alpha_i \in L} \sum_{k=1}^{t-(d-1)} g_{t-(k-1)} \alpha_i^{t-d+(1-k)} h_i ,$$

де $1 \leq d \leq t$. Причому повинна задовольнятися умова $cH^T = 0$, де c – шифротекст.

Кодування повідомлення у двійковому коді Гоппи [248].

Повідомлення можна закодувати, розбивши його на блоки довжиною k біт і помноживши кожен блок на породжуючу матрицю G:

$$(m_1, m_2, \dots, m_k) \cdot G = (c_1, c_2, \dots, c_k).$$

Отримане повідомлення (y)матиме вигляд:

$$y = mG + e. \qquad (15.56)$$

Алгоритм виправлення помилок ненаведеного коду Гоппи [248].

1) Розрахувати синдром $S(z)$:

$$S(z) = \sum_{i=1}^{n} \frac{y_i}{z - \alpha_i} .$$

2) Обчислити $\sigma(z)$, як показано нижче:

- Використовувати розширений алгоритм Евкліда для визначення $h(z)$ такого, що $\sigma(z)h(z) \equiv 1 \pmod{g(z)}$.

Якщо $h(z) = z$, то процес завершено і $\sigma(z) = z$, інакше продовжити:

- Обчислити $d(z)$так, щоб $d^2(z) \equiv h(z) + z \pmod{g(z)}$.

- Визначити $a(z)$ і $b(z)$ таке, що $d(z)b(z) \equiv a(z) \pmod{g(z)}$.

- Обчислити $\sigma(z) = a^2(z) + zb(z)$.

3) Розрахувати набір місць розташування помилки: $B = \{i|\sigma(\alpha_i) = 0\}$.

4) Вектор помилок $e = (e_1, e_2, \ldots, e_n)$ визначається, як $e_i = 1$ для $i \in B$ і нулями в іншому випадку.

5) Сформований шифротекст *c* має вигляд

$$c = y - e.$$

Розшифрування повідомлення [249].

Коли помилки виправлені, повідомлення може бути декодовано; закодоване повідомлення в рівнянні (15.56) можна подати у вигляді матриці:

$$G^T \cdot \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}.$$

Для обчислення повідомлення можна застосувати метод Гауса, щоб видалити породжуючу матрицю $\lceil n/8 \rceil$:

$$\left(G^T \middle| \begin{matrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{matrix} \right) \ldots \left(\begin{matrix} I_k \end{matrix} \middle| \begin{matrix} m_1 \\ \vdots \\ m_k \end{matrix} \right), \quad P$$

де I_k- це одинична матриця *c* розміром $k \times k$, а p - це матриця розміром $(n-k) \times (k+1)$.

2. Властивість нерозрізненості шифротексту

До криптосистем шифрування та інкапсуляції ключів з відкритим ключем висуваються такі вимоги [159, 246, 247, 248, 249]:

- *Семантична безпека* (також відома як поліноміальна безпека / нерозрізненість шифрів): якщо зловмисник має деяку інформацію про відкритий текст, він не може отримати на основі цих знань додаткові дані про параметри криптосистеми із шифротексту. Обчислювально неможливо розрізнити два повідомлення, одне з яких було зашифровано:

- *Було зашифровано.* Це означає, що схеми шифрування повинні бути імовірнісними;

- *Відсутність гнучкості.* З будь-якого заданого зашифрованого тексту зловмисник не може отримати новий зашифрований текст таким чином, щоб відкриті тексти, що лежать в основі двох зашифрованих текстів, були пов'язані. З іншого боку, зловмисник може використовувати безліч видів атак: він може просто отримати доступ до загальнодоступних даних, а потім зашифрувати будь-який відкритий текст за своїм вибором (атаки на основі підібраного відкритого тексту) або, крім того, використовувати алгоритм дешифрування (атаки на основі неадаптивно / адаптивно підібраного шифротексту).

Криптосистема може бути семантично захищеною від атак з підібраним відкритим текстом або навіть атак з неадаптивним підібраним зашифрованим текстом (CCA1), при цьому залишаючись гнучкою. Однак, захист від атак з

адаптивно підібраним шифротекстом (CCA2) еквівалентний відсутності гнучкості.

Властивість нерозрізненості шифротексту визначає криптостійкість алгоритму до атаки на основі підібраного відкритого тексту. Забезпечення такої властивості нерозрізненості на основі відкритого тексту (IND-CPA) вважається основною вимогою для більшості доказово захищених криптосистем з відкритим ключем [159,246,247,248,249], хоча деякі схеми також забезпечують криптографічну стійкість проти атак на основі підібраного шифротексту та адаптивних атак на основі підібраного шифротексту. Такі властивості нерозрізненості позначаються як IND-CCA1 і IND-CCA2 відповідно [159, 246, 247, 248, 249].

Для оцінки рівня криптостійкості в класичній і постквантовій криптографії для кожного алгоритму та його варіації використовується рівень криптостійкості, що забезпечується, і відповідні основні параметри перетворень. При описі вимог до алгоритмів, що подаються на конкурс, було визначено три рівні криптографічної стійкості [248,249]:

- Рівень 1: будь-яка атака, яка зламує IND-CCA-стійкий алгоритм, повинна вимагати обчислювальних ресурсів, порівнянних або перевищуючих необхідні для пошуку ключа на блоковому шифрі зі 128-бітним ключем (наприклад, AES-128);

- Рівень 3: якщо існує атака на IND-CCA-криптостійкий алгоритм, то для проведення такої атаки повинні забезпечуватися обчислювальні ресурси, які порівнянні або перевищують необхідні для пошуку ключа на блоковому шифрі з 192-бітним ключем (наприклад, AES-192);

- Рівень 5: будь-яка атака, яка порушує криптостійкість IND-CCA-стійкої схеми, повинна вимагати обчислювальних ресурсів, які порівнянні або перевищують необхідні для пошуку ключа на блоковому шифрі з 256 - бітним ключем (AES-256).

3. Класична та квантова атаки

За останні кілька років, після численних атак, доведена безпека стала представляти найбільший інтерес. Такий доказ проводиться в рамках теорії складності: робляться спроби поліноміально звести добре встановлену проблему до атаки. Отже, ефективний зловмисник не зможе вирішити складну проблему: інакше це призведе до суперечності. Дуже мало доведено схем із використанням тільки таких поліноміальних редукцій без будь-яких інших припущень. Крім того, вони насилу досягають ефективності. В останні роки так звана «випадкова модель оракула» активізувала дослідження, надавши цікавий інструмент для підтвердження безпеки дуже ефективних схем. Справді, ця модель, де ідеалізовані деякі конкретні криптографічні об'єкти, а саме хеш-функції, які вважаються дійсно випадковими, допомогли забезпечити докази безпеки для багатьох схем шифрування та схем цифрового підпису.

Криптографічна хеш-функція - це функція $h:\{0,1\}^m \to \{0,1\}^n$ (де зазвичай $n < m$), яку важко проаналізувати. Інтуїтивно зрозуміло, що бажано, щоб єдиний спосіб дізнатися інформацію про $h(x)$ будь-якого конкретного $x \in \{0,1\}^n$, це

фактично обчислити $h(x)$ і для кожного *x*, для якого h ще не обчислено, $h(x)$ повинен виглядати як нове випадкове значення у $\{0,1\}^n$.

Модель випадкового оракула - це спосіб формального моделювання цих інтуїтивних вимог при аналізі безпеки криптографічних схем (таких як схеми шифрування або цифрового підпису), які використовують криптографічну геш-функцію h як складову частину внутрішньої структури. У моделі випадкового оракула геш-функції h моделюється оракул, який необхідно запитати про $x \in \{0,1\}^m$, щоб об'єкт дізнався геш-значення $h(x)$, а випадкова відповідь оракула $h(x)$ відноситься до рівномірно випадкової функції $h:\{0,1\}^m \to \{0,1\}^n$.

Ця модель є дуже корисною для аналізу безпеки криптографічних схем. Наприклад, це дозволяє вести облік обчислювальних можливостей зловмисників, і тому можна точно визначити хеш-значення, які доступні зловмиснику. Тоді є ймовірність, що криптографічна схема, яка, як доведено, безпечна в моделі випадкового оракула, буде безпечною в принципі.

Виявляється, що модель випадкового оракула необхідно переглянути в контексті квантових атак, тобто, коли розглядаються зловмисники, які мають у своєму розпорядженні необмежені можливості проведення квантових обчислень. Така гіпотетична можливість дозволила б зловмиснику обчислити суперпозицію хеш-функції h за декількома *x*, тобто, він міг би створювати стани, які залежать від $\sum_x \alpha_x |x\rangle|h(x)\rangle$ для різних *x*, обчислюючи h тільки один раз. Однак можливості зловмисника щодо квантового криптоаналізу обмежені рядом теорем про заборону (наприклад, теорема про заборону клонування).

4. Криптосистема McEliece (1978)

Використовуючи той факт, що алгоритм швидкого декодування існує для загального коду Гоппи, у той час як його не існує для загального лінійного коду, в 1978 р. Робертом Мак-Елісом [118] було запропоновано створити криптосистему з відкритим ключем, яка була б досить безпечною, але водночас забезпечувала надзвичайно високу швидкість передачі. На думку автора, така криптосистема ідеально підходить для використання в розрахованих на багато користувачів комунікаційних мережах, таких як ті, які передбачені НАСА для поширення даних, отриманих з космосу, для передачі даних по незахищених каналах передачі даних.

Криптосистема Мак-Еліса - це асиметрична криптосистема з відкритим ключем, заснована на теорії алгебраїчного кодування. Криптосистема Мак-Еліса використовує неприводний двійковий код Гоппи, який досі вважався криптостійким, особливо з параметрами [1024, 524, 101], запропонованими Мак-Елісом. Криптосистема з таким набором параметрів страждає від великої матриці відкритих ключів, що ускладнює її застосування.

Криптосистема Мак-Еліса - одна з найбільш перспективних криптосистем з відкритим ключем, яка здатна протистояти атакам на основі квантових обчислень. Фактично, на відміну від криптосистем, що використовують цілочисельну факторизацію або дискретні логарифми, криптостійкість даного алгоритму покладається на складність декодування повних лінійних кодів (загальна задача декодування є NP-складною).

Після того, як Діффі та Хеллман ввели поняття криптосистеми з відкритим ключем, в якій безпека зв'язку досягається без необхідності періодичного розповсюдження особистого ключа відправнику та одержувачу, ця властивість робить такі системи ідеально підходящими для використання в розрахованих на багатокористувацьких мережах зв'язку.

Криптосистема Мак-Еліса складається з трьох основних алгоритмів:

– випадкової генерації відкритого та особистого ключів;

– випадкового шифрування відкритого тексту;

– детермінованого алгоритму розшифрування шифротексту.

Загальносистемними параметрами для всіх алгоритмів є довжина коду n бітової послідовності відкритого тексту, що виправляє здатність коду t.

Кожному непривідному багаточлену степеня t над $GF(2^m)$ відповідає двійковий код Гоппи довжини $n = 2^m$, розмірності $k \geq n - tm$, який здатний виправити будь-яку комбінацію t помилок чи менше.

Більше того, існує швидкий алгоритм декодування цих кодів (алгоритм, запропонований Паттерсоном, час роботи $O(nt)$).

Припустимо, що спочатку в системі вже обрані загальносистемні параметри, а саме n і t, а також випадковим чином обраний поліном степеня t над $GF(2^m)$. Оскільки ймовірність того, що випадково обраний багаточлен степеня t є неприводним, становить близько $\frac{1}{t}$ і оскільки існує швидкий алгоритм перевірки неприводимості, запропонований Берлекампом, цей вибір буде легко зробити.

Генерація ключів

Вхідні параметри:

відсутні.

Вихідні параметри:

відкритий ключ (G', t),

особистий ключ (S, G, P).

Алгоритм:

1) Сформувати $k \times n$ породжуючу матрицю G для коду, яка може бути у канонічній (наприклад, наведеного ступінчастого вигляду) формі.

2) Сформувати випадкову $k \times k$ невироджену матрицю S.

3) Сформувати випадкову $n \times n$ матрицю перестановок P.

4) Обчислити матрицю G' розміром $k \times n$: $G' = SGP$, яка генеруватиме лінійний код з тією ж швидкістю та мінімальною відстанню, що і матриця G. Матриця G' буде загальнодоступною породжуючою матрицею.

5) Сформувати набори параметрів в якості відкритого ключа (G', t) та особистого ключа (S, G, P).

Схематично цей алгоритм представлений на рис. 15.7.

Шифрування відкритого тексту

Вхідні параметри:

Відкритий текст m, загальнодоступна породжуюча матриця G'.

Вихідні параметри:

Шифротекст c.

Алгоритм:

1) Представити відкритий текст m у вигляді бітових послідовностей довжини k.
2) Сформувати так званий вектор помилок - випадковий вектор z довжини n та ваги t.
3) Обчислити шифротекст $c = mG' + z$.

Схематично цей алгоритм представлений на рис. 15.8.

Алгоритм розшифрування шифротексту

Вхідні параметри:

шифротекст c'.

Вихідні параметри:

відкритий текст m.

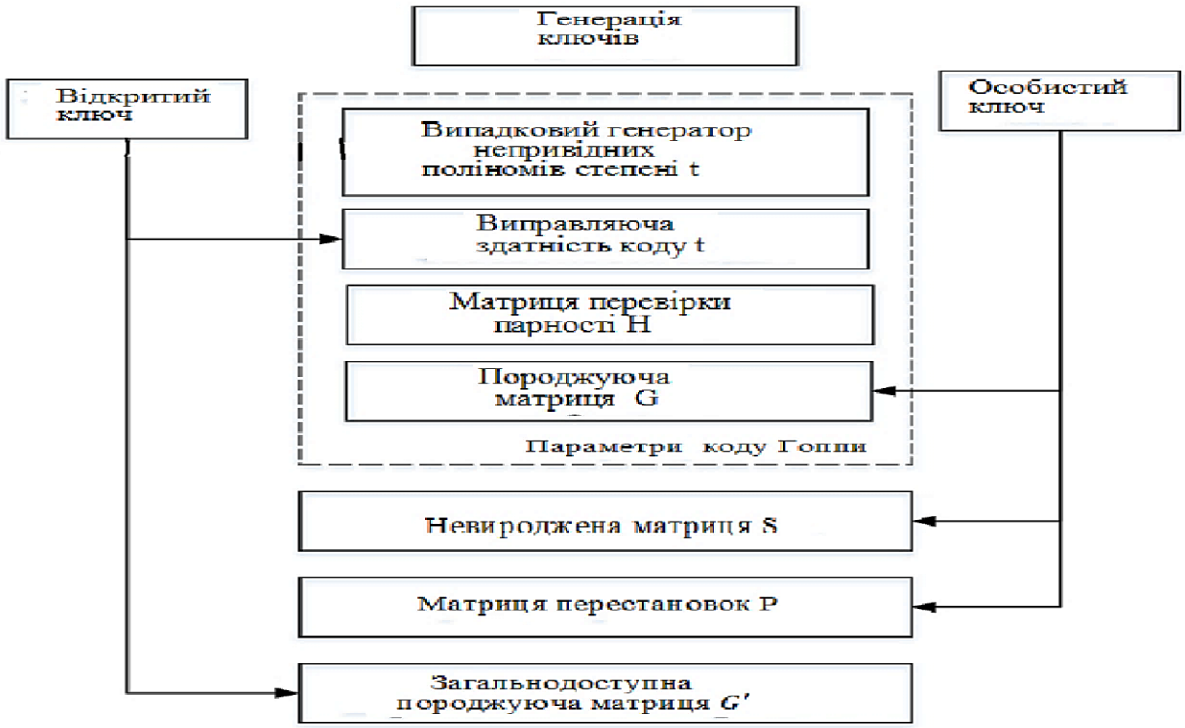


Рис. 15.7. Схема алгоритму випадкової генерації відкритого та особистого ключів

Алгоритм:

1) Обчислити зворотну матрицю перестановок P^{-1}.
2) Обчислити $c' = cP^{-1}$.
3) Використовувати алгоритм декодування кодів Гоппи для відновлення m' з отриманого c'.
4) Обчислити обернену матрицю S^{-1}, яка використовується в алгоритмі Паттерсона.
5) Обчислити відкритий текст $m = m'S^{-1}$.

Схематично цей алгоритм представлений на рис. 15.9.

Автором [248] доведено коректність системи, тобто, правильність відновлення відкритого тексту після розшифрування $Decrypt\big(Encrypt(m)\big) = m$. Сформований шифротекст має вигляд:

$$c = mG' + z = mSGP + z.$$

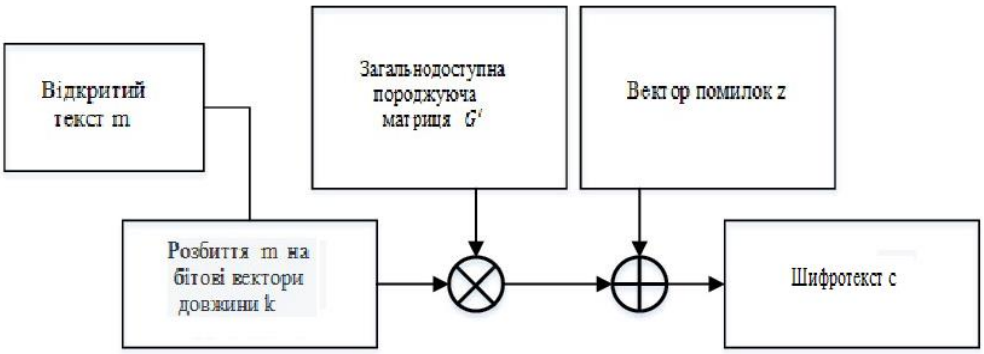


Рис. 15.8. Схема алгоритму шифрування відкритого тексту



Рис. 15.9. Схема алгоритму розшифрування шифротексту S' c' m'

При розшифровуванні формується

$$c' = cP^{-1} = mSG + zP^{-1}.$$

Як зазначалося раніше, z має вагу t, відповідно вага добутку zP^{-1} буде не більше t.

Отже, за допомогою коду Гоппи буде виправлено t помилок. Відстань Хеммінга для складових шифротексту не перевищує t , тобто, $d(mSG, cP^{-1}) \leq t$, що доводить відповідність вихідного і розшифрованого текстів $m = m'S^{-1} = mSS^{-1}$.

5. Загальна специфікація алгоритму Classic McEliece

5.1 Позначення та параметри

Загальносистемні параметри

n - Довжина коду, $n \leq q, n \in N$.

k - Розмір коду, $k = n - mt$.

t - Гарантується можливість корекції помилок $t \geq 2, mt < n$.

q - Розмір кінцевого поля, що використовується, $m - log_2 q$, $m > 0, m \in F_q$.

μ - Невід'ємне ціле число $\nu \geq \mu \geq 0, \nu \leq k + \mu$.

ν - Невід'ємне ціле число.

$f(z)$- Неприводимий поліном степеня m, $f(z) \in F_2[z]$. Це визначає представлення $F_2[z]/f(z)$ поля F_q.

$F(y)$- Неприводимий поліном степеня t, $F(y) \in F_q[y]$. Це визначає представлення $F_q[y]/F(y)$ поля $F_{q^t} = F_{2^{mt}}$.

Параметри симетричної криптографії

H - Криптографічна хеш-функція.

l - Довжина виходу криптографічної хеш-функції, $l > 0$.

σ_1 - Невід'ємне ціле число, $\sigma_1 \geq m$.

σ_2 - Невід'ємне ціле число, $\sigma_2 \geq 2m$.

G - Генератор псевдовипадкових бітів, який відображає рядок із l бітів у рядок із $(n + \sigma_2 q + \sigma_1 t + l)$ бітів.

Параметри особистого ключа

g - Багаточлен у полі $F_q[x]$.

α_i- Елемент кінцевого поля.

$\Gamma - (g, \alpha_1, \dots, \alpha_n)$.

s - Бітовий рядок довжини.

Параметри відкритого ключа

T - Матриця з $(n-k) \times k$ елементів над полем F_2.

Параметри шифротексту

e - Бітовий рядок довжини n та ваги Хеммінга t.

C - Зашифрований текст, який містить сеансовий ключ.

C_0 - Бітовий рядок довжини $n-k$.

C_1 - Бітовий рядок довжини l.

Елементи поля F_2^n, такі, як кодові слова та вектори помилок, завжди розглядаються як вектори-стовпці.

5.2 Вибір типу та розміру параметрів

Вибір параметрів симетричної криптографії:

– l - бітовий рядок $H(x)$ визначається як перші l бітів виведення $SHAKE256(x)$. Байтові рядки тут розглядаються як бітові рядки з прямим порядком байтів.

– Довжина бітового рядка $l = 256$.

– Ціле число дорівнює 16 (У всіх наборах параметрів $m \leq 16$, тому $\sigma_1 \geq m$).

– Ціле число σ_2 , що дорівнює 32.

– $(n + \sigma_2 q + \sigma_1 t + l)$ бітовий рядок $G(\delta)$ визначається, як перші $(n + \sigma_2 q + \sigma_1 t + l)$ виведення $SHAKE256(64, \delta)$. Тут $64, \delta$ означає 33-байтовий рядок, який починається з байта 64 і продовжується δ.

Всі входи $H(x)$, що використовуються в криптосистемі Classic McEliece, починаються з байта 0, 1 або 2 і, таким чином, не перекривають входи $SHAKE256(x)$, що використовуються в G.

5.3 Представлення параметрів у вигляді байтових рядків

Вектори над полем F_2

Якщо r кратно 8, то r - бітовий вектор $v = (v_0, v_1, \dots, v_{r-1}) \in F_2^r$ представляється у вигляді наступної послідовності з $r/8$ байтів:

$$\begin{pmatrix} v_0 + 2v_1 + 4v_2 + \cdots + 128v_7, v_8 + 2v_9 + 4v_{10} + \cdots \\ +128v_{15}, v_{r-8} + 2v_{r-7} + 4v_{r-6} + \cdots + 128v_{r-1} \end{pmatrix}.$$

Якщо r не кратно 8, то r - бітовий вектор $v = (v_0, v_1, \dots, v_{r-1}) \in F_2^r$ доповнюється 0 праворуч до довжини між $r+1$ і $r+7$, в такому випадку r стане кратно 8.

Сеансові ключі. Сеансовий ключ K, як елемент поля F_2^l, представляється $\lceil l/8 \rceil$ байтовим рядком.

Шифротекст. Шифротекст C складається із двох компонентів: $C_0 \in F_2^{n-k}$ і $C_1 \in F_2^l$. Шифротекст представлений як конкатенація $\lceil mt/8 \rceil$ - байтового рядка, який представляє C_0 і $\lceil l/8 \rceil$- байтового рядка, що представляє C_1.

Хеш-входи. В алгоритмі присутні три типи хеш-входів: $(2, v)$, $(1, v, C)$ і $(0, v, C)$. Тут $v \in F_2^n$ і C представляє шифротекст. Перші 0, 1 і 2 параметри представляються як байти. Вектор v представляє собою наступні $\lceil n/8 \rceil$ байт. Якщо шифротекст присутній, то він представляється як наступні $\lceil mt/8 \rceil + \lceil l/8 \rceil$ байт.

Відкриті ключі. Відкритий ключ *T*, який є матрицею $mt \times k$, представлений у вигляді рядків. Кожен рядок *T* представляється як $\lceil k/8 \rceil$ - байтовий рядок і відкритий ключ представлений як $mt \| \lceil k/8 \rceil$- байтова конкатенація цих рядків.

Елемент поля. Кожен елемент поля $F_q \cong F_2[z]/f(z)$ має форму $\sum_{i=0}^{m-1} c_i z^i$, де $c_i \in F_2$. Представлення вектора поля - це вектор $(c_0, c_1, \dots, c_{m-1}) \in F_2^m$.

Непривідні поліноми. Непривідні поліноми $g = g_0 + g_1 x + \dots + g_{t-1} x^{t-1} + x^t$ степеня *t* представлені $t\lceil m/8 \rceil$ байтами конкатенацією елементів поля $g_0, g_1, \dots, g_{t-1}$

Впорядковане поле. Представлення послідовності $(\alpha_1, \dots, \alpha_q)$ із q різних елементів поля F_q буде послідовність q елементів поля.

Автори [66, 159, 246–251] описують представлення, яке спрощує швидкі алгоритми декодування з постійним часом. «Локальна мережа Бенеша» (англ. in-place Benes network) представляє собою серію з $2m-1$ етапів заміни, що застосовуються до масиву з $q = 2^m$ об'єктів $(\alpha_0, \alpha_1, \dots, \alpha_{q-1})$. *Перший етап* змінює місцями α_0 і α_1, міняє місцями α_2 і α_3, міняє місцями α_4 і α_5 і т. д., як вказано послідовністю з $q/2$ керуючих бітів (біт 1 означає заміну місць, 0 означає відсутність необхідності в заміні). *Другий етап* змінює місцями α_0 і α_2, міняє місцями α_1 і α_3, міняє місцями α_4 і α_6 і т.д., як вказано наступними $q/2$

керуючими бітами. Це триває до стадії m, на якій змінюються α_0 і $\alpha_{q/2}$, змінюються на α_1 і $\alpha_{q/2+1}$ і так далі. *На* $(m+1)$ *етапі* точно все відбувається так само, як і на $(m-1)$ етапі (з новими керуючими бітами), $(m+2)$-й етап аналогічний $(m-2)$-му етапу, і так далі аж до $(2m-1)$ етапу.

Визначимо π як перестановку $\{0,1,\dots,q-1\}$ таку, що $a_{i+1} = \sum_{j=0}^{m-1} \pi(i)_i \cdot z^{m-1-j}$ для всіх $i \in \{0,1,\dots,q-1\}$. Порядок $(\alpha_1,\dots,\alpha_q)$ представлений як послідовність з $(2m-1)2^{m-1}$ керуючих бітів для локальної мережі Бенеша для π. Цей вектор представлений у вигляді $\lceil (2m-1)2^{m-4} \rceil$ байт як описано вище. Кожна перестановка має кілька варіантів керуючих бітових векторів. Щоб спростити тестування, потрібно, щоб перестановка π була перетворена конкретно в керуючі біти, визначені для π. Програмні біти керування зчитуванням не перевіряють унікальність. В якості простого захисту від помилок, при обчисленні керуючих бітів для π автори рекомендують розробникам після обчислення перевіряти, що застосування мережі Бенеша дає π, і перезапускати генерацію ключа, якщо цей тест не пройшов.

Вибір стовпця. Частина особистого ключа, згенерованого функцією $KeyGen$, представляє собою послідовність $c = (c_{n-k-\mu+1},\dots,c_{n-k})$ з μ цілих чисел у порядку зростання від $n-k-\mu+1$ до $n-k-\mu+\nu$. Ця послідовність c представлена, як рядок розміром $\lceil \nu/8 \rceil$ байт, формат представлення в порядку від молодшого до старшого (в оберненому порядку) байта наступний:

$$\sum_{i=0}^{\mu-1} 2^{c_{n-k-\mu+1}-(n-k-\mu+1)}.$$

Однак, для $(\mu,\nu) = (0,0)$ послідовність c замість цього представлена як 8-байтовий рядок, який є оберненим порядком байтів $2^{32}-1$, тобто, 4 байта значення 255, за якими слідують 4 байти значення 0. Спеціальна обробка $(\mu,\nu) = (0,0)$ розроблена таким чином, щоб особистий ключ, що використовує $(\mu,\nu) = (0,0)$, був сумісний з програмним забезпеченням декапсуляції, що використовує $(\mu,\nu) = (32,64)$, коли всі інші параметри співпадають.

Особистий ключ. Особистий ключ (δ,c,g,α,s) представляє собою конкатенацію п'яти частин:

- $\lceil l/8 \rceil$- байтовий рядок $\delta \in F_2^l$.

- Рядок, що представляє вибір стовпців c. Цей рядок має $\lceil \nu/8 \rceil$ байтів або 8 байтів, якщо $(\mu,\nu) = (0,0)$.

- $t\lceil m/8 \rceil$- байтовий рядок полінома g.

- $t\lceil m/8 \rceil$- байт, що представляє поле порядку α.

- $\lceil n/8 \rceil$- байтовий рядок $s \in F_2^n$.

6. Classic McEliece: алгоритм інкапсуляції ключів

6.1 Генерація непривідного полінома

Вхідні параметри:

рядок $d_{0,}d_{1,}\dots,d_{\sigma_1 t-1}$ із $\sigma_1 t$ бітів.

Вихідні параметри:

непривідний поліном $g \in F_q[x]$ степеня *t*.

Алгоритм:

1) Визначити $\beta_j = \sum_{i=0}^{m-1} d_{\sigma_1 j+iz^i}$ для кожного $j \in \{0,1,\dots,t-1\}$. У кожній групі σ_1 вхідних бітів використовуються тільки *m* перших бітів. Алгоритм ігнорує біти рядка, що залишилися.
2) Визначити $\beta = \beta_0 + \beta_1 y + \dots + \beta_{t-1} y^{t-1} \in F_q[y]/F(y)$.
3) Обчислити мінімальний поліном g із β над полем F_q. За визначенням g є непривідний поліном, такий що $g(\beta) = 0$.
4) Повернути g, якщо g має степінь *t*. В іншому випадку повернути помилку виконання.

6.2 Генерація впорядкованих полів

Вхідні параметри:

рядок $\sigma_2 q$.

Вихідні параметри:

послідовність $(\alpha_1, \alpha_2, \dots, \alpha_q)$ із q різних елементів у полі F_q.

Алгоритм:

1) Беруться перші α_2 вхідних бітів $b_0, b_1, \dots, b_{\sigma_2-1}$ як σ_2- розрядне ціле число $a_0 = b_0 + 2b_1 + \dots + 2^{\sigma_2-1} b_{\sigma_2-1}$, потім беруться наступні σ_2 бітів як σ_2- розрядне ціле число a_1 і так далі до a_{q-1}.
2) Якщо $a_0, a_1, \dots, a_{q-1}$ не різні, повертається помилка.
3) Сортуються пари (a_i, i) в лексикографічному порядку, щоб отримати пари $(a_{\pi(i),}\pi(i))$, де π – перестановка $\{0,1,\dots,q-1\}$.
4) Визначити $\alpha_{i+1} = \sum_{j=0}^{m-1} \pi(i)_j \cdot z^{m-1-j}$, де $\pi(i)$ визначає j – найменш значущий біт в $\pi(i)$. Слід відмітити, що кінцеве поле F_q побудоване як $F_2[z]/f(z)$.
5) Повернути $(\alpha_1, \alpha_2, \dots, \alpha_q)$.

6.3 Генерація ключів

Вхідні параметри:

не приймає ніяких вхідних даних (крім параметрів).

Вихідні параметри:

відкритий ключ та особистий ключ.

Алгоритм:

1) Генерується рядок δ із нормальним розподіленням довжиною l бітів, тобто, породжуюче значення.
2) Обчислюється $E = F(\delta)$ рядок довжиною $(n + \sigma_2 q + \sigma_1 t + l)$ бітів.
3) Визначається рядок δ' як останні *l* бітів від *E*.
4) Визначається *s* як перші *n* бітів від *E*.
5) Обчислюється $\alpha_1, \dots, \alpha_q$ з $\sigma_2 q$ наступних бітів від *E*, використовуючи описаний вище алгоритм генерації впорядкованих полів. Якщо генерація завершується помилкою, встановлюється $\delta \leftarrow \delta'$ та алгоритм виконується заново.
6) Обчислюється g із наступних $\sigma_1 t$ від E, використовуючи описаний вище алгоритм генерації непривідного полінома. Якщо генерація завершується помилкою, встановлюється $\delta \leftarrow \delta'$ і алгоритм виконується заново).
7) Визначається $\Gamma = (g, \alpha_1, \alpha_2, \dots, \alpha_n)$ (слід зазначити, що $\alpha_{n+1}, \dots, \alpha_q$ не використовується тут).
8) Обчислюється $(T, c_{n-k-\mu+1}, \dots, c_{n-k}, \Gamma') \leftarrow MATGEN(\Gamma)$. Якщо генерація завершується помилкою, встановлюється $\delta \leftarrow \delta'$ та алгоритм виконується заново.
9) Визначається Γ' як $(g, \alpha'_1, \alpha'_2, \dots, \alpha'_n)$.
10) Формується T як відкритий ключ і $(\delta, c, g, \alpha, s)$ як особистий ключ, де $c = (c_{n-k-\mu+1}, \dots, c_{n-k})$ і $\alpha = (\alpha'_1, \dots, \alpha'_n, \alpha_{n+1}, \dots, \alpha_q)$.

6.4 Генерація вектора фіксованої ваги

Вхідні параметри:

не приймає ніяких вхідних даних.

Вихідні параметри:

вектор $e \in F_2^n$ ваги *t*. Алгоритм використовує попередньо обчислене ціле число $\tau \geq t$.

Алгоритм:

1) Згенерувати $\sigma_1 \tau$ випадкових бітів $b_0, b_1, \dots, b_{\sigma_1 \tau - 1}$.
2) Визначити $d_j = \sum_{i=0}^{m-1} b_{\sigma_1 j + i} 2^i$ для кожного $j \in \{0, 1, \dots, \tau - 1\}$.
3) Визначити $a_0, a_1, \dots, a_{t-1}$ як перші *t* записів в $d_0, d_1, \dots, d_{\tau-1}$ в діапазоні $\{0, 1, \dots, n-1\}$. Якщо виходить визначити менш, ніж *t* таких послідовностей, алгоритм виконується заново.
4) Якщо $a_0, a_1, \dots, a_{t-1}$ не всі різні, алгоритм виконується заново.
5) Визначити $e = (e_0, e_1, \dots, e_{n-1}) \in F_2^n$ як вектор ваги *t* такий, що $e_{a_i} = 1$ для кожного *t*.
6) Результатом являється *e*.

Ціле число τ визначається як *t*, якщо $n = q$; як $2t$, якщо $(q/2) \le n \le q$; як $4t$, якщо $(q/4) \le n \le (q/2)$ і т.д. Всі вибрані параметри мають $(q/2) \le n \le q$, тому $\tau \in \{t, 2t\}$.

6.5 Інкапсуляція

Вхідні параметри:

відкритий ключ *T*.

Вихідні параметри:

шифротекст *C* та сеансовий ключ *K*.

Алгоритм:

1) Використовуючи алгоритм генерації вектора фіксованої ваги, згенерувати вектор $e \in F_2^n$ ваги *t*.

2) Обчислити $C_0 = ENCODE(e, T)$.

3) Обчислити $C_1 = H(2, e)$.

4) Сформувати $C = (C_0, C_1)$.

5) Обчислити $K = H(1, e, C)$.

6) Сформувати шифротекст *C* та сеансовий ключ *K*.

6.6 Алгоритм кодування

Вхідні параметри:

вектор-стовпець $e \in F_2^n$ з вагою t і відкритий ключ *T*, тобто, матриця $(n - k) \times k$.

Вихідні параметри:

вектор $C_0 \in F_2^{n-k}$.

Алгоритм:

1) Визначити $H = (I_{n-k}|T)$.

2) Обчислити $C_0 = He \in F_2^{n-k}$.

6.7 Алгоритм декодування

Вхідні параметри:

вектор $C_0 \in F_2^{n-k}$, набір параметрів Γ'.

Вихідні параметри:

вектор-стовпець $e, C_0 = He$.

Існує два можливі результати алгоритму:

- Якщо $C_0 = ENCODE(e, T)$, то $DECODE(C_0, \Gamma') = e$. Іншими словами, якщо існує вектор $e \in F_2^n$ вагою *t* такий, що виконується $C_0 = He \in F_2^{n-k}$ з $H = (I_{n-k}|T)$, то алгоритм декодування завершиться успішно.
- Якщо C_0 немає форми He для будь-якого вектора $e \in F_2^n$ вагою *t*, то алгоритм декодування завершиться помилкою.

Алгоритм:

1) Розширити C_0 до $v = (C_0, 0, \dots, 0) \in F_2^n$ додаванням *k* нулів.

2) Знайти унікальне кодове слово *c* у коді Гоппи, яке визначене Γ' з відстанню $\le t$ від v; якщо такого кодового слова немає, повернути помилку.

3) Встановити $e = v + c$.

4) Якщо $wt(e) = t$ і $C_0 = He$, повернути e, інакше повернути помилку.

7. Параметри Classic McEliece

Існує п'ять запропонованих наборів параметрів у систематичній формі: mceliece348864, mceliece460896, mceliece6960119, mceliece6688128 та mceliece8192128. Також авторами було запропоновано п'ять наборів параметрів, що використовують напівсистематичну форму: mceliece348864f, mceliece460896f, mceliece6960119f, mceliece6688128f та mceliece8192128f.

У табл. 15.24 наведено загальносистемні параметри варіацій криптосистеми Classic McEliece. У табл. 15.25 наведено фактичні розміри вхідних та вихідних параметрів для варіацій алгоритму. Усі дані наведено у байтах.

Таблиця 15.24.

Основні параметри криптосистеми Classic McEliece

Алгоритм	m	n	t	Поліноми
mceliece348864	12	3488	64	$f(z) = z^{12} + z^3 + 1$, $F(y) = y^{64} + y^3 + y + z$
mceliece348864f	12	3488	64	$f(z) = z^{12} + z^3 + 1$, $F(y) = y^{64} + y^3 + y + z$, $(\mu, \nu) = (32,64)$
mceliece460896	13	4608	96	$f(z) = z^{13} + z^4 + z^3 + 1$, $F(y) = y^{96} + y^{10} + y^9 + y^6 + 1$
mceliece460896f	13	4608	96	$f(z) = z^{13} + z^4 + z^3 + 1$, $F(y) = y^{96} + y^{10} + y^9 + y^6 + 1$, $(\mu, \nu) = (32,64)$
mceliece6688128	13	6688	128	$f(z) = z^{13} + z^4 + z^3 + z + 1$, $F(y) = y^{128} + y^7 + y^2 + y + 1$
mceliece6688128f	13	6688	128	$f(z) = z^{13} + z^4 + z^3 + z + 1$, $F(y) = y^{128} + y^7 + y^2 + y + 1$, $(\mu, \nu) = (32,64)$
mceliece6960119	13	6960	119	$f(z) = z^{13} + z^4 + z^3 + z + 1$, $F(y) = y^{119} + y^8 + 1$
mceliece6960119f	13	6960	119	$f(z) = z^{13} + z^4 + z^3 + z + 1$, $F(y) = y^{119} + y^8 + 1$, $(\mu, \nu) = (32,64)$
mceliece8192128	13	8192	128	$f(z) = z^{13} + z^4 + z^3 + 1$, $F(y) = y^{128} + y^7 + y^2 + y + 1$
mceliece8192128f	13	8192	128	$f(z) = z^{13} + z^4 + z^3 + 1$, $F(y) = y^{128} + y^7 + y^2 + y + 1$, $(\mu, \nu) = (32,64)$

Кожен із цих десяти наборів параметрів має еталонну реалізацію; реалізацію з використанням 64-бітної векторизації; реалізацію з використанням 128-бітних векторних інструкцій Intel/AMD); реалізацію з використанням 256-бітних векторних інструкцій Intel/AMD). Ці чотири реалізації є інтероперабельними і формують ідентичні тестові вектори.

Таблиця 15.25.

Розміри вхідних та вихідних параметрів для криптографічних функцій криптосистеми Classic McEliece.

Алгоритм	Категорія	Відкритий ключ	Особистий ключ	Шифротекст	Сесійний ключ
mceliece348864	1	261120	6492	128	32
mceliece348864f	1	261120	6492	128	32
mceliece460896	3	524160	13608	188	32
mceliece460896f	3	524160	13608	188	32
mceliece6688128	5	1044992	13932	240	32
mceliece6688128f	5	1044992	13932	240	32
mceliece6960119	5	1047319	13948	226	32
mceliece6960119f	5	1047319	13948	226	32
mceliece8192128	5	1357824	14120	240	32
mceliece8192128f	5	1357824	14120	240	32

Для наочності на гістограмі (рис. 15.10) наведено розміри вхідних та вихідних параметрів. Оскільки значення параметрів відрізняються на кілька порядків, для наочного представлення на гістограмах дані довжини особистих та відкритих ключів, а також довжини відповідних шифротекстів представлені у логарифмічному масштабі. Суть використання такого масштабування полягає в перетворенні довжин даних наступним чином: $x = \log_{10} X$, де X - параметр такий як довжина відкритого або особистого ключа, довжина шифротексту, який підлягає масштабуванню; x – результат обчислення десяткового логарифму над масштабованим значенням.

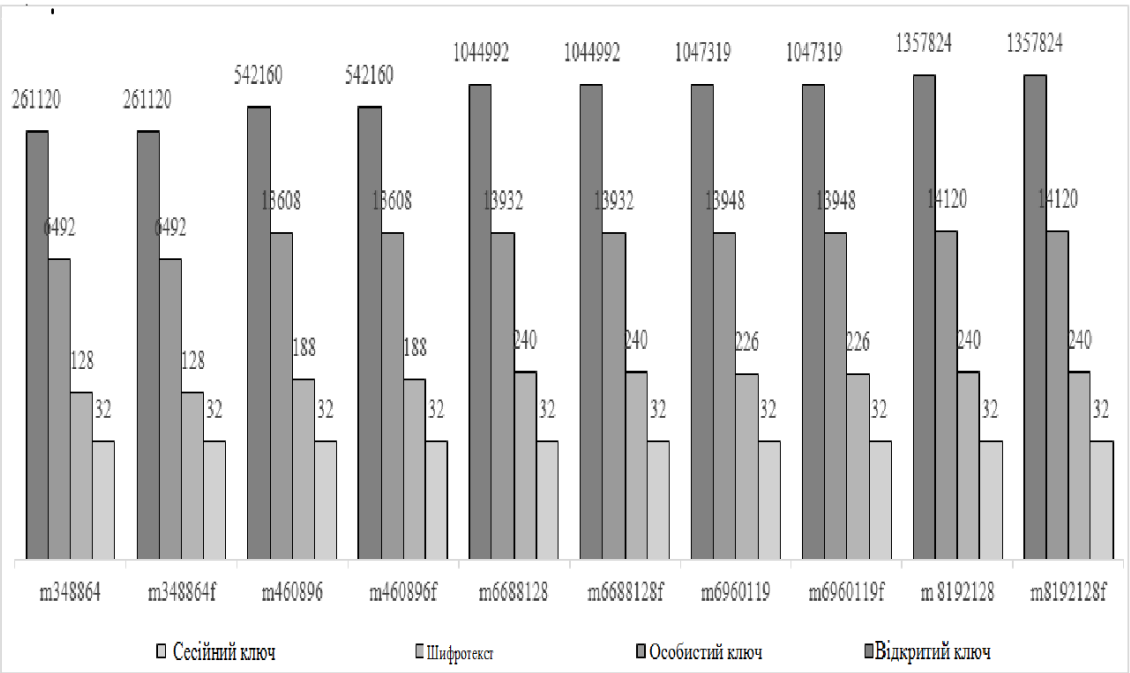


Рис. 15.10. Розміри вхідних та вихідних параметрів для криптографічних функцій криптосистеми Classic McEliece. Усі розміри представлені у байтах

Аналіз показників швидкодії

Вплив вибору основних параметрів на продуктивність. Розмір зашифрованого тексту становить $n-k$ біт. Зазвичай, співвідношення $R=k/n$ вибирається біля 0.8, тому розмір зашифрованого тексту становить близько $0.2n$ біт, тобто, $n/40$ байтів, плюс 32 байта для підтвердження. Розмір відкритого ключа складає $k(n-k)$ біт. Для $R\approx 0.8$ це, приблизно, $0.16n^2$ біт, тобто, $n^2/50$ байтів. Для генерації відкритого ключа використовується $n^{3+o(1)}$ операцій із стандартним методом Гауса. Існують асимптотично більш швидкі матричні алгоритми. У операціях із особистим ключем використовується всього $n^{1+o(1)}$операцій із стандартними алгоритмами.

Порівняльний аналіз показників швидкодії. Дана оцінка показників швидкодії представлена у форматі кількості циклів процесора, витрачених на виконання операції формування ключів, інкапсуляції та деінкапсуляції ключів. Всі виміри відбувалися без застосування будь-яких технологій оптимізації продуктивності. Зокрема, наводиться оцінка швидкодії, яка була проведена самими авторами при подачі алгоритму на всіх трьох раундах конкурсу постквантової криптографії.

Програмні вимірювання були виконані з використанням пакету SUPERCOP на платформі Intel Xeon E3-1275 v3 (Haswell) із тактовою частотою 3,50 ГГц [241]. Об'єм оперативної пам'яті платформи 32 Гб та працює під керуванням Ubuntu 16.04. Для варіацій із напівсистематичною формою швидкість виконання операцій інкапсуляції та деінкапсуляції не вказана. Наведені оцінки швидкодії можна порівняти за умови ігнорування інших (крім

наведених показників використовуваного центрального процесора) характеристик використовуваних обчислювальних систем. Така оцінка носить виключно первинний ознайомчий аналіз можливостей продуктивності перерахованих вище алгоритмів. Як вказують самі автори, показники продуктивності представлених ними оптимізованих реалізацій алгоритму можна ще поліпшити.

У табл. 15.26 наведено показники продуктивності для варіацій алгоритму інкапсуляції ключів Classic McElice, представленому на другому раунді конкурсу. У табл. 15.27 наведено показники продуктивності для Classic McElice, представленому на третьому раунді конкурсу [241]. Дані наведені у циклах процесора, які потрібно виконати щодо кожної операції. Як згадувалося раніше, кожна спроба генерації ключа (для систематичних варіантів) успішна з ймовірністю близько 29%, тому загальний час генерації ключа варіюється. Однак, остаточна успішна генерація ключів вимагає постійного часу, і вона використовує окремі випадкові числа з невдалих спроб генерації ключів.

Для наочності нижче на рис. 15.11 – 15.15 наведені гістограми параметрів швидкодії для всіх варіацій алгоритму.

Таблиця 15.26.

Показники продуктивності для варіацій алгоритму Classic McElice (раунд 2)

Назва	Формування ключів	Інкапсуляція	Деінкапсуляція
mceliece348864	208145574	46895	137503
mceliece348864f	82258215	-	-
mceliece460896	612458270	85410	274759
mceliece460896f	283980350	-	-
mceliece6688128	1344459257	156750	321536
mceliece6688128f	625501207	-	-
mceliece6960119	1202081992	156826	303207
mceliece6960119f	*565430070*	-	-
mceliece8192128	1277898472	185146	324803
mceliece8192128f	678901745	-	-

Таблиця 15.27.

Показники продуктивності для варіацій алгоритму Classic McElice (раунд 3)

Назва	Формування ключових даних	Інкапсуляція	Деінкапсуляція
mceliece348864	58034411	44350	134745
mceliece348864f	36641040	-	-

Продовження табл. 15.27.

mceliece460896	215785433	117782	271694
mceliece460896f	117067765	-	-
mceliece6688128	556495649	151721	323957
mceliece6688128f	284584602	-	-
mceliece6960119	438217685	161224	301480
mceliece6960119f	246508730	-	-
mceliece8192128	514489441	178093	326531
mceliece8192128f	316202817	-	-

Щоб адекватно продемонструвати ці показники, всі дані на гістограмах наведені з використанням логарифмічного масштабу, так як параметри відрізняються на кілька порядків. Слід зазначити, що менші значення циклів, що витрачаються на виконання однієї операції, є переважними. У той же час, великі значення кількості циклів, що витрачаються, вказують на низьку швидкість виконання операції.

На рис. 15.11 наведена зведена гістограма всіх показників швидкодії, що показує загальне співвідношення швидкості виконання трьох операцій (формування ключів, інкапсуляція і деінкапсуляція ключів) для кожної з варіацій алгоритмів, представлених на третьому раунді. Усі дані вказані у витрачених на виконання операції циклах. Приблизно порівнянну швидкість виконання операцій інкапсуліції та деінкапсуляції мають усі варіації. Проте варто відзначити досить великий розрив у продуктивності між формуванням ключів та інкапсуляцією (деінкапсуляцією).

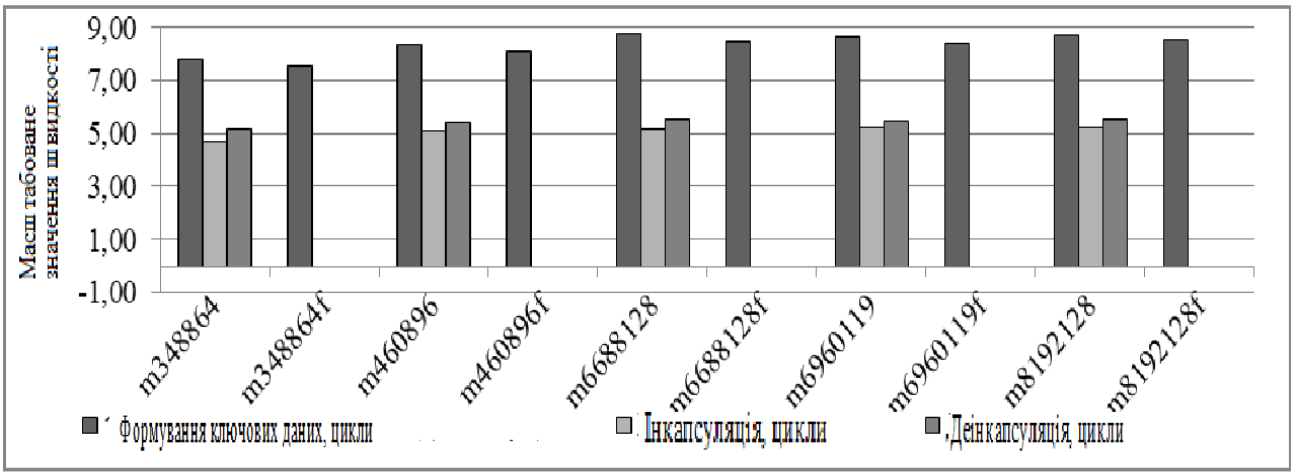


Рис. 15.11. Гістограма показників швидкодії (у логарифмічному масштабі)

Як вказувалося, автори для кожного з трьох раундів представляли еталонні та оптимізовані версії алгоритму. Однак, приріст продуктивності порівняно з першим раундом, на думку авторів, був майже подвійний. Приріст продуктивності для реалізації третього раунду порівняно з другим менш

значний. Невеликого прискорення вдалося домогтися для операції формування ключів, для інших операцій швидкість порівнянна, а для варіації mceliece460896 з третього раунду швидкість інкапсуляції зменшилася майже на третину. На рис. 15.12 і 15.13 наведені зведені гістограми показників швидкодії, що показують загальне співвідношення швидкості виконання реалізацій, представлених на другому та третьому раунді конкурсу.

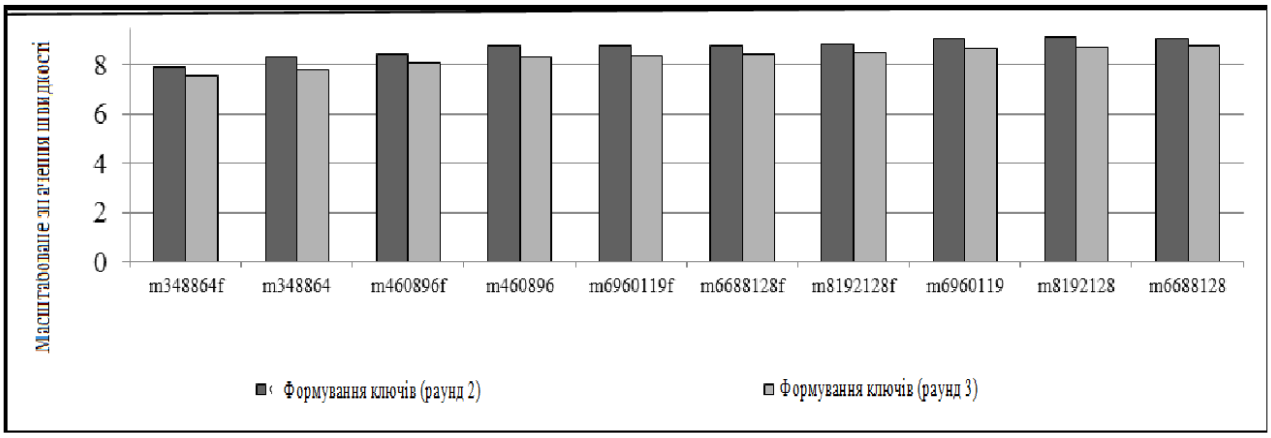


Рис. 15.12. Гістограма показника швидкодії: швидкість формування ключових даних (раунд 2 та 3), у циклах (у логарифмічному масштабі)

В рамках цього дослідження проведено порівняльний аналіз з алгоритмами BIKE та HQC. Для всіх трьох алгоритмів програмні вимірювання були виконані з використанням пакета SUPERCOP на платформі Intel Xeon E-2124 (CoffeeLake) з тактовою частотою 3,3 ГГц. На рис. 15.14 наведено гістограму показників швидкодії алгоритмів Classic McElice (раунд 3), BIKE та HQC.

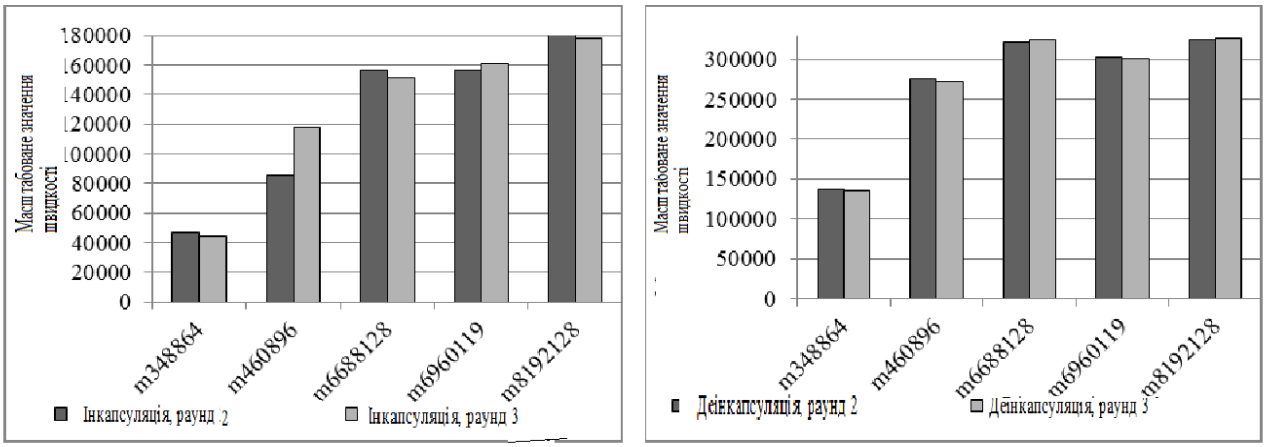


Рис. 15.13. Гістограма показників швидкодії: швидкість інкапсуляції (раунд 2 та 3) та швидкість деінкапсуляції (раунд 2 та 3), у циклах (у логарифмічному масштабі)

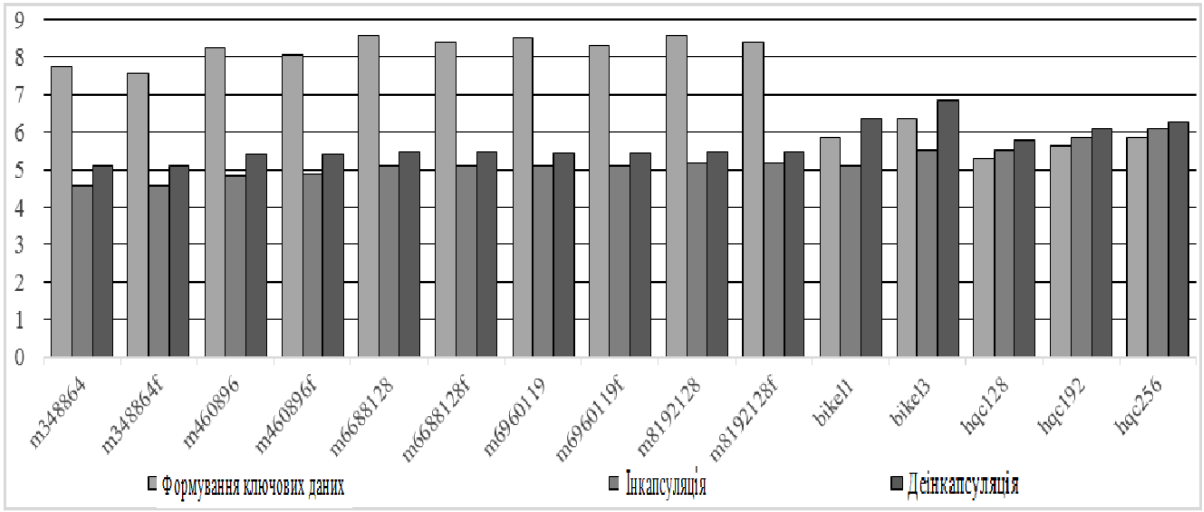


Рис. 15.14. Гістограма показників швидкодії алгоритмів Classic McElice, BIKE та HQC, у циклах (у логарифмічному масштабі)

Схема інкапсуляції ключів BIKE (BIt Flipping Key Encapsulation) представлена групою вчених з університетів країн Франції, США, Ізраїлю, Німеччини [253]. Схема заснована на квазіциклічних кодах з перевіркою на парність (QC-MDPC) з помірною щільністю. Алгоритм має IND-CPA криптостійкість. У третьому раунді авторами подано модифікацію алгоритму BIKE-2, представленого на попередніх раундах. В основі алгоритму BIKE-2 лежить криптосистема Нідеррайтера з перевірочною матрицею на парність. Алгоритм забезпечує всі три рівні безпеки. Схема HQC (Hamming Quasi-Cyclic) подана на конкурс групою вчених з університетів країн Франції, США, Ізраїлю, Німеччини [254]. Алгоритм має IND-CCA2 криптостійкість. Основними особливостями HQC є маленький розмір відкритого ключа та ефективні реалізації на основі класичних алгоритмів декодування. Варто зазначити, що, незважаючи на повільну операцію формування ключів для Classic McElice у порівнянні з BIKE і HQC, швидкість інкапсуляції та деінкапсуліції у алгоритму, що розглядається, менша.

У вимогах зазначено, що алгоритм повинен бути інтероперабельним і масштабованим. Тому в даному дослідженні був проведений аналіз продуктивності для досить слабкої обчислювальної платформи. Вимірювання були виконані з використанням пакета SUPERCOP на чотирьохядерній платформі Cortex-A9 із тактовою частотою 1,2 ГГц. Очікувано, швидкість виконання всіх операцій впала, для формування ключів знадобилася на чверть більша операція процесора, решті операцій потрібно було на 1/5 більше. Автори відзначають, що вдосконалення продуктивності для малоресурсних платформ є перспективним напрямом досліджень. На рис. 15.15 наведена гістограма показників швидкодії, отриманих при вимірах.

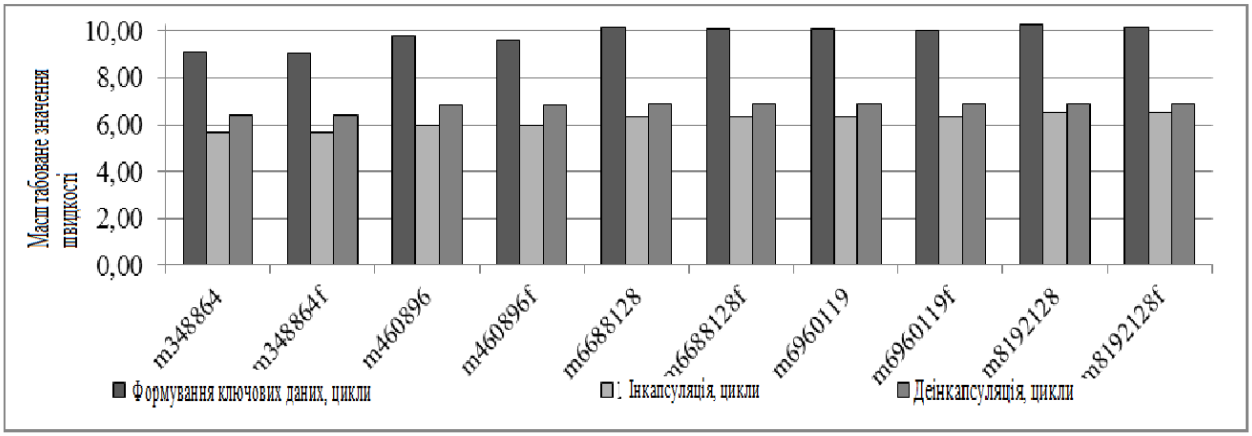


Рис. 15.15. Гістограма показників швидкодії алгоритму Classic McElice для обчислювальної платформи Cortex-A9, у циклах (в логарифмічному масштабі)

Внесені зміни

У алгоритм у міру просування конкурсу постквантової криптографії Classic McEliece вносилися зміни. Зокрема, перед початком другого раунду конкурсу NIST (30 березня 2019 р.) був суттєво змінений список наборів параметрів для підтримки необхідних рівнів безпеки. Список параметрів розширили так [241]:

– $(8192,\ 6528,\ 128)$, шифротексти довжиною 240 байт: приймають як n , так і t як степені двійки; як і в першому раунді.

- $(6960,\ 5413,\ 119)$, шифротексти довжиною 226 байт: оптимальна безпека в межах 2^{20} байт для відкритого ключа, як і першому раунді.

– $(6688,\ 5024,\ 128)$, шифротексти довжиною 240 байт: оптимальна безпека в межах 2^{20} байтів, якщо потрібно, щоб n і t були кратні 32.

- $(4608,\ 3360,\ 96)$, шифротексти довжиною 188 байт: оптимальна безпека в межах 2^{19} байт, якщо потрібно, щоб n і t були кратні 32.

- $(3488,\ 2720,\ 64)$, шифротексти довжиною 128 байт: оптимальна безпека в межах 2^{18} байтів, якщо потрібно, щоб n і t були кратні 32.

Оскільки Classic McEliece має такі маленькі зашифровані тексти, можна було б значно заощадити від зміни перетворення CCA, щоб виключити підтвердження відкритого тексту (32 байти).

Однак підтвердження у вигляді відкритого тексту має переваги в плані безпеки: наприклад, воно зупиняє атаки з вибраним зашифрованим текстом на більш ранньому етапі, ніж неявне відхилення, що спрощує захист побічних каналів. Тому автори вирішили зберегти існуюче перетворення CCA.

Також можливо змінити параметри для подальшого зменшення розмірів зашифрованого тексту на кожному рівні безпеки за рахунок використання більш довгих ключів.

Було змінено формат випадкового формування пари ключів так, щоб при необхідності можна було додатково стискати ключі до 256 – бітного рядка.

Також була описана, в якості можливого майбутнього, пропозиція альтернативного методу генерації ключів, який є більш складним (і вимагає великих стислих особистих ключів), але продуктивність якого значно вища.

При подачі документації (від 10 жовтня 2020 р.) до третього раунду набори параметрів для напівсистематичної форми алгоритму були задокументовані і реалізовані як можливі варіації алгоритму.

Як і в 2-му раунді, генерація ключа визначається з використанням різних випадкових об'єктів, таких як випадковий непривідний многочлен g над F_q степеня t. У 3-му раунді вказується явне відображення з випадкового рядка байтів у пари ключів за допомогою явних відображень із випадкових байтових рядків до випадкових об'єктів таких, як g. Це залишає відкритою можливість вказування додаткових способів відображень у майбутньому з перевіркою безпеки.

Відображення пар ключів починається з 32 байтів випадковості, що дозволяє стиснути особистий ключ до 32-байтового породжуючого значення. Породжуюче значення розширюється за допомогою *SHAKE*256 [252]. Вибірка відхилення при генерації ключів обробляється шляхом детермінованого зіставлення кожного породжуючого значення з новим породжуючим значенням. Особистий ключ тепер має наступний формат: 256 біт (остаточного) породжуючого значення, що генерує ключ (а не раніше відхилених породжуючих значень); 64-бітний рядок ваги 32, що визначає стовпці, що використовуються для напівсистематичної форми (або сумісна константа для систематичної форми); багаточлен g; керуючі біти для $(\alpha_1,...,\alpha_n)$ і n-бітний рядок s, що використовується для неявного відхилення. Це залишає відкритою можливість вказування інших форматів особистих ключів у майбутньому, таких як стислі формати (порядок об'єктів у цьому форматі розроблений, щоб забезпечити стиснення за допомогою простого усічення з ефективною декомпресією) або список $(\alpha_1,...,\alpha_n)$ замість керуючих бітів для середовищ, де перестановка через RAM не є проблемою. Будь-який інший формат особистого ключа з ефективними алгоритмами перетворення на цей формат особистого ключа і назад матиме таку ж математичну безпеку.

Висновки

Проведено аналіз фіналіста конкурсу постквантової криптографії NIST PQC алгоритму інкапсуляції ключів Classic McEliece. Розглянуто математичні моделі алгоритму Мак-Еліса та його сучасної варіації Classic McEliece. Розглянуто характеристики вхідних та вихідних параметрів, а також основні показники криптографічної стійкості та швидкодії. Оцінка швидкодії формувалася з урахуванням еталонних реалізацій на кількох обчислювальних платформах. Також проведено аналіз змін, внесених упродовж трьох раундів конкурсу. На основі цього проведено порівняльний аналіз показників швидкодії для варіацій алгоритму, представлених на трьох раундах.

У ході досліджень встановлено, що схема відповідає формальним вимогам до кандидатів на постквантові схеми інкапсуляції ключів, тобто, має різні

варіанти алгоритму, які забезпечують всі три рівні криптостійкості (1-й, 3-й та 5-й).

Алгоритму властиві довгі відкриті ключі внаслідок того, що ймовірність успішного виконання окремих функцій формування ключа досить низька, формування ключів - найбільш повільна та ресурсомістка операція. Незважаючи на це, операції інкапсуляції та деінкапсуляції в порівнянні з іншими конкурсантами більш продуктивні.

Перспективним напрямом дослідження є поліпшення властивостей масштабованості без втрати показників продуктивності та криптографічної стійкості.

16 МЕТОДИ АНАЛІЗУ ТА СИНТЕЗУ ПЕРСПЕКТИВНИХ КРИПТОПЕРЕТВОРЕНЬ ДЛЯ ПОСТКВАНТОВОГО ПЕРІОДУ

16.1 Особливості побудування та аналіз електронних підписів 5 рівня безпеки для постквантового періоду на основі алгебраїчних решіток

Вступ

Нині все більш актуальною стає проблема захисту інформації та ресурсів від існуючих та потенційних криптоаналітичних атак з використанням квантового комп'ютера та квантової математики. Це обумовлено розробкою математичних основ для квантового комп'ютера та безпосередньо стрімким розвитком теорії та практики квантових комп'ютерів [62]. На сьогодні вже створені та застосовуються 20, 53 та 72 – кубітні квантові комп'ютери [62]. Розуміючи цю проблему, технологічно розвинені держави суттєві зусилля направляють на аналіз криптографічної стійкості існуючих стандартів криптографічного захисту інформації у постквантовий період та ведуть пошук щодо створення постквантових стандартів асиметричної криптографії. По суті виникла необхідність розгляду та аналізу існуючих на сьогодні криптографічних алгоритмів, заміни їх параметрів або збільшення розміру цих параметрів, а також необхідність створення нових стандартизованих криптографічних алгоритмів з огляду на можливості квантового комп'ютера та його математичного забезпечення у постквантовий період. Вирішення цієї проблеми здійснюється на світовому рівні в процесі проведення NIST США міжнародного конкурсу [67,68,100].

NIST США, розуміючи необхідність пошуку нових стандартизованих асиметричних криптографічних примітивів, особливу увагу звертає на створення цифрового (електронного) підпису (ЕП). Завершено перший раунд (етап) та йде другий етап. На конкурс PQC NIST США було подано 22 алгоритми – кандидатів на постквантовий стандарт ЕП, що базуються на використанні різних математичних основ. В ході другого етапу проведено проміжний семінар, за рішенням якого NIST США рекомендував до подальших досліджень 9 криптопримітивів ЕП. Дані щодо них та асиметричних шифрів (АСШ) і протоколів інкапсуляції ключів (ПІК) наведено в таблиці 16.1 [332].

Необхідно відмітити, що 2-й раунд триватиме 12-18 місяців, після чого очікується, що відбудеться 3-й раунд. Також очікується прийняття проєктів в якості стандартів близько 2022 року (але залишається право змінити терміни).

Як показують попередні дослідження, надійною математичною основою, на якій можуть бути створені постквантові ЕП, нині вважаються алгебраїчні решітки [333]. По суті, такий математичний апарат пройшов випробовування часом у вигляді регіонального стандарту X9.98 [172]. На конкурс NIST було подано наступні механізми ЕП, що базуються на алгебраїчних решітках: Cristals-Dilithium, FALCON та qTESLA. Вони зараз аналізуються на другому етапі конкурсу NIST [334].

Таблиця 16.1.

Попередні підсумки 2-го етапу конкурсу NIST США

	ЕП	АСШ/ПІК	Загалом
На основі решіток	3	9	12
На основі кодів	0	7	7
На основі мультиваріативних перетворень	4	0	4
На симетричній основі	2		2
Інші	0	1	1
Всього	9	17	26

З огляду на вищевказане, актуальною є проблема аналізу, дослідження, оцінки та порівняння кандидатів на постквантові стандарти ЕП з використанням математики алгебраїчних решіток, а також розробки рекомендацій щодо вибору із них найбільш перспективних для майбутнього застосування.

При дослідженнях будемо спиратись на таке визначення постквантового періоду. Постквантовий період пропонується визначити як проміжок часу у майбутньому, коли будуть суттєво удосконалені класичні методи та створені квантові комп'ютери з необхідними для успішного криптоаналізу довжинами регістрів (в кубітах) та необхідне для їх реалізації математичне та програмне забезпечення.

Метою цього підрозділу є оцінка, порівняння, обгрунтування та вибір перспективного ЕП для постквантового періоду на основі алгебраїчних решіток, а також його більш детальне дослідження.

1. Попередній аналіз та вибір перспективного кандидата на постквантовий електронний (цифровий) підпис

В таблиці 16.2 наведено короткі подання кандидатів на ЕП на основі алгебраїчних решіток, які аналізуються на другому етапі конкурсу.

При порівняльному аналізі використовувалася сукупність безумовних і умовних оцінок згідно [334]. Умовними оцінками виступали наступні характеристики алгоритмів:

1) Іст. – рівень криптографічної стійкості;
2) lв.к – довжина відкритого ключа;
3) lо.к – довжина особистого ключа;
4) lрез. – довжина результату криптоперетворення;
5) Тпр. – швидкість прямого криптоперетворення;
6) Тзв. – швидкість зворотного криптоперетворення.

Таблиця 16.2.

Механізми ЕП на основі алгебраїчних решіток

№	Назва	Анотація
1.	CRYSTALS-DILITHIUM	Конструкція механізму ЕП Dilithium базується на підході "Fiat-Shamir з перериваннями" [Lyu09] та схожа з механізмами, що запропоновані в [GLP12, BG14]. Множення поліномів в механізмі виконується в кільці поліномів $R_q = \frac{Z_q[X]}{X^n+1}$. Всі алгебраїчні операції в механізмі Dilithium виконуються над кільцем полінома R_q, q=2^{23}–2^{13}+1 та n=256. Стійкість механізму ЕП Dilithium ґрунтується на складності пошуку коротких векторів у алгебраїчних решітках.
2.	FALCON	Механізм Falcon є розвитком ЕП NTRU та позначається як NTRU – Falcon. В ньому деталізовані швидкі компактні підписи на основі решітки Фур'є та є NTRU – Falcon механізмом ЕП на основі решітки. Конструкція ЕП Falcon є простою, в ньому реалізуються теоретичні границі, що описані в Gentry, Peikert і Vaikuntanathan, для гешування та ЕП на основі решітки. В механізмі використовується клас криптографічних решіток (клас решіток NTRU) та зразок «лазівки» засобом реалізації швидкої вибірки Фур'є. Спрощено механізм ЕП Falcon може бути поданий таким чином: Falcon = GPV рамка + решітка NTRU + швидка вибірка Фур'є.
3.	qTESLA	Механізм ЕП qTESLA базується на складності рішення проблеми навчання в кільці з помилками (R-LWE). На відміну від інших альтернатив, qTESLA ЕП є консервативним, але ефективним механізмом. Реалізації qTESLA захищені від класичних та квантових атак в квантовій випадковій моделі оракула. Особливістю механізму є несуттєве скорочення випадкової моделі оракула та жорстке зменшення квантової випадкової моделі оракула в R-LWE.

Експертні оцінки використовувалися для оцінки важливості кожної з наведених характеристик, а безпосередньо при порівнянні алгоритмів використовувалися об’єктивні числові значення, шкала оцінки та вагові коефіцієнти важливості характеристик, що були отримані при експертному оцінюванні (таблиця 16.3) [333].

Таблиця 16.3.

Експертні оцінки характеристик криптоалгоритмів

Показники / Експерти	Іс.т	Lв.к	lo.к	lрез.	Тпр.	Тзв.
1	0,316	0,179	0,101	0,045	0,179	0,179
2	0,209	0,342	0,038	0,078	0,124	0,209
3	0,111	0,242	0,027	0,051	0,111	0,457
4	0,107	0,239	0,045	0,107	0,045	0,458
5	0,206	0,125	0,041	0,140	0,171	0,317
W	0,190	0,225	0,050	0,084	0,126	0,324

В даному дослідженні при виборі алгоритмів висувалися додаткові безумовні вимоги:

1) алгоритм повинен гарантувати, щонайменше, 3 рівень безпеки за класифікацією NIST [67,68];

2) якщо існує декілька варіантів наборів параметрів для одного алгоритму, то в порівнянні бере участь варіант, що гарантує найбільшу безпеку.

В таблиці 16.4 наведені характеристики обраних для порівняння алгоритмів ЕП, що засновані на використані перетворень в алгебраїчних решітках [334, 335].

Таблиця 16.4.

Характеристики алгоритмів ЕП в алгебраїчних решітках

Алгоритми	Тип	Іст.	lв.к	lo.к	lрез.	Тпр.	Тзв.
Dilithium_very_high	Lattices	3	1 760	3 856	3 366	2 293 141	611 325
falcon1024	Lattices	5	1 793	8 193	1 330	19 884 364	1 384 574
falcon768	Lattices	3	1441	6145	1 077	13 058 641	1 117 624
qTesla_256	Lattices	5	8 224	8 256	6 176	8 143 869	1 436 949

У табл. 16.5 наведено результати оцінювання вибраних механізмів ЕП.

Таблиця 16.5.

Результати оцінювання механізмів ЕП в алгебраїчних решітках

Назва алгоритму	Значення оцінки	Назва алгоритму	Значення оцінки
Dilithium_very_high	0,0964	qTesla_256	0,0660
falcon1024	0,0715	falcon768	0,0712

На рисунку 16.1 наведено гістограму відносної переваги алгоритмів, причому для більш узагальненого порівняння ЕП на алгебраїчних решітках наведено також оцінки щодо механізмів ЕП на основі мультиваріантних криптоперетворень та функцій гешування. Як видно з рисунка, перше місце займає алгоритм Dilithium_very_high, а на другому місці розташувались відразу 3 алгоритми – picnicl5fs, falcon1024, falcon768.

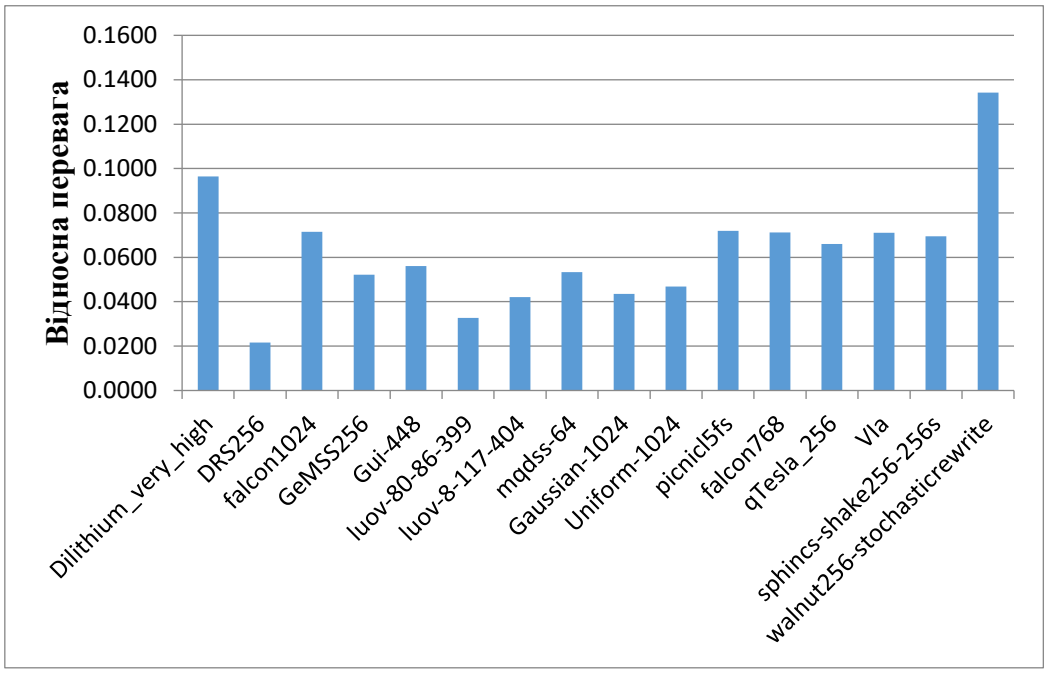


Рис. 16.1. Відносна перевага алгоритмів ЕП

Розглянемо детально результати порівняння. Особливістю даного порівняння стало те, що друге місце між собою розділили відразу три алгоритми (відмінність між їх значеннями лише у четвертому знаку після коми).

Необхідно відмітити, що механізм Walnut256-stochasticrewrite умовно є кращим через те, що має високі основні характеристики алгоритмів. Але на сьогодні є теоретична атака на алгоритм WalnutDSA, проте її успішність ще не доведена. Але зважаючи на цю атаку та те, що основним завданням нашого дослідження є порівняння ЕП на основі алгебраїчних решіток, механізм WalnutDSA при порівнянні не розглядається. Розглядаючи алгоритми, що зайняли третє місце, можна відмітити наступне. Алгоритм falcon768 відрізняється від falcon1024 більш низькою стійкістю на рівні Dilithium_very_high. Falcon1024 має більш високий рівень стійкості, а falcon768 має більш високі інші основні характеристики.

Таким чином, перше місце займає механізм Dilithium_very_high, тому в подальшому дослідження присвячені механізму Dilithium.

2. Модель порушника щодо постквантових механізмів ЕП

Побудова моделі порушника необхідна для того, щоб розробити комплекс заходів із забезпечення захищеності алгоритму. Така модель може бути побудована з урахування різних критеріїв.

Звичайно модель порушника розробляється з метою отримання відповідей на наступні питання:

- від кого необхідно захищати інформацію?
- якою є мета порушника?
- якими знаннями володіє порушник?
- які повноваження в системі має потенційний порушник?
- які методи і засоби використовує порушник?

По суті модель порушника – це опис можливих дій порушника, який формується на основі аналізу типу зловмисника, рівня його повноважень, знань, теоретичних та практичних можливостей. У якості порушника розглядається особа, що може отримати доступ до роботи з включеними до складу відповідної комп'ютерної системи (КС) засобами.

Порушники класифікуються за рівнем можливостей, що надаються їм штатними засобами КС. Виділяються чотири рівні таких можливостей. Класифікація є ієрархічною – кожний наступний рівень включає в себе функціональні можливості попереднього:

- 1-й рівень визначає найнижчий рівень можливостей проведення діалогу з КС – можливість запуску фіксованого набору завдань (програм), які реалізують заздалегідь передбачені функції обробки інформації;
- 2-й рівень визначається можливістю створення і запуску власних програм з новими функціями обробки інформації;
- 3-й рівень визначається можливістю управління функціонуванням КС, тобто впливом на базове програмне забезпечення системи, на склад та конфігурацію її устаткування;
- 4-й рівень визначається всім обсягом можливостей осіб, що здійснюють проєктування, реалізацію і ремонт апаратних компонентів КС, аж до включення до складу КС власних засобів з новими функціями обробки інформації [106].

Припускається, що в своєму рівні порушник – фахівець вищої кваліфікації, який має повну інформацію про КС і КЗЗ.

Така класифікація порушників є корисною для використання в процесі оцінки ризиків, аналізу вразливості системи, ефективності існуючих і планових заходів захисту [108].

При розробці моделі порушника необхідно визначити, що і у якій мірі має відображати ця отримана модель. Для цього необхідно визначитись з необхідними степенями деталізації моделі порушника.

На наш погляд, можливі наступні степені деталізації моделі порушника:

- змістовна модель порушників – відображає причини та мотивацію дій порушників, переслідувані ними цілі і загальний характер дій у процесі підготовки і здійснення порушення цілісності алгоритму;
- сценарії впливу порушників – визначають класифіковані типи порушень з конкретизацією алгоритмів і етапів, а також способи дії на кожному етапі.

Під час побудови моделі порушника необхідно спочатку проаналізувати усіх користувачів системи, розподілити їх за категоріями та визначити найбільш критичні. Користувачі таких категорій будуть прийняті як можливі внутрішні

порушники системи. Потім необхідно визначити, які категорії відвідувачів можуть бути зовнішніми порушниками.

Усіх можливих порушників необхідно класифікувати за різними показниками для того, щоб далі скласти модель порушника. Нижче наведені можливі види класифікацій порушників:

- За метою порушення. Проводиться для визначення мотивів порушника. Дії порушника в залежності від мети можуть бути спрямовані на алгоритм або на ключі, сервери тощо.

- За рівнем знань про алгоритм. Кожен порушник має певний рівень кваліфікації та поінформованості щодо алгоритму та його «секретів». Класифікація враховує знання можливого порушника та його практичні навички у роботі з комп'ютерними системами та інформаційними технологіями.

- За місцем дії. Класифікація проводиться для визначення розташування порушника відносно організації під час здійснення спроби несанкціонованого доступу до інформаційного ресурсу.

- За методами і способами, якими вони користуються. Порушник може отримати конфіденційну інформацію та інформацію з обмеженим доступом, користуючись при цьому різними методами та засобами. Порушення може бути скоєне з/без використання певних засобів для отримання інформації. Методи можуть бути різними, як дозволеними, так і забороненими. Дозволеним вважається отримання інформації без порушення прав власності.

З урахуванням вказаного вище приймемо наступну модель порушника:

- Захищати інформацію необхідно від зловмисника з квантовим комп'ютером, що має як мінімум декілька сотень кубітів потужності, або від добре укомплектованої хакерської групи. Потрібно намагатися забезпечити захист навіть проти атаки великих професійних груп.

- Порушник має за мету отримати інформацію будь-яким чином, використовуючи квантовий комп`ютер.

- У найкращому випадку порушник не має жодних знань ні про систему, ні про алгоритм. Але частіше за все порушник може знати тільки загальнодоступні дані або може перехватити частину інформації, або знати внутрішні дані (інсайдер).

- Також у найкращому варіанті порушник не має повноважень в системі. Але, якщо це інсайдер, або людина, що отримала адміністративний доступ, то він зможе підміняти процеси в системі і налаштувати її «під себе», при умові, що у адміністратора є відповідні права.

Вважатимемо, що порушник використовує усі доступні йому ресурси – найпотужніші комп'ютери і необмежний час.

Таким чином, у найгіршому випадку – порушник знає все про об'єкт та про всі засоби безпеки, встановлені на ньому. У найкращому – порушник не знає нічого про об'єкт. У нашому випадку це рівноімовірні варіанти.

3. Аналіз та оцінка механізму ЕП на алгебраїчних решітках Crystals-Dilithium

3.1 Загальні положення щодо ЕП на алгебраїчних решітках Crystals - Dilithium

Зважаючи на особливо жорсткі вимоги до ЕП постквантового періоду, розглянемо на наш погляд перспективний постквантовий механізм ЕП Crystals-Dilithium на алгебраїчних решітках [79].

Dilithium – це схема цифрового підпису, яка має надійну безпеку від атак на вибране повідомлення, що базуються на складності проблем решітки над модульними решітками. Поняття безпеки означає, що порушник, який має доступ до оракула підпису, не може виробити підпис повідомлення, підпис якого він ще не бачив, а також не може створити інший підпис повідомлення, яке він вже бачив підписаним. Dilithium – один із алгоритмів-кандидатів, який був представлений в проєкті постквантової криптографії NIST та досліджується на другому етапі конкурсу [67].

Конструкція Dilithium базується на методі «Fiat-Shamir з перериваннями» Lyubashevsky, який використовує вибірку відхилення для того, щоб зробити схеми Fiat-Shamir на основі решіток компактними та безпечними. Схема Dilithium була розроблена з урахуванням наступних критеріїв [62].

1. Простий для безпечної реалізації. Найбільш компактні схеми підпису на основі решіток [68] мають вирішальне значення для створення секретної випадковості. Механізм Dilithium, використовує рівномірну вибірку, як це було спочатку запропоновано в [100]. Крім того, всі інші операції (такі як множення поліномів і округлення) легко реалізуються за постійний час.

2. Консервативний за параметрами. Оскільки необхідно забезпечити довгострокову безпеку, в [79] було проаналізовано придатність атак на решітку з дуже сприятливої точки зору до точки зору атакуючого. Зокрема, розглядаються квантові алгоритми. Подібні алгоритми в даний час нереалістичні, і, як видається, існують серйозні перешкоди на шляху, що в майбутньому можуть відбутися покращення.

3. Зменшення розміру відкритого ключа та підпису. Оскільки багато додатків вимагають передачі як відкритого ключа, так і підпису (наприклад, ланцюжків сертифікатів), в [79] було розроблено схему, щоб мінімізувати суму цих параметрів. Наскільки відомо [67], Dilithium має найменше поєднання розміру підпису і розмірів відкритих ключів будь-якої схеми на основі решіток з однаковими рівнями безпеки [67].

4. Відносно легко змінювати безпеку. Дві операції, що складають практично всю процедуру підпису та перевірки, це розширення XOF (використовується SHAKE-128 і SHAKE-256) та множення у кільці полінома $R_q = Z_q[X]/(X^n+1)$. Тому дуже ефективні реалізації алгоритму [87] повинні оптимізувати ці операції та працювати за постійний час. Для всіх рівнів безпеки наша схема використовує одне і те ж кільце $q=2^{23}-2^{13}+1$ та $n=256$. Зміна безпеки полягає в тому, щоб робити більше/менше операцій над цим кільцем і робити більше/менше розширення XOF. Іншими словами, як тільки оптимізована

реалізація для певного рівня безпеки буде досягнута, вона майже тривіальна для отримання оптимізованої реалізації для вищого/нижчого рівня.

3.2 Огляд основного підходу ЕП на алгебраїчних решітках Crystals-Dilithium

Аналіз щодо вимог до параметрів та впровадження. Основна алгебраїчна операція, що виконується в механізмі – це множення матриці A, елементами якої є поліноми у $R_q = Z_q[X]/(X^n+1)$, на вектор таких поліномів. У механізмі ЕП , що рекомендується [94], параметр A є матрицею 5×4 і, отже, складається з 20 поліномів. Таким чином, множення A_v включає 20 поліноміальних множень. Як і в більшості механізмів на основі решіток, що ґрунтуються на операціях над кільцями поліномів, кільце поліномів обране так, що операція множення має мінімізовану складність. Це досягається на основі використання швидкого теоретико-числового перетворення (NTT), яке є лише версією цілочисленного множення (FFT), що працює над скінченим полем Z_q. Причому, для того, щоб використовувати NTT, модуль, просте ціле q, потрібно вибрати так, щоб у групі Z_q^* був елемент порядку $2n$ = 512, що еквівалентно $q \equiv 1 (mod\ 512)$.

Основні операції ЦП на алгебраїчних решітках Crystals-Dilithium.

В механізмі виконуються операції в кільці поліномів. Нехай R і R_q відповідно позначають кільця $Z[X]/(X^n+1)$ та $R_q = Z_q[X]/(X^n+1)$ для цілого числа q. При розгляді механізму ЕП значення степені полінома n буде 256 і модуль q простим числом $8380417 = 2^{23} - 2^{13} + 1$. Також в кільцях R або R_q включають елементи в Z і Z_q відповідно, а жирними рядковими буквами представляють векторні стовпці з коефіцієнтами в R або R_q. За замовчуванням всі вектори будуть векторними стовпцями. Жирні прописні літери позначають матриці. Для вектора v його транспонування позначається через V^T . При аналізі оператор формує значення 1, якщо значення true, та 0 – в іншому випадку.

Зведення за модулем. Для парного (непарного) додатного цілого числа α обчислене значення $r' = r\ mod^{\pm}\ \alpha$ є унікальним елементом r' у діапазонах відповідно

$$-\frac{\alpha}{2} < r' \le \frac{\alpha}{2}$$

або

$$-\frac{\alpha-1}{2} \le r' \le \frac{\alpha-1}{2} \qquad (16.1)$$

таке, що $r' = r\ mod\ \alpha$. Таке значення будемо називати центрованим зведенням за модулем q. Також, для будь-якого додатного цілого α, $r' = r\ mod^{+}\ \alpha$ є унікальний елемент r' у діапазоні $0 \le r' < \alpha$ такий, що $r' = r\ mod\ \alpha$.

Розміри елементів. Для елементу кільця $w \in Z_q$ позначення $|w \, mod^{\pm} \, q|$ означає $||w||_\infty$. Також для полінома $w = w_0 + w_1 X + \cdots + w_{n-1} X^{n-1} \in R$ значення $||w||_\infty$ отримуємо:

$$\left| ||w||_\infty = \max_i \right| ||w_i||_\infty, \text{ а } \quad ||w|| = \sqrt{||w_0||_\infty^2 + \cdots + ||w_{n-1}||_\infty^2}. \tag{16.2}$$

Аналогічно для полінома $w = w_1, \ldots, w_k \in R^k$ отримуємо у вигляді

$$\left| ||w||_\infty = \max_i \right| ||w_i||_\infty, \quad ||w|| = \sqrt{||w_1||^2 + \cdots + ||w_k||^2}. \tag{16.3}$$

Також S_η позначає, що для всіх елементів $w \in R$ виконується умова, що

$$\| w \|_\infty \leq \eta \, . \tag{16.4}$$

Вибір подання NTT швидкого перетворення. Модуль q вибирається таким чином, що існує 512-й корінь одиниці r за модулем q. Конкретно, в якості r приймемо r = 1753. Вказане означає, що циклотомічний поліном $X^{256} + 1$ розщеплюється на лінійні множники $X - r$ і за модулем q для i=1,3,5,...,511. Тому, відповідно до китайської теореми про лишки, вказане циклотомічне кільце R_q, є ізоморфним добутком кілець

$$Z_q[X] / (X - r^i) \cong Z_q \, . \tag{16.5}$$

За такої умови при знаходженні добутку складність суттєво зменшується, оскільки множення в NNT є точковим. А сам ізоморфізм:

$$a \mapsto (a(r), a(r^3), \ldots, a(r^{511})) : R_q \to \prod_i Z_q[X] / (X - r^i) \tag{16.6}$$

можна швидко обчислити за допомогою швидкого перетворення типу Фур'є (NTT).

4. Параметри та вимоги до механізму Crystals-dilithium та його властивості

Основні параметри:

- N= 256 – степінь поліному;
- Q =8380417 (QBITS = 23 - $\log_2$Q) .

Додаткові параметри:

- D = 14 -довжина молодшої частини коефіцієнта (бітів).

Цей параметр забезпечує зменшення довжини відкритого ключа за рахунок можливості зберігати тільки старшу частину коефіцієнтів поліному ($\log_2$Q – D) бітів замість $\log_2$Q та довжини особистого ключа, в якому зберігається тільки молодша частина коефіцієнтів поліному D бітів замість $\log_2$Q.

- OID – ідентифікатор методу, 3 байта.

Параметри, які використовуються при обчисленні норми

Ці параметри забезпечують правильність функціонування алгоритму ЕП:

- γ1 = ((Q - 1)/16) ;

- $\gamma 2 = \gamma 1 / 2$;
- $\alpha = 2^* \gamma 2$;
- T = 60 (кількість ненульових елементів в малому поліномі c).

Примітивний корінь

Примітивний корінь – це ціле число *R*, для якого виконується умова $R^{Q-1} mod\ Q = 1$

Для алгоритму CRYSTALS-DILITHIUM *ROOT_OF_UNITY* = 1753.

Довжини геш – значення, seed та засліплюючого полінома

Геш-функція, що використовується у схемі, повинна бути стійкою до колізій. Тому довжина її вектора значень повинна бути, як мінімум, HASHBYTES = 2λ/ BITSINBYTE.

Довжина seed повинна співпадати з криптостійкістю, тобто SEEDBYTES = λ/BITS_IN_BYTE.

Довжина коефіцієнтів засліплюючого поліному LOG2Z = 20 (біт).

Параметри, які можуть змінюватися

Ця група параметрів залежить від криптостійкості λ і визначена в таблиці 16.6 для λ=256 біт.

Таблиця 16.6.

Змінні параметри механізму Crystals-Dilithium

λ	mode	K	L	η	ζ	β(β < 60 η)	ω
	0	3	2	7	4	375	64
	1	4	3	6	4	325	80
	2	5	4	5	4	275	96
	3	6	5	3	3	175	120

Параметри які використовуються при перетворенні секретного, відкритого ключа та цифрового підпису в рядок октетів та зворотних перетворень:

- LOG2_2NJU = Ceil ($\log_2(2\eta-1)$;
- S1BYTES = L*N* LOG2_2NJU/BITSINBYTE;
- S2BYTES = K*N* LOG2_2NJU/BITSINBYTE;
- TLOWBYTES = K * N * D/ BITSINBYTE;
- SKBYTES = 2 * SEEDBYTES + HASHBYTES.

5. Вимоги до ключів для механізму Crystals-Dilithium

5.1 Узагальнений алгоритм

Під час генерування ключів повинні бути заданими наступні параметри:

- λ – криптостійкість;
- N – степінь поліному;

- Q – великий модуль перетворення;
- K, L – коефіцієнти, які визначають розміри матриці та векторів;
- D – параметр, який визначає кількість бітів в молодшій та старшій частині відкритого ключа;
- η - параметр, який визначає граничні значення для векторів S1, S2;
- SEEDBYTES – розмір seed (октетів);
- HASHBYTES - довжина геш – значення (октетів).

5.2 Вхідні та вихідні значення

Вхідні значення – немає.

Вихідні значення:

- pk – відкритий ключ (масив октетів);
- sk – секретний ключ (масив октетів).

5.3 Алгоритм генерування ключів

Генерування ключів здійснюється у такій послідовності:

1) Генерація випадкових послідовностей для ρ, ρ_1, k по SEEDBYTES байтів кожна (3 * SEEDBYTES байтів всього).

2) Генерація матриці A розміром [K][L], елементи якої є поліноми степеня *n* з коефіцієнтами за модулем Q: A := expand_a (ρ).

3) Генерація векторів S1[L][N] та S2[K][N]. Елементами векторів є поліноми степені N з коефіцієнтами в інтервалі [-η, η] (ці вектори визначає ρ_1): S1, S2 :=expand_s (ρ_1).

4) Обчислення вектора поліномів t := A * (S1 +S2) .

5) Представлення t = {t_{high}, t_{low}} згідно формули t = t_{high} * 2^D+ t_{low} (D = 14, довжина молодшої частини в бітах), функція t_{high}, t_{low} := gen_hl_w (t).

6) Формування відкритого ключа {ρ, t_{high}}.

7) Упаковка відкритого ключа, функція pk :=pack_pk (ρ, t_{high}).

8) Обчислення h_{pk}*:= hash(*p_k*)* - гешу для відкритого ключа. Довжина гешу повинна дорівнювати HASHBYTES.

9) Формування ctrhtnyjuj (особистого) ключа у складі: { ρ, key, h, S1, S2, t_{low}}.

10) Упаковка секретного ключа, функція sk :=pack_sk (ρ, key, hpk, S1, S2, t_{low}).

5.4 Склад та розміри відкритого ключа

- ρ -32 байти (випадкове значення – використовується для створення матриці A [K][L], кожний елемент якої є поліномом з елементами, які обмежені значенням Q.

- t_{high} обчислюється згідно формули t = t_{high} * 2^D+ t_{low} (D = 14).

Довжини відкритого ключа обчислюються в залежності від параметрів за формулою λ + K * (Log_2Q – D) * N (біт) і представлені в таблиці 16.7.

Таблиця 16.7.

Довжина значень відкритого ключа (байтів)

λ	N	K	$Log_2 Q – D$	PK BYTES
256	256	3	9	896
		4	9	1184
		5	9	1472
		6	9	1760

6. Алгоритм вироблення електронного підпису

6.1 Параметри алгоритму вироблення ЕП

Перед виконанням ЕП необхідно задати такі параметри:

- N – степінь поліному;
- Q – модуль, усі обчислення виконуються згідно цього модуля;
- L,K – цілі числа, які визначають розміри матриці та векторів S1, S2;
- CRYPTOBYTES – розмір криптографічної частини цифрового підпису;
- γ1, γ2, β – параметри, які використовуються для перевірки норм (16. 2) та (16.3).

При виконанні ЕП використовуються такі функції:

- PsevdoRandomInit – ініціалізація генератора псевдо – випадкових чисел;
- Hash – обчислення геш -значення (довжина геш – значення 2λ біт);
- unpack_sk – розпаковка секретного ключа;
- expand_matr – формування матриці A згідно ρ;
- matr_vector_mul – множення матриці на вектор S1;
- poly_mul, poly_add, poly_sub – функції множення, додавання та віднімання поліномів відповідно;
- int_array_to_os – перетворення коефіцієнтів поліному в рядок октетів;
- gen_hl – генерація старшої та молодшої частин для коефіцієнтів поліному;
- gen_y – генерація засліплюючого вектора Y;
- gen_c – генерація малого полінома (кількість ненульових коефіцієнтів 60);
- gen_h – функція для генерування вектора h з переносами в старшу частину;
- pack_ds – перетворення ЕП в рядок октетів;
- norma – обчислення норми вектора;
- get_count – кількість ненульових елементів.

6.2 Вхідні та вихідні дані

В якості *вхідних* повинні бути заданими такі параметри:

- m – повідомлення, рядок октетів;
- mlen – довжина повідомлення (байт);
- sk – особистий ключ, рядок байтів;
- sklen – довжина особистого ключа.

Вихідними є такі значення:

- success = ok, error (ok в разі успішного завершення і error в разі помилки);
- sm – цифровий підпис, рядок октетів довжиною CRYPTOBYTES

1 success := ok.

6.3 Алгоритм вироблення ЕП

1. Перевірка довжини цифрового підпису. Якщо dslen ≠ DSBYTES то success := error; вихід.

2. Перевірка довжини секретного ключа. Якщо sklen≠SKBYTES то success := error; вихід.

3. Розпаковка особистого ключа, отримання наступних компонентів: ρ, key, hpk, S1, S2, TLow (наприклад функція unpack_sk).

4. Обчислення геш- значення $\mu = Hash\ (h_{pk} \| m)$.

5. Відновлення матриці поліномів A[K][L] (функція expand_matr).

6. Ініціалізація генератора псевдо - випадкових чисел ctx := PsevdoRandomInit(key|| μ).

7. Обчислення засліплюючого вектора Y з L компонентами. Функція Y = gen_y (ctx).

8. Обчислення вектора W з K компонентами W := A * Y (функція matr_vector_mul).

9. Представлення вектора W у вигляді $W = \{W_{low}, W_{high}\}$, (функція get_hl);

10. Перетворення вектора W_{high} з K компонентами в рядок октетів $OS\_W_{high} := OS\_W_{high}[0] \| OS\_W_{high}[1] \ldots OS\_W_{high}[K-1]$ (функція int_array_to_os для обчислення $OS\_W_{high}[i]$.

11. i := 0..L-1).

12. Генерація поліному *c* (функція c =gen_c (μ || OS_W_{high}).

13. Обчислення вектора Z= c * S1 + Y (функції poly_mul, poly_add для L компонентів векторів).

14. Обчислення норми вектора Z, n_z := norma (Z, L).

15. Перевірка норми. If nz > (γ1 – β) то перейти на крок 7.

16. Обчислення вектора W_{CS2} := (W - C * S2) (функції poly_mul, poly_sub для K компонентів векторів).

17. Представлення W_{CS2} := { W_{CS2low}, $W_{CS2high}$} (функція get_hl для K компонентів вектора WCS2).

18. Обчислення норми вектора W_{CS2low}, nz := norma (W_{CS2low}, K).

19. Перевірка норми. If nz > γ2 – β то перейти на крок 7.

20. Перевірка умови $W_{CS2high} == W_{high}$. Інакше перейти на крок 5.

21. Обчислення вектора $CT_{low} = T_{Low} * c$.

22. Обчислення норми вектора CT_{low}, nz := norma (CT_{low}, K).

23. Перевірка норми. If nz > (γ2 – β), то перейти на крок 7.

24. Обчислення вектора t_{mp} := CT_{low} +W_{CS2} (функція poly_add для K компонентів векторів).

25. Обчислення вектора h (функція gen_h: h:= (CT_{low}, tmp))

26. Обчислення кількості ненульових елементів count в векторі h (функція get_count, count := get_count(h))

27. Якщо count > Ω, то перейти на крок 7.

28. Упаковка значення ЕП (функція pack_ds (Z, h, c)).

7. Алгоритм перевірки ЕП

7.1 Вхідні та вихідні значення

Вхідними значеннями для перевірки ЕП є такі:

- m – повідомлення, що перевіряється;
- mlen – довжина повідомлення, що перевіряється (байтів);
- sm – електронний підпис повідомлення, що перевіряється;
- smlen – довжина електронного підпису (байтів);
- pk – відкритий ключ перевірки ЕП;
- p_{klen} – довжина відкритого ключа;

Вихідним значення перевірки ЕП є результат перевірки:

0 – ОК, 1 – Помилка.

7.2 Перевірка ЕП

1. Перевірка довжини ЕП s_{mlen}. Якщо $s_{mlen} \neq$ CRYPTO_BYTES – повернути результат – помилка.
2. Перевірка значення p_{klen}. Якщо $p_{klen} \neq$ PKBYTES – повернути результат – помилка.
3. Розпакування відкритого ключа, визначення ρ, t1 (масив векторів).
4. Розпакування ЕП та визначення z, h, c.
5. Перевірка кількості ненульових елементів в поліномі c. Ця кількість не повинна перевищувати ω. Інакше повернути - помилка.
6. Перевірка кількості ненульових елементів в h. Кількість повинна бути від 60. Інакше повернути - помилка.
7. Перевірка норми вектора z, вона не повинна перевищувати значення GAMMA1 – β. Інакше повернути – помилка.
8. Обчислення геш – значення відкритого ключа h := Hash (p_k).
9. Створення буферу зі значенням buf := h || m
10. Обчислення гешу для буферу μ:= Hash (buf)
11. Генерація матриці A[K][L][N] згідно з ρ A:= ds_expand_matr(ρ)
12. Обчислення добутку матриці A та вектору z Az:= A * z
13. Відновлення старших частин коефіцієнтів поліномів.
14. i:= 0
15. Зсув старшої частини p1[i] := p1[i] << D;
16. Обчислення добутку cp1:= c * p1[i].
17. Обчислення Azcp [i] := (Az [i] – cp1);
18. Формування молодшої та старшої частин для Azcp [i] відносно α: low, high [i]:=gen_hl (Azcp [i], α)
19. Відкоректувати старшу частину для поліному high з урахуванням h та молодшої частини high [i]:=use_carry (high [i], low, h)
20. i:=i+1
21. if i < K то перейти на 14.
22. Відновлення поліному c. Позначимо відновлений поліном c’.
23. Перетворення старших частин high [i] вектора high в рядок байтів: os_high := int_array_to_os (high)
24. Сформувати буфер з μ та старшої частини buf := μ || os_high

25. Обчислити геш для buf: h := Hash (buf)

26. Формувати поліном c’: c’ := gen_c (h).

27. Порівняти відновлене c’ та отримане (пункт 4) задане значення поліномів c. success = ok, якщо поліноми співпадають, error, якщо не співпадають, то повернути значення success.

8. Оцінка фактичної стійкості механізму Cristals-Dilithium проти атак

Як показав аналіз значення параметрів механізму Cristals-Dilithium, що наведені в табл. 8 [171 нижче], *не забезпечують стійкість схеми електронного підпису на рівні 256 бітів*. Фактично нижні оцінки стійкості наведено в рядках табл. 16.8 з назвами Best Known Classical bit-cost і Best Known Quantum bit-cost окремо для кожної з двох задач, на складності яких базується стійкість механізму Cristals-Dilithium. Це задачі криптоаналізу, що отримали назву SIS та LWE [171].

Для кожної з них зазначені параметри захищеності від класичної атаки запропоновано обчислювати згідно формули:

$$\text{BKC (Best Known Classical bit-cost)} = 0{,}292b. \qquad (16.7)$$

Відповідно захищеність від квантової атаки на основі квантового комп’ютера запропоновано обчислювати згідно формули:

$$\text{BKQ (Best Known Quantum bit-cost)} = 0{,}265b, \qquad (16.8)$$

де b є довжиною блоку (BKZ bock-size b to break SIS або LWE [171]). В якості кінцевої оцінки стійкості використовується найменше з двох значень (16.7) та (16.8), обчислених для b, що є довжиною блоку для задачі SIS та задачі LWE відповідно.

В табл. 16.8 наведені значення довжини блоку, отримані за допомогою (16.7) та (16.8), та відповідні значення цього параметра [171 нижче].

Таблиця 16.8.

Довжини блоку, отримані за допомогою (16.7) та (16.8)

(k, l)	(3, 2)	(4,3)	(5,4)	(6, 5)
η	**7**	6	5	3
BKZ bock-size b to break LWE [171]	200	340	485	595
Формула (16.6)	220	373	525	639

Наприклад, для останнього стовпця табл. 8 в [171] (третя категорія стійкості за вимогами NIST) отримуємо, що

$$\text{BKZ(bock-size } b \text{ to break SIS)} = 605.$$

Значення (16.7) та (16.8) дорівнюють відповідно:

$$0{,}292\, b = 176,\ \ 0{,}265\, b = 160.$$

Наведемо значення для LWE.

$$\text{BKZ (bock-size } b \text{ to break LWE)} = 595.$$

Тому значення (16.7) і (16.8) в цьому випадку дорівнюють відповідно:

$$0{,}292\ b = 174 \text{ та } 0{,}265\ b = 158$$

(в табл. 16.8 наведені значення з округленням до цілих, причому не завжди у той самий бік).

Таким чином, кінцева нижня оцінка стійкості схеми ЕП механізму Cristals-Dilithium для випадку, що розглядається, є 174, тобто схема забезпечує стійкість відносно класичних атак на рівні не менше ніж 2^{174}, а відносно квантових – на рівні не менше ніж 2^{158}.

Відмітимо, що довжина блоку b обчислюється за параметрами k, l, η з табл. 8 [171] за допомогою окремих алгоритмів для кожної задачі SIS та LWE. Яких саме, потребує додаткового вивчення, але автори роботи [171] використовують для цього методику, наведену в [334, 335]. Вони зазначають ([171, с. 22], що ця методика дозволяє отримувати "консервативні" нижні оцінки стійкості, які "швидше" за все, не зміняться найближчим часом з появою більш ефективних алгоритмів розв'язання задач SIS та LWE, зокрема, на квантових комп'ютерах. Також автори [171] відмічають, що отримані ними оцінки формально є меншими в порівнянні з вимогами NIST, але вони вважають (belive), що значення для параметрів з табл. 8 механізму Cristals-Dilithium задовольняє цим вимогам.

Таким чином, найвищий рівень класичної стійкості, що забезпечує Cristals-Dilithium з параметрами, зазначеними в [171], табл. 1, є (принаймні) 2^{174}, а щодо квантових атак не менше ніж 2^{158}.

При подальшому аналізі приймемо *загальні обмеження, які треба врахувати при виборі параметрів для забезпечення стійкості на рівні* λ *бітів, де* $\lambda \in \{256\}$, наступні:

1. Геш-функція CRH, що використовується у механізмі (наприклад, в рядках 07, 10 на рис. 4 [171]), повинна бути стійкою до колізій. Отже, довжина її вектора значень повинна бути, як мінімум, 2λ бітів (зокрема, при $\lambda = 256$ функція CRH повинна приймати значення довжини 512, а не значення 384, як в оригінальній схемі на рис. [1]).

2. Криптографічна функція H, яка використовується в рядку 16 на рис. 4 [171], повинна бути стійкою відносно знаходження другого прообразу [336] і, отже, приймати, принаймні, 2^{λ} різних значень, що є поліномами з кільця R_q, які мають коефіцієнти $0, 1, -1$ та містять точно h ненульових коефіцієнтів.

Кількість таких поліномів дорівнює:

$$2^h \binom{n}{h}. \tag{16.9}$$

При $\lambda = 256$ для Cristals-Dilithium в [171] рекомендується використовувати параметри $n = 256$, $h = 60$, і умова:

$$2^h\binom{n}{h} \geq 2^\lambda \qquad (16.10)$$

виконується.

3. Довжини векторів ρ та K, що використовуються відповідно в рядках 01 та 02 на рис. 4 в [171], повинні бути не менше ніж λ.

Таким чином, *для забезпечення стійкості схеми електронного підпису на рівні* $\lambda \in \{256\}$ *необхідно*:

1) використовувати геш-функцію CRH, значеннями якої є двійкові вектори довжини 2λ;

2) використовувати двійкові вектори ρ та K, що мають довжину λ;

3) покласти $n = 256$, якщо $\lambda = 256$;

4) вибрати просте число $q \equiv 1 \, mod \, 2n$ та обчислити γ_1, γ_2 згідно наведених нижче значень параметрів:

$$q = 2^{23} - 2^{13} + 1,$$

$$\gamma_1 = (q-1)/16, \; \gamma_2 = \gamma_1/2; \qquad (16.11)$$

5) обчислити weight of c (див.[171],табл.1) як найменше натуральне h, що задовольняє умові (16.10).

Висновки

1. Наразі створені та застосовуються 20, 53 та 72 – кубітні квантові комп'ютери. Тому актуальною стає проблема захисту інформації та ресурсів від існуючих та потенційних криптоаналітичних атак.

2. В ході семінару другого етапу NIST США рекомендував до подальших досліджень 9 криптопримітивів ЕП. Дані щодо них, асиметричних шифрів (АСШ) та протоколів інкапсуляції ключів (ПІК) наведено в таблиці 16.1.

3. Дослідження можливостей створення ЕП, що можуть бути стійкими у постквантовий період, підтверджують, що надійною математичною основою для цього можуть бути алгебраїчні решітки. По суті, такий математичний апарат пройшов випробовування часом у вигляді регіонального стандарту X9.98.

4. На конкурс NIST США щодо стандарту ЕП було подано механізми ЕП, що ґрунтуються на алгебраїчних решітках, Cristals-Dilithium, FALCON та qTESLA. Наразі вони досліджуються на другому етапі конкурсу NIST США.

5. Побудування ЕП Cristals-Dilithium ґрунтується на підході "Fiat-Shamir з перериваннями". В ньому множення поліномів виконується в кільці поліномів $R_q = Z_q[X]/(X^n+1)$. Причому всі алгебраїчні операції в механізмі виконуються над кільцем полінома R_q, q=2^{23}–2^{13}+1 та n=256.

6. Оцінка та порівняння ЕП на алгебраїчних решітках здійснені на використанні узагальненої методики, що складається з методик на основі безумовних, умовних та прагматичних складових.

7. На основі аналізу даних рисунку 2 [171], можна зробити висновок. що кращим є механізм Cristals-Dilithium, уваги також заслуговують механізми picnicl5fs, falcon1024, falcon768.

8. Механізм Cristals-Dilithium є ЕП, що має надійну безпеку від атак на вибране повідомлення, його стійкість базуються на складності проблем решітки над модульними решітками. Поняття безпеки означає, що порушник, який має доступ до оракула підпису, не може виробити підпис повідомлення, підпис якого він ще не бачив, а також не може створити інший підпис повідомлення, яке він вже бачив підписаним.

9. Механізм і ЕП Cristals-Dilithium є консервативними за параметрами, що дозволяє зменшити розмір відкритого ключа та відносно легко дозволяє змінювати рівень криптографічної стійкості засобом зміни розмірів параметрів та ключів.

10. В $2^{23} - 2^{13} + 1 = 8380417$ алгебраїчні операції виконуються в кільці поліномів $Z[X]/(X^n+1)$ та $R_q = Z_q[X]/(X^n+1)$ для цілого числа q. В якості параметрів рекомендується в якості значення степені поліному n вибрати 256, а в якості модуля q просте число $2^{23} - 2^{13} + 1 = 8380417$. При даних параметрах може бути забезпечений 5 рівень криптографічної стійкості від класичних та квантових атак.

11. На основі детального аналізу обґрунтовані та вибрані розміри параметрів та ключів, при використанні яких застосування механізму Cristals - Dilithium забезпечує 5 рівень безпеки. Основними параметрами є розміри матриць перетворень K та L.

12. Фактично нижні оцінки стійкості наведено для вибраних параметрів в табл. 16.8 з назвами Best Known Classical bit-cost і Best Known Quantum bit-cost окремо для задач, що отримали назву SIS та LWE.

13. Кінцева нижня оцінка стійкості схеми ЕП механізму Cristals-Dilithium для випадку, що розглядається, є 174, тобто схема забезпечує стійкість відносно класичних атак на рівні не менше ніж 2^{174}, а відносно квантових – на рівні не менше ніж 2^{158}. Тобто, найвищий рівень класичної стійкості, що забезпечує Cristals-Dilithium з параметрами, зазначеними в табл. 16.8, є принаймні 2^{174}, а щодо квантових атак не менше ніж 2^{158}.

16.2 Основні положення та результати порівняння властивостей електронних підписів постквантового періоду на алгебраїчних решітках

Вступ

Постквантові проєкти стандартів електронних підписів (ЕП) Falcon [57] та Dilithium [256] є фіналістами, тобто, двома із трьох переможців другого раунду конкурсу NIST США [257]. Наразі вони досліджуються на третьому раунді і, при позитивних результатах дослідження, можуть бути прийняті в якості міжнародних постквантових стандартів ЕП. При їх побудуванні використано математичний апарат алгебраїчних решіток та відповідні методи. При

подальшому дослідженні та порівнянні вказаних постквантових проєктів стандартів ЕП як з теоретичних, так і практичних позицій основоположним є обґрунтування вимог до параметрів та ключів та у цілому обчислення основних показників згідно прийнятих умовних та безумовних критеріїв [258, 259]. Важливим при таких дослідженнях є визначення достатності забезпечення гарантованості їх захищеності від класичних, квантових, спеціальних та атак на основі помилок [57,256-259]. Вказане може бути забезпеченим, у тому числі, засобом обґрунтованого вибору розмірів загальних параметрів та ключів [260,261] та практичного їх побудування згідно з прийнятою моделлю безпеки [258,262,263]. Але при виборі розмірів загальних параметрів та ключів виникає суттєве протиріччя між властивостями проєктів стандартів ЕП Falcon та Dilithium щодо стійкості та складності перетворень. Так, збільшення розмірів загальних параметрів та ключів приводить до збільшення складності перетворень і навпаки [260,261].

Що стосується теоретичних методів щодо побудування загальних параметрів та ключів для ЕП Falcon та Dilithium, то вони у цілому зрозумілі, а якщо використовувати [67,256,260,261], то існує можливість їх побудувати і для більш високих рівнів безпеки, наприклад, запропонованих та реалізованих [260,261] у 6 та 7 рівнях безпеки, коли може бути забезпечено 384 та 512 біт захисту від класичних атак і 192 та 256 біт захисту від квантових атак, а також захищеність від спеціальних атак та атак на основі помилок [57,256,257,264–271]. У цілому, в певному плані, існує необхідність уточнення теоретичних питань та безумовна необхідність практичного побудування загальних параметрів та ключів для забезпечення 6 та 7 рівнів безпеки та оцінки і порівняння вказаних проєктів ЕП згідно із запропонованими безумовними, умовними та прагматичними критеріями [260,261].

Причому, вирішення цих проблемних питань зводиться до реалізації відповідних моделей безпеки, побудованих на основі реальних моделей порушника та загроз.

Наведемо основні питання, які будуть розглянуті в цьому розділі.

1) Аналіз проблемних питань вибору розмірів параметрів та ключів для постквантових проєктів ЕП, побудованих на основі математичних методів Falcon [57] та Dilithium [256], та особливості їх реалізації, в тому числі, реалізації відповідно до прийнятої моделі безпеки.

2) Порівняльний аналіз стійкості та складності проєктів стандартів ЕП Falcon та Dilithium у залежності від розмірів параметрів та ключів, в тому числі, для 6 та 7 рівнів безпеки.

3) Розробка пропозицій стосовно рішень щодо прийняття в якості національних постквантових стандартів ЕП на основі математичних методів Falcon та Dilithium.

4) Визначення впливу безумовних, умовних та прагматичних критеріїв на переваги при прийнятті рішення щодо стандартизації ЕП на основі математичних методів Falcon та Dilithium, в тому числі, з урахуванням наявності патентів та необхідності отримання ліцензій тощо.

16.2.1 Основні параметри та ключі ЕП Falcon та Dilithium

1. Основні параметри та ключі алгоритму Falcon та дані щодо їх розмірів

Нехай розмір відкритого ключа є (PK_SIZE), розмір ЕП є (DS_SIZE), а час (складність) перевірки підпису є (CHECK_TIME). В якості загального параметра виберемо степінь поліному ***n.*** Основний розгляд проведемо для п'ятого рівня безпеки щодо захищеності від квантових атак з використанням квантових комп'ютерів [57,79,188,256,272]. Вище прийняті позначення вибрані з урахуванням узгодженості з програмним забезпеченням.

При дослідженнях використано наступні формули для алгоритму Falcon [57].

Довжина відкритого ключа:

PK_SIZE=1+n*$\log_2 q$, де:

n – степінь полінома, модуль, що визначає основне поле x^n+1, n=1024 (5 рівень безпеки);

q – просте число, модуль для коефіцієнтів, q=12289, $\log_2 q$=14;

DS_SIZE – довжина ЕП.

ЕП обчислюється з використанням довжини випадкового компонента *nonce* (40 октетів), упакованого формату поліному, коефіцієнти якого задовольняють розподілу Гауса. Довжина коефіцієнта в упакованому форматі залежить від його значення і може займати від 8 до 23 бітів. Тому щодо довжини ЕП автори використовують середнє значення.

2. Основні параметри та ключі алгоритмів Dilithium та Falcon та дані щодо їх розмірів

При дослідженнях використаємо такі основні формули для алгоритму Dilithium [256].

Довжина відкритого ключа:

PK_SIZE=SEED_SIZE+n*k*($\log_2 q$–d), де:

SEED_SIZE – розмір випадкового компоненту для відновлення матриці A (32 октетів);

n – степінь полінома, модуль, визначає основне поле x^n+1, n=256;

q – просте число, модуль для коефіцієнтів, q=8380417;

d – кількість бітів в молодшій частині коефіцієнтів поліному (d=13);

k – довжина вектора, який складається з поліномів. Кожний поліном містить n коефіцієнтів розміром w=$\log_2 q$–d бітів; k=8, w=10.

Довжина ЕП Dilithium:

DS_SIZE=SEED_ SIZE+$l^* n^* \log_2 2\gamma_1 + (\omega + k)$, де:

SEED_ SIZE – розмір компоненти для відновлення полінома за модулем 3, який складається із заданої кількості ненульових елементів;

γ_1 – задає розмір коефіцієнта поліномів, $\gamma_1 = 2^{19}$;

l – довжина вектора, який складається з поліномів (l =7). Кожний поліном містить n коефіцієнтів $\omega = \log_2 2\gamma_1$ (ω=20);

ω – кількість ненульових елементів в бітовій матриці переносів h (ω=75);

k – кількість рядків в матриці переносів (k=8).

Результати розрахунків розмірів відкритих ключів (PK_SIZE), ЕП (DS_SIZE) та складностей (часових) перевірки підписів (CHECK_TIME) для алгоритмів Falcon та Dilithium для 5-го рівня безпеки наведені в табл. 16.9.

Таблиця 16.9.

Порівняння основних показників для алгоритмів Falcon та Dilithium

Алгоритм	PK_SIZE (октетів)	DS_SIZE (октетів)	CHECK_TIME (тактів)
Dilithium, n=256	2592	4595	279936
Falcon, n=1024	1793	1233.29^{1}	168498

Як видно з табл. 16.9, алгоритм Dilithium, згідно з наведеними параметрами, програє алгоритму Falcon.

Розмір ЕП змінюється в зв'язку з особливістю кодування полінома, коефіцієнти якого задовольняють розподілу Гауса. Кількість бітів для кодування одного коефіцієнта може змінюватися в інтервалі 8 – 24 біта, 8 бітів – найбільша імовірна довжина, 24 біта – найменш імовірна.

3. Порівняння основних параметрів та ключів ЕП Falcon та ЕП Dilithium для алгоритмів 7 рівня безпеки

На основі математичних методів Falcon [57] та Dilithium [256] розроблено проєкти стандартів "Вершина 1" та " Вершина 2" 7 рівня криптостійкості 512 біт захисту від класичних атак та 256 біт від квантових атак [260,261]. В табл. 16.10 наведені основні показники для цих алгоритмів в разі збільшення степені n та застосування решти необхідних параметрів.

Таблиця 16.10.

Порівняння основних показників для алгоритмів 7 рівня безпеки, побудованих на базі методів ЕП Falcon та Dilithium

Алгоритм	PK_SIZE (октетів)	DS_SIZE (октетів)
Dilithium, n=512	5824	10708
Falcon, n=2048	3585	4884

Також, аналогічно, як із табл. 16.9, із табл. 16.10 випливає, що алгоритм Dilithium відповідно до розглянутих параметрів програє алгоритму Falcon, але генерація ключів та формування ЕП для алгоритму Falcon та його прототипів потребує від розробника застосування багатьох різних алгоритмів та форматів, що може привести до значного звуження практичного застосування алгоритму ЕП Falcon.

4. Рекомендації щодо практичної реалізації алгоритму ЕП Falcon

Нижче розглянуто практичні рекомендації із генерації ключів для алгоритму Falcon [57]. На наш погляд, цей опис може значно полегшити реалізацію алгоритму для розробників. Ці рекомендації особливо актуальні в умовах збільшення значення степені полінома n до 2048 для забезпечення підвищення криптостійкості включно до 7 рівня [258,259]. Для цього, в якості решітки, в алгоритмі застосовується NTRU - решітка.

В якості секретних ключів для забезпечення 5 рівня безпеки [57,257] застосовуються:

- короткі вектори f, g, коефіцієнти яких задовольняють розподілу Гауса та за модулем не перевищують 32;
- короткі вектори F, G, коефіцієнти яких за модулем не перевищують 128 та задовольняють NTRU рівнянню:

$$f*G–Fg=q. \tag{16.12}$$

В якості відкритого ключа застосовується поліном, обчислений за формулою

$$h=f^{-1}*g \text{ в полі } Z[x]/(\phi, q), \tag{16.13}$$

де ϕ – поліном $x^{1024}+1$.

Для обчислення відкритого ключа h застосовується NTT- формат [57], що допустимо завдяки спеціальному вибору q. Але випадковий поліном f може не мати інверсію в полі $Z[x]/(\phi, q)$, тоді необхідно сформувати нову пару поліномів f, g. Обчислення ключів f, g, коефіцієнти яких задовольняють розподілу Гауса, нетривіальна задача, особливо в умовах забезпечення захисту від спеціальних атак на основі константного часу обчислень. Обмеження на значення квадратичної норми для поліномів f, g також може привести до необхідності повтору операцій, що впливає на час генерації ключів. Перевірка квадратичної норми виконується не тільки для самих поліномів f, g, а також для поліномів, обчислених за формулою

$$f' = \frac{qf^*}{ff^*+gg^*}, g' = \frac{qg^*}{ff^*+gg^*}. \tag{16.14}$$

Для обчислень (16.12) використовується комплексне подання чисел. Позначення f^*, g^* означають комплексно поєднані значення. Для поліномів f, g виконання операцій множення та ділення поліномів застосовують FFT формат. Таким чином, обчислення f, g, h потребує застосування двох спеціальних форматів NTT та FFT, а для множення за модулем застосовують арифметику Монтгомері [57,275].

Для рішення NTRU - рівняння в [57] запропонована така методика.

4.1 Поступовий перехід від поліномів для поля x^n+1 до поля $x+1$

Поліном в полі x^n+1 містить n коефіцієнтів. При поступовому переході з поля x^r+1 до поля $x^{r/2}+1$ кількість коефіцієнтів зменшується в 2 рази, а значення цих коефіцієнтів збільшується. Для перетворення поліному p, заданого в полі x^r+1, в поліном $\boldsymbol{p}$, який заданий в полі $x^{r/2}+1$, виконуються наступні операції:

4.1.1 Для полінома p формуються два поліноми p_1 p_2 розміром r/2.

Для цього в поліном p_1 записуються коефіцієнти поліному p з парними, а в поліном p_2 – з непарними номерами, тобто,

$$p_1\ [k]=p\ [2k];\ p_2\ [k] = p[2_k+1];\ (k = 0, \ldots\ r/2–1);$$

4.1.2 Обчислюються поліноми $e_2 = p_1^2$ та $o_2 = p_2^2$ за модулем $x^{r/2}+1$;

4.1.3 Обчислюються коефіцієнти поліному – результату p_ за формулами:

- $p\_[i+1]=e_2[i+1]–o_2[i]$; (i=0…r/2–2),
- $p\_[0]=e_2[0]+o_2[i])$; ($i=t/2–1$).

Однакові операції виконуються для поліномів *f, g*. Позначимо відповідні результати *pf_* та *pg_* для поліномів *f, g* відповідно.

Для виконання операцій над коефіцієнтами необхідно виконувати опрацювання довгих чисел. Ці операції виконуються без врахування модуля *q*. Так, при переході від поля $x^{1024}+1$ до поля x^1+1 отримаємо поліноми *f'', g',* які містять по одному коефіцієнту, значення якого для поля x^1+1 займає більше, ніж 6000 бітів, тобто, операції необхідно виконувати над даними, довжина яких значно перевищує довжину даних для інших несиметричних алгоритмів, наприклад, RSA.

Також в [57] не зберігають результатів перетворень, а для кожного поточного значення *r* для виконання зворотного перетворення обчислюють їх повторно, тому рекомендується їх зберігати, що збільшить навантаження на пам'ять, але зменшить час обчислення для кроку 3.

В результаті виконання цієї операції отримаємо два масиви поліномів, один - для поліному *f,* інший - для поліному *g*. Позначимо ці масиви відповідно *pf, pg, pf* [0], що дає співпадіння поліному *f, gp* [0] з поліномом *g*. *pf* [1] – результат перетворення полінома *f* в поліном в полі $x^{n/2}+1$, а *pg* [1] – результат перетворення полінома *g* з поліному в полі $x^{n/2}+1$, а поліном $pf[\log_2 n]$ – результат перетворення полінома *f* в поліном в полі $x+1$, $pg[\log_2 n]$ – результат перетворення полінома *g* в поліном в полі $x+1$. Поліноми $pf[\log_2 n]$, $pg[\log_2 n]$ містять по одному коефіцієнту. Позначимо їх *lf, lg* відповідно.

4.2 Для отриманих значень виконується розширений алгоритм Евкліда для вирішення діафантового порівняння і знаходження значень *s, t* та найбільшого спільного дільника для *lf, lg*:

4.2.1 $lf*s+lg*t=gcd\ (lf, lg)$;

4.2.2 Якщо $gcd\ (lf, lg) \neq 1$, то знову переобчислюються значення *f, g, h*. Якщо $gcd\ (lf, lg)=1$, то після обчислення $lG=q*s$ та $lF=-q*t$ отримаємо тотожність:
$lf*lG–lg*lF=q$ для поля $x+1$.

Значення *lG, lF* розглядаються, як поліноми *G, F* для поля x^1+1, вони записуються замість $pf[\log_2 n]$ та $pg[\log_2 n]$ і далі необхідно для них виконати зворотне перетворення поліномів для обчислення *F* та *G* для поля x^n+1.

4.3 Виконується поступовий зворотний перехід від поля $x+1$ до полів x^2+1, …, x^n+1. Вхідними даними для цього етапу є масиви поліномів *pf, pg,* отримані для полів $x+1$, x^2+1, …, x^n+1. Результатом є поліноми *F, G*.

Для кожного $r = 1, 2, 4, \dots n/2$ виконуються кроки 4.3.1–4.3.3.

4.3.1 Для зворотного переходу від поліному, заданому в полі x^r+1, до поліному p, заданому в полі $x^{2r}+1$, застосовуються поліноми pfr та pfr_2 з масиву pf для полів x^r+1, $x^{2r}+1$, поліноми pgr та pgr_2 з масиву $\underline{p\ g}$ для полів x^r+1, $x^{2r}+1$ відповідно. Далі, в процесі перетворення, виконуються такі операції:

- створюється 2 поліноми p_1, p_2 для поля $x^{2r}+1$. Поліном p_1 формується з поліному pfr за рахунок виконання операцій:

$$p_1[2k]=pfr\,[k];\ p_1[2k]=0;\ (k = 0, 1, \dots, r).$$

Поліном p_2 формується з поліному pg_2r за рахунок виконання таких операцій: $p_2[2k]=pg_2r[2k]$; $p_2[2k+1]=-pg_2r[2k+1]$; (k=0, 1, …,r);

- обчислюється поліном pF для поля $x^{2r}+1$: $pF_2r=p_1*p_2$ для поля $x^{2r}+1$;

- створюється 2 поліноми p_1, p_2 для поля $x^{2r}+1$. Поліном p_1 формується з поліному pgr за рахунок виконання операцій: $p_1[2k]=pgr[k]$; $p_1[2k]=0$; ($k = 0, 1, \dots, r$);

Поліном p_2 формується з поліному pf_2r за рахунок виконання операцій: $p_2[2k]=pf_2r\,[2k]$; $p_2[2k+1]=-pf_2r[2k+1]$; ($k = 0, 1,\dots, r$);

- обчислюється компонент масиву pG для поля $x^{2r}+1$: $pG_2r=p_1*p_2$ для поля $x^{2r}+1$.

4.3.2 Після переходу від поля x^r+1 до поля $x^{2r}+1$ виконується операція зменшення коефіцієнтів поліномів (reduce).

Вхідні дані для цієї операції:

значення r ;

поліноми для поля $x^{2r}+1$: $pf_2\,r$, $pg_2\,r$;

поліноми pF, pG.

Результатом виконання таких операцій є поліноми pF, pG.

В результаті виконання цієї операції довжини коефіцієнтів поліномів зменшуються таким чином, щоб вони могли записатися в число з плаваючою точкою без втрати значущих цифр (для поточної реалізації передбачається 53 біта). Зменшення довжини виконується за рахунок виділення старших цифр поліномів.

Операція спочатку виконується для пари поліномів (pf_2r, pg_2r). Для пари визначається максимальна довжина коефіцієнта поліному (max бітів), обчислюється параметр зсуву (scale=max–53) і виконується зсув усіх коефіцієнтів поліномів пари на задане значення в сторону молодших бітів (операція ділення коефіцієнтів на $2^{\text{scale}_\text{T}}$). В результаті отримаємо поліноми, в яких довжина кожного коефіцієнта не перевищує 53 біта.

Обчислюється загальна квадратична норма для поліномів pf_2r, pg_2r. Позначимо її pfg (операція виконується для FFT формату).

4.3.3 Далі виконується цикл поступового зменшення довжини коефіцієнтів для пари (pF, pG):

- для пари визначається максимальна довжина коефіцієнта полінома (Max бітів);

- якщо Max < max, то подальше зменшення неможливо, виконується вихід з циклу;

- обчислюється параметр зсуву (scale=Max–53) і виконується обчислення $lF_1=lF/2^{scale}$, $lG_1=lG/2^{scaleт}$;

- обчислюється загальна квадратична норма для поліномів lF_1, lG_1. Позначимо її *pFG* (формат FFT);

- обчислюється значення $d=pFG/pfg$ (формат FFT);

- перехід від FFT формату до звичайного формату і округлення коефіцієнтів поліному *d* до цілих коефіцієнтів, якщо усі коефіцієнти дорівнюють 0, то вихід з циклу;

- обчислення $lF_1=d*lf \bmod x^{2r}+1$; $lG_1=d*lg \bmod x^{2r}+1$ (*lf*, *lg* – поліноми в полі $x^{2r}+1$);

- обчислення $lF=lF–lF_1*2^{Max-max}$; $lG=lG–lG_1*2^{Max–max}$.

Таким чином, генерація ключів передбачає застосування наступних методів і форматів:

- формування поліномів з коефіцієнтами, розміри яких обмежені і задовольняють розподілу Гауса;

- застосування NTT формату для фіксованого простого *q* для формування відкритого ключа і множення поліномів в разі застосування CRT формату представлення великих чисел;

- застосування FFT формату для виконання операції ділення для поліномів.

Перехід на початок формування ключів виконується, якщо:

- поліном *f* не має зворотного значення (16.12);

- квадратична норма коефіцієнтів поліномів *f* і *g* перевищує задане значення;

- квадратична норма коефіцієнтів поліномів *f'* і *g'* (16.13) перевищує задане значення;

- найбільший загальний дільник поліномів *lf*, *lg* не дорівнює 1.

Як показує аналіз алгоритмів формування ключів, найбільш ресурсно важкими є операції множення поліномів за модулем з коефіцієнтами – довгими числами. Зазвичай ці операції виконуються для декількох поліномів, що спрощує задачу їх паралельного виконання. Зберігання проміжних даних в загальній пам'яті, що практикують автори [57], напроти, ускладнюють задачу паралельного виконання. Тому рекомендується для тимчасового зберігання даних в функціях застосовувати локальну пам'ять.

5. Методи та алгоритми побудови загальних параметрів для ЕП типу Falcon у залежності від їх розмірів

5.1 Постановка проблеми побудування загальносистемних параметрів для Falcon N для 256 та 512 біт безпеки

Одним із основних завдань конкурсу є розробка та прийняття постквантового чи постквантових стандартів ЕП. Фіналістами другого етапу конкурсу NIST стали три механізми ЕП – Dilithium, Falcon та Rainbow [264].

Попередні дослідження показали, що серед схем ЕП на решітках дещо відрізняється від інших кандидатів Falcon [57,273], він також має перспективи щодо прийняття в якості міжнародного стандарту ЕП. Домінуючим основним концептуальним підходом до проєктування механізму ЕП Falcon є використання перетворення типу «геш-і-підпис» [57,273]. Перевагою такого підходу є доведена стійкість в межах моделі квантового випадкового оракула. В процесі досліджень виявлено, що навіть його аналізу присвячено значно менше робіт, ніж щодо інших, наприклад, проєкту ЕП Dilithium [256]. Крім того, як і щодо інших, при проєктуванні ЕП Falcon були прийняті обмеження щодо максимальних рівнів безпеки у вигляді максимально 256 біт проти класичного та 128 біт проти квантового криптоаналізу. Тому, як з точки зору теорії, так і практики генерація загальносистемних параметрів для використання 384 і 512 біт безпеки проти класичного криптоаналізу та 192 і 256 біт безпеки проти квантового криптоаналізу є важливою проблемною задачею.

В якості основних властивостей розглянемо результати первинного аналізу щодо відомих атак на ЕП Falcon. При цьому врахуємо обмеження та практичні алгоритми обчислення загальносистемних параметрів та їх оптимізації для 256 та 512 біт безпеки проти класичного та не менше 128 та 256 біт проти квантового криптоаналізу. Важливим є проведення додаткових досліджень та прийняття рішення щодо реалізації рівнів 384 біт безпеки проти класичного та не менше 192 біт проти квантового криптоаналізу щодо ЕП Falcon. Сутність механізму Falcon наведено в [57,273] і нижче не розглядається.

5.2. Аналіз атак на ЕП Falcon

Перетворення GPV [57], що використане в ЕП Falcon, вимагає, щоб геш-функція H була захищена від колізій. Це означає, що розмір солі в бітах повинен бути не меншим за 2λ, де λ – рівень безпеки, що вимагається. Проте, за умовами конкурсу NIST [257,258], кількість запитів на вироблення ЕП (signature queries) є не більшою за $qs=2^{64}$, що вимагає цього розміру у вигляді $\lambda+\log_2(qs)$. В табл. 16.11 наведено вимоги щодо розмірів для 5 – 7 рівнів безпеки ЕП Falcon.

Таблиця 16.11.

Розмір (ентропія) початкового значення r (солі) в бітах

Безпека	Розмір r	Розмір r з врахуванням вимог NIST
256	512	320
384	768	448
512	1024	576

Попередній аналіз показав, що основними атаками щодо ЕП Falcon є атаки на відновлення особистого ключа з відкритого ключа ЕП та атаки на підробку ЕП. Розглянемо ці атаки.

5.3 Атаки на відновлення особистого ключа ЕП Falcon

Атаки на відновлення особистого ключа з відкритого ключа щодо Falcon можуть зводитися до вирішення проблеми NTRU [57]. У ряді криптосистем, стійкість яких ґрунтується на проблемі NTRU, коли поліноми f та g мають коефіцієнти з множини значень {0, 1, –1}, виникає проблема забезпечення стійкості проти комбінованих атак. Це робить можливим реалізувати різні комбінаторні атаки. Наприклад, найефективнішою атакою є гібридна атака, яка знаходить частину вектора комбінаторними шляхами, якщо не збільшувати розмір модуля n [276].

Тому для захисту від цієї атаки потрібно збільшити рівень безпеки до 2^{512} [276] . Щодо Falcon такі атаки неможливі, оскільки поліноми f, g змінюються, точніше, вибираються згідно з нормальним розподілом із заданими параметрами. В даному випадку простір можливих значень поліномів збільшується настільки, що застосування комбінаторних методів стає неефективним. Тому залишається прямий шлях відновлення особистого ключа з відкритого засобом редукції базису решітки [57,269-271]. При цьому, чим менші значення має норма найменшого вектора (f, g), тим більша криптостійкість системи. В криптосистемі Falcon поліноми генеруються над полем

$$Z_q[X]/(\phi(x)), \deg(\phi) = n$$

з математичним очікуванням, рівним 0. Перетворення спираються на [57,273], у яких детально досліджувалися можливості застосування алгоритмів вибірки нормально розподілених величин.

А основною умовою захисту ЕП Falcon від атак на відновлення особистого ключа ЕП з відкритого шляхом редукції є така:

$$\left(\frac{B}{2\pi e}\right)^{\frac{2B-2n+1}{2B-2}} < \sqrt{\frac{3eB}{8n}}.$$

5.4 Атаки на підробку ЕП

Атаки на безпосередньо підробку ЕП можуть бути найбільш загрозливими. Тому іншим методом є атака підробки ЕП. При реалізації такої атаки потрібно знайти достатньо короткий вектор s. Відповідно, це можливо зробити, редукувавши базис так, щоб виконувалася умова [260]:

$$\| b_1^* \| < \beta.$$

Причому оцінити $\| b_1^* \|$ можливо таким же чином, що і у попередньому випадку, тобто, у такій послідовності:

$$\| b_1^* \| \approx GH(B)^{\frac{2n+1-2}{2(B-1)}} \det(\Lambda)^{\frac{1}{2n}} = GH(B)^{\frac{2n-1}{2B-2}} \sqrt{q} = \left(\frac{B}{2\pi e}\right)^{\frac{2n-1}{2B-2}} \sqrt{q}.$$

Звідси отримуємо умову захисту від атак на підробку підпису:

$$\left(\frac{B}{2\pi e}\right)^{\frac{2n-1}{2B-2}}\sqrt{q}<\beta.$$

Для практичного прийняття рішення необхідно визначитися щодо того, як обирати параметр β. Розробники ЕП Falcon в якості N пропонують використовувати значення $\sigma=1.55\sqrt{q}$ для полінома $\phi=x^n+1$ і $\sigma=1.32*2^{1/4}*\sqrt{q}$ для полінома $\phi=x^n-x^{n/2}-1$. Такий вибір σ базується на результатах робіт [57,273] і параметр β для полінома $\phi=x^n+1$ обчислюється як:

$$\beta=1.2*\sigma\sqrt{2nq}.$$

Для полінома $x^n-x^{n/2}-1$ β обчислюється таким чином:

$$\beta^2=\frac{(1.2*\sigma*2n\sqrt{q})^2}{n}.$$

Якщо підставити значення β у рівняння для оцінки захищеності від атак на підробку підпису, то також обидві сторони будуть пропорційні значенню $\sqrt{q}$. Середньоквадратичні відхилення при вибірці поліномів з нормального розподілу підібрані таким чином, щоб від q складність атаки не залежала. Проте, на параметр q існує безліч інших наступних міркувань [57,263,268,273]:

- для захисту від алгебраїчних атак q має бути простим числом;
- якщо q буде занадто малим (порядку $q \approx n$), то будуть можливі BKW атаки;
- якщо q буде занадто великим ($q \approx n^{2.83}$), то будуть можливі атаки на підполе;
- якщо використовується поле x^n+1, то для реалізації ефективного множення повинне виконуватися рівняння $q \equiv 1 \bmod 2n$;
- якщо використовується поле $x^n-x^{n/2}-1$, то для реалізації ефективного множення повинне виконуватися рівняння $q \equiv 1 \bmod 3n$.

Необхідно відмітити, що стійкість найкращого алгоритму пошуку найменшого вектора оцінюється як 2^0.292B, де B – розмір блоку при редукції. Якщо при криптоаналізі застосовувати алгоритм Гровера, то нижня оцінка класичної стійкості в 256 біт складає 2^0.265B квантової стійкості (при класичній стійкості в 256 біт). Тому для ЕП на решітках квантова стійкість при класичній стійкості 256 біт набагато більша, ніж 128 біт [273,274].

5.5 Точність арифметики з плаваючою точкою

Останнім невизначеним параметром, який необхідно вибрати для забезпечення рівнів стійкості 384 та 512 біт, є необхідна точність виконання операцій у арифметиці з плаваючою точкою. Розробники Falcon для теоретичної оцінки використовували роботу [57], проте точність обиралася з практичних експериментів. З [57] видно, що рівень безпеки λ слабо впливає на потрібну точність. Основний вплив має кількість запитів на підпис $q_s=2^{64}$, тому є надія, що 53 бітів буде достатньо. Проте, це питання є предметом подальшого

дослідження. Також до недоліку ЕП Falcon необхідно віднести використання арифметики з плаваючою точкою. Разом з використанням деревоподібних структур це ускладнює аналіз схеми до атак сторонніми каналами. Іншою проблемою є складність реалізації на малоресурсних пристроях.

5.6 Генерація параметрів для 256, 384, 512 біт стійкості ЕП Falcon

У цілому, якщо підсумувати наведене вище, знайти параметри n, q, β можна з наступної системи нерівностей [261]:

$$\begin{cases} \left(\frac{B}{2\pi e}\right)^{\frac{2n-1}{2B-2}}\sqrt{q} < \beta \\ \left(\frac{B}{2\pi e}\right)^{\frac{2B-2n+1}{2B-2}} < \sqrt{\frac{3eB}{8n}} \end{cases}, \qquad (16.15)$$

де параметр β визначається як $\beta = 1.2*\sigma\sqrt{2nq}$, якщо використовується поліном x^n+1, і $\beta^2 = \frac{(1.2*\sigma*2n\sqrt{q})^2}{n}$, якщо використовується поліном $x^n–x^{n/2}–1$.

На основі (16.13) розроблено програмне забезпечення, з використанням якого були обчислені параметри n, q та β_2 для ЕП Falcon для відповідних поліномів для 256, 384, 512 біт безпеки, що наведені в табл. 16.12. Отриману криптостійкість наведено в табл. 16.13. Криптостійкість наведено у форматі «стійкість» λ / розмір блоку.

Оцінки криптостійкості отримувалися з розміру блоку, як $0{,}265B$ та $0{,}292B$. Такий підхід вважається класичним і використовувався як авторами Dilithium, так і авторами Falcon [57, 256]. Проте оцінки є досить грубими. В табл. 16.13 для деяких квантових атак для 256 і 512 біт отримані значення є трохи меншими за необхідні, проте через грубість оцінки можна вважати, що вони досягають потрібного порога. Для 256 біт згенеровані параметри співпадають із параметрами, що згенеровані авторами Falcon.

Таблиця 16.12.

Основні загальносистемні параметри Falcon *N*
для 256, 384, 512 біт безпеки

Безпека	$\phi(x)$	n	q	β_2
256	x^n+1	1024	12289	87070769
384	$x^n–x^{n/2}–1$	1536	18433	174141539
512	x^n+1	2048	12289	200928983

Обґрунтування та сутність оптимізації за параметром β наведено в [260]. У табл. 16.14 наведено значення для параметрів n, τ, які можливо використовувати на практиці, та відповідні значення β, причому параметр q має значення 12289.

Таблиця 16.13.

Криптостійкість Falcon *N* до атак на основі редукції решіток

Безпека	Стійкість до відновлення ключа (класична)	Стійкість до відновлення ключа (квантова)	Стійкість до підробки підпису (класична)	Стійкість до підробки підпису (квантова)
256	273/936	248/936	269/922	244/922
384	413/1417	375/1417	430/1474	390/1474
512	554/1899	503/1899	599/2053	544/2053

Таблиця 16.14.

Альтернативні значення параметра β

n	τ	Ймовірність повтору	$\lfloor \beta \rfloor$
1024	1.1	10^{-9}	8382
	1.08	10^{-6}	8230
	1.07	0.0007	8153
	1.05	0.064	8001
2048	1.07	10^{-9}	11710
	1.055	10^{-6}	11545
	1.045	0.0002	11436
	1.03	0.025	11272

Оптимізовані альтернативні набори загальносистемних параметрів для 256 біт класичної стійкості зведені у табл. 16.15.

Оптимізовані альтернативні набори загальносистемних параметрів для 512 біт класичної стійкості зведені у табл. 16.16.

Таблиця 16.15.

Оптимізовані альтернативні набори загальносистемних параметрів для 256 біт класичної стійкості

n	q	$\lfloor \beta^2 \rfloor$	Підробка ЕП клас. (біт)	Підробка ЕП квант. (біт)	Відновлення ключа клас. (біт)	Відновлення ключа квант. (біт)
1024	12289	70265242 (τ=1.1)	277	252	273	248
1024	12289	67733370 (τ=1.08)	279	253	273	248
1024	12289	66484856 (τ=1.07)	280	254	273	248
1024	12289	64022669 (τ=1.05)	282	255	273	248

Таблиця 16.16.

Оптимізовані альтернативні набори
загальносистемних параметрів для 512 біт класичної стійкості

n	q	$\lfloor\beta^2\rfloor$	Підробка ЕП клас. (біт)	Підробка ЕП квант. (біт)	Відновлення ключа клас. (біт)	Відновлення ключа квант. (біт)
2048	12289	137125015 (τ=1.07)	618	561	554	503
2048	12289	133307337 (τ=1.055)	621	563	554	503
2048	12289	130792161 (τ=1.045)	622	565	554	503
2048	12289	127064310 (τ=1.03)	625	567	554	503

6. Метод та алгоритми побудови загальних параметрів ЕП типу Dilithium у залежності від їх розмірів

В цьому підрозділі наводяться окремі результати теоретичних та практичних досліджень щодо створення постквантового ЕП Dilithium на алгебраїчній решітці [256,273,274]:

- обґрунтування перспективного постквантового національного стандарту ЕП Dilithium на основі алгебраїчних решіток з відхиленням;
- методи обчислення системних параметрів для ЕП Dilithium 128, 256, 384 та 512 біт рівнів безпеки;
- генерація системних параметрів ЕП Dilithium та «Вершина 1» для 128, 256, 384, та 512 біт стійкості.

6.1 Обґрунтування перспективного постквантового національного стандарту електронного підпису на основі алгебраїчних решіток з відхиленням

Одним із видів криптографічних перетворень типу ЕП, що може бути включений в національний стандарт ЕП постквантового періоду, на наш погляд, може стати ЕП на алгебраїчних решітках типу Dilithium [57,273].

6.2 Аналіз стійкості алгоритму Dilithium проти основних атак

Наразі в постквантовій криптології актуальними є завдання забезпечення криптографічної стійкості щодо квантових атак. Вона ґрунтується на вирішенні проблеми навчання з помилками.

На основі аналізу визначено [263–271,276], що стосовно LWE можливо застосування таких атак:

1. BKW, коли LWE зводиться до SIS атаки.
2. Primal attack (Search-LWE зводиться до BDD атаки).
3. Dual attack (Decision-LWE зводиться до SIS).
4. Зведення до uSVP атаки пошуку короткого вектора.

Деталі щодо кожної з атак, а також їх теоретичне обґрунтування, наведено у [256,261], а також авторами розглянуті у відповідних розділах підручника.

6.3 Захищеність алгоритму ЕП від атак сторонніми каналами

В процесі проведення конкурсу на постквантовий стандарт ЕП особлива вимога висунута до захищеності кандидата на ЕП від атак сторонніми каналами. Тому така проблемна задача є актуальною, в першу чергу, стосовно ЕП типу Dilithium та рішення для України «Вершина 1» [261].

Дослідження стосовно алгоритму ЕП Dilithium проведено за такими параметрами [259–261]:

- BKZ bock-size to break SIS=475;
- BKZ bock-size to break LWE=485;
- k=5; l=4; η=5; ς=4; β=275; ω=96.

Результати отримано методом програмного моделювання. Для проведення експерименту було згенеровано 10000 ключів та виконано 10000 підписів. Результат залежності часу підпису від номеру ключа наведено на рис. 16.2. Для 10000 ключів максимальне відхилення від нормалізованого середнього (дисперсія) усіх вимірів часу підпису повинно знаходитися в інтервалі 5.19676 <= d <= 6.62797(%), щоб вважати, що час підпису не залежить від ключа. Номери ключів, для яких було отримано мінімальне та максимальне значення при повтореннях вимірів, не повинні співпадати.

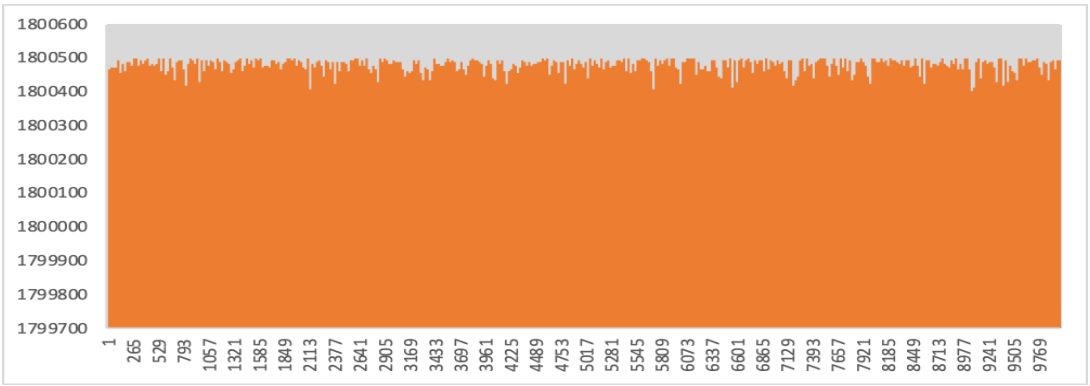


Рис. 16.2. Залежність часу підпису (у тактах процесора) від номеру ключа

Значення дисперсії $d \approx 2\%$, що свідчить про практично статистичну незалежність часу підпису від ключа, що є важливим з точки зору захищеності від атак сторонніми каналами.

6.4 Вибір параметрів 5-7 рівнів стійкості

Попередній аналіз показав, що значення параметрів, які наведені в [261], не забезпечують при застосуванні в механізмі Dilithium стійкість ЕП від класичних атак на рівні 256 бітів. Фактично нижні оцінки стійкості наведено в рядках табл. 9 [249] з назвами Best Known Classical bit-cost і Best Known Quantum bit-cost окремо для кожної з двох задач – класичної та квантової, на складності яких базується стійкість. Також вказані таблиці наведені нами в

посібнику, підрозділи 15.2.2, 16.1. Приведемо повторно ці таблиці, див. табл. 16.17 і 16.18.

Це задачі SIS та LWE. Для кожної з них зазначені параметри обчислюються за формулами:

$$\text{Best Known Classical bit-cost (класична атака)} = 0{,}292b, \tag{16.16}$$

$$\text{Best Known Quantum bit-cost (квантова атака)} = 0{,}265b, \tag{16.17}$$

де b є довжиною блоку (BKZ bock-size b to break SIS or LWE [261]).

Таблиця 16.17.

Нижні оцінки стійкості для задачі SIS

	I	II	III	IV
BKZ розмір блоку b для зламування SIS	235	355	475	605
Best Known Classical bit-cost	68	103	138	176
Best Known Quantum bit-cost	62	94	125	160

Таблиця 16.18.

Нижні оцінки стійкості для задачі LWE

	I	II	III	IV
BKZ розмір блоку b для зламування LWE	200	340	485	595
Best Known Classical bit-cost	58	100	141	174
Best Known Quantum bit-cost	53	91	128	158

В ролі кінцевої оцінки стійкості використовується найменше з двох значень, обчислених для b, що є довжиною блоку для задачі SIS та задачі LWE відповідно.

Загальні обмеження [259–261], які необхідно врахувати при виборі параметрів для забезпечення стійкості на рівні $\lambda \in \{256, 384, 512\}$ бітів, наступні:

1. Геш-функція CRH, що використовується у ЕП, повинна бути стійкою до колізій [256]. Отже, довжина її вектора значень повинна бути, як мінімум, 2λ бітів (зокрема, при λ=256 функція CRH повинна приймати значення довжини 512, а не 384, як в оригінальному методі [261]).

2. Криптографічна функція H, яка використовується, повинна бути стійкою відносно знаходження другого прообразу і, отже, приймати, як мінімум, $\mathbf{2\lambda}$ різних значень, що є поліномами з кільця R_q, які мають коефіцієнти 0, 1, –1 та містять точно h ненульових коефіцієнтів (зауважимо, що кількість таких поліномів дорівнює $2^h \binom{n}{h}$).

При λ=256 в [256,261] рекомендується використовувати параметри n=256, h=60, тоді умова

$$2^h \binom{n}{h} \geq 2^\lambda$$

виконується.

При $\lambda \in \{384,512\}$ та n=256 забезпечити виконання умови неможливо, якщо $h \leq n/2 = 128$. Отже, треба збільшити n до 512; при цьому числа q, γ_1, γ_2 можна залишити такими самими, як в [256,261]:

$$q=2^{23}-2^{13}+1,$$

$$\gamma_1=(q-1)/16,\ \gamma_2=\gamma_1/2. \qquad (16.18)$$

3. Довжини векторів ρ та K, що використовуються, повинні бути не менше, ніж $\boldsymbol{\lambda}$.

Таким чином, для забезпечення стійкості схеми ЕП на рівні $\lambda \in \{256,384,512\}$ необхідно:

1) використовувати геш-функцію CRH, значеннями якої є двійкові вектори довжини $\mathbf{2}\boldsymbol{\lambda}$ [261];

2) використовувати двійкові вектори ρ та K, що мають довжину $\boldsymbol{\lambda}$**;**

3) покласти n=256, якщо λ=256; n=512, якщо $\lambda \in \{384,512\}$;

4) вибрати просте число $q \equiv 1 \bmod 2n$ та обчислити γ_1, γ_2 за формулою (16.16);

5) обчислити вагу c , як найменше натуральне h.

Зауважимо, що при n=512, $q=2^{23}-2^{13}+1$ час формування підпису може виявитися занадто великим; у цьому випадку треба збільшити q (приблизно в два рази).

6.5 Генерація загальносистемних параметрів ЕП Dilithium для 128, 256, 384, 512 біт стійкості

Для генерації системних параметрів удосконаленого ЕП Dilithium використаємо результати аналізу відомих атак на криптосистему, що наведені вище, та встановимо умови, за яких забезпечується захист від них. Основними атаками є [261]:

- відновлення особистого ключа на основі відкритого ключа;
- засобом підробки ЕП.

Алгоритм генерації загальносистемних параметрів [256,261,273] наступний:

1. Визначити потрібний рівень безпеки $\lambda \in \{256,384,512\}$.

2. Обрати значення N. Якщо λ=256, то N=256, інакше N=512.

3. Обрати значення q: q=8389417.

4. Обчислити γ_1 та γ_2 за формулами $\gamma_1 = (q-1)/16$, $\gamma_2 = \gamma_1/2$.

5. Обчислити значення η. За замовченням встановити η=2. На наступних кроках значення буде уточнене.

6. Встановити значення (k, l)=(2, 1).

7. Обчислити λ_1-стійкість до Primal Attack. Якщо стійкість менша за λ, то оновити параметри $(k, l)=(k+1, l+1)$ та повернутися до кроку 7 або збільшити η та повернутися до кроку 7.

8. Обчислити λ_2-стійкість до Dual Attack. Якщо стійкість менша за λ, то поновити параметри $(k, l)=(k+1, l+1)$ та повернутися до кроку 7 або збільшити η та повернутися до кроку 7.

9. Обчислити λ_3-стійкість до SIS з ζ_1. Якщо стійкість менша за λ, то поновити параметри $(k, l)=(k+1, l+1)$ та повернутися до кроку 7 або збільшити $k=k+1$ та повернутися до кроку 7.

10. Обчислити λ_4-стійкість до SIS з ζ_2. Якщо стійкість менша за λ, то поновити параметри $(k, l)=(k+1, l+1)$ та повернутися до кроку 7 або збільшити $k=k+1$ та повернутися до кроку 7.

11. Обчислити h як найбільше ціле, для якого виконується нерівність $2^h \binom{n}{h} \geq 2^\lambda$.

12. Обчислити d як найбільше ціле, для якого виконується нерівність $2^{d-1}h+1 \leq 2\gamma_2$.

13. Встановити $\beta=\eta h$ та зменшувати β, щоб ймовірність повтору циклу була достатньо малою.

14. Обчислити $w=0{,}08nk$ (цей крок не впливає на криптостійкість та його можливо оптимізувати).

Алгоритм генерації загальносистемних параметрів авторами детально розглянутий в розділі 15.2, пункт 4.

У табл. 16.19 наведено значення параметрів для удосконаленого ЕП Dilithium.

Таблиця 16.19.

Значення параметрів для 256, 384, 512 біт стійкості

Набір	N, q	γ_1	γ_2	k, l	η	β	d	h	ω
256	(256,8380417)	523776	261888	(9,8)	2	144	14	60	184
384	(512, 8380417)	523776	261888	(7,5)	5	100	13	77	286
512	(512, 8380417)	523776	261888	(9,8)	2	74	13	118	368

Ймовірність повтору циклу при цьому складає для 256 біт – 0,15442678312246608, для 384 біт – 0,15609624568669475 і для 512 біт – 0,15247678668181552.

Результати оцінки криптостійкості для удосконаленого ЕП Dilithium (в бітах) з використанням параметрів з табл. 16.19 наведено в табл. 16.20.

6.6 Особливості застосування методики оцінки та результати порівняння постквантових алгоритмів ЕП на алгебраїчних решітках

В даному підрозділі наводяться пропозиції щодо застосування методики з оцінки та порівняння перспективних криптографічних перетворень типу ЕП, в першу чергу, щодо криптографічної стійкості.

В табл. 16.21 наведені характеристики обраних для порівняння алгоритмів (значення швидкості криптоперетворень та генерації ключів наведено в тактах).

Таблиця 16.20.

Оцінки криптостійкості для удосконаленого ЕП Dilithium

Набір	Primal атака (клас.)	Primal атака (квант.)	Dual атака (клас.)	Dual атака (квант.)	SIS (класич.)	SIS (квант.)
256	298	270	296	269	293	266
384	440	399	438	397	503	456
512	582	*527*	579	525	590	535

В порівнянні приймали участь проєкти стандартів «Вершина 1» та «Вершина 2», а також алгоритм Dilithium, який за попередніми дослідженнями мав кращі результати серед постквантових алгоритмів підпису, що засновані на перетвореннях на алгебраїчних решітках. Стійкість алгоритмів «Вершина» 128 біт відповідає 3-му рівню стійкості NIST, 256 – 5-му, тому пропорційно для виконання порівняння згідно шкали оцінок попарного порівняння параметрам 384 був наданий 7-й рівень, а 512 – 9-й.

Таблиця 16.21.

Характеристики алгоритмів ЕП, що засновані на перетвореннях на алгебраїчних решітках

Алгоритми	$I_{ст.}$	$l_{в.к.}$	$l_{о.к.}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$	$T_{гк.}$
Dilithium_round3sec2	2	1 312	3 504	2 420	259 172	118 412	124 031
Dilithium_round3_sec3	3	1 952	3 856	3 293	428 587	179 424	256 403
Dilithium_round3_sec5	5	2 592	5 792	4595	538 986	279 936	298 050
Вершина_1_128	3	1 472	3 488	2 693	133 340	109 818	90 328
Вершина_1_256	5	2 624	5 792	5 345	259 103	233 712	229 669
Вершина_1_384	7	4 528	9 088	6762	411 040	398 029	317 324
Вершина_1_512	9	5 824	11 008	10708	643 744	620 989	485 471
Вершина_2_128	3	897	4097	666	655 672	139 620	33 696 000
Вершина_2_256	5	1 793	8193	1 280	1 338 825	285 714	107 055 000
Вершина_2_512	9	3 585	5121	2 515	2 600 053	265 416	28493603229

В табл. 16.22 наведені результати досліджень – відносна перевага алгоритмів ЕП, що отримана методом попарних порівнянь за кожною з характеристик.

На рис. 16.3 відображено гістограму загальної відносної переваги алгоритмів ЕП з урахуванням вагових коефіцієнтів характеристик. Як видно, найбільшу перевагу має алгоритм «Вершина 1» з параметрами стійкості 128 біт, для більш стійких параметрів перевага вже у алгоритму «Вершина 2». «Вершина 2» на даному етапі програє через свою низьку швидкодію, якщо її реалізація буде більш оптимізована, то вже «Вершина 2» буде на першому місці.

Таблиця 16.22.

Відносна перевага алгоритмів ЕП за кожною з характеристик

Алгоритми	$I_{ст.}$	$l_{в.к.}$	$l_{о.к.}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$	$T_{гк.}$
Dilithium_round3_sec2	0,0198	0,1770	0,1816	0,1131	0,1583	0,1857	0,2090
Dilithium_round3_sec3	0,0299	0,0965	0,1475	0,0606	0,0849	0,0984	0,1082
Dilithium_round3_sec5	0,0697	0,0655	0,0768	0,0507	0,0666	0,0506	0,0915
Вершина_1_128	0,0299	0,1395	0,1816	0,0800	0,3006	0,2195	0,2696
Вершина_1_256	0,0697	0,0655	0,0768	0,0339	0,1583	0,0716	0,1388
Вершина_1_384	0,1453	0,0327	0,0407	0,0261	0,0975	0,0348	0,0797
Вершина_1_512	0,2681	0,0233	0,0296	0,0173	0,0479	0,0218	0,0608
Вершина_2_128	0,0299	0,2487	0,1212	0,3211	0,0479	0,1466	0,0192
Вершина_2_256	0,0697	0,1108	0,0467	0,1989	0,0238	0,1130	0,0146
Вершина_2_512	0,2681	0,0406	0,0973	0,0984	0,0143	0,0581	0,0086

Таким чином, зроблено порівняння алгоритмів, що пройшли до третього етапу NIST, а також алгоритмів проєктів стандарту «Вершина 1» та «Вершина 2». При порівнянні використовувалися два методи порівняння для отримання більш точної оцінки в залежності від вимог до алгоритмів ЕП.

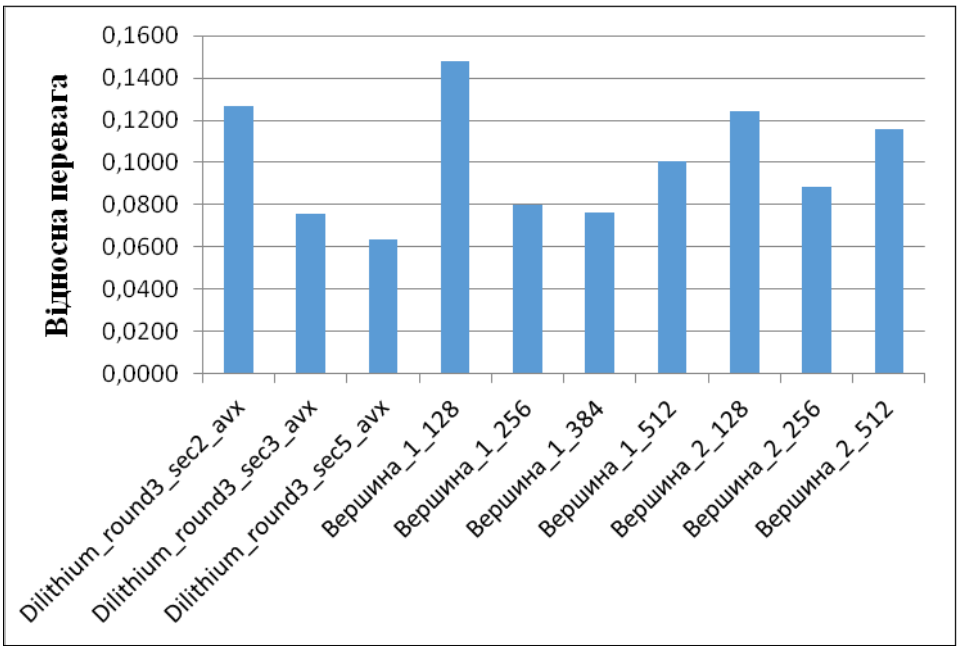


Рис. 16.3. Переваги алгоритмів ЕП

Висновки

1. Математичні методи Falcon та Dilithium є теоретичними основами для створення постквантових проєктів стандартів ЕП Falcon та Dilithium. Ці ЕП є фіналістами міжнародного конкурсу NIST США. Наразі вони досліджуються на третьому раунді і при позитивних результатах дослідження можуть бути прийняті в якості міжнародних постквантових стандартів ЕП. При їх побудуванні використано математичний апарат алгебраїчних решіток та відповідні методи.

2. Як з теоретичних, так і практичних позицій основоположним щодо цих ЕП є обґрунтування вимог до параметрів та ключів та їх побудування за умови достатності забезпечення гарантованості їх захищеності від класичних, квантових, спеціальних та атак на основі помилок.

3. При виборі розмірів загальних параметрів та ключів виникає суттєве протиріччя між властивостями проєктів стандартів ЕП Falcon та Dilithium щодо стійкості та складності перетворень. Збільшення розмірів загальних параметрів та ключів приводить до збільшення складності перетворень і навпаки.

4. Що стосується теоретичних методів побудування загальних параметрів та ключів для ЕП Falcon та Dilithium, то вони у цілому зрозумілі, а якщо використовувати [57,256,280,261], то існує можливість їх побудувати і для більш високих, але обґрунтованих рівнів безпеки – 6-го та 7-го, коли може бути забезпечено 384 і 512 біт захисту від класичних атак та 192 і 256 біт захисту від квантових атак, а також захищеність від спеціальних атак та атак на основі помилок.

5. Результати розрахунків розмірів відкритих ключів (PK_SIZE), електронних підписів (DS_SIZE) та складностей (часових) перевірки підписів (CHECK_TIME) для алгоритмів ЕП Falcon та Dilithium для 5-го рівня безпеки наведені в табл. 16.17, які свідчать про те, що ЕП Dilithium для 5-го рівня безпеки програє ЕП Falcon.

6. На основі математичних методів Falcon та Dilithium розроблено проєкти стандартів ЕП «Вершина 1» та «Вершина 2» 7-го рівня криптостійкості 512 біт захисту від класичних атак та 256 біт захисту від квантових атак. В табл. 16.22 наведені основні показники для цих алгоритмів в разі збільшення степені n та застосування решти необхідних параметрів.

7. Генерація ключів та формування електронного підпису для алгоритму Falcon та його прототипів потребує від розробників застосування багатьох різних алгоритмів та форматів, що може привести до значного звуження практичного застосування алгоритму ЕП Falcon.

8. У [57] не зберігають результатів перетворень, а для кожного поточного значення r для виконання зворотного перетворення обчислюють їх повторно, тому рекомендується їх зберігати, що збільшить навантаження на пам’ять, але зменшить час обчислення. Для тимчасового зберігання даних в функціях рекомендується застосовувати локальну пам'ять.

9. Попередній аналіз дозволяє зробити висновки, що Falcon все-таки має перспективи щодо прийняття в якості міжнародного стандарту ЕП.

10. Домінуючим основним концептуальним підходом до проєктування механізму ЕП Falcon є використання перетворення типу «геш-і-підпис». Перевагою такого підходу є доведена стійкість в межах моделі квантового випадкового оракула.

11. Безпосередньо ЕП Falcon та його аналізу присвячено значно менше робіт, ніж щодо інших, наприклад, проєкту ЕП Dilithium. Крім того, як і щодо інших, при проєктуванні ЕП Falcon були прийняті обмеження щодо максимальних рівнів безпеки у вигляді максимально 256 біт проти класичного та 128 біт проти квантового криптоаналізу.

12. Попередній аналіз показав, що основними атаками щодо ЕП Falcon є атаки на відновлення особистого ключа з відкритого ключа ЕП та атаки на підробку ЕП.

13. Атаки на відновлення особистого ключа з відкритого ключа щодо Falcon можуть зводитися до вирішення проблеми NTRU. У ряді криптосистем, стійкість яких ґрунтується на проблемі NTRU, коли поліноми f та g мають коефіцієнти з множини значень {0, 1, –1}, виникає проблема забезпечення стійкості проти комбінованих атак.

14. Атаки безпосередньої підробки ЕП Falcon можуть бути найбільш загрозливими. Тому іншим впливом є атака підробки ЕП. При реалізації такої атаки потрібно знайти достатньо короткий вектор s. Відповідно, це можливо зробити, редукувавши базис так, щоб виконувалися необхідні умови.

15. До недоліку ЕП Falcon необхідно віднести використання арифметики з плаваючою точкою. Разом з використанням деревоподібних структур це ускладнює аналіз методу до атак сторонніми каналами. Іншою проблемою є складність реалізації перетворень на малоресурсних пристроях.

16. Одним із видів криптографічних перетворень типу ЕП, що може бути включений в національний стандарт ЕП постквантового періоду, на наш погляд, може стати ЕП на алгебраїчних решітках типу Dilithium.

17. Наведені у табл. 16.21 та 16.22 результати порівняння проєктів стандартів ЕП «Вершина 1» та «Вершина 2», а також алгоритму Dilithium, показали, що кращі результати серед постквантових алгоритмів ЕП, що засновані на перетвореннях на алгебраїчних решітках, мають проєкти ЕП «Вершина 1» та «Вершина 2».

18. Кращі показники у алгоритмів «Вершина 1» та «Вершина 2» («Вершина 2» на даному етапі програє через свою низьку швидкодію, якщо її реалізація буде більш оптимізована, то вже «Вершина 2» буде на першому місці).

16.3 Основні положення щодо моделі безпеки для асиметричних перетворень типу ЕП з урахуванням вимог та загроз постквантового періоду

Вступ

Стосовно криптографічної стійкості (безпеки) асиметричних криптоперетворень, як і симетричних криптоперетворень, для оцінки використовуються існуючі методи, методики та різні системи криптоаналізу. Такий вузький підхід при аналізі стійкості залишає без необхідного врахування

можливі моделі порушника, моделі загроз, а також не дає можливих варіантів протидії. На наш погляд, продуктивним може бути введення та використання узагальненої (комплексної) моделі криптографічної стійкості. Приймемо в якості складових комплексної моделі безпеки щодо асиметричних криптоперетворень типу ЕП такі часткові моделі:

- порушника щодо асиметричних криптоперетворень типу ЕП;
- загроз щодо асиметричних криптоперетворень типу ЕП;
- безпеки щодо асиметричних криптоперетворень типу ЕП.

При викладенні сутності, призначення, можливостей застосування та обмежень щодо такої комплексної моделі будемо орієнтуватись та використовувати результати теоретичного обґрунтування та розробки вказаних часткових моделей безпеки та комплексної моделі безпеки у цілому. Метою цього розділу є обґрунтування та розробка пропозицій щодо побудування комплексної моделі безпеки стосовно асиметричних криптоперетворень типу перспективний ЕП, що може та повинна застосовуватись в постквантовий період.

1. Визначення моделей порушника, загроз та безпеки щодо перспективного ЕП

Побудова моделі порушника необхідна для того, щоб розробити комплекс заходів із забезпечення захищеності механізмів ЕП, в тому числі, з урахуванням вимог та умов їх застосування в постквантовий період. Така модель порушника може бути побудована з урахуванням різних критеріїв. Звичайно модель порушника розробляється з метою отримання відповідей на наступні питання:

- Від кого необхідно захищати інформацію?
- Якою є мета порушника?
- Якими знаннями володіє порушник?
- Які повноваження в системі має потенційний порушник?
- Які методи, системи та засоби використовує порушник?

По суті модель порушника – це опис можливих дій порушника, який формується на основі аналізу типу зловмисника, рівня його повноважень, знань, теоретичних та практичних можливостей. У якості порушника розглядається особа, що може отримати доступ до роботи з включеними до складу відповідної комп’ютерної системи (КС) засобами.

Модель загроз ЕП (далі – модель загроз) повинна бути документом, яким закріплено найбільш повний перелік загроз щодо існуючих та перспективних ЕП, що може застосовуватись в постквантовий період відповідно до Законів України "Про захист інформації в інформаційно-телекомунікаційних системах", "Про електронні довірчі послуги" та "Про захист персональних даних". Перелік можливих загроз безпеці застосування існуючих та перспективних ЕП сформований з числа загроз, наявних у IT-Grundschutz Catalogues [283] з урахуванням апаратних, програмних та апаратно-програмних ресурсів, технологій обробки даних та механізмів криптографічного захисту при застосуванні ЕП, в тому числі, з урахуванням вимог та умов синтезу перспективних ЕП та застосування ЕП в постквантовий період.

Основними загрозами (атаками) із застосуванням квантових математичних методів, які можуть бути реалізованими на квантовому комп'ютері (звичайно, якщо він буде побудований та доступний для застосування), є такі:

- квантовий алгоритм факторизації Шора;
- квантовий алгоритм Гровера;
- квантовий алгоритм Шора вирішення дискретного логарифму в полі;
- квантовий алгоритм Шора вирішення дискретного логарифму в групі точок еліптичної кривої;
- квантовий алгоритм криптоаналізу для перетворень в фактор-кільці тощо.

Стосовно перспективного ЕП можливо виділити та необхідно, як мінімум, розглядати наступні атаки (загрози):

1. Атака грубої сили, тобто, повного перебору.
2. Традиційна атака зустріч посередині.
3. Атака на основі алгоритму Arora-Ge.
4. Диференційні атаки.
5. BKW, коли LWE зводиться до SIS атаки.
6. Primal attack (Search-LWE зводиться до BDD атаки).
7. Dual attack (Decision-LWE зводиться до SIS).
8. Зведення до uSVP атаки пошуку короткого вектора.

Для однозначного розуміння та використання також введемо поняття щодо моделі безпеки (криптографічної стійкості) стандартизованих криптографічних примітивів ЕП. Модель безпеки (криптографічної стійкості) криптографічних примітивів (в тому числі типу ЕП) – це сукупність організаційно-технічних, програмно-технічних та логічних механізмів (методів) та заходів, законодавчих та нормативно-правових норм та правил, що визначають вимоги до методів синтезу, оцінки криптографічної стійкості та застосування стандартизованих криптографічних примітивів (типу ЕП) з урахуванням умов їх реалізації при інтенсивній протидії порушника 4-го рівня [78,278] із застосуванням методів, систем та засобів класичного та квантового криптоаналізу.

Аналіз показує, що NIST США планує стандартизувати один чи декілька стандартизованих криптопримітивів типу ЕП, що будуть забезпечувати екзистенційну стійкість ЕП щодо атак на основі адаптивно підібраного повідомлення. Така безпека позначається в науковій літературі як EUF-CMA безпека [94,277]. Представлені кандидати ЕП оцінюються на підставі того, наскільки такі можливості існують для кандидата на постквантовий стандарт, щоб забезпечити таку його властивість. Але така вимога не є обов'язковою, заявники не зобов'язані надавати докази безпеки щодо атаки на основі адаптивно підібраного повідомлення, хоча такі докази вже розглядаються та розглядатимуться при подальших дослідженнях, якщо вони будуть наявні (доступні).

Важливим є те, що у якості апріорних даних запропоновано вважати, що зловмисник має доступ не більше, ніж до 2^{64} обраних повідомлень. Проте, атаки за умови більшої кількості повідомлень також можуть розглядатись. Крім того, скоріше всього, NIST розглядає як класичні, так і квантові атаки.

2. Загальні положення щодо моделі безпеки перспективного ЕП

Модель безпеки EUF-CMA визначає екзистенційну непідроблюваність від атак на основі адаптивно вибраних повідомлень [94,277,280]. Зокрема, безпека в сенсі EUF-CMA повинна протидіяти порушнику 4-го рівня виробляти ЕП для повідомлень, що залежать від ключів (застосуванні особистого s_k ключа). По суті, при застосуванні механізму безпечного ЕП, згідно моделі EUF-CMA, вона є безпечною для EUF-CMA у випадку, коли не застосовуються запити повідомлення. Але при наявності хоча б одного запиту повідомлення, що залежить від ключів, безпека механізму ЕП порушується.

Існує два загальних формальних визначення для забезпечення безпеки схеми ЕП [282]. Кожне з цих визначень представлено як експеримент, який виконується між атакуючим (attacker) та деяким чесним претендентом (challenger).

Експеримент щодо моделі EUF-CMA (екзистенційна непідроблюваність при атаці на основі підібраних повідомлень) виконується у такій послідовності:

1. Претендент генерує дійсну пару ключів (p_k, s_k) і надає p_k атакуючому.

2. Далі атакуючий може повторно запросити підписи на підібраних (вибраних) повідомленнях (M_1,...,M_q) за своїм вибором і отримує дійсні підписи (σ_1,...., σ_q) у відповідь.

3. По завершенню експерименту зловмисник повинен вивести повідомлення та підпис M*, σ* такі, щоб одне повідомлення було не одним із повідомлень, які вимагали попереднього кроку (1), а повідомлення/підпис (2) перевіряється правильно з відкритим ключем.

Така схема вважається безпечною, якщо жоден (ефективний) зловмисник не має ні найменшої переваги у виконанні вищезазначених умов. Зазвичай, кількість повідомлень q обмежується лише часом дій атакуючого, однак, для спеціального випадку одноразових ЕП, зловмисник обмежується запитом лише одного підпису на кроці (2).

Така властивість є досить сильною, але не настільки, як можливо. Дещо сильнішим визначенням є визначення моделі безпеки SUF-CMA [282].

Експеримент із застосуванням моделі SUF-CMA (сильна екзистенційна непідроблюваність при атаці на основі підібраних повідомлень) виконується у такій послідовності:

1. Те саме, що і в попередньому експерименті.

2. Те саме, що і в попередньому експерименті.

3. Після завершення п. 2 експерименту атакуючий повинен вивести повідомлення та підпис M*, σ* такі, щоб пара (M*,σ*) (1) не була одним із запитаних повідомлень, а підпис повернувся на попередньому кроці, повідомлення / підпис (2) перевіряється правильно з відкритим ключем.

Атакуючий виграє, якщо модель SUF-CMA задовольняє вищенаведеним умовам.

Головна відмінність моделі SUF-CMA полягає в тому, що це більш сильне визначення гарантує, що атакуючий не зможе підібрати підпис. Так, схема, в якій атакуючий може повторно рандомізувати дійсний підпис, щоб він залишався

дійсним, але виглядав інакше, ніж вихідне значення, не задовольнила би умові SUF-CMA.

Визначення 1 [282].

Нехай EUF–CMA–GAME(Gen, Sign, Ver, A, n), причому виконуються пункти 1 – 7.

1. Gen(1n) (s_k, v_k);
2. A отримує v_k;
3. A створює повідомлення m;
4. A отримує s Sign(s_k, m);
5. A повторює кроки 3 та 4, якщо це необхідно;
6. A виводить (m*, s*);
7. A виграє, якщо Ver (v_k, m*, s*)=accept та m* не було раніше підписано Sign.

Тоді дійсним є *визначення 1*.

Визначення 2. Схема підпису, що виконується за поліноміальний час, є EUF-CMA безпечною, якщо для будь-якого зловмисника A ймовірність виграшу EUF-CMA-GAME є незначною.

3. Попередні дані та визначення

Як вже зазначалося і як буде показано далі, для цілей безпеки вкрай важливо допустити алгоритм підпису з відслідковуванням стану (stateful).

Це забезпечується на основі використання та врахування наступного:

- KDM – повідомлення, що залежне від ключа;
- KD – залежність від ключа;
- KDS – підписи, що залежні від ключів;
- EUF-CMA – екзистенційна непідроблюваність при атаці на основі адаптивно підібраних (вибраних) повідомлень;
- Aeuf – зловмисник (імовірнісний алгоритм поліноміального часу) при UF-CMA;
- Akds – зловмисник (імовірнісний алгоритм поліноміального часу) при KDS-CMA;
- Akd – зловмисник (імовірнісний алгоритм поліноміального часу) при KD - EUF;
- OS – оракул підпису при Aeuf;
- безпека розглядається в сенсі KDS-CMA, що означає безпеку в сенсі EUF CMA;
- CMA – атака на основі адаптивно підібраних (вибраних) повідом лень.

Також використовується «пряма» секретність (forward security, forward secrecy), як властивість криптосистем зберігати конфіденційність минулих сеансових ключів при компрометації довгострокового ключа.

Також застосовується досконала пряма секретність (perfect forward secrecy (PFS)), яка означає, що сеансовий ключ, який генерується з використанням довгострокових ключів, не буде скомпрометований при умові, якщо один або декілька з цих довгострокових ключів будуть скомпрометовані у майбутньому.

Аналіз показав [281,282], що моделі безпеки щодо підписів засновуються на принципах та понятті теорії ігор. Поняття теорії ігор може використовуватись у такому сенсі.

Теорія ігор -це теорія математичних моделей прийняття оптимальних рішень в умовах конфлікту. Оскільки сторони, що беруть участь в більшості конфліктів, зацікавлені в тому, щоб приховати від супротивника власні наміри прийняття рішень в умовах конфлікту, зазвичай, відбувається в умовах невизначеності. Фактор невизначеності можна інтерпретувати як противника суб'єкта, який приймає рішення. Логічною основою теорії ігор є формалізація трьох понять, які входять в її визначення, і є фундаментальними для всієї теорії: конфлікт; прийняття рішення у конфлікті; оптимальність прийнятого рішення.

4. Аналіз схем підписів та екзистенційна непідроблюваність

Подальший аналіз можна провести засобом використання визначень 3 та 4 із [282].

Визначення 3 [282] (Схема підпису). Схема підпису S є трійкою поліноміальних алгоритмів $S = (K, S, V)$, де:

K – імовірнісний алгоритм генерації ключа, який при введенні параметра безпеки 1k повертає пару (s_k, p_k) ключів − відкритий ключ перевірки p_k з відповідним секретним ключем підпису $s_k\{0,1\}^*$. У випадку підписувача з відслідковуванням стану (stateful) s_k інтерпретуємо як початковий стан підписувача, тобто, вся секретна інформація підписувача є частиною його стану;

S – імовірнісний алгоритм підпису, який при введенні повідомлення M $\{0,1\}^*$ та стану s_k − який у випадку підписувача без стану (stateless) є лише секретним ключем − повертає підпис σ $\{0,1\}^*$ на M або символ помилки. Крім того, оновлюється стан значення s_k ;

V − це детерміністичний алгоритм перевірки, який при введенні відкритого ключа p_k, повідомлення M та підпису кандидата σ для M повертає true або false, вказуючи, чи є σ дійсним підписом для M при відкритому ключі p_k.

Далі, для пар ключа (s_k, p_k), що виводяться за допомогою K, вимагаємо, щоб із переважною ймовірністю мала місце очевидна умова правильності – для всіх повідомлень M маємо Vpk(M, Ssk(M))=true.

Стандартна вимога до безпеки для схем підписів EUF-CMA означає екзистенційну непідроблюваність при атаці адаптивно підібраних повідомлень.

Визначення 4 [282]. (EUF-CMA). Нехай S=(K, S, V) буде схемою підпису і Aeuf – імовірнісним алгоритмом, що виконується за поліноміальний час. Сценарій атаки буде такий:

1. Обчислити пару ключів (s_k, p_k) K(1k) та p_k , як вхідні дані Aeuf.

2. Зловмиснику Aeuf надається необмежений доступ до оракула підпису OS для виконання Ssk(M).

3. Зрештою, Aeuf виводить повідомлення M та підпис σ.

Зауваження 1. Вищезгадане визначення безпеки EUF-CMA є дійсними одноразовими схемами підпису в очевидний спосіб – єдиною модифікацією є те, що Aeuf може запитувати оракул підпису OS лише один раз.

Зокрема, безпека в сенсі EUF-CMA не дозволяє зловмиснику отримувати підписи повідомлень, залежних від ключів, як підпис на повному секретному ключі (стані) s_k. Фактично, враховуючи EUF-CMA-безпечну схему підпису, легко скласти схему підписів, яка все ще є EUF-CMA-безпечною, але один запит на повідомлення, залежного від ключів, порушує безпеку схеми.

5. Аналіз безпеки за наявності підписів, залежних від ключів

В ряді випадків схема підпису S=(K, S, V) називається KDS-CMA-безпечною, якщо вона є захищеною, незважаючи на здатність зловмисника отримувати підписи на довільних (таких, що ефективно обчислюються) функціях g стану s_k підписувача. Зокрема, g має доступ до секретного ключа, що зберігається під час підписування.

Визначення 5 [282] (KDS-CMA). Нехай трійка S=(K, S, V) буде схемою підпису, і Akds – це імовірнісний алгоритм поліноміального часу. Нехай реалізується такий сценарій атаки:

1. Обчислити пару ключів (s_k, p_k) K(1k) та p_k– як вхідні дані до Akds.

2. Зловмисник Akds отримує необмежений доступ до оракула підпису. Оракул приймає як вхідну функцію g, представлену, як логічна схема поліноміального розміру, і виконує алгоритм підпису S із поточним станом s_k і повідомлення $g(s_k)$, як вхідними даними (у моделі випадкового оракула g може викликати випадковий оракул).

3. Зрештою, Akds виводить повідомлення M {0,1}* і підпис σ.

Необхідно відмітити, що оцінка ефективності моделі безпеки KDS -CMA в сенсі KDS-CMA означає безпеку в сенсі EUF-CMA, і виникає питання, чи / як може бути досягнута безпека в сенсі *Визначення* 5.

6. Неможливість KDS-CMA з алгоритмом підпису без стану (stateless)

Як перший (негативний) результат необхідно відмітити, що жодна схема підпису з алгоритмом підписування без стану не може відповідати цілі безпеки KDS-CMA-безпечності.

Зауваження 2. [282] Нехай S=(K, S, V) буде схемою підпису з алгоритмом підписування без стану *S*, тобто, секретний ключ підпису s_k не змінюється шляхом виконання *S*. Тоді схема підпису *S* не є безпечною в сенсі KDS-CMA.

Незважаючи на свою простоту, атака в доказі зауваження 2 [282] є досить руйнівною і це може виявитися незрозумілим.

Як правило, для моделі прямої безпеки розглядаються так звані схеми підпису з ключами, що розвиваються, та компрометація поточного секретного ключа не дозволяє зловмиснику підроблювати попередні підписи. Підписи для повідомлень, підписаних раніше при фіксованому відкритому ключі, дійсні, навіть якщо поточний секретний ключ розкрито. Крім того, зловмисник не може підробити підпис з "датою" перед розголошенням ключа.

7. Схеми підпису з ключами, що розвиваються, та пряма безпека

Визначення 6. Схема підпису з ключем, що модифікується. Схема підпису з ключем, що «розвивається». Така схема Sf є кортежем з чотирьох елементів поліноміальних алгоритмів S=(Kf, Uf, Sf, Vf):

1. Kf – імовірнісний алгоритм генерації ключа, у якому, при введенні параметра безпеки 1k, загальна кількість періодів часу T (i, можливо, інші параметри) повертає пару $(s_k 0, p_k)$ ключів – відкритий ключ перевірки p_k з відповідним (базовим) секретним ключем підпису $s_k 0$.

2. Uf – детермінований алгоритм оновлення секретного ключа, який приймає в якості вхідних даних секретний ключ підпису sk_{j-1} попереднього періоду $j-1$ і повертає секретний ключ підпису sk_j для періоду j.

3. Sf – імовірнісний алгоритм підпису, у якого вхідними даними є повідомлення M {0, 1}* та секретний ключ підпису sk_j поточного періоду часу j, повертає підпис для M для періоду j або повертає символ помилки.

4. Vf – це детермінований алгоритм перевірки, у якого вхідними даними є відкритий ключ p_k, повідомлення M і підпис, повертає true або false, що вказує на те, що підпис прийнято або відхилено відповідно.

Можна припустити, що sk_j зберігає саме значення j для періоду $j\{1,...,T\}$, а також загальне число T періодів часу. Далі, приймається рішення про те, що skT+1 – це порожній рядок, і що Uf(skT) повертає skT+1. І поточний період часу j, і загальна кількість періодів T є загальновідомими та доступними для зловмисника Afwd разом із атакованим відкритим ключем p_k. Фактична гра атаки, яка використовується для визначення прямої безпеки схеми підпису з ключем, що «розвивається», включає в себе три етапи – етап атаки підібраного повідомлення (cma), етап розриву (breakin) та етап підробки (forge).

Визначення 7 (FWD-CMA – пряма безпека). Нехай Sf=(Kf, Uf, Sf, Vf) – це схема підпису з ключем, що розвивається, і нехай Afwd – це імовірнісний поліноміальний алгоритм. Тоді важливим для рішення є сценарій:

1. CMA стадія

Встановити j←0, і згенерувати пару ключів $(s_k 0, p_k)$ Kf(1k,...,T). (Тут '...' вказує, що додаткові вхідні параметри можуть бути присутніми).

Алгоритм CMA стадії наступний:

repeat
    j←j+1; skj←Uf(skj−1)
    (cma, pk)
until (d = breakin) or (j = T)
if d ≠ breakin and j = T
then j = T + 1
end if

2. Breakin стадія

Зловмиснику Afwd передається поточний секретний ключ s_{kj}.

3. Forge стадія

Зрештою, Afwd виводить повідомлення M та підпис з $b<j$.

Нехай QueriedEarlier – це подія, що Afwd виводить повідомлення M, яке вже було запитано до оракула підпису.

Процес у *визначенні* 6 суворо упорядкований тим, що як тільки зловмисник відмовляється від оракула підпису для s_{kj}, він не "може" знову отримати доступ до цього оракула. У якийсь момент зловмисник Afwd вирішує скористатися своїм привілеєм розриву, і повертає поточний секретний ключ s_{kj}. Щоб бути успішним, Afwd повинен підробити підпис з s_{kb} для деяких $b < j$ і нового повідомлення M.

Зауваження 3 [282]. За визначенням, безпечна схема FWD-CMA дозволяє зловмиснику Afwd подавати поліноміальну кількість запитів до його оракула підпису протягом одного часового періоду j. Таким чином, за наявності повідомлень, залежних від ключів, атака, як показано в доказі *зауваження* 2, може розкрити повний секретний ключ, перед тим, як з'явиться оновлення секретного ключа. Іншими словами, безпека у сенсі FWD-CMA не передбачає сильних гарантій безпеки при наявності повідомлень, залежних від ключів.

На противагу вищезазначеному негативному твердженню, після застосування деяких технічних модифікацій для отримання синтаксично правильної схеми підпису з ключем, що «розвивається», компілятор (який був розроблений для забезпечення безпеки KDS-CMA), може бути використаний для безпеки EUF-CMA одноразової схеми підпису S, включно до прямої безпеки схеми підпису з ключем, що розвивається, S_f.

Висновки

1. В результаті проведених досліджень визнано, що продуктивним може бути введення та використання узагальненої (комплексної) моделі криптографічної стійкості ЕП.
2. В якості складових узагальненої (комплексної) моделі криптографічної стійкості асиметричних криптоперетворень типу ЕП обґрунтовані та визначені такі часткові моделі:

- порушника щодо асиметричних криптоперетворень типу ЕП;
- загроз щодо асиметричних криптоперетворень типу ЕП;
- безпеки щодо асиметричних криптоперетворень типу ЕП.

3. Побудова моделі порушника необхідна для того, щоб розробити комплекс заходів із забезпечення захищеності механізмів ЕП, в тому числі, з урахуванням вимог та умов їх застосування в постквантовий період. Така модель порушника може бути побудована з урахуванням різних критеріїв.
4. Порушники класифікуються за рівнем можливостей, що надаються їм штатними засобами КС, в тому числі, для класичного та квантового криптоаналізу. Виділяються чотири рівні таких можливостей. Класифікація є ієрархічною – кожний наступний рівень включає в себе функціональні можливості попереднього рівня [280]. Прийнято, що, відносно перспективного ЕП для постквантового періоду, безпека повинна бути забезпечена в умовах протидії порушнику 4 рівня.
5. У найгіршому випадку безпека повинна бути забезпечена проти криптоаналітика 4-го рівня можливостей; він знає все про метод синтезу

перспективного ЕП, криптографічні властивості методу ЕП, а також всі механізми безпеки, що виконуються під час синтезу та застосування. Виключенням є те, що криптоаналітик не знає особистого ключа чи відповідним чином обґрунтовану частину особистого ключа. У найкращому випадку, порушник не знає нічого про системні параметри та ключі. У нашому випадку можливі варіанти є рівноймовірними.

6. Модель загроз щодо криптоперетворення ЕП повинна бути документом, яким закріплено найбільш повний перелік загроз щодо існуючих та перспективних ЕП. Відповідно до Законів України, інформація у основних інформаційних ресурсах поділяється на відкриту і конфіденційну. Інформація у підтримуючих інформаційних ресурсах є технологічною інформацією.

7. При застосуванні ЕП, незалежно від видів додатків, використовуються асиметричні пари ключів, для кожної пари особистий та відкритий. В подальшому при реальному застосуванні ЕП відкритий ключ, як правило, є сертифікатом відкритого ключа та доступним усім користувачам інфраструктури відкритого ключа.

8. Стосовно відкритого ключа ЕП повинна бути можливість забезпечення його цілісності, справжності, доступності, неспростовності та захист від несанкціонованих дій, які можуть привести до випадкової чи умисної модифікації, нав’язування хибного значення чи знищення.

9. Стосовно особистого ключа повинна забезпечуватись його цілісність, справжність, доступність, неспростовність та захист від несанкціонованих дій, а також його захист від несанкціонованого та неконтрольованого ознайомлення, модифікації, знищення, копіювання та поширення, тобто, конфіденційність.

10. Перелік можливих загроз безпеці застосування існуючих та перспективних ЕП сформований з числа загроз, що визначені у IT-Grundschutz Catalogues Німеччини з урахуванням апаратних, програмних та апаратно-програмних ресурсів, технологій обробки даних та механізмів криптографічного захисту при застосуванні ЕП, в тому числі, з урахуванням вимог та умов синтезу перспективних ЕП та застосуванні ЕП в постквантовий період.

11. Безумовно, що щодо обґрунтованої та вибраної на основі IT-Grundschutz Catalogues бази Німеччини моделі загроз, при синтезі та застосуванні існуючих стандартизованих та перспективних ЕП повинно бути зроблено перекриття названих загроз з необхідною якістю. Для цього, у залежності від механізмів та засобів, що застосовуються для протидії, повинні бути розроблені відповідні нормативно-правові документи.

12. Детально загрози щодо застосування класичного криптоаналізу при синтезі та застосуванні ЕП розглянуті в [78,99,277-282]. Їх перекриття повинно бути зроблене на всіх етапах синтезу та застосування перспективних ЕП. Вказані загрози з точки зору застосування математичних методів синтезу та застосування перспективних методів ЕП залежать від математичних методів, що застосовуються, та умов їх функціонування.

13. Загрози (атаки) сторонніми каналами є виділеним класом атак, основною особливістю яких є спрямованість на вразливості практичної реалізації криптосистем, тобто, на відміну від теоретичного криптоаналізу.

Загрозою такого класу атак над традиційними є менша потужність та більш висока дієвість.

14. Загрози (атаки) сторонніми каналами є надзвичайно небезпечними, якщо їх не перекривати. Концепція цих атак існує доволі тривалий час, але реалізація захищеності від них вимагає знань не лише у сфері криптографії, а й у сферах технічного характеру. Тому переважна більшість перспективних ЕП розрахована на використання у пристроях, які не можуть захистити від сторонніх атак, бо не мають відповідних програмних рішень щодо захисту від витоку сторонніми каналами.

15. Основними загрозами (атаками) із застосуванням квантових математичних методів, які можуть бути реалізованими на квантовому комп'ютері (звичайно, якщо він буде побудований), є такі: квантовий алгоритм факторизації Шора; квантовий алгоритм Гровера; квантовий алгоритм Шора вирішення дискретного логарифму в полі; квантовий алгоритм Шора вирішення дискретного логарифму в групі точок еліптичної кривої; квантовий алгоритм криптоаналізу для перетворень в фактор - кільці.

16. У залежності від математичних методів, що застосовуються для синтезу та застосування ЕП, можуть застосовуватись різні методи, системи та засоби. Наприклад, для криптоперетворень на алгебраїчних решітках необхідно вирішувати проблему навчання з помилками (LWE).

17. Наразі в постквантовій криптології актуальними завданнями є забезпечення криптографічної стійкості щодо квантових атак.

18. Стосовно атак на LWE можливо виділити та необхідно розглядати наступні атаки (загрози): атаки грубої сили, тобто, повного перебору; традиційні атаки зустріч посередині; атаки на основі алгоритму Arora-Ge; BKW, коли LWE зводиться до SIS атаки; Primal attack (Search-LWE зводиться до BDD атаки); Dual attack (Decision-LWE зводиться до SIS); зведення до uSVP атаки пошуку короткого вектора.

19. Для однозначного розуміння та використання пропонується ввести та використовувати нове поняття щодо моделі безпеки (криптографічної стійкості) стандартизованих криптографічних примітивів.

20. Модель безпеки (криптографічної стійкості) криптографічних примітивів (в тому числі типу ЕП) – це сукупність організаційно-технічних, програмно-технічних та логічних механізмів (методів) та заходів, законодавчих та нормативно-правових норм та правил, що визначають вимоги до методів синтезу, оцінки криптографічної стійкості та застосування стандартизованих криптографічних примітивів (типу ЕП), з урахуванням умов їх реалізації при інтенсивній протидії порушника 4-го рівня із застосування методів, систем та засобів класичного та квантового криптоаналізу.

21. Модель безпеки EUF-CMA визначає екзистенційну непідроблюваність від атак на основі адаптивно вибраних повідомлень [282]. Зокрема, безпека в сенсі EUF-CMA повинна протидіяти порушнику 4-го рівня виробляти ЕП для повідомлень, що залежать від ключів (при застосуванні особистого sk ключа). По суті, при застосуванні механізму безпечного ЕП, згідно моделі EUF-CMA, вона є безпечною для EUF-CMA випадку.

22. Головна відмінність моделі SUF-CMA полягає в тому, що це більш сильне визначення гарантує, що атакуючий не зможе підібрати підпис. Так, схема, в якій атакуючий може повторно рандомізувати дійсний підпис, щоб він залишався дійсним, але виглядав інакше, ніж вихідне значення, не задовольнила би умові SUF-CMA.

23. В ряді випадків схема підпису S=(K, S, V) називається KDS-CMA-безпечною, якщо вона є захищеною, незважаючи на здатність зловмисника отримувати підписи на довільних (таких, що ефективно обчислюються) функціях g стану sk підписувача. Зокрема, g має доступ до секретного ключа, що зберігається під час підписування.

24. Оцінка ефективності моделі безпеки KDS-CMA в сенсі KDS-CMA означає безпеку в сенсі EUF-CMA, і виникає питання, чи може бути досягнута за цих умов безпека.

25. Після застосування деяких технічних модифікацій для отримання синтаксично правильної схеми підпису з ключем, що «розвивається», компілятор може бути використаний для безпеки EUF-CMA одноразової схеми підпису S, включно до «прямої» безпеки схеми підпису з ключем, що розвивається, S_f.

16.4 Аналіз та дослідження алгоритму цифрового підпису PICNIC

Вступ

Важливою особливістю постквантового періоду у криптографії є суттєва невизначеність щодо вихідних даних для криптоаналізу та протидії в частині можливостей квантових комп'ютерів, їх математичного та програмного забезпечення, а також застосування квантового криптоаналізу до існуючих криптоперетворень та криптопротоколів. В якості основних методів обрано математичні методи цифрового підпису (ЦП), що пройшли суттєвий аналіз та обґрунтування в процесі широких досліджень криптографами та математиками на найвищому рівні. Вони детально описані та пройшли дослідження на першому етапі міжнародного конкурсу NIST США. В процесі другого етапу прийнято ряд рішень стосовно об'єднання деяких кандидатів на постквантовий стандарт ЦП [259].

Для подальших досліджень на 2-му етапі залишили 9 кандидатів: CRYSTALS-DILITHIUM, FALCON, GeMSS, LUOV, MQDSS, Picnic, qTESLA, Rainbow та SPHINCS+. Три з них (Dilithium, FALCON, qTeSLA) засновані на стійкості алгебраїчних решіток (Lattice-based), чотири (GeMSS, LUOV, MQDSS, Rainbow) – на основі багатовимірних перетворень (MQ-перетворення), один (SPHINCS+) – на стійкості геш-функції, один (Picnic) – на стійкості геш-функції та блокових потокових шифрів [263].

За результатами досліджень перспективних постквантових криптографічних алгоритмів типу цифровий підпис протягом 2-го раунду конкурсу NIST США було отримано наступні результати – обрані алгоритми-фіналісти та альтернативні алгоритми. У якості алгоритмів-фіналістів були обрані такі алгоритми ЦП, як CRYSTALS-DILITHIUM, FALCON та Rainbow. У якості альтернативних алгоритмів – GeMSS, Picnic та SPHINCS+ [247].

Далі у цьому розділі більш детально розглянемо перспективний постквантовий алгоритм ЦП, що входить до переліку альтернативних алгоритмів – Picnic.

1. Опис та параметри алгоритму Picnic

1.1 Опис алгоритму

Алгоритм цифрового підпису Picnic – це схема підпису, яка не використовує теоретико-числових або структурованих припущень складності. Зменшення безпеки відносяться до геш-функцій і симетричних блокових шифрів. Підпис Picnic базується на неінтерактивному нульовому доказі знання секретного ключа. Підписується відкритий текст таким чином (через гешування), що тільки власник секретного ключа може вивести доказ і перевірити текст на правильність. Алгоритм ЦП Picnic має невеликий розмір відкритого ключа, але великі підписи. Підпис, як і перевірка підписаного тексту, досить повільні. Генерація ключів досить ефективна. Довжина підпису залежить від мультиплікативної складності схеми шифрування і від конкретної методики побудови доказу нульового знання (з області безпечних багатоваріантних обчислень). Picnic має модульну схему проєктування. Криптографічні примітиви – геш-функції та блоковий шифр – можуть бути створені різними способами. Представлена конструкція використовує LowMC, блоковий шифр з низькою мультиплікативною складністю. LowMC не вивчався так багато, як AES, і, отже, потребує набагато більшого аналізу. Ефект використання AES замість LowMC в Picnic полягає в розширенні довжини підпису на коефіцієнт, який коливається від 6 до 9, в залежності від розміру блоку. Поліпшення обчислень призведе до зменшення підписів. Варто зазначити, що вимоги безпеки для базового блочного шифру менш суворі, ніж загальні вимоги безпеки блокового шифру, тому що тільки одна (випадковий відкритий текст, шифротекст) пара коли-небудь розкривається [285].

1.2 Порівняння постквантових алгоритмів

В табл. 16.23 порівнюються алгоритми, що пройшли до другого етапу міжнародного конкурсу NIST США. Алгоритм ЦП Picnic є унікальним серед представлених кандидатів, бо він єдиний, заснований на стійкості геш-функції та блокових потокових шифрів сімейства LowMC [247]. З табл. 16.23 можна зробити висновок, що алгоритм Picnic досить збалансований за усіма критеріями. Але він має набагато більший розмір підпису порівняно з іншими алгоритмами, що, у свою чергу, безперечно впливає на швидкодію.

Таблиця 16.23.

Порівняння постквантових алгоритмів

Схема підпису	Безпека	Захищеність від часових атак	Цикли підписання	Цикли перевірки підпису	Розмір відкритого ключа, байти	Розмір підпису, байти
Dilithium	125	Так	789	209	1472	2701
FALCON NTRU-GVP	>> 128	Ні	-	-	1792	1200

Продовження табл. 16.23.

qTESLA	98	Так	143402	19284	12582912	2444
GeMSS (HmFEv)	128	Так	1497	15	83100	61
LUOV	128	Так	659000	290000	34100	421
MQDSS	128	Так	8510	5752	72	40952
Rainbow	128	Так	68	22	145500	48
Picnic	128	Так	1034	194	64	195458
SPHINCS+	128	Так	51636	1451	1056	41000

1.3 Параметри Picnic

У цьому підрозділі коротко описано кожен з наборів параметрів для Picnic.

В табл. 16.24 наведені параметри для трьох рівнів безпеки L1, L3 і L5, відповідних безпеці AES-128, AES-192 і AES-256. Для кожного з трьох рівнів безпеки існує два алгоритми підпису, які використовують систему перевірки ZKB, засновану на перетворенні Fiat-Shamir (FS): pincic-L1-FS, picnic-L3-FS і picnic-L5-FS та засновану на перетворенні Unruh (UR): picnic-L1-UR, picnic-L3-UR і picnic-L5-UR. Існує також три набори налаштувань, що використовують перетворення FS і систему підтвердження з алгоритму покращення неінтерактивних нульових знань за допомогою додатків для постквантових підписів [285]. В табл. 16.25 наведені розмір ключа і підпису (в байтах) за рівнем безпеки. Всі параметри обрані так, що очікується, що вони забезпечать S біт безпеки від класичних атак і, як мінімум, S/2 біт безпеки від квантових атак [190].

Параметр u – кількість оспорюваних повторень застосовано лише до наборів параметрів picnic2. Параметр N - кількість сторін в імітації MPC завжди 64 для наборів параметрів picnic2 [285,285].

Для варіантів FS довжина підпису змінюється в залежності від виклику, тому вказується максимально можливий розмір разом із середнім розміром і стандартним відхиленням, що обчислюється за 100 підписами [286].

Таблиця 16.24.

Параметри рівня безпеки

Набір параметрів	S	n	s	r	Hash/KDF	L_h	T	u
picnic-L1-FS	128	128	10	20	SHAKE128	256	219	-
picnic-L1-UR							219	-
picnic2-L1-FS							343	27
picnic-L3-FS	192	192	10	30	SHAKE256	384	329	-
picnic-L3-UR							329	-
picnic2-L3-FS							570	39
picnic-L5-FS	256	256	10	38	SHAKE256	512	438	-
picnic-L5-UR							438	-
picnic2-L5-FS							803	50

Таблиця 16.25.

Розмір ключа і підпису (в байтах) за рівнем безпеки

Набір параметрів	Відкритий ключ	Секретний ключ	Підпис (max)	Підпис (avg., std. dev.)
picnic-L1-FS	32	16	34016	32838, 107
picnic-L1-UR			53945	-
picnic2-L1-FS			13786	12359, 213
picnic-L3-FS	48	24	76764	74134, 198
picnic-L3-UR			121837	-
picnic2-L3-FS			29742	27173, 315
picnic-L5-FS	56	32	132856	128176, 315
picnic-L5-UR			209506	-
picnic2-L5-FS			54732	46282, 613

2. Аналіз відомих атак на алгоритм

У цьому підрозділі аналізується схема підпису Picnic стосовно відомих атак. По-перше, надається спостереження, що коли ми маємо справу з ідеальними примітивами, Corollary 5.4 вже дає нам перевірену прив'язку до безпеки EUF-CMA (Existentially Unforgeable under Chosen Message Attacks – екзистенційна непідроблюваність при атаках на основі (адаптивно) вибраних повідомлень). Оскільки це примітиви, що створені за допомогою конкретних будівельних блоків, ми розглядаємо конкретні атаки на ці будівельні блоки. У цій схемі використовується класичний підхід, щоб перетворити Σ-протоколи в схеми підпису з випадковою моделлю оракула. На основі того факту, що з моменту введення випадкової моделі оракула не було виявлено жодної атаки, яка виникає з припущення, що геш-функція веде себе як випадкова модель оракула (за виключенням деяких штучних контрприкладів), тому можна стверджувати, що найбільшими атаками на підпис є атаки, які роблять недійсними твердження, зроблені для базових симетричних примітивів [285].

Всі криптографічні примітиви, за винятком односторонньої функції LowMC, спираються на SHA-3 функцію SHAKE, визнаний і стандартизований примітив, і вона використовується стандартним способом. Що стосується цих примітивів, то вже досягнуто значної згоди в питаннях безпеки завдяки широкому криптоаналізу всередині спільноти. Тому ці будівельні блоки не розглядаються як центральна поверхня атаки. Покращення атак на ці примітиви також призводить до покращення атак на схему підпису [285].

2.1 Використання та безпека LowMC

Отже, надалі робиться акцент на атаки на одностороннюю функцію *f*. По суті, функція *f* може бути будь-якою односторонньою функцією, але в Picnic використовується сімейство блочних шифрів LowMC, так як саме ці блочні шифри дали в результаті найбільш ефективні підписи. Зокрема, потрібно виходити з того, що використання LowMC, як вказано нижче, дає відповідне сімейство односпрямованих функцій $\{f_u\}_u \in K_\kappa$. Ці функції використовуються

для встановлення відповідних відносин між секретними та відкритими ключами. Зокрема, нехай

$$f_u(x) := E(x, u)$$

і нехай *E* позначає шифрування LowMC щодо одного блоку *u* на ключі *X*. Ключі у використаній схемі підпису генеруються наступним чином. По-перше, вибирають ключ шифрування LowMC – *x*, а також один блок *u* рівномірно випадковим чином. Потім відкритий ключ підпису p_k, а також секретний ключ підпису s_k визначаються наступним чином:

$$p_k := (y, u) = (f_u(x), u),\ s_k := (p_k, x).$$

Вибір кількості раундів у LowMC поставляється з показовим запасом безпеки. Для L1 рівня безпеки з 20 раундами найбільш відома атака на 12 раундів. Для L3 рівня безпеки з 30 раундами найбільш відома атака на 19 раундів. Для L5 рівня безпеки з 38 раундів найбільш відома атака – 26 раундів. І навіть ці атаки вимагають від зловмисника знання двох пар відкритого зашифрованого тексту для однієї й тієї самої пари ключів, в той час як у використаній схемі підпису зловмисник бачить лише одну пару вводу-виводу для кожного ключа [285].

2.2 Атаки в режимі, розрахованому на одного користувача

В режимі, розрахованому на одного користувача, зловмисник завжди бачить тільки одну пару ключів схеми підпису Picnic, тобто, одну пару відкритий текст-шифртекст ($f_u(x)$, u) LowMC відносно рівномірно випадкового ключа *x* і рівномірно випадкового блоку *u*. Отже, в цьому налаштуванні криптоаналітичні результати для LowMC також безпосередньо застосовані до схеми, що використовується. Слід зазначити, що можна навіть глобально використовувати *x* для подальшого скорочення розміру ключа p_k загальнодоступної версії [285].

2.3 Атаки в режимі, розрахованому на багато користувачів

Багатокористувацький режим точніше моделює реальність тому, що є кілька користувачів, кожен з відкритим ключем і зловмисник вважається успішним, якщо він може атакувати будь-якого з користувачів.

Багатокористувацький EUF-CMA. Приділяється особлива увага сценаріям атак, які стають можливими при переході до налаштування, розрахованого на багато користувачів. Тут зловмисник може побачити багато пар ключів підписання, і варто бути обережним у відношенні більш складних атак, які можуть мати місце. Зокрема, на відміну від атак в режимі, розрахованому на одного користувача, рішення вибрати незалежний і рівномірно випадковий блок *u*, будучи криптографічною функцією $E(\cdot, u)$ LowMC, на одну пару підписуючих ключів, виявляється важливим. В цих атаках один з *n* блочних ключів шифрування може бути відновлений за менший час, ніж час відновлення єдиного ключа, і атаки стають дуже ефективними для великого *n*. Інтуїтивно випадковий блок, що вибирає унікальну функцію за користувача і роботу для атаки на одного користувача (функцію), не можна використовувати для

одночасної атаки на іншого користувача (функцію). Крім того, Банегас і Бернштейн [285] нещодавно показали, що паралельні атаки пошуку співпадінь також можуть бути застосовані для квантових значень параметрів, які також підтримують випадковий вибір u на користувача. Можна вибрати менше значення, яке буде унікальним для кожного користувача (з потенційним зниженням безпеки), щоб зменшити розмір відкритого ключа. Однак, оскільки відкриті ключі в схемах вже малі (максимум 64 байта), в конструкції використовується повний випадковий блок, щоб бути максимально консервативним [285].

2.4 Атаки із заміною ключа

Це атаки, де є зловмисник, який має підпис σ_A повідомлення M з відкритим ключем $p_{k\,A}$, що дозволяє створити відкритий ключ $p_{k\,E}$ (з $p_{k\ A}$) таким чином, що підпис σ_A підтвердиться з ключем $p_{k\,E}$ та повідомленням *M*. Menezes і Smart [286] надають формальну модель для захисту від таких атак, які не покриваються моделлю безпеки EUF-CMA. Безпека від цих типів атак може бути досягнута в цілому. Це було показано в їх роботі і нижче подана їх теорема.

Теорема 1: Нехай (Gen, Sign, Verify) є EUF-CMA безпечною схемою підпису. Потім, (Gen, Sign', Verify) з Sign': = Sign (s_k, p_km) та p_k є однозначним кодуванням відкритого ключа і безпечною схемою підпису в налаштуванні, розрахованому на багато користувачів [286].

Вищевикладене призводить, зокрема, до утримання захищеності схем підпису sEUF-CMA. Відкритий ключ додається до повідомлення про підписання, а опис забезпечує однозначне кодування (оскільки відкритий ключ є парою бітових рядків, кодування є тривіальним). Отже, ми маємо наступний наслідок.

Наслідок 1: picnic-FS і picnic-UR забезпечують безпеку в режимі, розрахованому на багато користувачів [288].

2.5. Багатоцільові атаки

Дінур і Надлер [286] описують атаки на Picnic версії 1.0 (набори параметрів picnic-FS і picnic-UR). В той час набори параметрів Picnic2 не були визначені, але атака в рівній мірі відноситься до прямого створення примітивів протоколу KKW. Їхні атаки являють собою багатоцільові атаки, де у атакуючого є список значень виду y_i : = H (x_i),..., y_S : = H (x_S), а відновлення будь-якого з x_i призводить до успішної атаки. Спеціально значення x_i мають довжину k біт і зловмисник може відновити ключ k біт. Значення y_i можна взяти з [286]:

- одного підпису (у підписі є близько 2^7 значень);
- з декількох підписів при умові одного й того самого підписувача;
- з декількох підписів при умові різних підписувачів.

В атаці Дінура і Надлера значення x_i є параметром, що використовується для кожної сторони, в кожному примірнику MPC [286]. Функція H розширює x_i до випадкової довжини, що використовується під час моделювання протоколу MPC. В Picnic нащадки двох із трьох сторін розкриваються перевіряючому, а в Picnic2 розкриваються n–1 з n. Якщо зловмисник дізнається про відсутнього

нащадка, він може відновити секретний та загальний ключі підпису. Вихідна довжина залежить від набору параметрів, але завжди перевищує 600 біт. Що робить атаку неочевидною, так це те, що вихід *H* (випадкової довжини) не розкривається безпосередньо. Вони показують, що це – власність протоколів MPC, яка не робить звичайне поняття безпеки MPC (протокол MPC повинен гарантувати секретність вихідних даних, але припустима деяка хаотичність, якщо через це не виникають проблеми з секретністю даних). Вони також визначають кількість випадкових бітів, які можуть бути відновлені з примірника MPC. У всіх випадках можна ефективно відновити понад *k* бітів, так що може бути виконано тестування значення кандидата x_i. Однак у бітів від кожної цілі є різні позиції у випадковій довжині, що ускладнює ефективність проведення атаки. При типовій багатоцільовій атаці атакуючий обчислює x_0, обчислює $y_0 = H(x_0)$, потім порівнює y_0 з y_i, ефективно використовує структури даних (наприклад, геш-таблицю або дерево пошуку). Але тут порівняння довжини кандидата y_0 (з усіма відомими бітами) з цільовою довжиною y_i (де для кожної з них відома підмножина бітів) не є очевидним. Дінур і Надлер показують, що його можна зробити таким і точно кількісно оцінити витрати при різних налаштуваннях. У кращому випадку, коли всі підписи створюються одним підписувачем, їх атаки коштують 2k–7/S (інформація теоретично оптимальна). В інших випадках атака коштує дорожче, але все одно нижче очікуваного рівня безпеки [286].

Пом'якшення наслідків. Версія специфікації 2.0 вносить зміни, щоб пом'якшити ці атаки (у всіх наборах параметрів). Зміна додає додаткову інформацію до входу *H* (названою *salt*) так, щоб нащадок кандидата x_0 не міг бути перевірений, витримавши порівняння y_0 до всього y_i, оскільки кожний y_0 вираховується з використанням різної *salt*. *Salt* гарантує, що y_0 потрібно буде перерахувати з правильною *salt* перед кожним порівнянням. Для вирішення всіх трьох варіантів атаки *salt* повинна бути унікальною для кожного підпису, для кожного підписувача і для кожного виклику *H*. Першою зміною є вибір випадкової завади для кожного підпису довжиною 256 біт. Це гарантує, що (з високою ймовірністю) *salt* унікальна. Потім, щоб гарантувати, що *salt* є унікальною у підписах, також включаємо пару лічильників; перше значення відповідає номеру примірника MPC, а друге відповідає номеру виклику *H*. Спеціалізація вже використовує метод поділу домену, де різні геш-функції створюються для різних цілей наступним чином: $H_i(x)=H(i||x)$, де || - символ конкатенації. Цей механізм також допомагає гарантувати, що *salt* є унікальною, наприклад, при обчисленні дерева нащадків ми можемо використовувати H_i та при обчисленні дерева Меркле використовувати H_j, і не турбуватися про те, що (salt, counters) пари повторюються в обох деревах. Деякі додаткові дані також повинні бути гешовані, але витрати на ЦП в еталонних показниках суттєво не зросли. Ймовірно, це пов'язано з тим, що геш-входи були короткими для початку, і залишаються короткими (меншими, ніж розмір блока геш-функції) навіть з *salt* [286].

Висновки

1. У ході семінару другого етапу NIST США рекомендував до подальших досліджень 9 криптографічних примітивів типу ЦП. Порівняння постквантових алгоритмів можна дослідити за табл. 16.23.

2. За результатами досліджень перспективних постквантових криптографічних алгоритмів типу цифровий підпис протягом 2-го раунду конкурсу NIST США було отримано наступні результати: обрані алгоритми – фіналісти та альтернативні алгоритми. У якості алгоритмів - фіналістів обрані такі алгоритми ЦП, як CRYSTALS-DILITHIUM, FALCON та Rainbow. У якості альтернативних алгоритмів – GeMSS, Picnic та SPHINCS+.

3. На конкурс NIST США щодо стандарту ЕП було подано механізм ЕП, що ґрунтується на стійкості геш-функції та блокових потокових шифрів (Picnic). Наразі він досліджується на третьому етапі конкурсу NIST США у якості альтернативного алгоритму.

4. Побудова ЕП в алгоритмі Picnic заснована на перетворенні Fiat-Shamir (FS) та на перетворенні Unruh (UR).

5. Оцінка параметрів алгоритму говорить про те, що сам алгоритм досить збалансований за усіма критеріями, але має великий розмір підпису. Спроби зменшення розміру підпису призводять до зменшення безпеки алгоритму.

6. Після аналізу відомих атак на алгоритм можна зробити висновок, що алгоритм є безпечним і його можна рекомендувати для використання.

7. Великою перевагою алгоритму є великий простір для покращення та змін, які постійно проводяться, та, коли буде представлена кінцева версія алгоритму, його можна затверджувати для застосування.

16.5 Аналіз стійкості постквантового електронного підпису DILITHIUM до атак на помилки

Вступ

На сьогодні поряд із задачею стандартизації постквантових алгоритмів стоїть задача поступового заміщення класичних алгоритмів, які використовуються у реальних додатках та пристроях на квантовостійкі. Критичною задачею у цьому процесі переходу є забезпечення відповідного рівня доказової стійкості та захищеності від атак не тільки в теорії, але і від атак на реалізацію та атак по стороннім каналам. Цей розділ присвячений дослідженню саме такої, практичної, стійкості Dilithium до атак на помилки.

Схема ЕП Dilithium [265] пройшла до третього раунду процесу постквантової стандартизації NIST в якості одного з трьох фіналістів. Статус «фіналіст» означає, що дана схема придатна до застосування у багатьох додатках та випадках використання [266]. Характерними рисами Dilithium є маленькі розміри ключів та підпису, достатня швидкість, простота та гнучкість (адаптація до різних рівнів стійкості).

Очікується, що незабаром, після закінчення третього раунду, відбудеться процес стандартизації одного з двох фіналістів ЕП (Dilithium або Falcon, обидві схеми засновані на математиці алгебраїчних решіток).

1. Основні відомості про стійкість Dilithium

Доцільно сказати, що теоретична та практична стійкість повинні доповнювати одна іншу, тому спочатку наведемо рівні доказової стійкості для України та сумісність з ними Dilithium.

Для використання в Україні пропонуються наступні чотири рівні доказової стійкості (класична / квантова):

0-й рівень – 128 біт / 64 біт,

1-й рівень – 256 біт / 128 біт,

2-й рівень – 384 біт / 192 біт,

3-й рівень – 512 біт / 256 біт.

Dilithium забезпечує 1-й (AES128), 2-й (SHA256/ SHA3-256) та 3-й (AES192) рівні стійкості згідно NIST. Наприклад, стійкість AES128 означає, що зламати схему так само важко, як зламати 128-бітний AES повним перебором.

Важливо розуміти можливості таких атак на помилки на алгоритми ЕП, але до того, як буде прийнято рішення щодо стандартизації постквантового ЕП в Україні, щоб забезпечити їх безпечне впровадження у перехідний та постквантовий періоди.

Стійкість Dilithium заснована на складності задачі модульного навчання з помилками (MLWE) та модульного короткого цілого рішення (MSIS). Схема підпису Dilithium використовує структуру Фіат – Шаміра з перериваннями. Структура Фіат – Шаміра потребує використання нонсу (одноразового випадкового значення), через це вона має добре відому вразливість – при використанні однакового нонсу для різних повідомлень стає можливим відновити особистий ключ. Автори Dilithium запропонували контрзахід до цієї атаки – вироблення нонсу шляхом гешування повідомлення та ключа, то ж кожен раз нонс є унікальним. Таке рішення забезпечує детермінованість схеми ЕП, однак воно не стає у нагоді, коли зловмисник застосовує атаку на помилки. Він може ввести обчислювальну помилку в одному з підписів, то ж два повідомлення будуть мати однаковий нонс.

2. Параметри Dilithium

Спершу наведемо основні параметри Dilithium, які знадобляться при опису атаки на помилки (див. підрозділ 4), причому c, h, z – частини підпису, де:

- c – значення після застосування гешування;
- σ – цифровий підпис;
- s – малий елемент особистого ключа;
- y – шум (компонент маскування);
- tr – геш-значення відкритого ключа.

Функція XOF у підпису Dilithium застосовується для детермінованого розширення початкового значення матриці A, використовуючи функцію розширюваного виводу SHAKE128. Ця функція зменшує розміри відкритого та особистого ключів, оскільки потрібно зберігати тільки ρ замість повного значення матриці A .

3. Сутність атаки на помилки

Атаки на помилки вперше були представлені в 1996 р. Боне та ін. [291] для реалізації атаки на RSA-CRT, а незабаром Біхам і Шамір описали диференціальну атаку на помилки стосовно шифру DESblock [292]. Атаки на помилки – це розповсюджений метод криптоаналізу, який полягає у внесенні помилок на будь-якому етапі дії електронної схеми, у нашому випадку, схеми ЕП Dilithium. Даний тип атак належить до атак по стороннім каналам.

Атаки на помилки на електронні системи вивчалися більше 40 років. Відтоді до реального обладнання застосовувались різні форми внесення помилок, такі як зміна напруги, подача високих температур на апаратні засоби, використання рентгенівських променів тощо [293].

Через необхідність фізичного доступу до пристрою внесення помилок в основному загрожували захищеному відмовостійкому апаратному забезпеченню як смарт-картки, так і жорстким апаратним модулям безпеки. Це змінилося з появою апаратної помилки Rowhammer, що впливає на основну пам'ять комп'ютера [294]. У 2014 р. Кім та інші показали, що зловмисне програмне забезпечення може використовувати пошкодження пам'яті, щоб викликати обернення бітів у сусідніх областях пам'яті, минаючи захист пам'яті обладнання [294]. За допомогою Rowhammer помилки стало можливо вносити дистанційно і за допомогою лише програмного забезпечення. Такий спосіб став потужною можливістю внесення помилки.

Сутність диференційної атаки на помилки на схему Dilithium заключається у наступному. Зловмисник створює таку ситуацію, в якій жертва двічі підписує одне і те ж повідомлення, але в одному з підписів внесена обчислювальна помилка (наприклад, за рахунок збоїв годинника). Це призводить до того, що різні підписи використовують однаковий нонс, таким чином стає можливим відновити особистий ключ. При диференційних атаках на помилки різниця між помилковим та правильним виводом використовується для визначення інформації про особистий ключ.

4. Можливості здійснення атаки на помилки

Спочатку визначимо, якою може бути помилка. Помилкою може бути пропуск інструкцій, арифметична помилка, збій у сховищі тощо. Помилки можуть застосовуватися протягом великого періоду часу виконання, а не лише у конкретних операціях.

Коротко проведення атаки можна описати наступним чином. Підписувач двічі виконує підписання одного повідомлення M. Перший раз помилка не вноситься та отримуємо дійсний і коректний підпис $\sigma = (\mathrm{z}, \mathrm{h}, c)$. Другий раз вноситься помилка (змінні при підписі з помилкою позначаються штрихом $'$). Більш точно, помилка вноситься така, що за рахунок детермінованості y' є непорушеним та дорівнює y, проте $c' \neq c$, таким чином $\mathrm{z}' = \mathrm{y} + c'\mathrm{s}_1$.

Однак структура Фіат – Шаміра з перериванням створює додаткові перешкоди для здійснення атаки. Існує вимога, щоб обчислення як дійсного, так

і помилкового підпису закінчувалось однаковою ітерацією циклу переривання. Іншими словами, коли k_f позначає кінцеве значення лічильника циклу k, потрібно, щоб $\Delta k_f = k_f - k'_f = 0$. Зауважимо, що в алгоритмі підпису лічильник циклу k вводиться до детермінованої вибірки. Отже, щоб виконувалася тотожність $\mathrm{y} = \mathrm{y}'$, постає вимога $\Delta k_f = 0$. Через проміжні помилки та вплив тестів відхилення це, очевидно, не гарантується.

Далі описуються можливі сценарії здійснення атаки на помилки.

4.1 Випадкова помилка при зверненні до геш-функції

При даному способі внесення помилки до обчислення $\mathrm{c} \in B_{60} := H(\mu, w_1)$ вводиться випадкова помилка. Наприклад, можна використати один із входів μ, w_1 безпосередньо перед їх використанням у H, або внести помилку безпосередньо у саму геш-функцію H. У [295] показано, що зловмисник може внести помилку в правильну ітерацію k_f, тобто, в останню при обчисленні помилки. Якщо крок відхилення проходить з різним c', секретний елемент s_1 може бути відновлений.

Атака відбулась успішно, якщо відновлений елемент особистого ключа s_1 є малим, а Δc є оборотним. Це вірно з дуже високою ймовірністю. Частка оборотних поліномів у R_q дорівнює $(1-1/q)^n$ [296], що для параметрів Dilithium приблизно становить $1-2^{-15}$. Знаючи s, за допомогою лінійної алгебри можна відновити повний секретний ключ. Також можна використати малу норму cs, обчислюючи $\|\Delta \mathrm{z}\|_2$ і перевірити, чи вона не нижче певного порогу.

4.2. Випадкова помилка у множенні поліномів геш-функції

Ця помилка зводиться до зміни складової підпису $c := \mathrm{H}(\mu, \mathrm{w}_1)$ у геш-значенні H, яке використовується при обчисленнях її вхідних параметрів μ, w_1. Геш-значення повідомлення або відкритого ключа μ також використовується як початкове значення (seed) для детермінованої вибірки, отже, помилки в обчисленні $\mu := (tr, M)$ не можна використати. Але можна використати помилки в обчисленні $\mathrm{w} := \mathrm{Ay}$, які призводять до неправильного w_1. Необхідні множення n поліномів у q можна ефективно реалізувати за допомогою теоретико-числового перетворення (NTT). Тим не менше, час множення є більшим, ніж час гешування, тому множення може підходити для атак на помилки.

Варто зазначити, що зменшення кількості коефіцієнтів помилки збільшує ймовірність успіху, оскільки непорушені коефіцієнти w чітко проходять відхилення, а одного зміненого достатньо для досягнення $\Delta c \neq 0$. Таким чином, на відміну від інших сценаріїв з [295], таке застосування помилки має набагато сильніший вплив. Більш імовірно, що атака буде успішною, якщо вносити помилку не у прямий NTT від y, а у зворотній, застосований до w, у цьому випадку порушені лише два коефіцієнти. Однак, однокоефіцієнтні помилки

трохи рідше призводять до успішного відновлення ключів. Це пов'язано з тим, що коефіцієнт помилки w' може округлитися до правильного $w_1' = w_1$, що призводить до $\Delta c = 0$ і помилку не можна використати.

4.3 Випадкова помилка при завантаженні особистого ключа та розширенні початкового значення

Сутність даної помилки – виконати розширення початкового значення ρ до матриці A. Дана операція є привабливою з трьох причин:

- помилка вноситься перед входом у цикл переривання і, таким чином, завжди виконується за один час;
- розширення до A дає головний внесок до загального часу виконання;
- A має більший слід (20 кБ у Dilithium-III), ніж інші змінні, і потенційно зберігається в пам'яті протягом тривалого часу. Тому, при обчисленні його кеш-значення не потрібно повторно запускати Expand A для кожної операції підпису. Також існує атака на пам'ять Rowhamer, яка має схожий принцип дії.

Різниці (диференціали) у матриці A можна досягти, використовуючи початкове значення ρ, наприклад, під час завантаження особистого ключа або вносячи помилку у розширення $A \in R_q^{k \times l} := \mathrm{ExpandA}(\rho)$. У першому випадку поліноми $k \cdot l$, що містять A, створюються за допомогою незалежних викликів SHAKE у методі розширення ρ до A. І помилка у перестановці SHAKE призводить лише до одного полінома з помилкою. Отже, після множення матриці на вектор Ay маємо n різних коефіцієнтів w. У другому випадку помилка вводиться у напрямку ρ під час імпорту або при зберіганні значення, що призводить до зовсім іншого значення A і, таким чином, до w. Цей тип помилок варто застосовувати, якщо використовуються контрзаходи, такі, як подвійне обчислення, тому що у цьому випадку помилку складно виявити.

Хоча ймовірність успіху другого методу менше у порівнянні з першим [295].

4.4 Повторне використання нонсу

Задача даної атаки – створити сценарій повторного використання нонсу з використанням помилок для генерації екземпляру LWE, який буде мати тривіальне розв'язання. Спільною рисою ряду схем (NewHope, Frodo, Dilithium, Kyber) на основі LWE є процедура вибірки помилок.

Логіка атаки. У процедурі генерації ключів як компонент помилки, так і компонент секрету екземпляру LWE генеруються з використанням початкових значень (seed), які відрізняються один від одного лише одним значенням нонсу. Ці початкові значення далі вводяться до функції вибірки, яка генерує необхідні поліноми за рахунок використання функцій XOF з родини SHA3. Найбільш важливим спостереженням є те, що початкові значення відрізняються лише значенням нонсу, щоб згенерувати компоненти секрету та помилки екземпляру LWE. Спосіб виконання даної атаки полягає у тому, щоб пересвідчитись, що використовуються однакові нонси як для генерації секрету, так і помилки. А робиться це за допомогою внесення відповідних помилок. Далі показано, як

знайдені вразливості можуть призвести до атак відновлення особистого ключа Dilithium. У [297] було досліджено можливість здійснення атаки по відновленню ключа на Dilithium на схему, яка виконує операції з модулями (матрицями / векторами / поліномами у кільці). Атака націлена на процедуру генерації ключів. Визначимо екземпляр LWE t як тип $\mathrm{LWE}_{\mathrm{PK}}$. Кратні поліноми у відповідних компонентах секрету та помилки створюються з використанням дуже схожих початкових значень (seed), які відрізняються лише значенням нонсу, яке детерміновано збільшується. Якщо до реалізації повторного використання нонсу при генерації компонентів секрету і помилки можна внести кратні помилки, то $\mathbf{t}$ зводиться до множини визначених лінійних рівнянь ($k \times n$ невідомих, оскільки $k > l$). Існує достатньо специфічна, але істотна відмінність щодо розкриття екземплярів LWE у схемі Dilithium. Відкритий ключ розкриває лише $\mathbf{t}_1$, d верхніх біт $\mathbf{t}$ тоді як $\mathbf{t}_0$ (компонент нижнього порядку) є частиною особистого ключа. Навіть при повторному використанні нонсу не має можливості тривіально отримати секретне значення s з помилкового відкритого ключа. Але відзначимо, що аналіз стійкості підпису Dilithium робиться за припущення, що усі t можна відновити лише при перегляді декількох підписів, які були згенеровані на одному ключі. Таким чином, доцільно припустити, що, після успішного внесення помилок до процедури генерації ключів, це призводить до успішної атаки відновлення ключа у схемі підпису Dilithium.

4.5 Часткове повторне використання нонсу

Даний спосіб атаки спочатку застосовувався до підписів на еліптичній кривій [298], потім був перенесений на схеми на решітках та для Dilithium було отримано 100 % ймовірність успіху атаки [295].

Алгоритм атаки

1) Внести помилку $\mathrm{y}' \in S^{l}_{\gamma_1 - 1}$, але таку, що лише один коефіцієнт $\left(\underline{y'}_u\right)_v \in \mathrm{y}$ (з індексом $u \in \{0,...,l-1\}$) та $v \in \{0,...,n-1\}$ змінюється на випадкове значення. Це все ще призводить до зовсім іншого w_1 а, отже, до різного c' та $\mathrm{z}' = \mathrm{y}' + c'\mathrm{s}_1$.

2) Обчислити $\mathrm{s}_1 = \Delta c^{-1} \cdot \Delta \mathrm{z}$.

3) Визначити u, просто використовуючи один індекс, для якого $s_{1,u} \notin S_\eta$. Для всіх $i \neq u$ маємо, що $\Delta y_i = 0$, таким чином вдається відновлення цих поліномів ключа.

4) Встановити поріг $\delta = 60\eta$ такий, що $\|c\mathrm{s}_1\|_\infty \leq \delta$ виконується для будь-яких c і s_1.

5) Визначити індекс v і далі положення помилки з допомогою індексу максимального $|\Delta z_u|$. Оскільки введена помилка $\left(\underline{y'}_u\right)_v$ може спричинити будь-яку різницю від значення, але в середньому вона буде великою. Як

очікувалось, $\left|\left(\underline{y'}_u\right)_v - \left(\underline{y}_u\right)_v\right|$ буде $\frac{2_{\gamma_1}-1}{3}$, оскільки обидва ці коефіцієнти є випадковими значеннями в $\left[-(\gamma_1-1),(\gamma_1-1)\right]$ (за припущенням).

6) Відновити $S_{1,u}$. Для цього потрібно видалити рядок v лінійної системи $s_{1,u} = \Delta c^{-1} \cdot \Delta z_u$, відгадати значення $\left(\underline{s}_{1,u}\right)_v$ (вичерпним пошуком), розв'язати для повного $S_{1,u}$ і перевірити, чи є він у S_η.

5. Заходи щодо протидії можливим атакам

Після аналізу ряду джерел було знайдено п’ять заходів протидії атакам на помилки. Далі наведено сутність заходів, їх переваги та недоліки.

5.1 Повторне обчислення підпису

Для того, щоб виявити помилку у підписі, можна виконати його повторне обчислення та порівняти два підписи. Але якщо внести помилку двічі або помилка є постійною (якщо внести її при завантаженні особистого ключа), то цей захід не захистить від атаки. Також його недоліком є подвійний час виконання.

5.2 Перевірка після підпису

Більшість способів атаки призводять до того, що підписи є недійсними. Таким чином, якщо перевірити підпис, то буде виявлено помилку підпису, що є ефективним заходом протидії. При цьому час перевірки приблизно у три рази менше часу вироблення підпису [299], тому цей варіант є більш ефективним, ніж повторне обчислення підпису.

Але недоліком є те, що даний захід не може виявити помилки, внесені до вибірки y, оскільки при цьому виробляються дійсні підписи.

5.3 Додаткова випадковість

Ще одним контрзаходом є внесення додаткової випадковості до детермінованої вибірки шуму y. Можна просто взяти вибірку випадкового $r \leftarrow 0{,}1^{256}$, а потім викликати y:= $DeterministicSample(K, \mu, k, r)$. Це ефективно пом’якшує диференційну атаку на помилки, оскільки виклик помилки до алгоритму підпису використовує різні y і, отже, $\Delta y \neq 0$. Більше того, цей метод також може перешкодити подальшим атакам по стороннім каналам, що стають побічним ефектом детермінізму. Як зазначають [300 - 302], змішування відомого повідомлення μ з таємним початковим значенням K у геш-функції (y Dilithium - це SHAKE у DeterministicSample) відкриває двері для DPA-подібних атак. Функції гешування важко захистити від таких атак; використання додаткового випадкового значення може бути альтернативою. Додатковими перевагами цього контрзаходу є його простота та незначна тривалість виконання. Крім того, на відміну від прямих реалізацій двох попередніх контрзаходів, він є однопрохідним і тому не потребує збереження копії повідомлення в пам'яті.

Наприклад, автори qTESLA вимагають використовувати цей контрзахід через наявність зазначених вище атак.

Недоліком даного заходу протидії є його ймовірнісний характер та необхідність у генераторі випадкових чисел. Такий генератор може бути доступний не на всіх пристроях, особливо малоресурсних. Також, введення цього контрзаходу скасовує гарантії стійкості, хоча конкретні атаки невідомі. Автори Dilithium "все ще рекомендують використовувати детерміновані підписи, за винятком середовищ, які можуть бути вразливими до вищезгаданих атак по стороннім каналам" [299]. Однак визначити, чи є вразливим середовище, чи ні, непросто, як це чітко показано помилкою Rowhammer.

5.4 Перевірка значення нонсу

Щодо атаки повторного використання нонсів у еталонній реалізації Dilithium може стати легкою мішенню за рахунок атак на помилки. Основна причина, однак, пов'язана з використанням початкових значень (seed) для створення компонентів секрету та помилки, які змінюються лише на один або два байти через нонс. Значення нонсу насамперед визначає різницю між компонентами секрету та помилки. Таким чином, важливо провести перевірку значення нонсу, що може унеможливити атаку. Існує маса відомих вразливостей задачі LWE. В [295] реалізували одну із вразливостей за допомогою помилок. Таким чином, проведення простих перевірок секретних та помилкових компонентів екземплярів LWE на предмет відомих тривіальних слабкостей також може бути потенційним контрзаходом проти даної атаки.

5.5 Обчислення середнього значення та дисперсії вибірки

Наступний захід протидії, запропонований у [295], обчислює середнє значення вибірки та дисперсію вибірки, одночасно перевіряючи наявність повторень. Перевірка дисперсії вибірки є особливо важливою, оскільки цей параметр пов’язаний зі складністю задачі LWE криптографічної схеми. Зменшення дисперсії вибірки помилок може спростити вирішення екземплярів LWE. Ці контрзаходи також виявляють помилки у вибірці та будь-які зловмисні реалізації. В апаратному забезпеченні цей тест буде обчислюватися після обсягу вибірки в два рази, тому не має необхідності явно обчислювати розподіл. Це зручно для такої схеми як Dilithium, яка вимагає $n = 256$ для кожного підпису.

Обчисливши середнє значення та дисперсію, можна провести статистичні тести, щоб побачити, чи ці значення є допустимими. Для середнього значення проводиться t-тест, щоб побачити, чи середнє значення цієї вибірки знаходиться в допустимих межах (попередньо розрахованих). Для дисперсії проводиться тест x_i-квадрата, щоб побачити, чи дисперсія вибірки знаходиться в межах допустимого, заздалегідь визначеного, значення.

Даний контрзахід додає приблизно 44 % необхідних ресурсів до вибірки CDT. Однак, він має фіксовану вартість та необхідності у додаткових апаратних ресурсах не буде, якщо дискретизатор помилок був повільнішим чи більшим. Цей контрзахід також має мінімальний вплив на продуктивність, оскільки більшість розрахунків обчислюються в паралельному розподілі помилок.

В цілому, для завершення обчислень потрібен лише один додатковий такт після того, як розподіл помилок завершить генерацію *n* вибірок.

Висновки

1. Із наведеного можна зробити висновок, що диференційні атаки на помилки становлять загрозу сучасним та перспективним схемам електронного підпису. Це стосується і фіналіста конкурсу на постквантовий стандарт Dilithium.

2. Слабкими властивостями схеми Dilithium щодо атаки на помилки є геш-функція (можливо внести випадкову помилку при зверненні до геш-функції та в операцію множення поліномів), а також функція розширення початкового значення та стадія завантаження особистого ключа. Також з'являється вразливість через використання нонсу – порушник може використати двічі однакове значення нонсу (частково чи повністю).

3. Основними методами захисту від вказаних атак можуть бути:

- повторне обчислення підпису;
- перевірка підпису після підписання, що є втричі швидшим, ніж попередній метод;
- внесення додаткової випадковості до детермінованої вибірки шуму;
- перевірка значення секретних та помилкових компонентів (нонсу);
- обчислення середнього значення та дисперсії вибірки та перевірка на приналежність заданому діапазону.

Так, наприклад, зменшення дисперсії свідчить про те, що варіант LWE, на якому заснована стійкість Dilithium, значно простіше вирішити.

4. Тому розробникам постквантових схем ЕП доцільно врахувати знайдені вразливості для забезпечення стійкості до атак на помилки та застосувати знайдені рішення щодо захисту від даних атак.

5. Потрібно взяти до уваги можливі побічні ефекти після застосування контрзаходів, такі як збільшення необхідних ресурсів, порушення вимоги детермінованості схеми, зменшення ефективності.

16.6 Аналіз апаратних реалізацій алгоритмів електронного підпису QTESLA, CRYSTALS-DILITHIUM I MQDSS на різних рівнях безпеки

Вступ

Як відомо, існуючі алгоритми криптографії з відкритим ключем, що засновані на RSA та еліптичних кривих, надають гарантії безпеки, які супроводжуються складністю; також можна говорити про неможливість вирішення завдань цілочисельної факторизації і дискретного логарифму. Але експерти прогнозують, що створення квантового комп'ютера зможе зламати класичні криптографічні алгоритми. Через цю майбутню проблему Національний інститут стандартів і технологій (NIST) разом із провідними вченими у галузі криптографії розпочав відкритий процес стандартизації алгоритмів з відкритим ключем для квантових атак. Метою цього розділу є аналіз

та порівняння трьох апаратних реалізацій кандидатів 2-го раунду конкурсу NIST на алгоритм електронного підпису.

Криптографія з відкритим ключем – це фундаментальний протокол безпеки для всіх форм цифрового зв'язку, дротового або бездротового. Криптографія з відкритим ключем має три основні криптографічні функції, а саме: шифрування з відкритим ключем, електронні підписи (ЕП) і обмін ключами. Алгоритми криптографії з відкритим ключем на основі RSA і еліптичних кривих забезпечують гарантії безпеки, засновані на складності вирішення завдань цілочисельної факторизації і дискретного логарифму. Пітер Шор показав, що квантові комп'ютери можуть факторизувати цілі числа за поліноміальний час, роблячи традиційні алгоритми криптографії з відкритим ключем неефективними. Після цього криптографи стали шукати надійні альтернативи, такі як криптографічні алгоритми на основі решіток та математичних кодів.

З історичної точки зору у 1997 р. NIST запросив рекомендації у громадськості для визначення заміни стандарту шифрування даних (DES) Advanced Encryption Standard (AES) [306]. Відтоді відкриті криптографічні оцінки стали способом вибору криптографічних стандартів. Наприклад, NESSIE (2000-2002), eSTREAM (2004-2008), CRYPTREC (2000-2002), SHA-3 (2007-2012) і CAESAR (2013-2019) прийняли цей підхід. У цих оцінках головним параметром була безпека. Продуктивність у програмному забезпеченні, продуктивність у прикладних специфічних інтегральних схемах (ASIC), продуктивність у FPGA та можливість реалізації з використанням обмежених ресурсів (невеликих мікропроцесорів та малопотужних апаратних засобів) є вторинними критеріями. У конкурсі AES Rijndael мав найшвидшу ASIC і другу найшвидшу реалізацію FPGA з тими ж гарантіями безпеки, що й конкуренти [307].

У розділі описується порівняння апаратного забезпечення трьох алритмів підпису (qTesla, Crystals-Dilithium, MQDSS), які, зокрема, є кандидатами 2-го раунду конкурсу NIST PQC, а алгоритм Crystals-Dilithium – фіналістом цього конкурсу.

1. Постквантова криптографія

Експерти та вчені у галузі криптографії почали розробляти постквантові криптографічні алгоритми, які зможуть вистояти проти атак постквантових комп’ютерів. Безперечно, алгоритми постквантової криптографії поділяються на класи. У цьому розділі розглядаються два алгоритми на основі алгебраїчних решіток (qTesla та Crystal-Dilithium) і один алгоритм на основі багатовимірної криптографії (MQDSS) [303-310]. Криптографія на основі решіток заснована на підході побудови алгоритмів асиметричного шифрування з використанням задач теорії решіток, тобто, задач оптимізації на дискретних адитивних підгрупах, що задані у множині $\mathbb{R}^n$. Такі алгоритми забезпечують найкращу продуктивність, але є найменш консервативними серед усіх. Криптографія на основі решіток ґрунтується на вирішенні наступних обчислювально важких задач:

• Задача знаходження найкоротшого вектора (SVP, Shortest Vector Problem) – знайти в заданому базисі решітки ненульовий вектор по відношенню до визначеної нормалі.

• Задача знаходження ідеального найкоротшого вектора (ISVP, Ideal Shortest Vector Problem) не вважається NP-складною. Однак не існує відомих решіток, що засновані на методі редукції, значно більш ефективних на ідеальних структурах, чим на загальних.

• Задача знаходження (приблизно) найкоротшого незалежного вектора (SIVP, Shortest Independent Vector Problem), в якій є базис решітки *B* і необхідно знайти *n* лінійно незалежних векторів.

• Задача знаходження найближчого вектора (CVP, Closest Vector Problem) – знаходження вектора за заданим базисом і деяким вектором, який не належить решітці, при цьому максимально схожий за довжиною із заданим вектором.

Навіть з квантовим комп'ютером SVP представлений багаточленом у степені *n*. Інші криптографічні алгоритми на основі алгебраїчних решіток засновані на проблемі короткого цілочисельного рішення (SIS).

Багатовимірна криптографія або багатовимірна криптографія відкритого ключа – це загальний термін, що описує асиметричні криптографічні схеми, побудовані на вирішенні рівнянь, заснованих на багатовимірних поліномах над кінцевим полем *F*. Безпека багатовимірної криптографії заснована на припущенні, що вирішення системи квадратичних багаточленів над кінцевим полем *F*, в загальному випадку, є NP-повною задачею в сильному сенсі або просто NP-повна. Ці схеми часто вважаються гарними кандидатами для постквантової криптографії також через те, що вони пропонують найкоротші підписи.

Існують й інші класи постквантових криптографічних алгоритмів, такі як: криптографія на основі математичних кодів, ЕП на основі гешування та інші криптографічні методи. Але у розділі розглядаються лише алгоритми, що засновані на багатовимірній криптографії та криптографії на основі решіток через те, що вони є найбільш використовуваними, що можна зрозуміти з табл. 16.26, де показано, що алгоритми ЕП цих двох класів мають найбільше представників у конкурсі NIST.

Наведена кількість алгоритмів для оцінки NIST PQC раунду 1 і раунду 2 (всередині круглих дужок).

Таблиця 16.26. Математика ЕП і КЕМ, що представлені на конкурс PQC

Математика	Проблема, що вирішується	Підписи	КЕМ	Всього
Решітки	Пошук найкоротшого вектору, найбільш близького вектору CVP (Closest Vektor Problem)	5 (3)	23 (9)	28 (12)
Коди	Декодування випадкового лінійного коду	3 (0)	17 (7)	20 (7)
Багатовимірна поліноміальна	Розв'язання багатовимірних квадратичних рівнянь	8 (4)	2 (0)	10 (4)

Продовження табл. 16.26.

Геш	Стійкість до атаки знаходження прообразу геш-функції	3 (2)	0 (0)	3 (2)
Ізогенії	Пошук ізогенного відображення між еліптичними кривими з однаковою кількістю точок	0 (0)	1 (1)	1 (1)
Інші	–	2 (0)	5 (0)	7 (0)
Всього	–	21 (9)	48 (17)	69 (26)

1.1 Електронні підписи

Поточний процес стандартизації PQC NIST консолідує невразливих кандидатів після кожного наступного раунду. Кожен кандидат у конкурсі PQC NIST реалізує одну з трьох функцій: шифрування відкритого ключа, електронний підпис і механізм інкапсуляції ключа (KEM). Табл. 16.26 показує кількість представлень PQC раунду 2 і підсумовує їх функціональність і математичну складність.

NIST розраховує стандартизувати кілька алгоритмів, щоб забезпечити різні компроміси залежно від програми (швидкість і потужність пам'яті тощо). Пряме порівняння гарантій безпеки, що забезпечуються алгоритмами PQC-кандидатів, є складним завданням у відсутність стандартної платформи квантових обчислень, відмінностей у математичних функціях, що лежать в основі алгоритмів, і складного алгоритму специфікації порівняно з попередніми криптографічними конкурсами [304].

Електронні підписи PQC працюють за принципом, що відправник підписує повідомлення за допомогою закритого ключа, а одержувач перевіряє підпис, використовуючи відкритий ключ відправника. Ці алгоритми використовують три функції:

- *crypto_sign_keypair* генерує відкритий ключ p_k та секретний ключ s_k;
- *crypto_sign* приймає s_k і повідомлення *m* плюс його довжина *mlen* і видає підпис s_m , доданий до повідомлення;
- *crypto_sign_open* приймає *pk, sm* та довжину *smlen* і видає повідомлення *m.*

1.2 Класифікація безпеки алгоритмів PQC

NIST виділяє п’ять категорій потужності безпеки:

Перший рівень безпеки ⇒ еквівалентний пошуку ключа AES-128.

Другий рівень безпеки ⇒ еквівалентний пошуку колізій SHA-256/SHA3- 256.

Третій рівень безпеки ⇒ еквівалентний пошуку ключа AES-192.

Четвертий рівень безпеки ⇒ еквівалентний пошуку колізій SHA-384/SHA3-384.

П’ятий рівень безпеки ⇒ еквівалентний пошуку ключа AES-256.

Рівні безпеки PQC алгоритмів другого раунду NIST наведені у табл. 16.27.

2. Оцінка апаратних засобів PQC

Алгоритми ЕП PQC, їх реалізація і характеристики безпеки узагальнені в табл. 16.27. Розглядається програмна реалізація із використанням Xilinx Virtex-7 FPGA.

Розглядаються такі показники продуктивності: затримка, область і область затримки. Затримка – це час, необхідний системі для отримання вихідного сигналу з моменту подачі вхідного сигналу. Пропускна здатність – це максимальна швидкість, з якою можна забезпечити виведення. Мінімальна кількість тактових циклів між двома послідовними входами є інтервалом ініціювання і є мірою пропускної здатності [304].

Алгоритми, що не підтримують визначений рівень безпеки, позначені символом "–".

У табл. 16.28 і 16.29 представлені службові дані апаратних засобів і синхронізації для реалізації алгоритмів підпису, верифікації і генерації ключової пари PQC відповідно при синтезі без будь-яких додаткових обмежень (затримки).

Серед алгоритмів підпису рівня безпеки 1 Crystals-Dilithium і qTesla мають затримки менше мільйона циклів. Crystals-Dilithium (для генерації підпису) є гарним кандидатом для пристроїв Інтернету речей. Він має рівень безпеки 1 і є другим найшвидшим і другим найменшим серед алгоритмів підпису.

Таблиця 16.27.

Алгоритми ЕП PQC, які досліджуються

Алгоритм	Вирішувальна задача	Примітив PQC	Підтримуваний рівень безпеки (розмір відкритого ключа в байтах)				
			1	2	3	4	5
Crystals-Dilithium	Решітки	Алгоритм ЕП	1184	1472	1760	–	–
MQDSS	Багатовимірна криптографія	Алгоритм ЕП	–	62	–	88	–
qTESLA	Решітки	Алгоритм ЕП	1504 14880	–	3104 2976 39712	–	–

Таблиця 16.28.

Безпека в порівнянні з площею залежно від часу
алгоритмів підпису PQC, без оптимізації (тобто базового рівня)

Алгоритм	Рівень безпеки	Тригери	LUT	Тактовий цикл (нс)	Затримка
Алгоритми електронного підпису					
Crystals-Dilithium	1	25926	133461	10	609828
qTELSA	1	41978	232582	10	125374
MQDSS	2	35263	193320	15	49365597

Таблиця 16.29.

Співвідношення безпеки і площі та часу алгоритмів верифікації PQC без оптимізації (тобто базового рівня)

Алгоритм	Рівень безпеки	Тригери	LUT	Тактовий цикл (нс)	Затримка
Алгоритми електронного підпису					
Crystals-Dilithium	1	20865	108878	10	5380
qTELSA	1	29875	168570	10	71223
MQDSS	2	26423	147359	15	25124450

Серед алгоритмів підпису qTesla і Crystals-Dilithium мають затримку менше мільйона циклів. Crystals-Dilithium є найшвидшим серед алгоритмів підпису. Crystals-Dilithium (для створення підпису) є гарними кандидатами для пристроїв Інтернету речей. Жодна з верифікацій безпеки рівня 5 в цьому дослідженні не має низької затримки і, отже, не підходить для серверів. Всі алгоритми з низькою затримкою мають безпеку рівня 1.

У табл. 16.30 наведено та розглянуто критичні функції і цикли, які призводять до високої затримки.

Використовується розв'язка циклу для підпису і верифікації, результати наведені в табл. 16.31 і 16.32 відповідно. Серед алгоритмів з рівнем безпеки 1 Crystals-Dilithium – найшвидший алгоритм підпису.

Таблиця 16.30.

Критичні функції алгоритмів електронного підпису

Алгоритм	Критична функція	# Цикли
Алгоритми електронного підпису		
Crystals-Dilithium	*expand_mat*	2
MQDSS	*crypto_sign*	2
qTELSA	*sparse_mul16*	1

Crystals-Dilithium посідає найбільше місце серед алгоритмів підпису. Серед алгоритмів підпису PQC розгортання циклу не зменшує затримки MQDSS. Розгортання циклу зменшує затримку всіх алгоритмів підпису PQC. Однак це також призводить до збільшення площі.

Серед інших алгоритмів високої безпеки MQDSS (підпис, рівень безпеки 2) може використовуватися для пристроїв Інтернету речей, оскільки забезпечує відносно низьке апаратне навантаження.

Таблиця 16.31.

Співвідношення безпеки і площі порівняно із синхронізацією алгоритмів підпису PQC, після завершення циклу

Алгоритм	Рівень безпеки	Тригери	LUT	Тактовий цикл (нс)	Затримка
Алгоритми електронного підпису					
Crystals-Dilithium	1	158313	584742	10	18525
qTELSA	1	97235	328106	10	59854
MQDSS	2	45135	230273	15	49365597

Таблиця 16.32.

Залежність безпеки від площі залежно від часу алгоритмів верифікації PQC, після завершення циклу

Алгоритм	Рівень безпеки	Тригери	LUT	Тактовий цикл (нс)	Затримка
Алгоритми електронного підпису					
Crystals-Dilithium	1	108154	388991	10	5380
qTELSA	1	77567	247726	10	36423
MQDSS	2	93945	323734	15	25084906

Розгортання циклу Crystals-Dilithium забезпечує суттєве скорочення затримок. За винятком MQDSS, алгоритм не має затримки понад 1 мільйон циклів. Подібно підпису, розгортання циклу зменшує затримку верифікації PQC. Це поставляється з додатковим обладнанням. Для пристроїв Інтернету речей, де високий рівень безпеки не потрібен, можна використовувати алгоритм рівня безпеки 1 з низьким рівнем службових даних, наприклад, qTesla (підпис). З іншого боку, Crystals-Dilithium (підпис), що забезпечує безпеку рівня 1, може використовуватися в серверах через його низьку затримку.

Результати підпису з використанням конвеєрного циклу представлені в табл. 16.33.

Подібно до розгортання циклу, конвеєризація також зменшує загальну затримку для алгоритмів підпису PQC. Основна відмінність з таблицею 16.33 пов'язана з алгоритмом підпису MQDSS. Хоча при розгортанні циклу не вдалося змінити його затримку, конвеєрна обробка може скоротити затримку на 50 %. Конвеєрний цикл зменшує затримку алгоритмів підпису PQC, збільшуючи апаратну область. Поліпшення затримки порівняно з розгортанням циклу не погоджено. Жоден з алгоритмів підпису не забезпечує низьку затримку після конвеєрної обробки [304].

Таблиця 16.33.

Співвідношення безпеки і площі порівняно із синхронізацією алгоритмів ЕП PQC, після конвеєризації циклу

Алгоритм	Рівень безпеки	Тригери	LUT	Тактовий цикл (нс)	Затримка
Алгоритми електронного підпису					
Crystals-Dilithium	1	146076	1327355	10	155166
qTELSA	1	112657	346020	10	63736
MQDSS	2	47441	270713	15	25825918

3. Поглиблені дослідження з використанням трьох підписів PQC

У цьому підрозділі розглянуто три алгоритми PQC на основі підпису і проаналізовано апаратні реалізації на різних рівнях безпеки. Крім того, розглянуто вплив різних оптимізацій конструкції як для процедури підпису, так і для процедури верифікації. Всі досліджувані реалізації виконані на платі Xilinx Artix-7 FPGA. Це платформа, яку NIST розглядає в якості платформи порівняння [304,305].

3.1 Порівняння за рівнем безпеки 1

Порівнюється область і продуктивність трьох алгоритмів для рівня безпеки 1. Аналізовані параметри – кількість тригерів, кількість LUT, затримка і LAP. Порівняння частин підпису та верифікації наведено в табл. 16.34 і 16.35 [304 – 306].

У той час, як накладні витрати на розмірі більш-менш однакові (MQDSS займає на 83 % менше місця, ніж інші), загальна затримка і LAP для MQDSS надзвичайно високі. MQDSS займає в 71 раз більше затримки і в 61 раз більше LAP порівняно з Crystals-Dilithium. На рівні безпеки 1 Crystals-Dilithium є найкращим алгоритмом для "crypto_sign", оскільки займає найменший LAP.

Таблиця 16.34.

Аналіз апаратних реалізацій компонентів "crypto_sign" алгоритмів PQC на основі підпису

Алгоритм	FF	LUT	Такти (нс)	Затримка	LAP
qTesla	26299	126732	12.65	537092	6.8×10^{10}
Crystals-Dilithium	27132	123655	8.738	485963	6.0×10^{10}
MQDSS	21841	106035	17.05	34502428	3.6×10^{10}

Таблиця 16.35.

Аналіз апаратних реалізацій "crypto_sign_open"
компонентів алгоритмів PQC на основі підпису

Алгоритм	FF	LUT	Такти (нс)	Затримка	LAP
qTesla	17780	87067	12.58	80422	7.0×10^9
Crystals-Dilithium	14712	63863	8.738	149950	9.5×10^9
MQDSS	23072	117097	17.045	25686731	3.0×10^{12}

Crystals-Dilithium має найменший розмір, тоді як qTesla має найменшу затримку і показує кращу продуктивність. На рівні безпеки 1 qTesla є найкращим алгоритмом для "crypto_sign_open", оскільки займає найменший LAP [304].

3.2 Порівняння за рівнем безпеки 3

У цьому підрозділі порівнюється вартість і продуктивність впровадження апаратного забезпечення як для "crypto_sign", так і для "crypto_sign_open" для трьох алгоритмів на основі підпису на рівні безпеки 3. Результати для підпису та верифікації наведено у табл. 16.36 та 16.37 відповідно.

У той час, як накладні витрати на розмір більш-менш однакові для всіх алгоритмів, загальна затримка і LAP для MQDSS надзвичайно високі. На рівні безпеки 3 qTesla є найкращим алгоритмом для "crypto_sign", оскільки займає найменший LAP [304,305].

Crystals-Dilitium має найменший розмір, тоді як qTesla має найменшу затримку. На рівні безпеки 3 qTesla є найкращим алгоритмом для "crypto_sign_open", оскільки займає найменший LAP. З іншого боку, MQDSS має найгірші показники LAP [304].

Таблиця 16.36.

Аналіз апаратних реалізацій "crypto_sign"
компонентів алгоритмів PQC на основі підпису на рівні безпеки 3

Алгоритм	FF	LUT	Такти (нс)	Затримка	LAP
qTesla	26011	126311	12.65	347655	4.3×10^{10}
Crystals-Dilithium	27308	123933	8.738	826832	1.0×10^{11}
MQDSS	24748	123170	16.378	119353597	1.4×10^{13}

Таблиця 16.37.

Аналіз апаратних реалізацій "crypto_sign_open"
компонентів алгоритмів PQC на основі підпису

Алгоритм	FF	LUT	Такти (нс)	Затримка	LAP
qTesla	17754	86142	12.65	201027	1.7×10^{10}
Crystals-Dilithium	14783	63980	8.738	297592	1.9×10^{10}
MQDSS	23149	117574	17.045	87861777	1.0×10^{13}

3.3 Співставлення між різними рівнями безпеки

У цьому підрозділі порівнюються службові дані області та продуктивність алгоритмів на різних рівнях безпеки [304].

Порівнюється область (у термінах тригерів і LUT), продуктивність (у термінах затримки) і LAP для реалізації "crypto_sign" частини трьох алгоритмів для рівнів безпеки 1 і 3. Результати показано на рис. 16.4. Для qTesla і Crystals-Dilithium площа накладних витрат на рівнях безпеки 1 і 3 аналогічна. Для MQDSS, накладні витрати на рівень безпеки 3 на 13 – 16 % більше, ніж на рівень безпеки 1. Для qTesla затримка знижується на рівні безпеки 3 порівняно з рівнем 1. Для двох інших алгоритмів затримка різко збільшується при більш високому рівні безпеки. Порівнюючи як площу, так і продуктивність, можна помітити, що тільки qTesla має більш низькі LAP на рівні безпеки 3 порівняно з рівнем 1. qTesla на рівні безпеки 3 має найнижчий LAP і забезпечує найвищу безпеку серед усіх альтернатив, що обговорюються в цьому підрозділі.

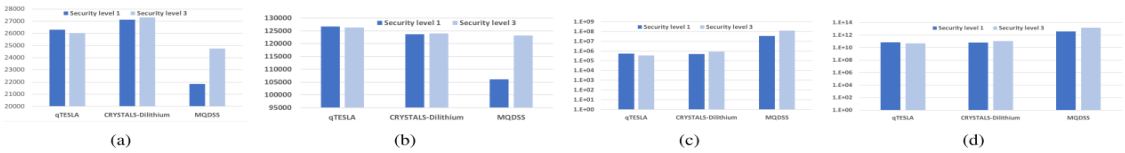


Рис. 16.4. Порівняння (*a*) тригерів, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на різних рівнях безпеки

Для qTesla і Crystals-Dilithium різниця в кількості тригерів не є істотною, в той час як для MQDSS кількість тригерів для реалізації на рівні безпеки 3 збільшується на 13% порівняно з рівнем безпеки 1. Для qTesla і Crystals-Dilithium різниця в кількості LUT не є істотною, в той час як для MQDSS кількість тригерів для реалізації на рівні безпеки 3 збільшується на 16 % порівняно з рівнем безпеки 1. Оскільки затримка MQDSS на два порядки більша, ніж qTesla або Crystals-Dilithium, цей графік зображено в логарифмічній шкалі. qTesla на рівні безпеки 3 і Crystals-Dilithium на рівні безпеки 1 мають найменшу затримку. qTesla на рівні безпеки 3 має найкращий LAP, а MQDSS на обох рівнях безпеки має найгірший LAP.

Далі порівнюється область, продуктивність і LAP для верифікації підпису, тобто, компонент "crypto_sign_open" трьох алгоритмів на двох рівнях безпеки. Результати показано на рис. 16.5. Для всіх алгоритмів службові дані області залишаються однаковими для двох рівнів безпеки. З іншого боку, послідовно затримка і LAP вища на рівні безпеки 3 порівняно з рівнем безпеки 1. qTesla має найменший LAP на обох рівнях безпеки.

У той час, як на рис. 16.5 Crystals-Dilithium займає найбільшу кількість тригерів, при реалізації "crypto_sign_open" потрібна найменша кількість тригерів. Порівняно з рис. 16.4, різниця в кількості LUT не є значною для жодного алгоритму. qTesla на рівні безпеки 1 вимагає найменшої затримки. MQDSS вимагає найбільших затримок для "crypto_sign" і "crypto_sign_open."

qTesla на рівні безпеки 1 має найкращий LAP, а MQDSS на обох рівнях безпеки має найгірший LAP [304].

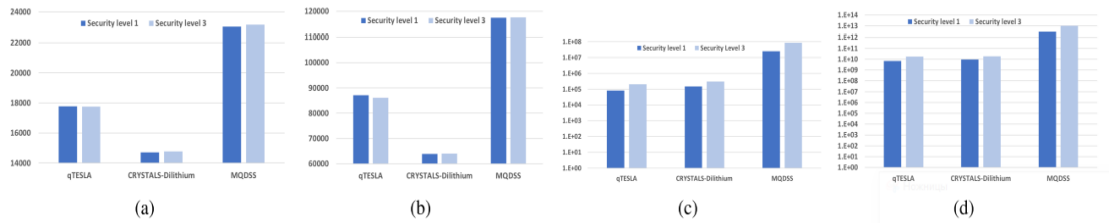


Рис. 16.5. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на різних рівнях безпеки

3.4 Оптимізація

У цьому підрозділі проаналізовано, як різні методи оптимізації (розбиття циклу і конвеєрування циклу) допомагають зменшити загальну затримку алгоритмів PQC. Результати реалізації "crypto_sign" і "crypto_sign_open" показані на рис. 16.6 і 16.7 відповідно.

Конвеєрний цикл дає найменшу кількість тригерів по всьому циклу, в той час як розгін циклу вимагає максимальної кількості тригерів.

Конвеєрний цикл дає найменшу кількість LUT для всіх конструкцій.

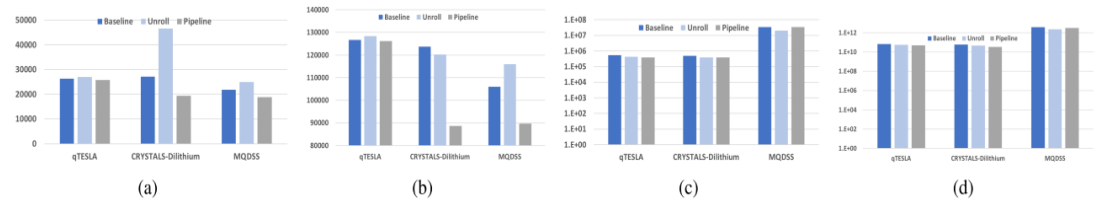


Рис. 16.6. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних оптимізацій

За винятком Crystals-Dilithium, кількість LUT збільшується з розмотуванням циклів. Розгортання циклу і конвеєрна обробка зменшують затримку порівняно з базовою лінією. Crystals-Dilithium і qTesla мають найменше значення LAP при конвеєризації циклів. Для MQDSS найменше значення LAP виходить, коли цикли не розгорнуті [304,305].

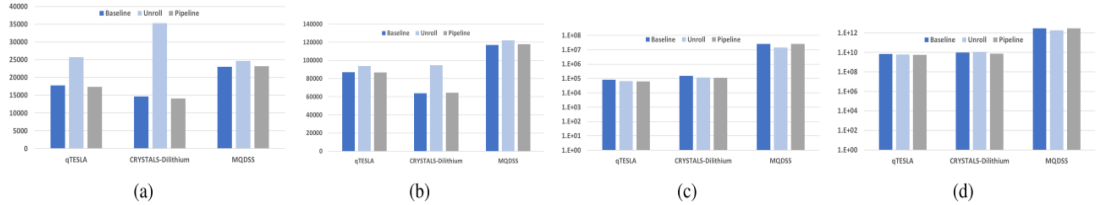


Рис. 16.7. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на рівні безпеки 1 для різних оптимізацій

Оптимізація з конвеєрними циклами займає найменшу кількість тригерів і LUT у всіх конструкціях, в той час як розукрупнення циклу вимагає максимальної кількості тригерів і LUT. Оптимізація за допомогою розмотування циклів і конвеєрів дозволяє скоротити затримки порівняно з базовою реалізацією. За винятком MQDSS, два інших алгоритми мають найменше значення LAP при виконанні конвеєризації циклу. Для MQDSS найменше значення LAP виходить, коли цикли не розгорнуті [304].

3.5 Наслідки апаратно оптимізованого Keccak

Keccak – сімейство sponge-функцій, що використовують три алгоритми другого раунду NIST PQC. Протягом багатьох років дослідники розробляли апаратно оптимізовані варіанти Keccak. У цьому підрозділі розглянемо вплив одного з них і спостерігаємо, як змінюється розмір і накладні витрати на продуктивність. Використовується реалізація Keccak [305]. Автори [305] розвивали впровадження функції *KeccakF1600_StatePermute* з окремими функціями, що використовуються для внутрішніх операцій як θ, ρ, π, χ та ι. На рис. 16.8 порівнюються параметри для реалізації "crypto_sign", а на рис. 16.9 порівнюється "crypto_sign_open". Для забезпечення узгодженості всі реалізації належать до рівня безпеки 1.

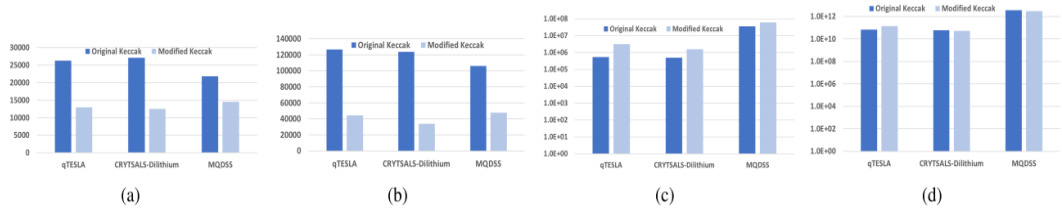


Рис. 16.8. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для двох версій Keccak

Кількість тригерів і LUT значно зменшується порівняно з початковим Keccak. Crystals-Dilithium має 54 % зниження кількості тригерів і 73% зниження LUT. З іншого боку, затримка збільшується з модифікованим Keccak. Найсильніше постраждав qTesla, який має збільшення затримки в 5,8 разів, в той час як MQDSS несе тільки збільшення затримки в 1,7 рази. З іншого боку, значення LAP зменшується для MQDSS і Crystals-Dilithium і збільшується тільки для qTesla. Для MQDSS LAP скорочується на 21 % [305].

Як і на рис. 16.7, на рис. 16.8 кількість тригерів і LUT значно зменшується порівняно з вихідним Keccak. MQDSS має максимальне 56 % зниження кількості тригерів і Crystals-Dilithium має близько 96 % зниження LUT. Затримка збільшується з модифікованим Keccak. Найсильніше це стосується qTesla, яка має збільшення затримки в 7 разів, в той час як MQDSS має збільшення затримки в 1,7 рази. LAP значно зменшується для всіх алгоритмів, крім qTesla. Crystals-Dilithium має 82 % зниження LAP порівняно з вихідним Keccak.

Кількість тригерів і LUT значно зменшується порівняно з оригінальною версією Keccak. qTesla має приблизно 48 % зниження кількості тригерів і 64% скорочення кількості LUT. З іншого боку, затримка збільшується з реалізацією

модифікованого Keccak. Найсильніше вражає qTesla, яка має збільшення затримки в 7,15 разів. На відміну від рис. 16.7, значення LAP послідовно збільшуються для всіх трьох алгоритмів. Для MQDSS збільшення LAP мінімально – 20 % [305].

На рис. 16.10, 16.11 порівнюються накладні витрати з розмотуванням циклів. Для чесного порівняння використовуються одні й ті ж директиви для обох реалізацій. Аналогічне порівняння для конвеєризації циклу показано на рис. 16.12, 16.13 [305].

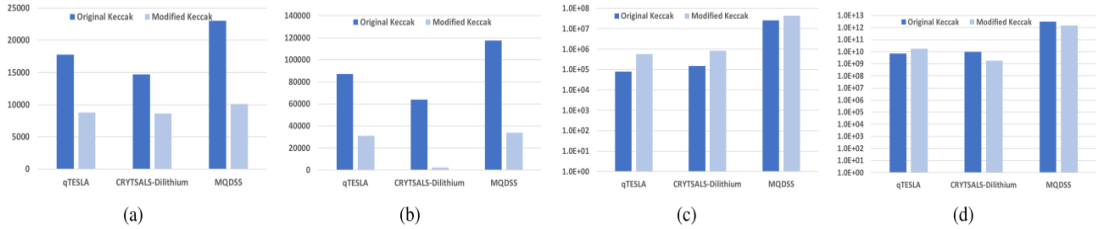


Рис. 16.9. Порівняння (*a*) тригерів, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів на рівні безпеки 1 для різних версій Keccak

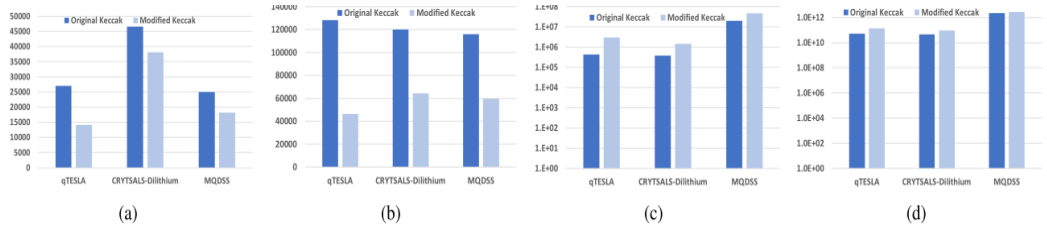


Рис. 16.10. Порівняння (*a*) тригери, (*b*) LUT, (*c*) Latency і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних версій Keccak після завершення циклу

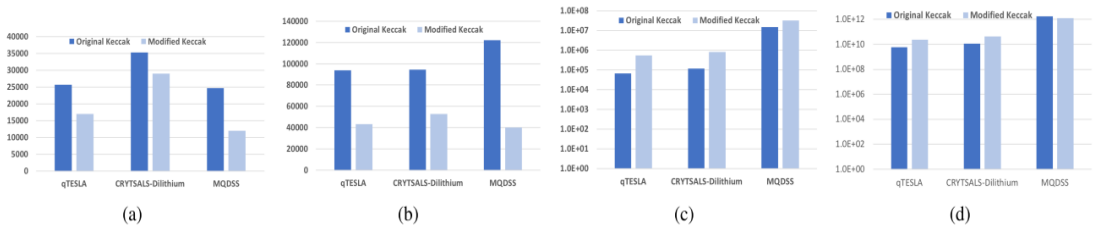


Рис. 16.11. Порівняння (*a*) тригери, (*b*) LUT, (*c*) LAP і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на рівні безпеки 1 для різних Keccak після розгортання циклу

Кількість тригерів і LUT значно зменшується порівняно з початковим Keccak. MQDSS має приблизно 52 % зниження кількості тригерів і 61 % зменшення кількості LUT. З іншого боку, затримка збільшується з модифікованим Keccak. Найсильніше постраждав qTesla, який має збільшення в 8,4 рази. LAP збільшується для qTesla і Crystals-Dilithium, але знижується на 28 % для MQDSS.

Скорочення кількості тригерів і LUT значне порівняно з оригінальним Keccak. qTesla має 50 % зниження кількості тригерів і 65 % скорочення кількості LUT. З іншого боку, затримка збільшується з реалізацією модифікованого Keccak. Найсильніше це торкнулося qTesla, що має збільшення затримки в 7,7 разів. Значення LAP збільшуються для qTesla і Crystals-Dilithium, але зменшуються на 5 % для MQDSS. В цілому, відносне збільшення LAP менше, ніж при розгортанні циклу на рис. 16.9.

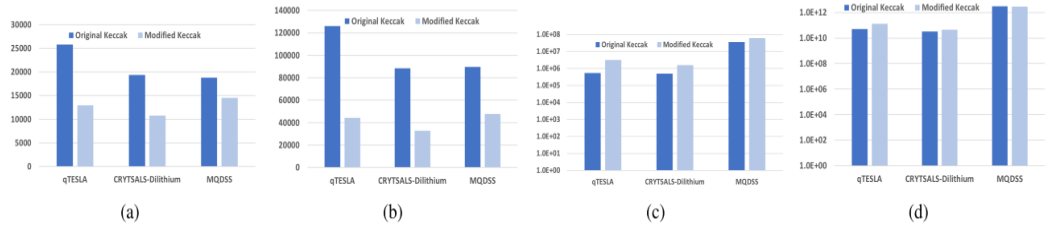


Рис. 16.12. Порівняння (*a*) тригери, (*b*) LUT, (*c*) затримка і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних версій Keccak після конвеєризації циклу

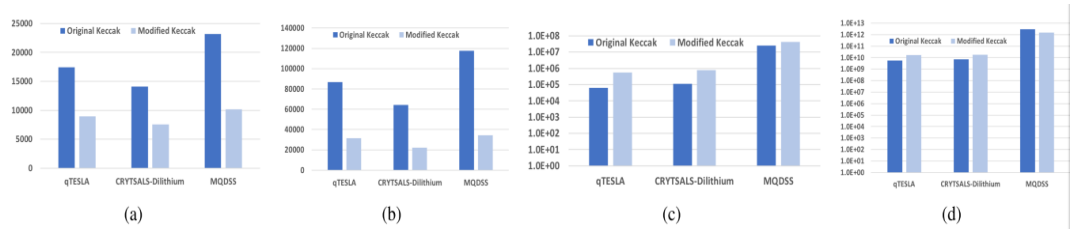


Рис. 16.13. Порівняння (*a*) тригери, (*b*) LUT, (*c*) затримка і (*d*) LAP для реалізації "crypto_sign_open" трьох алгоритмів на рівні безпеки 1 з використанням різних версій Keccak після конвеєризації циклу

Кількість тригерів і LUT значно зменшується порівняно з базовим Keccak. MQDSS має 56 % зниження кількості тригерів і 70 % скорочення кількості LUT. Хоча затримка MQDSS збільшується на 67 % при модифікованому Keccak, загальне значення LAP зменшується на 52 % [304].

Висновки

1. Наразі однією із важливих проблем сучасної криптографії є створення стандартів асиметричних криптоперетворень ЕП, які були б безпечними у постквантовий період. Вирішення цієї проблеми здійснюється в процесі міжнародного конкурсу NIST, завданням якого є розробка такого ЕП, який би був стійким як до квантових, так і до класичних атак.

2. Були проведені та проаналізовані специфічні тематичні дослідження з використанням трьох алгоритмів підпису – qTesla, Crystals-Dilithium і MQDSS. Відзначено розмір і продуктивність накладних витрат, коли апаратні реалізації цих алгоритмів виконуються для різних рівнів безпеки, а також для різних оптимізацій.

3. Алгоритми ЕП qTesla і Crystals-Dilithium мали найменший LAP при конвеєрному циклі, в той час як MQDSS мав найменший LAP при розгортанні циклу. Майбутні дослідження визначать, яка оптимізація підходить для якого алгоритму.

4. Навіть, якщо одні й ті самі директиви оптимізації застосовуються до двох різних версій Keccak, модифікована функція Keccak зменшує загальні витрати області. Однак у цьому випадку LAP знижуються не так сильно (фактично LAP збільшується в більшості випадків) порівняно з базовою лінією (без оптимізації). Зменшення LAP більше для конвеєрного циклу порівняно з розмотуванням циклу.

5. У цьому дослідженні було використано одну і ту ж плату (Virtex або Artix) для реалізації варіантів алгоритмів PQC як з маленьким розміром, так і з низькою затримкою. У майбутньому буде вивчено, які архітектури FPGA ідеально підходять для реалізації з маленьким розміром (низькою затримкою).

16.7 Моделі та методи створення криптосистем, заснованих на використанні ізогеній еліптичних кривих

Вступ

З появою квантового комп'ютера криптостійкість існуючих криптографічних систем значно зменшиться, бо існуючі системи для методів квантового криптоаналізу мають лише поліноміальну складність. Тому виникає необхідність створення систем на основі криптографічних перетворень, що є стійкими до квантового криптоаналізу, тобто, будуть мати експоненційну чи субекспоненційну складність. Одним з таких методів є використання ізогеній. Ізогенія – це відображення точок однієї кривої в іншу. Для криптосистем з відкритим ключем планується, що сама ізогенія – це особистий ключ, а крива, до якої вона веде – відкритий. Тому стійкість залежить від складності розрахунку ізогенії між двома відомими кривими. Ця складність має експоненційний характер.

1. Загальні відомості про ізогенії

Ізогенія – це раціональне відображення $\varphi: E_1(K) \to E_2(K)$, де $E_1(K)$ та $E_2(K)$ є еліптичними кривими, а $\varphi(P_\infty) = P_\infty$ [312,313]. Нульова ізогенія – це ізогенія, що відображає усі точки однієї кривої у точку на нескінченності іншої. Ядром ізогенії є $Ker\,(\varphi) = \{K_i \in E_1\};\ \varphi\,(K_i) = P_\infty$. Ізогенії ініціюють відображення полів функцій на кривих. Степінь розширення $\left(K(E_1): \varphi * K(E_2)\right)$ називається степеню ізогенії. Для ізогенії $\varphi: E_1(K) \to E_2(K)$ існує дуальна ізогенія $\hat{\varphi}: E_2(K) \to E_1(K)$, така, що $\hat{\varphi}\varphi = [l]$, де l – множення точки кривої E_1 на число l, аналогічно $\varphi\hat{\varphi} = [l]$, де l – множення точки кривої E_2 на число l. Дуальні ізогенії мають однакову степінь.

У випадку алгебраїчно замкненого поля операція множення точки на число l задає ендоморфізм еліптичної кривої з ядром l^2 точок. Оскільки ізогенія відповідає квадратному кореню з операції множення на l, то ядро ізогенії

складається з l точок порядку l , що створюють циклічну групу (однією з них є точка P_∞).

Ізогенії складних степенів можуть використовуватися, як композиція ізогеній простих степеней.

Властивості ізогеній, що використовуються при створенні криптосистем, наступні:

1) $\gamma\left(\varphi\left(A\right)\right) = \varphi\left(\gamma\left(A\right)\right) = \gamma\varphi(A)$;

2) $k * A_\varphi = \varphi\left(k * A\right)$.

2. Знаходження ізогеній по ядру

Для знаходження ізогенії $\varphi: E_1(K) \to E_2(K)$ з заданим ядром використовується формула Велу [314,315].

Для еліптичної кривої, що задана формулою Вейєрштрасса, ізогенія має наступний вигляд:

$$E_1 : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 .$$

Нехай C – группа точок еліптичної кривої, що буде ядром ізогенії. Приведемо алгоритм знаходження ізогенії за її ядром:

1) Формуємо набір точок S:

а) Виключаємо з С точку на нескінченності.

б) Нехай C_2 – усі точки C, в яких координата y дорівнює нулю, а R – усі інші точки C.

в) Розділимо точки набору R на R_+ та R_- таким чином, що для кожної точки P, що належить R_+ , $-P$ належить R_-.

г) $S = R_+ \cup C_2$.

2) Для кожної точки $Q \in S$ виконуємо наступні обчислення:

а) $g_Q^x = 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q$;

б) $g_Q^y = -2y_Q - a_1x_Q - a_3$;

в) $v_Q = \begin{cases} g_Q^x, & 2Q = \infty \\ 2g_Q^x - a_1g_Q^y, & 2Q \neq \infty \end{cases}$;

г) $u_Q = \left(g_Q^y\right)^2$;

ґ) $v = \sum_{Q \in S} v_Q$;

д) $w = \sum_{Q \in S} \left(u_Q + x_Qv_Q\right)$.

3) Розраховуємо формулу еліптичної кривої $E_2(K)$:

а) $A_1 = a_1$;

б) $A_2 = a_2$;

в) $A_3 = a_3$;

г) $A_4 = a_4 - 5v$;

г) $A_6 = a_6 - (a_1^2 + 4a_2)v - 7w$;

д) $E_2: y^2 + A_1xy + A_3y = x^2 + A_2x^2 + A_4x + A_6$.

4) Розраховуємо координати точки $(x_\varphi, y_\varphi) = \varphi(x, y)$:

а) $x_\varphi = x + \sum_{Q \in S}\left(\frac{v_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2}\right)$;

б) $y_\varphi = y - \sum_{Q \in S}\left(u_Q \frac{2y + a_1x + a_3}{(x - x_Q)^3} + v_Q \frac{a_1(x - x_Q) + y - y_Q}{(x - x_Q)^2} + \frac{a_1u_Q - g_Q^x g_Q^y}{(x - x_Q)^2}\right)$.

Отримані формули на 4-му кроці й будуть ізогенією із заданим ядром.

Оскільки в криптографії еліптичних кривих прийнято використовувати спрощену формулу Вейєрштрасса, то й ми приведемо її до прийнятого вигляду.

$$E_1 : y^2 = x^3 + a_4x + a_6 .$$

Нехай *C* – группа точок еліптичної кривої, що буде ядром ізогенії. Тоді:

1) Формуємо набір точок *S*:

а) Виключаємо з C точку на нескінченності.

б) Нехай C_2 – усі точки *C*, в яких координата *y* дорівнює нулю, а *R* – усі інші точки *C*.

в) Розділимо точки набору *R* на R_+ та R_- таким чином, що для кожної точки *P*, що належить R_+ , *-P* належить R_- .

г) $S = R_+ \cup C_2$.

2) Для кожної точки $Q \in S$ виконуємо наступні обчислення:

а) $g_Q^x = 3x_Q^2 + a_4$;

б) $g_Q^y = -2y_Q$;

в) $v_Q = \begin{cases} g_Q^x, 2Q = \infty \\ 2g_Q^x, 2Q \neq \infty \end{cases}$;

г) $u_Q = \left(g_Q^y\right)^2$;

ґ) $v = \sum_{Q \in S} v_Q$;

д) $w = \sum_{Q \in S} \left(u_Q + x_Q v_Q\right)$.

3) Розраховуємо формулу еліптичної кривої $E_2(K)$:

а) $A_4 = a_4 - 5v$;

б) $A_6 = a_6 - 7w$;

в) $E_2: y^2 = x^2 + A_4x + A_6$.

4) Розраховуємо координати точки $(x_\varphi, y_\varphi) = \varphi(x, y)$:

а) $x_\varphi = x + \sum_{Q \in S}\left(\frac{v_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2}\right)$;

б) $y_\varphi = y - \sum_{Q \in S}\left(u_Q \frac{2y}{(x - x_Q)^3} + v_Q \frac{y - y_Q}{(x - x_Q)^2} - \frac{g_Q^x g_Q^y}{(x - x_Q)^2}\right)$.

3 Вибір еліптичних кривих для криптографічних систем на основі ізогеній

Складність знаходження ізогенії за її ядром, використовуючи приведений алгоритм, становить $O(l^3)$ [314,315], де l – порядок ізогенії. Щоб зменшити складність обчислення, пропонується використання суперсингулярних кривих, що дозволить розбивати ізогенію на композицію ізогеній малого порядка. Наприклад, обчислення ізогенії порядка 2^{64} має складність 2^{192}, а якщо її розбити на композицію ізогеній порядка 2, складність обчислення фінальної ізогенії становить $64 * 2^3 = 2^9$. Оскільки наші криптосистеми будуть засновані на використанні ізогеній, то вразливість суперсингулярних кривих до вирішення задачі знаходження дискретного логарифму не впливає на стійкість криптографічної системи на основі ізогеній при використанні таких кривих.

За теоремою Сільвермана [313,314] кількість суперсингулярних еліптичних кривих над полями характеристики p становить $\frac{p+1}{12} + 1$. Рекомендується використовувати квадратичні розширення полів F_{p^2}, що дозволить досить легко за алгоритмом Брокера будувати суперсингулярні криві [312].

4. Проблема створення криптосистем з відкритим ключем на ізогеніях еліптичних кривих (ЕК) та складність криптоаналізу

Планується, що в криптосистемах на ізогеніях еліптичних кривих при спільному параметрі ЕК E_1 таємним ключем буде ізогенія $\varphi: E_1(K) \to E_2(K)$, а ЕК E_2 – відкритим ключем. На рис. 16.14 зображена спрощена схема криптоперетворень на ізогеніях ЕК [314].

На цій схемі E_0, E_φ, E_ψ, $E_{\varphi\psi}$ – еліптичні криві, φ , ψ, φ' , ψ' – ізогенії, причому $Ker(\varphi') = \psi\big(Ker(\varphi)\big)$ та $Ker(\psi') = \varphi\big(Ker(\psi)\big)$. Проблема використання ізогеній в криптосистемах з відкритим ключем полягає в тому, що якщо криптоаналітику відома пара (ψ, φ') чи (φ', ψ'), він може знайти φ – особистий ключ [316].

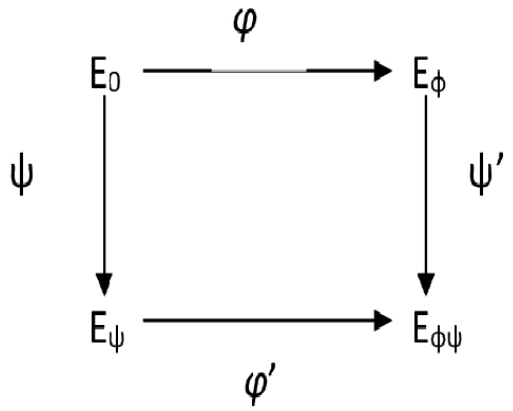


Рис. 16.14. Спрощена схема криптоперетворень на ізогеніях ЕК

Це витікає з того, що пошук зворотної ізогенії має низьку складність. Для цього на еліптичній кривій обирається точка, що не належить ядру ізогенії та має

порядок, який дорівнює порядку ізогенії. Ця точка відображається через ізогенію і з її відображення будується нова ізогенія, що веде до еліптичної кривої, яка ізоморфна до початкової. Тоді обчислення відповідного ізоморфізму є простою задачею. Далі наведемо алгоритм відновлення секретної ізогенії через знання її відображення через відкриту ізогенію.

1) Нехай криптоаналітику відома пара (ψ, φ'). Тоді він будує зворотню ізогенію $\hat{\psi}$, тоді $Ker(\varphi) = \hat{\psi}\big(Ker(\varphi')\big)$.

2) Нехай криптоаналітику відома пара (φ', ψ'). Тоді він будує зворотні ізогенії ($\widehat{\varphi'}, \widehat{\psi'}$), далі знаходить $Ker\big(\hat{\psi}\big) = \widehat{\varphi'}\left(Ker\big(\widehat{\psi'}\big)\right)$, після цього вже можна знайти $Ker(\varphi) = \hat{\psi}\big(Ker(\varphi')\big)$.

Звідси випливає, що якщо криптоаналітику стають відомі сусідні ребра ромбічної структури графів ізогеній, то, незалежно від їх направленості, він може відновити усю схему криптосистеми і встановити невідомі ізогенії. Тому в криптосистемі повинно зберігатися в таємниці не лише секретна ізогенія, а й протилежна їй у графі ізогеній. Саме це є однією з основних проблем при побудові схем електронного підпису (ЕП) на ізогеніях еліптичних кривих.

Атака на ізогенії полягає в пошуку ізогенії між двома відомими ЕК. Для цього від кожної еліптичної кривої будується дерево ізогеній, доки вийде співпадання в спільній вершині ЕК. Тобто, примусове створення другого випадку наведеної вище ситуації, де E_φ, E_ψ – наші початкові еліптичні криві, а $E_{\varphi\psi}$ – спільна вершина. Ця атака називається «зустріч посередині». Побудову ізогеній відразу з двох вершин створюють для того, щоб зменшити складність, бо складність побудови двох ізогеній порядка l/2 менша складності побудови ізогенії порядку l в 4 рази. Складність атаки «зустріч посередині» [314,316] становить $O(p^{1/4})$ для класичного комп'ютера і $O(p^{1/6})$ для квантового, що має експоненційний характер.

5. Ізогенії на кривих Едвардса

Останнім часом набуває популярності дослідження на тему використання еліптичних кривих Едвардса у криптографії. Перевагою таких ЕК вважається значне збільшення швидкодії. Тому необхідно розглянути можливість використання ізогеній для кривих Едвардса.

Криві Едвардса мають наступний вигляд:

$$E_d: x^2 + y^2 = 1 + dx^2y^2,$$

$$E_{a,d}: ax^2 + y^2 = 1 + dx^2y^2,$$

де $a, d \neq \{0, 1\}$. Рівняння суми точок наступне:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right).$$

Позначимо через ϵ відображення кривих Едвардса на криві Вейєрштрасса $E: y^2 = x^3 + 2(a + d)x^2 + (a - d)^2x$:

$$\epsilon\colon (x,y) \to \left((a-d)\frac{1+y}{1-y}, (a-d)\frac{2(1+y)}{x(1-y)} \right),$$

та зворотнє:

$$\epsilon^{-1}\colon (x,y) \to \left(\frac{2x}{y}, \frac{x-(a-d)}{x+(a-d)} \right).$$

Для побудови ізогеній розглянемо два випадки:

1) Побудова ізогеній порядка 2 на кривих Едвардса.

Нехай є - крива Едвардса E_d та три елемента поля u, o, i такі, що $u^2 = 1-d$, $o^2 = d$, $i^2 = -1$, тоді існує три ізогенії порядка 2 [317]:

$$\omega_1(x,y) \to \left((u \mp 1)xy, \frac{(u \mp 1)y^2 \pm 1}{(u \pm 1)y^2 \mp 1} \right),$$

$$\omega_2(x,y) \to \left((iu \pm o)\frac{x}{y}, -\frac{oy^2 \mp iu - o}{oy^2 \pm iu - o} \right),$$

$$\omega_3(x,y) \to \left(i(o \mp 1)\frac{x(1-dy^2)}{y(1-d)}, \frac{(d \mp o)(oy^2 \pm 1)}{(d \pm o)(oy^2 \mp 1)} \right),$$

де $\omega_1\colon E_d \to E_{d'}$, $\omega_2\colon E_d \to E_{d''}$, $\omega_3\colon E_d \to E_{d'''}$,

для яких $d' = \left(\frac{u \pm 1}{u \mp 1}\right)^2$, $d'' = \left(\frac{iu \mp o}{iu \pm o}\right)^2$, $d''' = \left(\frac{o \pm 1}{o \mp 1}\right)^2$.

2) Побудова ізогеній непарного порядка на кривих Едвардса.

Нехай F - підгрупа кривої Едвардса $E_{a,d}$ порядка $l = 2s+1$, точками якого є $\{(0,1), (\pm\alpha_1, \beta_1), \dots, (\pm\alpha_s, \beta_s)\}$. Далі за цим ядром обчислюються наступні значення [317]:

1) $A = \prod_{i=1}^{s} \alpha_i$;

2) $B = \prod_{i=1}^{s} \beta_i$;

3) $a' = \frac{A^4}{B^4} a^l$;

4) $d' = A^4 B^4 d^l$.

Тоді ізогенія $\omega\colon E_{a,d} \to E_{a',d'}$ має вигляд:

$$\omega(x,y) = \left(-1^s \frac{x}{A^2} \prod_{i=1}^{s} \frac{\beta_i^2 x^2 - \alpha_i^2 y^2}{1 - d^2 \alpha_i^2 \beta_i^2 x^2 y^2}, \frac{y}{B^2} \prod_{i=1}^{s} \frac{\beta_i^2 y^2 - \alpha_i^2 x^2}{1 - d^2 \alpha_i^2 \beta_i^2 x^2 y^2} \right).$$

Обчислення ізогенії для кривих Едвардса виконується в 4 рази швидше, ніж для звичайних кривих, тому використання кривих Едвардса має потенціал для подальшої оптимізації обчислення ізогеній.

Але вимагається дослідження використання кривих Едвардса для криптосистем на основі ізогеній на предмет появи вразливостей. Крім того, якщо збільшується швидкість звичайного обчислення, то і збільшуються швидкі

виконання атаки «зустріч посередині». До того ж невідомо, чи буде присутній весь обсяг ізогеній еліптичних кривих на їх відображенні у криві Едвардса, бо якщо кількість можливих ізогеній буде інша, то це, можливо, буде говорити про збільшення чи зменшення криптостійкості.

6. Схема розподілу ключів Діффі-Хеллмана на ізогеніях еліптичних кривих

Загальносистемні параметри [314,316]:

Еліптична крива $E_0(F_{p^2})$.

Секретні параметри:

Елементи поля m, n та ізогенія φ.

Відкриті параметри:

Точки $P, Q \in E_0$ та крива E_φ .

Алгоритм:

1) Абонент A формує параметри P_A, Q_A, m_A, n_A. Обчислює точку:

$G_A = m_A P_A + n_A Q_A$,

де G_A є точкою, що породжує групу точок - ядро ізогенії φ_A. За формулою Велу будується ізогенія $\varphi_A: E_0 \rightarrow E_A$.

E_A – відкритий ключ.

2) Абонент B робить аналогічні кроки для параметрів P_B, Q_B, m_B, n_B, G_B, φ_B, E_B .

3) Абоненти A і B обмінюються відкритими параметрами.

4) Абонент A обчислює та надсилає $\varphi_A(P_B)$, $\varphi_A(Q_B)$.

5) Абонент B обчислює та надсилає $\varphi_B(P_A)$, $\varphi_B(Q_A)$.

6) Абонент A обчислює точку:

$$G_{AB} = m_A \varphi_B(P_A) + n_A \varphi_B(Q_A),$$

де G_{AB} є точкою, що породжує групу точок – ядро ізогенії φ_{AB}. За формулою Велу будується ізогенія $\varphi_{AB}: E_B \rightarrow E_{AB}$.

E_{AB} – розподілена таємниця.

7) Абонент B обчислює точку:

$$G_{BA} = m_B \varphi_A(P_B) + n_B \varphi_A(Q_B),$$

де G_{BA} є точкою, що породжує групу точок – ядро ізогенії φ_{BA}. За формулою Велу будується ізогенія $\varphi_{BA}: E_A \rightarrow E_{BA}$.

E_{BA} – розподілена таємниця.

Загальна схема розподілу ключів зображена на рис. 16.15.

$$E_{AB} = E_{BA}.$$

Так як кожному з абонентів відомі лише протилежні ізогенії ромбічного графу, то вимога секретності підрозділу 4 виконується.

7. Схема електронного підпису на ізогеніях еліптичних кривих

Схема підпису, що описується в даному розділі, може бути перевірена будь-якою стороною, але для запобігання вразливості, що описана вище,

вимагається участь підписувача в інтерактивному режимі. Для того, щоб відрізнити неправильний підпис, механізм має можливість підписувачу в інтерактивному режимі довести підробку підпису. На відміну від спрощеної схеми, схема підпису має три проходи – додається ще ізогенія – ключ сесії. Тому зовнішній вигляд графу має форму куба.

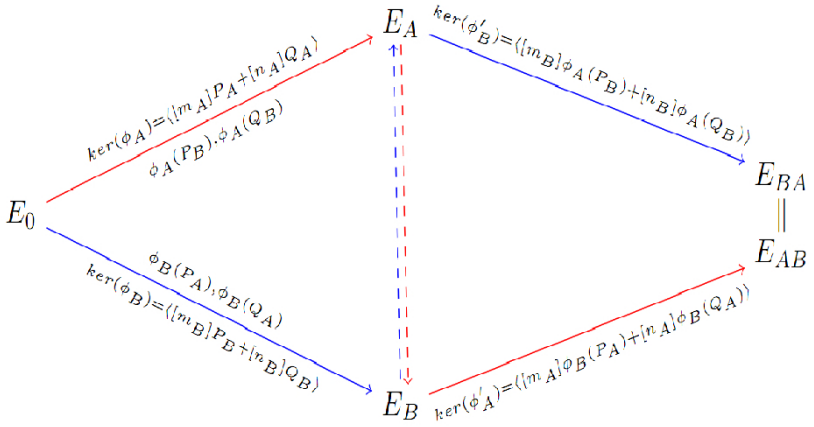


Рис. 16.15. Схема протоколу Діффі-Хеллмана на ізогеніях

Обирається ініціалізуюча еліптична крива $E(F_{p^2})$. На цій кривій обираються пари точок (P_A, Q_A), (P_M, Q_M), (P_C, Q_C) [316]. Точки (P_A, Q_A) будуть використовуватися для генерації ключа підписувача. Точки (P_M, Q_M) для генерації ізогенії, яка буде прив'язана до повідомлення, що підписується. Точки (P_C, Q_C) будуть використовуватися для генерації сесійного ключа для протоколів підтвердження чи спростування підпису.

Підписувач випадково генерує елементи поля m_A, n_A. Отримує ключову точку $K_A = [m_A]P_A + [n_A]Q_A$, точка K_A – точка генерування $Ker(\varphi)$, де ізогенія $\varphi: E \to E_A$ буде секретним ключем підписувача.

Відкриті параметри системи: p, $E(F_{p^2})$, (P_A, Q_A), (P_M, Q_M), (P_C, Q_C) і геш-функція, що використовується.

Відкритий ключ підписувача: E_A, $\varphi(P_C)$, $\varphi(Q_C)$.

Особистий ключ підписувача: m_A, n_A.

Алгоритм підпису наступний:

1) Підписувач отримує геш-значення h повідомлення M. Формує точку $K_M = P_M + [h]Q_M$ [316], що буде точкою генерування $Ker(\gamma)$ для ізогенії $\gamma: E \to E_M$.

2) Формує граф підпису, що представлений на рисунку 16.16, для цього виконує розрахунок ізогеній:

1) $\gamma: E \to E_M$;

2) $\gamma_A: E_A \to E_{AM}$, $Ker(\gamma_A) = \varphi\big(Ker(\gamma)\big)$;

3) $\varphi_M: E_M \to E_{AM}$, $Ker(\varphi_M) = \gamma\big(Ker(\varphi)\big)$.

3) Обчислює значення точок ($\varphi_M\gamma(P_C)$, $\varphi_M\gamma(Q_C)$). Ці дві точки, а також значення кривої E_{AM} і буде підписом повідомлення M на ізогенії φ.

Під час виконання протоколу підтвердження підпису нам необхідно довести правильність кривої E_{AM} без передачі ізогенії, що використовувалася для її створення [316]. Для цього генерується засліплююча ізогенія ε, яка використовується для побудови графа підтвердження підпису, що представлений на рисунку 16.17, і яка передається за спеціальним протоколом підписувачем перевірнику. Ця інтерактивність є основним недоліком використання схеми електронного підпису на ізогеніях ЕК. Єдина система, в якій можливо не звертати увагу на цей недолік, це хмара.

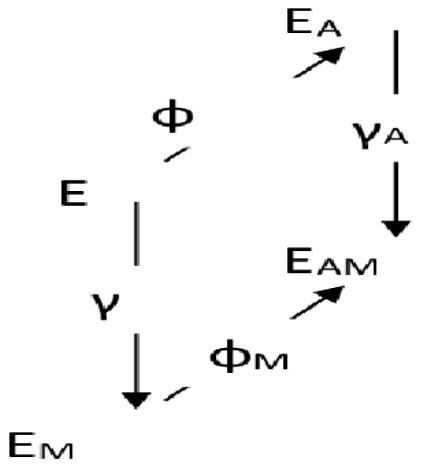


Рис. 16.16 Граф підпису

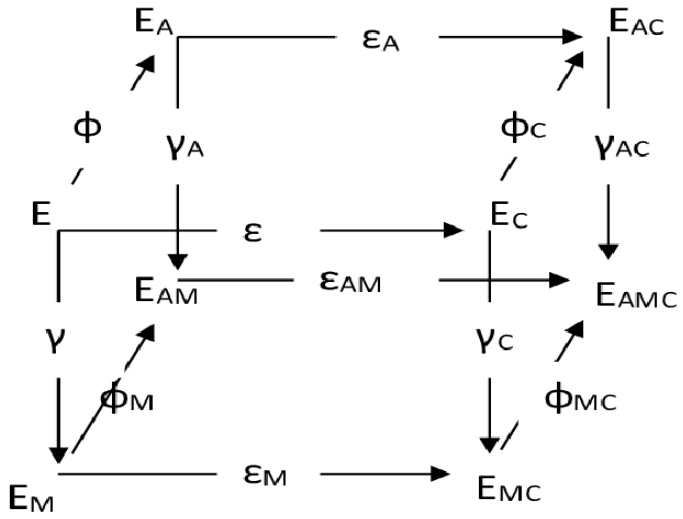


Рис. 16.17. Граф підтвердження підпису

Висновки

1. Використання ізогеній є одним з шляхів забезпечення стійкості до квантового криптоаналізу. Вже існує декілька схем узгодження ключів та електронного підпису на ізогеніях еліптичних кривих.

2. Складність обчислення ізогенії ЕК становить $O(l^3)$, для її зменшення пропонується використання суперсингулярних кривих. Тоді можливо розбити ізогенію великого порядку на композицію ізогеній низького порядку.

3. Додатковим варіантом оптимізації є використання кривих Едвардса, що дозволить збільшити швидкодію ще в 4 рази. Але це вимагає подальшого дослідження на предмет наявності вразливостей при використанні кривих Едвардса в криптосистемах на ізогеніях ЕК.

4. Складність криптоаналізу схем, заснованих на ізогеніях ЕК, становить $O(p^{1/4})$ для класичного комп'ютера і $O(p^{1/6})$ для квантового, що має експоненційний характер.

5. Схема узгодження ключів – класична схема використання ізогеній в системах з відкритим ключем. В цій схемі дотримуються усі вимоги до криптографічних систем на основі ізогеній.

6. Схема електронного підпису дозволяє підтвердити правильність підпису в інтерактивному режимі з підписувачем. Недоліком є ймовірність підтвердження недійсного підпису. Щоб запобігти цьому, необхідно виконувати повторні перевірки при інших параметрах.

16.8 Процеси та методи вибору загальносистемних параметрів та аналізу стійкості до атак сторонніх каналів для ключового механізму інкапсуляції ДСТУ 8961:2019

Вступ

В останні роки спостерігається значний прогрес у створенні квантових комп'ютерів. Якщо масштабовані квантові комп'ютери будуть реалізовані, це поставить під загрозу безпеку найбільш широко використовуваних криптосистем відкритого ключа. Найбільш уразливими є ключові схеми, тобто, цифрові підписи, засновані на факторизації, дискретних логарифмах і криптографії еліптичної кривої. Основним завданням зараз є розробка, оцінка, дослідження та стандартизація асиметричних крипто - трансформацій на міжнародному рівні, включаючи ключові механізми інкапсуляції (KMI), стійкі до атак порушників постквантового періоду. Ще одним важливим завданням є подальше вивчення вже прийнятих національних стандартів АСУ та ГРД щодо стійкості до атак сторонніх каналів, зокрема, для оцінки залежності часу конвертації з використанням закритого ключа від структури ключових бітів.

Основні зусилля міжнародної криптографічної спільноти з розробки стандартизації та впровадження нових постквантових крипто-трансформацій зосереджені навколо конкурсу NIST US – процесу стандартизації NIST PQC, який розпочався в грудні 2016 року [318]. З 82 поданих кандидатів 69 були допущені до 1-го туру конкурсу. У січні 2019 року, на основі відкритого обговорення та зворотного зв'язку з криптографічною спільнотою, NIST відібрав 26 алгоритмів для другого туру [319], включаючи 17 асиметричних шифрувань та/або KMI.

1. Стан розробки та стандартизації ключових протоколів інкапсуляції на міжнародному рівні та в Україні

З критеріїв оцінки NIST найважливішим є критерій безпеки алгоритму. Для алгоритмів KMI NIST в програмному положенні конкурсу висуваються

вимоги до «семантичної безпеки» алгоритмів в плані стійкості до атак з адаптивно обраним шифротекстом, що еквівалентно моделі безпеки IND-CCA2. Враховуючи, що задоволення більш жорстких вимог моделі IND-CCA2 для деяких алгоритмів може вплинути на продуктивність [320], NIST також прийняла алгоритми, які забезпечують захист від атак з вибраним шифротекстом в IND-CPA.

У липні 2020 року завершився 2-й етап змагань і оголошено старт 3-го туру [320]. П'ятнадцять кандидатів пройшли до 3-го раунду, з яких 7 були обрані фіналістами і 8 як альтернативні кандидати. Зокрема, фіналістами були обрані наступні алгоритми KMI: Classic McEliece, SABER, CRYSTALS-KYBER, NTRU. Альтернативними кандидатами були: BIKE, FrodoKEM, ВККК, NTRU Prime, SIKE. З 4 основних фіналістів NTRU, CRYSTALS-KYBER, SABER засновані на алгебраїчних решітках, що послужило основою для припущення про включення хоча б одного з них в стандарт.

У доповіді [320] другою за важливістю вимогою є швидкість, а для кандидатів 3-го туру буде розглядатися як швидкість генерації ключів вперед і зворотні перетворення, а також просторова складність відкритих ключів, цифрових підписів і шифротекстових обчислень. Для алгоритмів KMI час генерації ключів вважається нарівні з часом переадресації та зворотного перетворення, оскільки велика кількість додатків використовує нову пару ключів для кожного сеансу, щоб забезпечити ідеальну безпеку вперед. В результаті, вимоги до безпеки та продуктивності в даний час є основними, які розглядаються для рішення NIST у 3-му раунді конкурсу PQC.

У [321] NIST рекомендувала розробникам зосередитися на розробці параметрів для рівнів стабільності 1-5, визначених у [322], тобто, для рівня стабільності 128 біт квантової та 256 біт класичної безпеки.

Незважаючи на це, на національному рівні було обґрунтовано [323] і поставлено завдання розробити алгоритм асиметричного перетворення типів і наборів параметрів KMI, який забезпечив би 7 рівнів стійкості, тобто, 256 біт квантової і 512 біт класичної безпеки. У [324] обґрунтовано основні алгоритми генерації загальносистемних параметрів, шифрування та дешифрування для передового алгоритму НТУ Прайм ІМТ Україна KMI.

Далі детально розглянуті питання побудови загальносистемних параметрів для 7-го рівня стійкості, що доводять правильність алгоритму, а також криптографічної стійкості в [324-326]. Згодом на основі досліджень, проведених у [323-326], був прийнятий національний стандарт ДСТУ 8961:2019 [329].

1.1 Основні параметри ДСТУ 8961:2019 та метод генерації параметрів для 7-го рівня стійкості

Основні параметри алгоритму представлені в таблицях 16.38 і 16.39. Весь набір параметрів можна знайти в [325,333].

Таблиця 16.38.

Загальні параметри

Визначення	Опис
$(Z/q)[x]$	Кільце з багаточленів. Всі коефіцієнти зменшені на модуль q.
$N \geq \max(3, 2t)$	Порядок багаточлена. Має бути простим числом, для якого багаточлен $x^n - x - 1$ незворотній. Порядок визначає кількість його коефіцієнтів.
$P = x^n - x - 1 \in (Z/q)[x]$	Монічний поліном **N** степеня, незвідний за полем $(Z/q)[x]$, на який зводяться поліноми – елементи з R/q.
$Z[x]/(x^n - x - 1)$	Кільце поліномів $Z[x]$ з модулем $x^n - x - 1$.
$(Z/3)[x]/(x^n - x - 1)$	Кільце поліномів $(Z/3)[x]$ з модулем $x^n - x - 1$.
$(Z/q)[x]/(x^n - x - 1)$	Кільце з багаточленів $(Z/q)[x]$ з модулем $x^n - x - 1$.
$p = 3$	Менший модуль, всі коефіцієнти зменшені цим модулем в багаточлені R/3.
$q \geq 48t + 3$	Більший модуль, всі коефіцієнти зменшені цим модулем в багаточлен r/q.
$t \geq 1$	Натуральне число, кількість ненульових елементів багаточлена залежить від цього параметра.
$\lambda \in \{256, 384, 512\}$	Рівень криптосили класичної безпеки.
$m \in R/3$	Приватне повідомлення. Число 0, 1, -1 більше або дорівнює **t**.
$e \in R/q$	Зашифроване повідомлення.
$G \in R/3$	Випадковий t-малий елемент (багаточлен), оборотний в полі R/3. Числа з 1 і -1 не обов'язково рівні. Секретний параметр, який використовується для обчислення відкритого ключа.
$f \in R/q; f = (1 + 3F) \bmod q$	Невеликий багаточлен, незвідний в R/q, є секретним ключем.
$F \in R/3$	Випадковий багаточлен, який ідентифікує закритий ключ.
$dF = 2t/3$	Число 1 і -1 в багаточлені f.
$df = 2t; df = dr$	Число 1 і -1 в секретному ключі, не обов'язково рівні.
$dg = \begin{cases} dg_1 = n/3 + 1 \\ dg_{-1} = n/3 \end{cases}$	Кількість 1 і -1 в багаточлені G.

Таблиця 16.39.

Додаткові параметри

Визначення	Опис
$qBits = \lceil \log_2 q \rceil$	Кількість бітів у q.
$r \in R/3$	Засліплюючий багаточлен, випадковий t-малий.
$dr = n/3$	Число 1 і -1 в засліплюючий багаточлен.
b	Випадковий компонент (сіль), щоб додати в повідомлення.
$db=256$	Довжина випадкового компонента (бітів).
$bLen = db/8$	Довжина випадкового компонента (октетів).
$\max MLen = \frac{\frac{3(N-1)}{2} - db}{8} - 1$	Максимальна довжина повідомлення для шифрування (байтів).
$Hlen = \begin{cases} 256, \lambda = 256 \\ 512, \lambda = 384 \| 512 \end{cases}$	Довжина значення хешу (біти).

Комплексний алгоритм опису загальних параметрів розрахунку представлений в [324,325,330, 331]. Спрощена послідовність кроків може бути представлена наступним чином.

Алгоритм:

Крок 1. Вибрати просте число N. Як просте число вибирається просте число, для якого порядок N-1 або (N-1) / 2.

$$2^{\lambda} < (3/2^{N})..$$

Крок 2. Сформувати ключовий простір для закритих ключів.

Для визначення ключового простору необхідно визначити кількість ненульових N / 3 (1 і -1) елементів в багаточлені f = F1 * F2 + F3 і клавішах G.

Тоді максимальна кількість ненульових елементів у ключі, визначеному багаточленом F = F1 * F2 + F3 з урахуванням чисел (1) і (-1), дорівнює [325]:

$$2d_1 * 2d_2 + 2d_3 = 4d_1d_2 + 2d_3.$$

Для того, щоб знайти ключі на багаточленах, необхідно, щоб F1, F2, F3 були приблизно тієї ж складності, що ми вибираємо

$$d_1 \approx d_2 \approx d_3. \qquad (16.19)$$

У [330,331] пропонується обчислити значення (16.19) за наступними формулами:

$$d_1 = \left\lceil \frac{-1+\sqrt{1+\frac{8N}{3}}}{4} \right\rceil, d_2 = \left\lceil \frac{\left\lceil \frac{N}{3} \right\rceil - d_1}{2d_1} \right\rceil, d_3 = \max\left(\left\lceil \frac{d_1+1}{2} \right\rceil, \left\lceil \frac{N}{3} - 2d_1d_2 \right\rceil \right),$$

$$d_g = \frac{N}{3}.$$

Крок 3. Розрахувати параметри безпеки з урахуванням ключового простору і атаки зустрічі посередині (верхні межі безпеки).

Для розрахунку параметра безпеки з урахуванням ключового простору і атаки зустрічі посередині кількість ключів визначається з урахуванням їх форми представлення і атаки зустрічі посередині [331].

Для визначення мінімального простого числа, що забезпечує потрібну стабільність λ, використовується наступна нерівність [330]:

$$2^{\lambda} \le \sqrt{\frac{\binom{N}{d_1\ d_1} * \binom{N}{d_2\ d_2} * \binom{N}{d_3\ d_3}}{N}}.$$

Якщо λ < необхідного, вибрати більше просте N і перейти до кроку 2.

Крок 4. Обчислити максимальну кількість ненульових елементів в повідомленні (d_m).

Під час шифрування даних закодованого повідомлення, перетвореного у форму малого багаточлена, воно повинно містити ряд ненульових елементів, визначених загальним параметром d_m для запобігання атак. Однак, у разі, якщо кількість таких ненульових елементів більше якогось порогу, то ймовірність повторного виділення маски і множення на засліплюючий багаточлен будуть високими [325]. Тому цей параметр істотно впливає на продуктивність алгоритму шифрування.

Для усунення помилки дешифрування достатньо наступної умови:

$$1 - \frac{\sum_{i=d_m}^{N-2d_m-1}\left(\sum_{j=d_m}^{N-d_m-i}\left(\binom{N}{i}\binom{N-i}{j}\right)\right)}{3^N} \le 2^{-10}.$$

Крок 5. Обчислити q. Повинно бути простим і задовольняти умові з [330, 331].

Аналіз показав, що значення модуля q впливає на ймовірність помилки дешифрування і використовується при оцінці безпеки решітки.

Для розрахунку значення q, яке забезпечує максимальну ймовірність помилки, яка визначається стабільністю, можна використовувати наступну нерівність:

$$q \ge 24(2d_1d_2 + d_3) + 3.$$

Крок 6. Обчислити Tmintm. Обчислити мінімальне значення 0 <r ≤ N, яке задовольняє наступним умовам:

$$T_{MINTM}(N, r, d_g) > \lambda;$$

$$T_{MINTM}(N, r, d_m) > \lambda.$$

Якщо не вдається виконати хоча б одну з умов, виберіть більше просте N і перейдіть до кроку 2.

Крок 7. Обчислити розмір решітки:

$$S = 2N - r\,.$$

Крок 8. Розрахувати $T_{lattice}$ - кількість операцій для створення Коркіна-Золотарьова зменшеної основи [330] повної решітки розмірності *S* за наступною формулою:

$$T_{Lattice} = 2^{E(S,m,\beta)}\,,$$

де $E(S,m,\beta) = 0,366098\beta + 0,000784314\beta^2 + 0,875$.

Це значення має бути більшим значення, яке відповідає надійно рівню λ**.** Якщо, $T_{Lattice} < \lambda$**,** виберіть більше значення простого N і перейдіть до кроку 2. В іншому випадку, в той час як $T_{Lattice} > T_{MINTM}$ - приріст *r* і перейдіть до кроку 6.

Обчислювані значення параметрів N, t, q наведені в таблиці 16.32. Вони можуть бути застосовані для інкапсуляції ключів і прямого шифрування. Виділені рядки використовуються в стандарті ДСТУ 8961:2019 [352].

Таблиця 16.40.

Розрахункові значення параметрів для різних рівнів стійкості стандарту ДСТУ 8961:2019

	N	t	q
СКЕЛЯ 256/128	881	159	7673
	883	168	8089
	907	160	7727
	907	183	8807
	953	132	6343
	953	171	8237
	967	171	8243
	971	101	4871
	971	198	9551
	977	120	5783
	977	162	7817
	991	194	9349
	997	112	5393
	1013	149	7177
	1019	139	6691
	1021	112	5393
	1021	183	8819
СКЕЛЯ 384/192	1201	192	9221
СКЕЛЯ 512/256	1259	210	10103
	1283	214	10289
	1289	215	10331
	1291	215	10331

Продовження табл. 16.40.

	1297	216	10453
	1301	217	10427
	1303	217	10429
	1307	218	10499
	1319	220	10567
	1321	220	10597
	1327	221	10613
	1361	227	10957
	1373	229	11057
	1381	230	11059
СКЕЛЯ 512/256	1399	233	11213
	1409	235	11299
	1423	237	11383
	1471	255	12251

2. Порівняльний аналіз наборів параметрів стандарту ДСТУ 8961 відносно критеріїв стабільності та складності

Показники ефективності для різних рівнів стабільності наведені в таблиці 16.41.

Таблиця 16.41.

Показники ефективності для різних рівнів стабільності

Рівень стабільності λ	Шифрування	Дешифрування
256	89224	102982
384	138237	146128
512	163658	188235

Існує прямий зв'язок між рівнем стабільності і часом прямого і зворотного перетворення. На рис. 16.18 показано графік цієї залежності. Очевидно, що при підвищенні рівня стійкості складність перетворень збільшується, тому важливо вибрати достатній рівень стійкості на основі наявної обчислювальної потужності.

2.1 Аналіз стійкості стандарту ДСТУ 8961:2019 проти атак бічних каналів

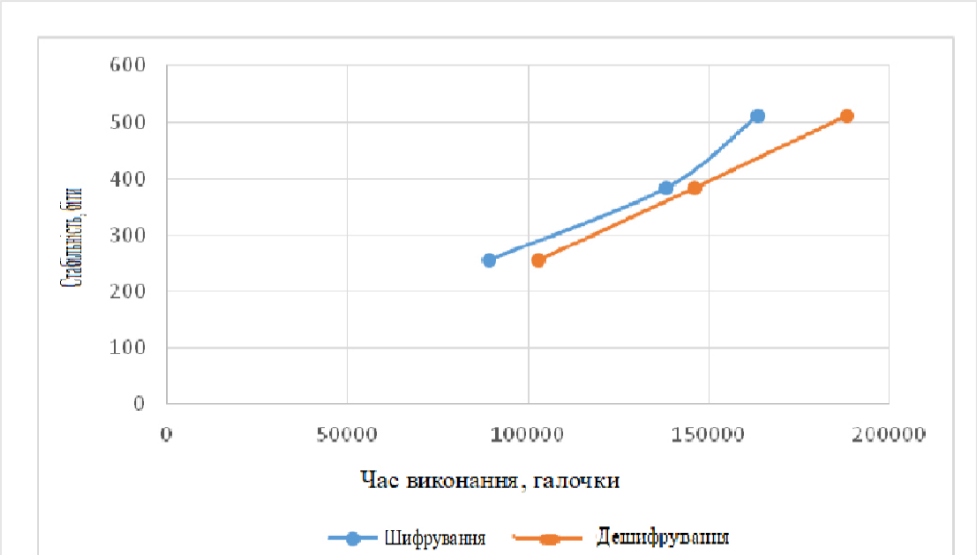


Рис. 16.18. Час шифрування та дешифрування

Важливим питанням є стабільність національного стандарту ДСТУ 8961:2019 проти атак бічних каналів, а також проти атак на впровадження. В таблиці 16.42 розглянуто аналіз залежності часу прямих та зворотних перетворень (інкапсуляції /декапсуляції ключів) від структури довгострокового ключа, а саме, від кількості одиниць у довгостроковому ключі.

Хеш-функція, яка визначена в національному стандарті ДСТУ 7564:2014, використовувалася як хеш-функція. Алгоритм, визначений в ДСТУ 8845:2019, використовується як алгоритм симетричного потокового перетворення.

Для експерименту було створено і відсортовано 10 000 ключів за збільшенням кількості. Для кожної клавіші було здійснено по 100 дзвінків на кожну з перевірених функцій і розраховано середнє значення часу виконання в процесорних годинниках для такого ключа. На рис. 16.19 і 16.20 випадкові відхилення можуть бути викликані іншими процесами в операційній системі і не залежати від кількості одиниць в ключі. Однак, деякі оптимізації реалізації все ж можливі і від реалізації може залежати кількість циклів.

Таблиця 16.42.

Значення коефіцієнтів кореляції між часом виконання функцій інкапсуляції / декапсуляції ключів і кількістю одиниць в ключі для стандарту ДСТУ 8961:2019

	Номер ключа	Кодування КМІ	Декодування КМІ
Номер ключа	1.000000	0.106736	0.153254
Кодування КМІ	0.106736	1.000000	0.768122
Декодування КМІ	0.153254	0.768122	1.000000

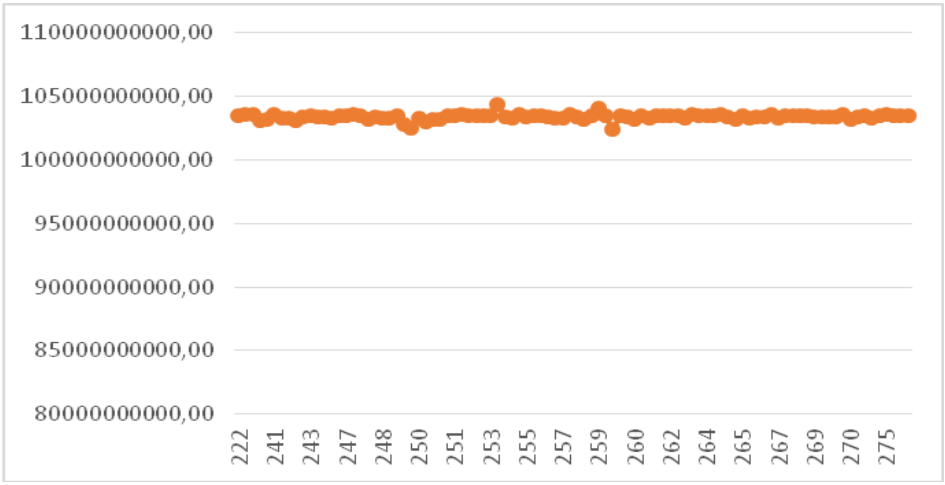


Рис. 16.19. Фрагмент графіка залежності часу виконання інкапсуляції від кількості одиниць у ключі до стандарту ДСТУ 8961:2019

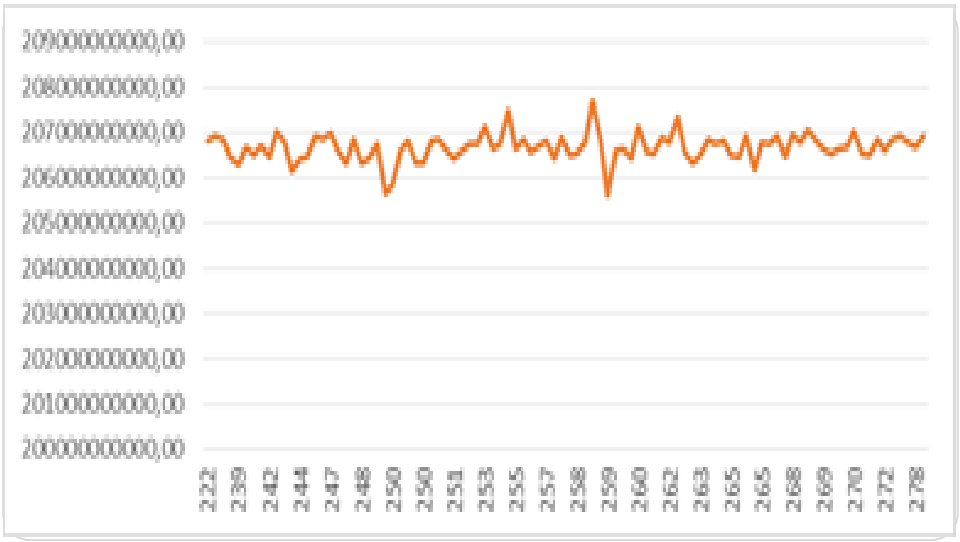


Рис.16.20. Фрагмент графіка залежності часу декапсуляції від кількості одиниць у ключі від стандарту ДСТУ 8961:2019

Коефіцієнти кореляції між часом виконання функцій інкапсуляції / декапсуляції ключів і кількістю одиниць в ключі до стандарту ДСТУ 8961:2019 знаходяться в діапазоні [0,106 ... 0,153), що свідчить про практичну самостійність часу виконання, перетвореного з кількості одиниць у ключі.

Висновки

1. Основними проблемами асиметричної криптографії є розробка стабільних асиметричних криптоперетворень, таких як КМІ проти як класичних, так і квантових атак, а також побудова загальносистемних параметрів 5-7 рівнів стійкості [324,325].

2. Особливе значення має надання криптографічної стійкості до атак сторонніми каналами, що вимагає принципово нового підходу до здійснення тестування криптографічних рішень постійного часу, а також аналізу існуючих стандартів уразливості до цього класу атак і контрзаходів.

3. Проведено дослідження національного стандарту ДСТУ 8961:2019 щодо протидії атакам сторонніх каналів. Алгоритми КМІ, викладені в ДСТУ 8961:2019, захищені від атак сторонніми каналами в разі правильної і точної реалізації, що підтверджено експериментально.

4. Рівень стабільності безпосередньо впливає на продуктивність, тому важливо вибрати достатній рівень стабільності на основі пристрою, наприклад, для смарт-карт і токенів; було б доцільно вибрати більш низький рівень стабільності. У розділі представлено залежність між продуктивністю та рівнем стабільності.

5. Одночасне використання кінцевого шифрування ключів дозволяє налаштувати симетричні ключі шифрування, такі як ключі AES, для збільшення швидкості безпечної передачі даних через канали зв'язку.

СУТНІСТЬ СТАНДАРТИЗАЦІЇ ТА ВПРОВАДЖЕННЯ КВАНТОВО-СТІЙКИХ СТАНДАРТІВ У США ТА УКРАЇНІ

Необхідно відмітити, що наразі на міжнародному та національному рівнях інтенсивно впроваджуються квантовостійкі асиметричні та симетричні крипто-перетворення. Для цього NIST США провів в 2017 - 2022 роках міжнародний конкурс щодо створення та стандартизації квантовостійких електронних підписів (ЕП), асиметричних шифрів (АСШ) та протоколів інкапсуляції ключів (ПІК) . В липні 2022 року за підсумками 3 раунду конкурсу в якості проєктів міжнародного конкурсу рекомендовано ПІК CRYSTALS - RYBER та ЕП: CRYSTALS- DILITHIUM, FALCON та SPHINCS. Надалі після тестування та дороблення на протязі майже року вони прийняті в якості чернеток федеральних стандартів США відповідно: CRYSTALS - RYBER- Fips IR - 203; CRYSTALS-DILITHIUM - Fips IR - 204, а SPHINCS - Fips IR - 205. Проєкт FALCON та Сокіл мають вразливості та удосконалюються.

Незважаючи на складність впровадження вказаних федеральних стандартів, президент США вимагає оперативного впровадження їх в федеральних органах. Це в основному викликано потенційним існуванням квантових атак на існуючі асиметричні примітиви ЕП, АСШ та ПІК.

На національному рівні проведені в той же період також прийняті національні квантово стійкі стандарти. У якості АСШ та ПІК - ДСТУ 8961-2019 та у якості ЕП - ДСТУ - 9212-2023. Також і на національному рівні стоїть складна проблема впровадження їх в інфраструктуру відкритого ключа та інші системи асиметричного та симетричного криптографічного захисту інформації.

В таблиці наведені особливості реалізації міжнародних та національних квантовостійких асиметричних стандартів.

9.5 Генерація ПВП для постквантових алгоритмів

Алгоритм	PRF (для вихідних послідовностей відомої довжини)	XOF (для вихідних послідовностей невідомої довжини)
FIPS 203	SHAKE256 SHA3_512	SHAKE128
FIPS 204	SHAKE256	SHAKE128 SHAKE256
FIPS 205	SHAKE256(mode 1) SHA2-256 (mode 1) SHA2-256 + SHA2-512 + MGF1 + HMAC (mode3, 5)	
ДСТУ 8961-2019	Струмок (ДСТУ 8845-2019)	Струмок (ДСТУ 8845-2019)
ДСТУ 9212-2023	Струмок (ДСТУ 8845-2019)	Струмок (ДСТУ 8845-2019)

82

ВИСНОВКИ

Квантова інформатика - це новий розділ науки, присвячений використанню квантових об'єктів для обробки та передачі інформації. В даний час докладається великих зусиль для розробки квантового комп'ютера. Створюються квантові елементи, будуються квантові алгоритми і розробляється архітектура квантового комп'ютера. Ще одним перспективним напрямком квантової інформатики є квантова криптографія. Квантові методи передачі гарантують неможливість розшифрування повідомлення. Ідея створення переплутаних станів, висловлена в свій час Ейнштейном, Подольським і Розеном, дозволяє передавати повідомлення через квантовий канал - без прямого зв'язку між передавачем і приймачем. Проте один біт інформації повинен передаватися через класичний канал.

Безпека сучасних інформаційних систем та технологій грунтується на стійкості криптографічних перетворень, які вони використовують для криптографічної обробки інформації. Криптографічна стійкість базується на складності розв’язання певних математичних задач (факторизації великого цілого числа, розв’язку дискретного алгоритму тощо), для таких задач характерна субекспоненційна або експоненційна складність розв’язання на сучасних (класичних) комп’ютерах. Проте, використовуючи квантові алгоритми Шора та Гровера, певні математичні задачі можна розв’язувати навіть з поліноміальною складністю. Ідеї використання потужностей квантового середовища висунули Пауль і Фейман. Важливим стало розроблення у 1992 р. Дойчем та іншими першого квантового алгоритму, можливості якого значно перевищували можливості звичайних комп’ютерів. У випадку появи квантового комп’ютера, на якому може бути запущений квантовий алгоритм криптоаналізу Шора або алгоритм пошуку в неупорядкованій базі даних Гровера, можуть виникнути великі загрози у інформаційній сфері відносно забезпечення криптографічної стійкості як для асиметричних криптоперетворень, так і для певних симетричних. Важливим є не тільки сам факт побудови такого комп’ютера, а й технічні характеристики, якими володітиме квантовий комп’ютер.

В квантовому комп'ютері ключ шифрування передається за допомогою елементарних часток світла – фотонів. Будь-який прилад, за допомогою якого третя сторона спробує перехватити дані, обов'язково вплине на стан фотона, і ключ буде недійсним. Передавати фотони можна за допомогою виділених оптоволоконних ліній, на обох кінцях такої лінії необхідні спеціальні шифрувальні пристрої. Недоліком квантової криптографії є те, що вона вимагає великих затрат на інфраструктуру. При цьому поки що ключі удається передавати на відстань (50-100 км), швидкість їх генерації досить низька, а на передачу фотонів впливає маса зовнішніх факторів.

Постквантова (квантово – безпечна) криптографія, як і класична, основана на розв'язку математичних задач. Проте нові алгоритми шифрування повинні бути іншими, щоб їх не могли розв'язати за допустимий час не тільки звичайні, але і квантові комп'ютери.

Також активні дослідження ведуться у напрямі квантових протоколів та квантових генераторів випадкових послідовностей.

Алгоритм Шора, використовуючи можливості квантових комп'ютерів, здатний виконати факторизацію числа не просто за поліноміальний час, а за час, що не набагато перевершує час множення цілих чисел (тобто, практично так само швидко, як відбувається саме шифрування). Таким чином, реалізація масштабованого квантового комп'ютера в разі її успіху поставила б хрест на більшій частині сучасного криптографічного захисту (мова не тільки про схему RSA, що прямо спирається на складності факторизації, а й про інші подібні схеми, які квантовий комп'ютер здатний зламати аналогічним чином).

Як і інші алгоритми для квантових комп'ютерів, алгоритм Шора ймовірнісний: він дає вірну відповідь з високою ймовірністю. Імовірність помилки може бути зменшена при повторному використанні алгоритму. Тим не менш, так як можлива перевірка запропонованого результату (множенням) в квадратичний час, алгоритм може бути модифікований так, що відповідь буде вірною з одиничною ймовірністю.

Програмний засіб, представлений в посібнику, може чудово моделювати алгоритм Шора та може застосовуватися у навчальних цілях.

Нині суттєві результати досягнуті в частині розроблення, стандартизації та застосування симетричного криптоперетворення. Разом з тим продовжується розвиток та здійснюються спроби розробити більш ефективні методи криптоаналізу симетричних криптосистем – симетричних блокових перетворень (СБП), симетричних потокових перетворень (СПП) та функцій гешування (ФГ). Одним із основних є метод Гровера. Освоєння та застосування квантових систем та квантових математичних методів, а також відповідне їх програмування носить як методологічний аспект, так і психологічний – складність сприйняття. Метод Гровера полягає в проведенні вичерпного пошуку специфічного (унікального) елемента у несортованій базі даних, що складається з $N = 2^n$ елементів, де n довжина квантового регістру (кількість кубітів). Для криптоаналізу СБП специфічність елементу може зводитись до сеансового чи довгострокового ключа, синхропослідовності тощо. Особливістю, наприклад, ключа є те, що при

його застосуванні зашифровані дані можуть бути розшифрованими за поліноміальний час. У порівнянні з найкращими класичними методами метод Гровера передбачає проведення пошуку з квадратичним прискоренням. Для отримання такого прискорення використовується квантова суперпозиція станів. Конкретна реалізація квантового оракула залежить від кожного окремого випадку та задачі. Наприклад, у випадку криптоаналізу СБП оракул виконує розшифрування на ключі та повертає результат - успіх чи ні.

Метод Гровера може бути застосований як на класичному, так і квантовому комп'ютері, хоча реалізація на класичному комп'ютері не є рентабельною.

Наведені параметри схеми електронного підпису Rainbow, відомості про атаки на нього та оцінки складності атак, набори параметрів, що відповідають різним категоріям безпеки NIST, вказують на наявність можливості реалізації атаки стосовно електронного підпису Rainbow та на те, що складність атак з використанням квантового комп'ютера значно менша за складність класичних реалізацій цих атак. Також слід зазначити, що вплив від застосування квантового комп'ютера на атаки на різні набори параметрів не є постійним. Для набору параметрів I найкращою атакою з використанням квантового комп'ютера є атака HighRank, а для наборів параметрів III та V – пряма атака. Також важливо відмітити, що використання квантових методів в наведених атаках зводиться до виконання певної частини атаки, окрім квантової атаки «грубої сили».

Таким чином, нині вирішується дві важливі проблеми – створення квантового комп'ютера та розробки і реалізації математичних методів квантового криптоаналізу.

Аналіз досліджень показує, що основна проблема у побудові квантового комп'ютера полягає в тому, що необхідно створити систему, яка б задовольняла майже несумісним таким вимогам:

- кубіти (елементи квантового комп'ютера) повинні бути максимально ізольовані один від одного та від навколишнього середовища;
- можливість корельованого впливу на пару кубітів, тобто, потрібно не тільки вміти змінювати стан одного кубіту, але й ще вмикати та вимикати взаємодію між парою сусідніх кубітів;
- система кубітів повинна бути досить стабільною, щоб зберігати корельованість станів, але одночасно й легко відновлюваною для нового циклу обчислень;
- реалізовувати таку сукупність зворотних перетворень над системою кубітів, які б дозволили виконати будь-яку потрібну логічну операцію;
- під час обчислень система повинна зберігати квантові властивості, але наприкінці треба зробити вимірювання, яке б однозначно визначило стан системи і у такий спосіб звело б квантову інформацію в класичну.

Дуже важливою властивістю квантових об'єктів є можливість здійснювати паралельні операції. Так, для системи із N кубітів, що перебуває в переплутаному стані, ефективно кодується відразу 2N чисел. Тому операція над нею, завдяки когерентності станів різних кубітів, впливає на всі доданки в сумі і це дозволяє обробляти відразу всі 2N чисел.

На сьогодні вже існують квантові алгоритми, які дають змогу проводити атаки на такі існуючі асиметричні криптосистеми:

- системи, що базуються на складності факторизації великого цілого числа (RSA);

- системи, що базуються на складності вирішення дискретного логарифму в скінченному полі Галуа (DSA) ;

- системи, що базуються на складності вирішення дискретного логарифму в групі точок еліптичної кривої (ECC) ;

- системи на базі алгебраїчних решіток (NTRU).

Усі вказані криптосистеми відносяться до класу ймовірно-стійких. А ця ймовірна стійкість як раз і визначається можливостями появи квантових комп’ютерів, і, як наслідок, вирішення задачі повного розкриття.

Значних здобутків досягла фірма D-Wave, яка стала першою компанією, що продала комерційну версію квантового комп’ютера. Так, з 20 травня 2011 D-Wave Systems продає за $ 11 млн. квантовий комп'ютер D- Wave One з 128-кубітним чіпсетом, який виконує тільки одну задачу - дискретну оптимізацію.

Квантовий комп'ютер – це пристрій, процеси обчислень та передачі даних у якому грунтуються на явищі квантової суперпозиції і квантової переплутаності. На нинішній час повноцінний квантовий комп'ютер є ще поки гіпотетичним пристроєм, можливість побудови якого пов'язана з вирішенням складних теоретичних та практичних проблем квантової фізики та складних експериментів. Дослідження в цьому напряму знаходяться на передньому краї сучасної фізики. Важливою проблемою є також обгрунтування вимог та створення мови програмування для квантового комп’ютера.

Відносно поняття «квантовий паралелізм» в обчисленні можна трактувати так: «Дані в процесі обчислень є квантовою інформацією, яка після закінчення процесу перетворюється в класичну шляхом вимірювання кінцевого стану квантового регістра із заданим числом кубітів. При цьому виграш в квантових алгоритмах досягається за рахунок того, що при застосуванні однієї квантової операції велике число коефіцієнтів суперпозиції квантових станів, які у віртуальній формі містять класичну інформацію, перетворюється одночасно».

З критеріїв оцінки NIST найважливішим є критерій безпеки алгоритму. Для алгоритмів KMI NIST в програмному положенні конкурсу висуваються вимоги до «семантичної безпеки» алгоритмів в плані стійкості до атак з адаптивно обраним шифротекстом, що еквівалентно моделі безпеки IND-CCA2. Враховуючи, що задоволення більш жорстких вимог моделі IND-CCA2 для деяких алгоритмів може вплинути на продуктивність, NIST також прийняла алгоритми, які забезпечують захист від атак з вибраним шифротекстом в IND-CPA

У липні 2020 року завершився 2-й етап змагань і оголошено старт 3-го туру. П'ятнадцять кандидатів пройшли до 3-го раунду, з яких 7 були обрані фіналістами і 8 як альтернативні кандидати. Зокрема, фіналістами були обрані наступні алгоритми KMI: Classic McEliece, SABER, CRYSTALS-KYBER, NTRU. Альтернативними кандидатами були: BIKE, FrodoKEM, BKKK, NTRU Prime, SIKE. З 4 основних фіналістів NTRU, CRYSTALS-KYBER, SABER засновані на

алгебраїчних решітках, що послужило основою для припущення про включення хоча б одного з них в стандарт.

Другою за важливістю вимогою є швидкість, а для кандидатів 3-го туру буде розглядатися як швидкість генерації ключів вперед і зворотні перетворення, а також просторова складність відкритих ключів, цифрових підписів і шифротекстових обчислень. Для алгоритмів КМІ час генерації ключів вважається нарівні з часом переадресації та зворотного перетворення, оскільки велика кількість додатків використовує нову пару ключів для кожного сеансу, щоб забезпечити ідеальну безпеку вперед. Тому вимоги до безпеки та продуктивності є основними, які розглядаються NIST у 3-му раунді конкурсу PQC.

NIST рекомендувала розробникам зосередитися на розробці параметрів для рівнів стабільності 1-5, тобто, для рівня стабільності 128 біт квантової та 256 біт класичної безпеки.

Незважаючи на це, на національному рівні було обґрунтовано і поставлено завдання розробити алгоритм асиметричного перетворення типів і наборів параметрів КМІ, який забезпечив би 7 рівнів стійкості, тобто, 256 біт квантової і 512 біт класичної безпеки. Обґрунтовано основні алгоритми генерації загальносистемних параметрів, шифрування та дешифрування для передового алгоритму НТУ Прайм ІМТ Україна КМІ. Далі детально розглянуті питання побудови загальносистемних параметрів для 7-го рівня стійкості, а також криптографічної стійкості. Згодом на основі досліджень, проведених в Україні, був прийнятий національний стандарт ДСТУ 8961:2019.

Сучасний розвиток квантових комп’ютерів та результати і дослідження, які ведуть гіганти технічної індустрії, надзвичайно сильно стосуються і криптографії. В свою чергу від квантової інформатики вже давно була відокремлена і квантова криптографія. Так, алгоритм Шора, що безумовно є її частиною, дуже сильно може змінити сучасну «класичну криптографію».

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Einstein A., Can Quantum – Mechanical Description of Physical Reallity be Considered Complete? / A. Einstein, B. Podolsky, N. Rosen // Phys. Rev. -V. 47. - 1935.
2. Feynman R.P. Quantum mechanical computers / Found. Phys.- 16.- 1986.- P. 507-531.
3. Orlov Yu.F., The wave logic of consciousness: A hypothesis, Int. J. The or. Phys., 21, p.37-53, 1982.
4. Параллельные алгоритмы решения задач вычислительной матема тики / [Химич А.Н., Молчанов И.Н., Попов А.В., Чистякова Т.В., Яковлев М.Ф.].- Киев: Наукова думка, 2008 г. – 248 с.
5. Feynman R.P., The Feynman Lectures of Physics, Quantum Mechanics, Ad discmWesley / R.P. Feynman, R.B. Leighton, M. Sands, Reading, Vol. Ill, 1965.
6. E. Schrödinger. Die gegenwiirtige Situation in der Quantenmechanik, Naturwisseuschaftei, 23, p.807-812; p.823-828; p,844-849, 1935. English translation, <The Present Situation in Quantum Mechanics», Proc. of the American Philosophical Society, 124, 323, 1980; reprinted in Quantum Theory and Measurement edited by .J.A. Wheeler and W.H. Zurek, Princeton, 152, 1983.
7. J.S. Bell, Speakable and Unspeakable in Quantum Mechanics, Cambridge U.P., Cambridge, 1987.- 397 p.
8. A. Barenco, A universal twobit gate for quantum computation, Proc. R. Soc. London A, 449, p.679-683, 1995.
9. D. Deutsch, Quantum computational networks, Proc. Roy. Soc. London, Ser A, 425, p.73-90, 1989.
10. D. Aharonov, Quantum computation, In D. Stauffer, editor, Annual Reviews of computational Physics VI. , World Scientific, Singapore, 1999.
11. R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, Quantum algorithms revis ited, roc .Roy. Soc. London A, 454(1969), p.339-354, 1998.
12. R.B. Griffiths, C.S. Niu, Semiclassical Fourier transform for quantum computation, Phys. Rev. Lett., 76(17), p.3228-3231, 1996.
13. C.H. Bennet, Quantum Information and computation, Phys. Today, 48, 10, p. 24- 30.
14. D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eible, H. Weinfurter, A. Zeilinger. Experimental quantum teleportation, Nature, 390, p. 575-579, 1997.
15. X. Zhou, D.W. Leung, I.L. Chuang, Methodology for quantum logic gate construction, Phys. Rev. A 62, p.052316-052328, 2000.
16. D. Gottesman, I. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, Nature, 402, p.390-393, 1999.
17. A.S. Bradley, M.K. Olsen, S.A. Haine, J.J. Hope, Teleportation of massive particles without shared entanglement, ar Xiv:quanph/0706.0062.
18. S.A. Haine, J.J. Hope, A multi-mode model of a non-classical atom laser produced by outcoupling from a Bose-Einstein condensate with squeezed light, Las. Phys. Lett., 2, p.597-602, 2005.

19. A.K. Ekert, Quantum cryptography based on Bell's theorem, Phys. Rev. Lett., 67, p. 661-663, 1991.

20. A.M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett., 77, p. 793-797, 1996.

21. R. Galderbank, E.M. Rains, P.W. Shor, N.J.A. S1 0ane, Quantum error correction and orthogonal geometry, Phys. Rev. Lett., 78, p.405-408, 1997.

22. B. Misra, E.C.G. Sudarshan, The Zeno's paradox in quantum theory, J. Math. Phys., v.18, p.756-763, 1977.

23. C.B. Chiu, E.C.G. Sudarshan, B. Misra, Time evolution of unstable quantum states and a resolution of Zeno's paradox, Phys. Rev., v. D16, p.520-529, 1977. 175.

24. R. Laflamme, C. Miquel, J.P. Paz and W.H. Zurek, Perfect quantum error correcting code, Phys. Rev. Lett., 77, p. 198-201, 1996.

25. D. Gottesman, I. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, Nature, 402, p.390-393, 1999.

26. P. Shor, Fault-tolerant quantum computation, In Proc. 37th Annual Symposium on Foundations of Computer Science, page 56, Los Alamitos, CA.

27. Gisin N. Quantum cryptography / N. Gisin, G. Ribordy, W. Tittel H. Zbinden // Reviews of Modern Physics. - 2002. - V. 74, issue 1. - P. 145-195.

28. Кибертерроризм на авиационном транспорте / [В.П. Харченко, Ю.Б. Чеботаренко, А.Г. Корченко, Е.В. Пацира, С.А. Гнатюк] // Проблеми інформатизації та управління. - К. : НАУ, 2009. - Вип. 4 (28).- С. 131-140. DOI: https://doi.org/10.18372/2073-4751.4.589

29. Wegman M.N. New hash functions and their use in authentication and set equality / M.N. Wegman, J.L. Carter // Journal of Computer and System Science. - 1981. - V. 22. - P. 265-279.

30. Ознаковий принцип формування класифікацій кібератак / [О.Г. Корченко, Є.В. Паціра. С.О. Гнатюк, В.М. Кінзерявий, С.В. Казмірчук] // Вісник Східноукраїнського національного університету імені Володимира Даля. -№ 4 (146). - Ч. 1, 2010. - С. 184-193.

31. Korchenko O. Modern quantum teclinologies of information security against cyber-terrorist attacks / O. Korchenko, Y. Vasiliu, **S** Gnatyuk // Aviation. Vilnius: Technika, 2010, Vol. 14, No. 2, p. 58-69.

32. Василиу Е.В. Стойкость квантовых протоколов распределения ключей типа "приготовление-измерение" / Е.В. Василиу // Georgian Electronic Scientific Journal: Computer Science and Telecommunications. -2007, No. 2 (13). P. 50-62.

33. Vasiliu E.V. Noncoherent attack on the ping-pong protocol with com pletely en tangled pahs of qutrits / Eugene V. Vasiliu // Quantum Information Processing. - 2011. - V. 10.- Num. 2. - P. 189-202.

34. Hwang W. Eavesdropper's optimal information in variations of Bennett-Brassard 1984 quantum key distribution in the coherent attacks / W. Hwang. D. Aim. S. Hwang /7 Physics Letters A. - 2001. - V. 279, issue 3-4. -P. 133-138.

35. Cai Q.Y. The "ping-pong" protocol can be attacked without eavesdropping / Q.Y. Cai // Physical Review Letters. - 2003. - Vol. 91, issue 10. - 109801.

36. Deng F.G. Robustness of two-way quantum communication protocols against Trojan horse attack / F.G. Deng, P. Zhou. X.H. Li et al // [Електронний ресурс]. - Rezhym dostupa: http://arxiv.org/abs/quant-ph/0508168

37. Li X.-H. Improving the security of secure direct communication based on the secret transmitting order of particles / X.-H. Li. F.-G. Deng. H.-Y. Zhou // Physical Review A. - 2006. - V. 74, issue 5. - 054302.

38. Toshiba Quantum Key Distribution System. Toshiba Research Europe Ltd. - [Електронний ресурс]. - Режим доступу: <http:/7www.toshiba-europe.cont/research/crl/QIG/quanmmkeysener.html >

39. MAGIQ QPN 8505 Security Gateway (QPN - 8505). - [Електронний ресурс]. - Режим доступу: <htф://www.magiqtech.coln/^lagiQ/Products_flles/8505_Data_Sheet.pdf>

40. Cerberis. - |Електронний ресурс]. - Режим доступу: <http://idquantique.coni/products/cerberis.htm>

41. Lutkenliaus N. Quantum key distribution with realistic states: photon-number statistics in the photon-number splitting attack / N. Lutkenliaus, M. Jahma // NJP. - 2002. - V. 4.- P. 441-449.

42. Williamson M. Eavesdropping on practical quantum cryptography / M. Williamson,V. Vedral // Journal of Modern Optics. - 2003. - V. 50, issue 13. - P. 1989-2011.

43. Niederberger A. Photon-number splittmg versus cloning attacks in practical implementations of the Bennett-Brassard 1984 protocol for quantum cryptography / A. Niederberger, V. Scarani. N. Gisin // Physical Review A. - 2005. - V. 71, issue 4.-042316.

44. Scarani V. The security of practical quantum key distribution / V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf et al // Review of Modern Physics. - 2009. - V. 81, issue 3. - P. 1301-1350.

45. Calsamiglia J. Conditional beam splitting attack on quantum key distribution / J. Calsamiglia, S.M. Barnett, N. Lutkenhaus // Physical Review A. -2001. - V. 65, issue 1. - 012312.

46. Faint laser quantum key distribution: Eavesdropping exploiting multiphoton pulses / [S. Felix, N. Gisin, A. Stefanov. H. Zbinden]//7 Journal of Modem Optics. - V. 48. issue 13. - P. 2009-2021.

47. Nauerth S. Information leakage via side channels in freespace BB84 quantum cryptography / S. Nauerth, M. Furst, et al //New Journal of Physics. - 2009. - V. 11, issue 6. - 065001.

48. Lamas-Linares A. Breaking a quantum key distribution system through a timing side channel /A. Lamas-Linares. C. Kurtsiefer // Optics Express. - 2007. - V. 15, issue 15. - P. 9388-9393.

49. Юдін О.К. Захист інформації в мережах передачі даних / О.К. Юдін, О.Г. Корченко, Г.Ф. Конахович // Підручник. - К. : Видавництво DIRECTLINE, 2009. - 714 с.

50. Корченко О.Г. Квантові технології конфіденційного зв'язку / О.Г. Корченко, С.О. Гнатюк. В.М. Кінзерявий // Защита информации: Сб. науч. трудов. - К. : НАУ, 2010. - Вип. 1. - С. 179-184.
51. Задірака В.К. Комп'ютерна криптологія: підручник / В.К. Задірака, О.С. Олексюк. – Київ-Тернопіль: Збруч, 2002. – 504 с.
52. Shor, Peter W. (1997), "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM J. Comput. 26 (5).
53. Shor P. W., Polynomial-time algorithms for prime factorization and discretelogarithms on a quantum computer //Proceedings of the 35th Annual Symposiumon the Foundations of Computer Science, S. Goldwasser ed., IEEE Computer Society Press, Los Alamitos CA, 1994.
54. Neal Koblitz and Alfred J. Menezes A. Riddle wrapped in an Enigma. Department of Mathematics, Box 353.350, University of Washington, Seattle, WA 98195 U.S.A. – Access mode: https://eprint.iacr.org/2015/1018.pdf.
55. Report on Post-Quatum Cryptography. NISTIR 8105 (DRAFT) /[Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone] // – Access mode: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.
56. Горбенко Ю.І. Методи побудування та аналізу, стандартизація та застосування криптографічних систем: монографія / Ю.І. Горбенко. Х. : Форт, 2016. – 959 с.
57. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. Round 3 Submissions. [Електронний ресурс]. Режим доступу: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.
58. Emma Strubell. An Introduction to Quantum Algorithms. COS498 – Chawathe. https://people.cs.umass.edu/~strubell/doc/quantum_tutorial.pdf
59. Горбенко Ю.І. Сутність та особливості реалізації методу Гровера на класичному комп'ютері для симетричного криптоаналізу / Ю.І. Горбенко, Є.Ю. Каптьол // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. – 2018.- Вип. 195. – С. 89-100.
60. Горбенко, І.Д. Прикладна криптологія. [Текст]: монографія / І.Д. Горбенко, Ю. І. Горбенко; ХНУРЕ. – Х.: Форт, 2012. - 868 с.
61. Lov K. Grover. A fast quantum mechanical algorithm for database search. 3C-404A, Bell Labs 600 Mountain Avenue Murray Hill NJ 07974. https://arxiv.org/pdf/quant-ph/9605043.pdf
62. Квантовые компьютеры.- Google.
63. Feyman, R.P. Quantum mechanical computers [Text] / R.P. Feyman // Opt. News – 1985. - February.11. – P.P. 11-39.
64. Andreas Hülsing. W-OTS+ – shorter signatures for hash-based signature schemes. In Amr Youssef, Abderrahmane Nitaj, and Aboul-Ella Hassanien, editors, Progress in Cryptology – AFRICACRYPT 2013, volume 7918 of LNCS, pages 173–188.Springer, 2013.
65. Daniel J. Bernstein, Grover vs. McEliece, The University of Illinois at Chicago, March 03, 2010. URL: http://cr.yp.to/codes/grovercode-20100303.pdf

66. Vera Blomkvist Karlsson, Philip Strömberg, 4-qubit Grover's algorithm implemented for the ibmqx5 architecture, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, June 6, 2018. URL: http://kth.diva-portal.org/smash/get/diva2:1214481/FULLTEXT01.pdf

67. Post-Quantum Cryptography PQC. Round 3 Submissions. NIST Computer Security Resource Center (CSRC). Режим доступу: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

68. PQC Standardization Process: Third Round Candidate Announce ment. NIST Computer Security Resource Center (CSRC). July 22, 2020. Режим доступу: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

69. Jintai Ding. Rainbow – Algorithm Specification and Documentation. The 3rd round Proposal. Department of Mathematical Sciences, University of Cincinnati.

70. J.C. Faugere A new efficient algorithm for computing Grobner Bases (F4). Journal of Pure and Applied Algebra, 139 (1999) 61-88. Режим доступу: https://doi.org/10.1016/S0022-4049(99)00005-5

71. L. Bettale, J.-C. Faugere, L.Perret. Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology, 3, pp. 177-197, 2009.

72. D. Coppersmith, J. Stern, S. Vaudenay. Attacks on the birational signature scheme. CRYPTO 1994, pp.435 – 443. Springer, 1994.

73. A. Kipnis, J. Patarin, L. Goubin. Unbalanced Oil and Vinegar schemes. EUROCRYPT 1999, pp. 206-222. Springer, 1999.

74. A. Kipnis, A. Shamir. Cryptanalysis of the Oil and Vinegar signature scheme. CRYPTO 1998, pp. 257-266. Springer,1998.

75. P. Schwable, B. Westerbaan Solving. Binary MQ with Grover’s Algorithm. SPACE 2016, pp. 303-322. Springer 2016.

76. A RIDDLE WRAPPED IN AN ENIGMA. NEAL KOBLITZ AND ALFRED J. MENEZES. Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A.

77. Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT)/[Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone]. https://www.google.com.ua/search?

78. Горбенко Ю.І. Методи побудування та аналізу, стандартизація та застосування криптографічних систем: Монографія / За загальною редакцією професора Горбенко Івана Дмитровича. – Харків: Форт, 2015. – 959 с.

79. Горбенко Ю.І. Аналіз можливостей квантових комп’ютерів та квантових обчислень для криптоаналізу сучасних криптосистем / Ю.І. Горбенко, Р.С. Ганзя. // Східно-європейський журнал передових технологій. – 2014. – № 1/9 (67). – С. 8–15.

80. Shor P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer [Text] / P.W. Shor //SIAM J. Comput. – 1997. – 26 (5). – P.P. 1484–1509. (67).

81. Grover, L. K. A fast quantum mechanics algorithm for database search [Text] / L. K. Grover // - Proceeding of the 28th annual ACM symposium on the theory of computing (STOC, 96). ACM Press, New York. – 1996. – P. P. 212–219.

82. Quantum computer built inside diamond [Electronic resource] / Futurity Research news from top universities – Режим доступу: www/ URL: – http://www. futurity. org/quantum-computer-built-inside-diamond/ – 09. 04

83. Lenstra H.W. Analysis and Comparison of Some Integer Factoring Algorithms, in Computational Methods in Number Theory [Text] / H.W. Lenstra, Jr. and R. Tijdeman, eds. // Math. Centre Tract 154 – 1946 – P.P. 89.

84. Горбенко Ю.І. Аналіз стійкості популярних криптоситем проти квантового криптоаналізу на основі алгоритму Гровера / Ю.І. Горбенко, Р.С. Ганзя // Науково-практичний журнал «Захист інформації». – 2014.Том 16, №2. – С. 22–28.

85. Ганзя Р.С. Квантовий криптоаналіз перетворень в групі точок еліптичних кривих/ Р.С. Ганзя, Ю. І. Горбенко // Спеціальні телекомунікаційні системи та захист інформаціцї. Збірник наукових праць. Випуск 2(22). Київ, 2012, С 17-30.

86. Ганзя Р.С. Методи здійснення атак на NTRU на основці квантових алгоритмів та їх порівняльний аналіз / Р.С. Ганзя, Ю.І. Горбенко // Спеціальні телекомунікаційні системи та захист інформації. Збірник наукових праць. Випуск 2(23). Київ 2013, С 9-17.

87. Горбенко Ю.І. Аналіз стійкості постквантових систем / Ю.І. Горбенко, Р.С. Ганзя // Прикладная радиоэлектроника. – Х: ХНУРЭ, 2014. – Том 13. – № 3. – С. 268–274.

88. Горбенко Ю.І. Аналіз шляхів розвитку криптографії після появи квантових комп'ютерів / Ю.І. Горбенко. Р.С. Ганзя // Компютерні системи та мережі: Вісник національного університету «Львівська політехніка». – 2014. – № 806. – С. 40–49.

89. Wang X.A quantum algorithm for searching a target solution of fixed weight [Text] / Wang X.W.S. Bao and X.Q. Fu// Chinese Sci Bull. - 2002. - Vol. 55(29). – P.P. 484–488.

90. Wang, H. An efficient quantum meet-in-the-middle attack against NTRU-2005 [Text] / Wang Hong, MA Zhi, MA ChuanGui // Chinese Science Bulletin. – 2013. – Vol. 58, No. 28–29. – P.p. 3514–3518.

91. Deputy Chief Designer JSC Institute of Information Technology.-www.iit.com.ua

92. Quantum Safe Cryptography and Security. An introduction, benefits, enablers and challenges. ETSI White Paper, No. 8 – 2015.

93. ETSI GR QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework.

94. Moody D. Post-Quntum Cryptography: NIST's Plan for the Future. The Seventh International Conference on Post-Quntum Cryptography, Japan, 2016. Режим доступу: [https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf]

95. D-Wave Systems: official site (англ.)

96. Quantum entanglement. The requested URL /content/view/40/60/ was not found on this server.

97. McGrew D., Curcio M. Hash-Based Signatures draft-mcgrew-hash-sigs-00 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступу: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-00

98. McGrew D., Curcio M. Hash-Based Signatures draft-mcgrew-hash- sigs-01 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступу: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-01

99. ETSI GR QSC 001 V.1.1.1 (2016-07). Quantum-Safe Cryptography (QSC); Quantum-safe algorithmic framework. Режим доступу: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/001/01.01.01_60 /gr_QSC001v010101p.pdf.

100. NIST PQC workshop: SAFEcrypto Project, M. O'Niell. – 2015.

101. Інтернет-ресурс. – Goole.

102. Feyman R.P. Quantum mechanical computers / Opt. News, February. - 1985. - 11.-P.p. 11-39.

103. Reinier Broker. Constructing supersingular elliptic curves. J. Comb. Number Theory, (3): p.p. 269–273, 2009.

104. Steven D. Galbraith. Constructing isogenies between elliptic curves over Finite Fields. LMS J. Comput. Math,. 2: pp. 118–138 (electronic), 1999.

105. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, PQCrypto, volume 7071 of Lecture Notes in Computer Science, p.p. 19–34. Springer, 2011.

106. A Note on Quantum Security for Post-Quantum Cryptography Fang Song Department of Combinatorics & Optimization and Institute for Quantum Computing University of Waterloo. https://eprint.iacr.org/2010/428.pdf

107. Random Oracles in a Quantum World /[Dan Boneh1, Ozgur Dagde- len2, Marc Fischlin2, Anja Lehmann3, Christian Schaner, and Mark ZhandRy] // Stanford University, USA. https://eprint.iacr.org/2014/709.pdf

108. Towards a Game Theoretic View of Secure Computation/ Gilad Asharov, Ran Canetti, Carmit Hazay http://gir.bmstu.com /data/ksbp_data/pdf/EUROCRYPT/EC_2011.00024_Towards_a_Game_Theoretic_View_of_Secure_Computation.pdf

109. ETSI White Paper No. 8, Quantum Safe Cryptography and Security: An Introduction, Benefits, Enablers and Challenges, June 2015. Режим доступу: http://www.etsi.org/ - 24.07.2016.

110. Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010, Proceedings NSA acknowledges need for quantum-safe crypto. [Електронний ресурс]. Режим доступу: http://www.idquantique.com/nsa-quantum-safe-crypto/- 24.07.2016.

111. NISTIR 8105 DRAFT Report on Post-Quantum Cryptography. National Institute of Standards and Technology Internal, Report 8105. [Електронний ресурс]. Режим доступу: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf - 24.07.2016

112. Evaluating Post-Quantum Asymmetric Cryptographic Algorithm Candidates / Tolga Acar, Josh Benaloh, Craig Costello and Dan Shumow. - MSR Security and Cryptography Group. [Електронний ресурс]. Режим доступу:

http://csrc.nist.gov/groups/ST/post-quantum-2015/presentations/session7-shumowdan.pdf - 24.07.2016.

113. Bernstein D. Post-quantum cryptography [Text] / D. Bernstein, J. Buchmann, E. Dahmen. – Berlin: Springer, 2009. – 246 p.

114. Sphincs: practical stateless hash-based signatures. Cryptology ePrint Archive, Report 2014/795, 2014 /Daniel J. Bernstein, Daira Hopwood, Andreas Helsing. Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Peter Schwabe and Zooko Wilcox O'Hearn// [Електронний ресурс]. Режим доступу: http://eprint.iacr.org/- 24.07.2016.

115. Ray A. Perlner and David A. Cooper Quantum resistant npublic key cryptography: a survey. In Proceedings of the 8th Symposium o Identity and Trust on the Internet (IDtrust '09), Kent Seamons, Neal McBurnett, and Tim Polk (Eds.). - New York, ACM, 2009. P. 85-93.

116. Enhanced Lattice-Based Signatures on Recon gurable Hardware Extended Version [Електронний ресурс]. – Режим доступу: https://eprint.iacr.org/2014/254.pdf

117. M. Finiasz Parallel-CFS - Strengthening the CFS McEliece-Based Signature Scheme. In A. Biryukov, G. Gong, and D.R. Stinson, editors, Selected Areas in Cryptography, vol. 6544 of Lecture Notes in Computer Science. - Springer Berlin Heidelberg, 2011. P. 159 – 170.

118. McEliece R.J. A Public-Key Criptosystem Based on Algebraic Theory // DGN Progres Report 42-44, Jet Propulsi on Lab. Pasadena, CA. January – February, 1978. – P. 114-116.

119. W. Whyte. IEEE P1363.1 Draft 10: Draft Standard for Public Key Cryptographic Techniques Based on Hard Problems over Lattices [Електронний ресурс] / N. Howgrave-Graham, J. Hostein, J. PIpher, J.H. Silverman. Режим доступу: http://grouper.ieee.org/groups/1363/ WorkingGroup/contact.html

120. Neal Koblitz and Alfred J. Menezes. A Riddle Wrapped in an Enigma. https://eprint.iacr.org/2015/1018.pdf

121. [Електронний ресурс] https://csrc.nist.gov/projects/post-quantum-cryptography.

122. Jean-Phillippe Aumasson and Guillaume Endignoux.: Gravity- SPHINCS – Submission to the NIST's post-quantum cryptography standardization process (2017).

123. Daniel J. Bernstein et al.: SPHINCS+ – Submission to the NIST's post-quantum cryptography standardization process (2019).

124. [Електронний ресурс] https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.

125. Daniel J. Bernstein et al. Sphincs: practical stateless hash-based signatures. Cryptology ePrint Archive, Report 2014/795, 2014.

126. Leslie Lamport. Constructing digital signatures from a one-way function. Technical. Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.

127. Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, CRYPTO, volume 435 of LNCS, pages 218–238. Springer, 1989.

128. Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hulsing and Markus Ruckert. On the security of the Winternitz one-time signature scheme. In A. Nitaj and D. Pointcheval, editors, Africacrypt 2011, volume 6737 of Lecture Notes in Computer Science, pages 363-378. Springer Berlin / Heidelberg, 2011.

129. Andreas Huülsing. W-OTS+ – shorter signatures for hashbased signature schemes. In Amr Youssef, Abderrahmane Nitaj, and Aboul-Ella Hassanien, editors, Progress in Cryptology // AFRICACRYPT 2013, volume 7918 of LNCS, pages 173–188. Springer, 2013.

130. Leonid Reyzin and Natan Reyzin. Better than biba: Short one-time signatures with fast signing and verifying. In Lynn Batten and Jennifer Seberry, editors, Information Security and Privacy, volume 2384 of Lecture Notes in Computer Science, pages 1–47. Springer Berlin/ Heidelberg, 2002.

131. Andreas Huülsing, Lea Rausch, and Johannes Buchmann. Optimal parameters for XMSSMT. In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl and Lida Xu, editors, Security Engineering and Intelligence Informatics, volume 8128 of Lecture Notes in Computer Science, pages 194–208. Springer Berlin Heidelberg, 2013.

132. Горбенко Ю.І. Аналіз потенційних постквантових механізмів електронних підписів на основі геш-функцій / Ю.І. Горбенко, Т.В. Мельник, І.Д. Горбенко // Радиотехника. -2017. -Вып. 189.

133. M.A. Kudinov, E.O. Kiktenko, A.K. Fedorov. Security analysis of the W-OTS+ signature scheme: Updating security bounds – 2020.

134. Chen L, Jordan S, Liu Y-K, Moody D, Peralta R, Perlner RA, Smith-Tone D (2016). Report on Post-Quantum Cryptography (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8105. https://doi.org/10.6028/NIST.IR.8105

135. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8309 / [Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone] // 22 July 2020. Режим доступу: https://doi.org/10.6028/NIST.IR.8309.

136. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehlé / ЕП CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation. Access mode: https://pq-crystals.org/dilithium/data/dilithium-specification.pdf

137. Post-Quantum Cryptography. Round 2 Submissions. [Електронний ресурс]. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions

138. ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2019. 445 p.

139. The problem of cryptographic transformations standardization and the state of its solution / [Ivan Gorbenko, Olena Kachko, Oleksandr Kuznetsov, Yurii Gorbenko, Maryna Yesina] // VII Міжнар. наук.-техн. конф. "Захист інформації і

безпека інформаційних систем” : Праці Науково-технічної конференції, 30–31 травня 2019 р. Львів : Нац. ун-т “Львівська політехніка”, 2019. С. 84-85.

140. Горбенко І.Д. Порівняння, оцінювання, дослідження можливості використання та переваг постквантових алгоритмів / І.Д. Горбенко, В.А. Пономар, М.В. Єсіна // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. Київ: Нац. техн. ун-т України "Київський політехнічний інститут імені Ігоря Сікорського", 2017. Вип. 2(34). С. 9-32.

141. Постквантова криптографія та механізми її реалізації /[І.Д. Горбенко, О.О. Кузнецов, О.В. Потій, Ю.І. Горбенко, Р.С., Ганзя, В.А. Пономар]// Радиотехника. 2017. Вып. 186. С. 32–52.

142. ДСТУ 8961:2019. Інформаційні технології. Криптографічний захист інформації. Алгоритми асиметричного шифрування та інкапсуляції ключів. [Електронний ресурс]. Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=88056.

143. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In ASIACRYPT, pages 598–616, 2009.

144. Vadim Lyubashevsky. Lattice signatures without trapdoors. In EUROCRYPT, pages 738–755, 2012.

145. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In CHES, pages 530–547, 2012.

146. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. Cryptology ePrint Archive, Report 2017/916, 2017. Access mode: https://eprint.iacr.org/2017/916

147. Lyubashevsly V (2009). Fiat-Shamir with aborts: Applications to Lattice and Factoring-Based Signatures. International Conference on the Theory and Application of Cryptology and Information Security – ASIACRYPT (Springer), pp. 598-616. https://doi.org/10.1007/978-3-642-10366-7_35

148. Don J, Fehr S, Majenz C, Schaffner C (2019). Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. Annual International Cryptology Conference – CRYPTO (Springer), p. 356-383. https://doi.org/10.1007/978-3-030-26951-7_13

149. Єсіна М.В. Моделі безпеки постквантових криптографічних примітивів / М.В. Єсіна // Міжнародний науковий симпозіум “Питання оптимізації обчислень (ПОО-XLVI)”, 2019 р. Математичне та комп’ютерне моделювання. Серія: Технічні науки. Вип. 19. С. 49-55.

150. V. Shoup On Formal Models for Secure Key Exchange, Theory of Cryptography Library, 1999. [Електронний ресурс]. Режим доступу: http://philby.ucsd.edu/cryptolib/1999/9912.html

151. M. Bellare, R. Canetti, H. Krawczyk. A modular approach to the design and analysis of authentication and key-exchange protocols. 30th STOC 1998.

152. Privacy for Code-Based Encryption in the Standard Model. In: Lange T., Takagi T. (eds) Post-Quantum Cryptography. PQCrypto 2017. Lecture Notes in Computer Science, vol 10346. Springer, Cham.

153. M. Bellare, A. Boldyreva, A. Desai, D. Pointcheval / Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248. P. 566–582. Springer, Heidelberg (2001). doi:10.1007/3-540-45682-1.

154. Державна служба спеціального зв'язку та захисту інформації України. Наказ від 20.07.2007 №141 «Положення про порядок розроблення, виробництва та експлуатації засобів криптографічного захисту конфіденційної та відкритої інформації з використанням електронного цифрового підпису» № 862/14129.

155. Закон України «Про основні засади забезпечення кібербезпеки України» // Відомості Верховної Ради (ВВР). 2017. № 45. Ст. 403.

156. Закон України «Про електронні довірчі послуги» // Відомості Верховної Ради (ВВР), 2017. № 45. Ст. 400.

157. Закон України "Про захист інформації в інформаційно-телекомунікаційних системах".

158. Закон України "Про захист персональних даних".

159. Горбенко І.Д. Прикладна криптологія: Монографія / Горбенко І.Д., Горбенко Ю.І. 2-ге вид. Харків : Форт, 2012.- 868 с.

160. Revisiting the expected cost of solwing uSVP and applications to LWE / [Albrecht M.R., Goepfert F., Virdia F., Wunderer T.] // Cryptology ePrint Archive, Report 2017/815. Access mode: http://eprint.iacr.org/2017/815

161. Albrecht M.R., Player R., Scott S. On the concrete hardness of learning with errors // Cryptology ePrint Archive, Report 2015/046. Access mode: http://eprint.iacr.org/2015/046

162. Rachel Player Parameter selection in lattice-based cryptography. Access mode: https://pure.royalholloway.ac.uk/portal/files/29983580/2018playerrphd.pdf

163. Gottfried Herold, Elena Kirshanova and Alexander May. On the asymptotic complexity of solving LWE // Designs, Codes and Cryptography, Jan 2017.

164. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE // Willy Susilo and Yi Mu, editors, ACISP 14, vol. 8544 of LNCS, pages 322–337. Springer, Heidelberg, July 2011.

165. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors // Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, ICALP 2011, Part I, vol. 6755 of LNCS, pages 403–415. Springer, Heidelberg, July 2011.

166. Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, and Ludovic Perret. Algebraic algorithms for LWE. Cryptology ePrint Archive, Report 2014/1018, 2014. Access mode: http://eprint.iacr.org/2014/1018

167. Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the BKW algorithm on LWE // Designs, Codes and Cryptography, 74:325–354, 2015.

168. Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy modulus switching for the BKW algorithm on LWE // Hugo Krawczyk, editor, PKC 2014, vol. 8383 of LNCS, pages 429–445. Springer, Heidelberg, March 2014.

169. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE based encryption // Aggelos Kiayias, editor, CT-RSA 2011, vol. 6558 of LNCS, pages 319–339. Springer, Heidelberg, February 2011.

170. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model // Journal of the ACM, 50(4): 506–519, July 2003.

171. Leon Groot Bruinderink1 and Peter Pessl2 . Differential Fault Attacks on Deterministic Lattice Signatures.

172. Lyubachevsky V., Ducas L., Kiltz E. [et all]. CRYSTALS–Dilithium. Techn. rep. NIST (2017). [Electronic resource]. Access mode: https://crc.nist.gov./projects/post-quantum-cryptogtraphy / round-1-submissions.

173. Albrecht M.R., Goepfert F., Virdia F., Wunderer T. Revisiting the expected cost of solwing uSVP and applications to LWE // Cryptology ePrint Archive, Report 2017/815. Access mode: http://eprint.iacr.org/2017/815

174. Peikert C., How (not) to instanaite Ring-LWE // Cryptology ePrint-Achive 2016/351, 2016.

175. Особливості побудування та аналіз електронних підписів 5 рівня безпеки для постквантового періоду на основі алгебраїчних решіток / [І.Д. Горбенко, О.Г. Качко, А.М. Олексійчук, Ю.І. Горбенко, В.П. Звєрєв, М.В. Єсіна, В.А. Пономар] // Прикладная радиоэлектроника. Харьков : ХНУРЭ, 2019. Т. 18, № 3, 4. С. 123–136.

176. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography – CRC Press, 1997. – 794 p.

177. Standards catalogue. 2016. URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm? Csnumber =18199

178. Arto Salomaa. Public-Key Cryptography, Second, Enlarged Edition. – Springer-Verlag, Berlin, Heidelberg, New York, 1996. – x+271 p.p.

179. Nigel Smart. Cryptography: An Introduction (3rd Edition). – 432 p.p. https://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf

180. David Deutsch and Richard Jozsa. Rapid solutions of problems by quantum computation / Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 439, no. 1907.–1992.– P. 553-558.

181. Cleve R., Ekert A., Macchiavello C., Mosca M. Quantum algorithms revisited / Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1969. – 1998. – P. 339-354.

182. Simon D. R. On the power of quantum computation // Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium. – P. 116-123.

183. Grover L. A framework for fast quantum mechanical algorithms / Proceedings of the 13th annual ACM symposium on theory of computing (STOC' 98). ACM Press, New York. – 1998. – P. 53–62.

184. Shor P. W. Algorithms for quantum computation: discrete logarithms and factoring / Foundations of Computer Science : Conference Publications. – 1994. – P. 124-134.

185. Committee on National Security Systems, Use of public standards for the secure sharing of information among national security systems, Advisory Memorandum 02-15, July 2015. https://cryptome.org/2015/08/CNSS_Advisory_Memo_02-15.pdf

186. Bernstein, Daniel J., Buchmann, Johannes, and Dahmen, Erik. Post-Quantum Cryptography. – 2009, Springer-Verlag, Berlin-Heidleberg. – 245 p.

187. Niederreiter H. Knapsack-type cryptosystems and algebraic coding theory // Problem Control and Inform Theory, 1986, v. 15. P. 19-34.

188. T. R. N. Rao and K. H. Nam. Private-key algebraic-coded cryptosystem. Advances in Cryptology – CRYPTO 86, New York. – NY: Springer. – pp. 35–48.

189. Yu. V. Stasev, A. A. Kuznetsov. Asymmetric Code-Theoretical Schemes Constructed with the Use of Algebraic Geometric Codes // Cybernetics and Systems Analysis, Volume 41, Issue 3, May 2005, Pages 354 – 363.

190. Gorbenko I.D., Zamula A.A., Semenko Ye.A. Ensemble and correlation properties of cryptographic signals for telecommunication system and network applications // Telecommunications and Radio Engineering. - Volume 75, 2016 Issue 2. pages 169-178.

191. Кузнецов А.А. Исследование эффективности криптосистем на алгебраических блоковых кодах // Системи обробки інформації. – Харків: ХУПС. –2005. – Вип. 4. – С. 202 –206.

192. Кузнецов А.А. Исследование помехоустойчивости и криптостойкости теоретико-кодовых схем // Моделювання та інформаційні технології. – Київ: НАНУ. – 2005. – №33. – С. 81-84.

193. Fisher Jean-Dernard , Jacques Stern. An efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding / Jean-Dernard Fisher, Stern Jacques // EUROCRYPT’96 Proceeding, LNCS 1070. P. 245 – 255.

194. Improvement for distinguisher efficiency of the 3-round Feistel network and a random permutation / Oliynykov R., Gorbenko I., Dolgov V., Kaidalov D. // Proceedings of the 6th IEEE International ConfeRence on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2011, 2011, pp 743 - 746.

195. Gaborit, P., Laudaroux, C., and Sendrier, N.: Synd: a very fast code-based cipher stream with a security reduction. In IEEE Conference, ISIT’07, pages 186–190.

196. Kirill Morozov, Tsuyoshi Takagi. Zero-Knowledge Protocols for the McEliece Encryption // Information Security and Privacy Volume 7372 of the series Lecture Notes in Computer Science, p.p. 180-193.

197. Methods of Information Protection in Communications Systems and Methods of Their Cryptoanalysis / Gorbenko I.D., Dolgov V.I., Rublinetskii V.I., Korovkin K.V. // Telecommunications and Radio Engineering. - Volume 52, 1998 Issue 4. pages 89-96.

198. Courtois, N., Finiasz, M., and N.Sendrier. How to achieve a McEliece-based digital signature scheme. In Advances in Cryptology - ASIACRYPT 2001, volume 2248, pages 157–174.

199. Finiasz, M.: Parallel-CFS: Strengthening the CFS McEliece-based signature scheme. In Biryukov, A., Gong, G., Stinson, D., eds.: Selected Areas in Cryptography. Volume 6544 of LNCS., Springer (2010), p.p. 159 -170.

200. Stern, J.: A new identification scheme based on syndrome decoding. In Advances in Cryptology - CRYPTO'93, volume 773 of LNCS. Springer Verlag (1994).

201. Veron, P.: Improved identification schemes based on error-correcting codes. Appl. Algebra Eng. Commun. Comput., 8(1):57–69 (1996).

202. Anne Canteaut and Nicolas Sendrier. Cryptanalysis of the original McEliece cryptosystem. In Kazuo Ohta and Dingyi Pei, editors, Advances in cryptology— ASIACRYPT'98, volume 1514 of Lecture Notes in Computer Science, pages 187– 199.

203. Vladimir M. Sidelnikov and Sergey O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Mathematics and Applications, 1992. – P. 439–444.

204. Minder L., Shokrollahi A. Cryptanalysis of the Sidelnikov Cryptosystem // Advances in Cryptology — EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007. Proceedings — Springer Berlin Heidelberg, 2007. — P. 347—360.

205. Daniel J. Bernstein and Tanja Lange and Christiane Peters. Attacking and defending the McEliece cryptosystem. https://cr.yp.to/codes/mceliece-20080807.pdf

206. E. Berlekamp, R. McEliece, H. van Tilborg. On the Inherent Intractability of Certain Coding Problems. // IEEE Transactions on Information Theory, vol. IT-24, No. 3, May 1978. – P. 384-386.

207. Clark G.C., Cain J.B. Error-Correction Coding for Digital Communications. – Springer, 1981, - 432 p.

208. Blahut R.E. Theory and Practice of Error Control Codes. – Addison Wesley Publishing Company, Inc., Reading, Massachusetts, 1983, – 500 p.

209. F.J. Mac Williams and N.J.A. Sloane. The theory of error-correcting codes. – North-Holland, Amsterdam.

210. Claude E. Shannon. Communication in the Presence of Noise. Proceedings of the IRE, vol. 37, no. 1, pp. 10–21, Jan. 1949.

211. National Institute of Standards and Technology, "FIPS-197: Advanced Encryption Standard", November 2001: http://csrc.nist.gov/publications/ fips/fips197/fips-197.pdf

212. A New Encryption Standard of Ukraine: The Kalyna Block Cipher. https://eprint.iacr.org/2015/650.pdf

213. Raphael Overbeck, Nicolas Sendrier, Code-based cryptography. In: Daniel J. Bernstein, et al. (eds). First International Workshop on Post-quantum Cryptography, PQ Crypto 2006, Leuven, The Netherland, May 23-26, 2006. Selected papers, pp. 95-145.

214. D.J. Bernstein. Grover vs. McEliece. In N. Sendrier, editor, Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt,

Germany, May 25-28, 2010. Proceedings, volume 6061 of Lecture Notes in Computer Science, pages 73–80. Springer, 2010.

215. A. Menezes, P. van Oorschot, S. Vanstone. Chapter 14. Efficient Implementation // Handbook of Applied Cryptography. – CRC-Press, 1996. – 816 p.

216. Kerry Maletsky. RSA vs ECC Comparison for Embedded Systems. White Paper. Atmel Corporation – 2015, 5p. http://www.atmel.com/images/atmel-8951-cryptoauth-rsa-ecc-comparison-embedded-systems-whitepaper.pdf

217. John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. arXiv.quant-ph/0301141, v2, 2004.

218. Ziatdinov M. Using frequency analysis and Grover's algorithm to implement known ciphertext attack on symmetric ciphers // Lobachevskii Journal of Mathematics 2013, vol.34, N4, pages 313-315.

219. Метод недвійкового рівновагового кодування / В.Б. Дудикевич, О.О. Кузнєцов, Б.П. Томашевський // Сучасний захист інформації.- 2010. - № 3. - С. 57-68. http://nbuv.gov.ua/UJRN/szi_2010_3_10

220. Tietavainen A., Perko A. There are no unknown perfect binary codes. — Annales Universitatis Turkuensis. — Ser. A, I 148, 3-10, 1971.

221. Lint van J. H. Nonexistence theorems for perfect error-correcting codes. - Computers in Algebra and Number Theory. -Vol. IV, 1971.

222. ETSI GR QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework.

223. NIST Workshop on Cybersecurity in a Post-Quntum World (2015). PQCrypto project, T Lange.

224. PQCrypto. Initial recommendation of Long-term secure post-quantum systems. 2015. Email: coordinator@pqcrypto.eu.org www.pqcrypto.eu.org

225. Gonzáles Vasco M. I. On minimal length factorizations of_nite groups / M. I. Gonzáles Vasco, M. Rotteler, R. Steinwandt // Experimental Mathematics. 2003. Vol. 12 (1). P. 1–12.

226. Singhi N. Minimal logarithmic signatures for _nite groups of Lie type / N. Singhi, S. Magliveras // Designs, Codes and Cryptography. 2010. Vol. 55 (2). P. 243–260.

227. Magliveras S. New approaches to designing public key cryptosystems using one-way functions and trap-doors in nite groups / S. Magliveras, D. Stinson, T. van Trung // Journal of Cryptology. 2002. Vol. 15. P. 285–297.

228. Lempken W. A public key cryptosystem based on non-abelian _nite groups / W. Lempken, T. van Trung, S.S. Magliveras, W. Wei // Journal of Cryptology. 2009. Vol. 22 (1). P. 62–74.

229. Goldreich O. Foundations of Cryptography: Basic Tools / O. Goldreich // Cambridge University Press. 2001.

230. Nuss A. On group based public key cryptography [Electronic resource] : Phd thesis. Access mode : http://nbn-resolving.de/urn:nbn:de:bsz:21-opus-63659.

231. Blackburn S. R. Cryptanalysis of the MST 3 public key cryptosystem / S. R. Blackburn, C. Cid, C. Mullan // Journal of Mathematical Cryptology. 2009. Vol. 3 (4). P. 321–338.

232. Bohli J. Weak keys in MST / J. Bohli, M.I. Gonzáles Vasco, C.J.M. Martínez, R. Steinwandt // Designs, Codes and Cryptography. 2005. Vol. 37 (3). P. 509–524.

233. Caranti A. The round functions of cryptosystem PGM generate the symmetric group / A. Caranti, F. D. Volta // Designs, Codes and Cryptography. 2006. Vol. 38 (1). P. 147–155.

234. Magliveras S. Algebraic Properties of Cryptosystem PGM / S. Magliveras, N.D. Memon // Journal of Cryptology. 1992. Vol. 5 (3). P. 167–183.

235. Khalimov G. MST3 Cryptosystem Based on a Generalized Suzuki 2-Groups [Electronic resource] / G. Khalimov, Y. Kotukh, S. Khalimova. Access mode : http://ceur-ws.org/Vol-2711/paper1.pdf

236. Khalimov G., MST3 cryptosystem based on the automorphism group of the hermitian function field' / G. Khalimov, Y. Kotukh, S. Khalimova // IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019 – Proceedings, 2019, pp. 865 – 868.

237. Khalimov G., Encryption scheme based on the automorphism group of the Ree function field / G. Khalimov, Y. Kotukh, S. Khalimova // 2020 7th International Conference on Internet of Things: Systems, Management and Security, IOTSMS 2020, 2020, 9340192.

238. McGrew D., Hash-Based Signatures draft-mcgrew-hash-sigs-00 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступу: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-00

239. McGrew D., Curcio M. Hash-Based Signatures draft-mcgrew-hash-sigs-01 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступу: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-01

240. McGrew D., Curcio M. Hash-Based Signatures draft-mcgrew-hash-sigs-03 [Електронний ресурс] / D. McGrew, M. Curcio - Режим доступу: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-03

241. Classic McEliece: conservative code-based cryptography [Електронний ресурс]. Режим доступу: https://classic.mceliece.org/nist/mceliece-20201010.pd

242. Інтернет-ресурс. – Goole.

243. Інтернет-ресурс. Режим доступу http://www.win.tue.nl/diamant/symposium05/abstracts/wolf.pdf

244. Матеріали V міжнар. наук.-техн. конф. «Захист інформації і безпеки інформаційних систем» / [Потій О.В, Горбенко Ю.І., Ганзя Р.С., Пономар В.І.] // Львів, 2016, 02.06 – 03.06. С. 52.

245. Reinier Broker. Constructing supersingular elliptic curves // J. Comb. Number Theory, (3): pp. 269–273, 2009.

246. Єсін В.І. Безпека інформаційних систем і технологій / В.І. Єсін, О.О. Кузнецов, Л.С. Сорока.- Харків: ХНУ ім. В.Н. Каразіна, 2013.- 632 с.

247. Electronic signature mechanisms. The Current State, the Existing Contradictions and Prospects of Practical Use for the Post-Quantum Period / [I. Gorbenko, A. Kuznetsov, Yu. Gorbenko, S. Kavun, O. Kachko, M. Yesina] // ASC Academic Publishing Minden, Nevada, USA, 2017. 165 p.

248. Report on Post-Quantum Cryptography [Електронний ресурс]. https://csrc.nist.gov/publications/detail/nistir/8105/final

249. Daniel J. Bernstein Johannes Buchmann Erik Dahmen. Post-Quantum Cryptography. [Електронний ресурс]. Режим доступу: https://www.researchgate.net/profile/Nicolas_Sendrier/publication/226115302_Code-Based_ Cryptography/links/540d62d50cf2df04e7549388/Code-Based-Cryptography.pdf.

250. Menezes J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. Boca Raton, Florida. CRC Press. 1997. 816 p.

251. Katz, Jonathan; Lindell, Yehuda. Introduction to Modern Cryptography: Principles and Protocols // Chapman and Hall/CRC, 2007. 552 pages.

252. FIPS PUB 180-4, Secure Hash Standard (SHS) [Електронний ресурс]. Режим доступу: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

253. BIKE: Bit Flipping Key Encapsulation [Електронний ресурс]. Режим доступу: https://bikesuite.org/files/v4.1/BIKE_Spec.2020.10.22.1.pdf

254. Hamming Quasi-Cyclic (HQC) [Електронний ресурс]. Режим доступу: http://pqc-hqc.org/doc/hqc-specification_2020-10-01.pdf

255. SUPERCOP [Електронний ресурс]. Режим доступу: https://bench.cr.yp.to/supercop.html

256. Léo Ducas. Crystals-Dilithium: Algorithm Specifications and Suppor ting Documentation. Round 3 Submissions. [Електронний ресурс]. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.

257. Gorjan Alagic. NISTIR 8309 Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process / [Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone]. Режим доступу: https://doi.org/10.6028/NIST.IR.8309.

258. Chen L. Report on Post-Quantum Cryptography / Chen L, Jordan S, Liu Y-K, Moody D, Peralta R, Perlner RA, Smith-Tone D // (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8105 (2016). Режим доступу: https://doi.org/10.6028/NIST.IR.8105.

259. Єсіна М.В. Аналіз та дослідження алгоритму цифрового підпису PICNIC / М.В. Єсіна, Б.С. Шахов.- Радіотехніка. 2020. Вип. 203. С. 19 - 24. DOI:10.30837/rt.2020.4.203.02

260. Генерація загальносистемних параметрів для криптосистеми Falcon для 256, 384, 512 біт безпеки / [І. Д. Горбенко, С.О. Кандій, М.В. Єсіна, Є.В. Острянська] // Радіотехніка. 2020. Вип. 202. С. 57-63.

261. Методи обчислення системних параметрів для електронного підпису «Crystals-Dilithium» 128, 256, 384 та 512 біт рівнів безпеки / [І.Д. Горбенко, А.М. Олексійчук, О.Г. Качко, Ю.І. Горбенко, М.В. Єсіна, С.О. Кандій] // Радіотехніка. 2020. Вип. 202. С. 5-28.

262. Comparative Analysis of Key Encapsulation Mechanisms / [Maryna Yesina, Mikolaj Karpinski, Volodymyr Ponomar, Yurij Gorbenko, Tomasz Gancarczyk, Uliana Iatsykovska] // Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems:

Technology and Applications (IDAACS). September 18-21, 2019, Metz, France. Volume 1. – P. 7-12.

263. Моделі загроз щодо асиметричних криптоперетворень перспективного електронного підпису / [Ю.І. Горбенко, М.В. Єсіна, В.В. Онопрієнко, Г.А. Малєєва] // Радіотехніка. 2020. Вип. 202. С. 72-78.

264. Горбенко Ю.І. Аналіз стійкості постквантового електронного підпису Dilithium до атак на помилки / Ю.І. Горбенко, О.С. Дроздова // Радіотехника. 2020. Вип. 202. С. 49 – 56.

265. Daniel J. Bernstein, Tanja Lange, Christiane Peters. Attacking and defending the McEliece cryptosystem. [Електронний ресурс]. Режим доступу: https://link.springer.com/chapter/10.1007/978-3-540-88403-3_3.

266. Martin Albrecht, Shi Bai, Leo Ducas A. Subfield Lattice Attack on Overstretched NTRU Assumptions. In Matthew Robshaw and Jonathan Katz, editors, Advances in Cryptology – CRYPTO 2016, pages 153–178, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

267. Paul Kirchner, Pierre-Alain. Fouque Revisiting Lattice Attacks on Overstretched NTRU Parameters. In Jean-Sebastien Coron and Jesper Buus Nielsen, editors, Advancesin Cryptology – EUROCRYPT 2017, pages 3-26, Cham, 2017. Springer International Publishing.

268. Vadim Lyubashevsky, Daniel Wichs. Simple lattice trapdoor sampling from a broad class of distributions. In Jonathan Katz, editor, PKC 2015, volume 9020 of LNCS, pages 716 – 730, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.

269. On the complexity of the BKW algorithm on LWE / [Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, Ludovic Perret] // Designs, Codes and Cryptography, 74: 325-354, 2015.

270. Ronald Cramer, Léo Ducas, Benjamin Wesolowski Short stickelberger class relations and application to ideal-SVP. In Coron and Nielsen, pages 324-348.

271. Avrim Blum, Adam Kalai, Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM, 50(4): 506-519, July 2003.

272. Каптьол Є.Ю. Аналіз можливостей та особливості програмування задач криптології на квантовому комп'ютері /Є.Ю. Каптьол, І.Д. Горбенко // Радіотехніка. 2020. Вип. 202. С. 37-48.

273. Vadim Lyubashevsky. CRYSTALS-Dilithium. Submission to the NIST Post-Quantum Cryptography Standardization [NIS] / Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint.- Journal Submission to the NIST Post-Quantum Cryptography Standardization [NIS].-2017.

274. Обґрунтування перспективного постквантового національного стандарту електронного підпису на основі решіток / [А.М. Олексійчук, В.А. Кулібаба, М.В. Єсіна, С.О. Кандій, Є.В. Острянська, І.Д. Горбенко] // Радіотехніка. 2020. Вип. 200. С. 5-14.

275. Оптимізація алгоритму множення поліномів для NTRU– побітних алгоритмів / [О.Г. Качко, Ю.І Горбенко, В. А. Пономар, М.В. Єсіна, С.О. Кандій] // Радіотехніка. 2020. Вип. 200. С. 15-24.

276. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, CRYPTO 2007, volume 4622 of LNCS, pages 150–169, Santa Barbara, CA, USA, August 19-23, 2007. Springer, Heidelberg, Germany.

277. Report on Post-Quatum Cryptography. NISTIR 8105 (DRAFT) / [Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone] // Access mode: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.

278. Аналіз проблем криптографічного захисту інформації у постквантовий період та можливі шляхи їх вирішення / [Горбенко І.Д., Кузнєцов О.О., Олійников Р.В., Горбенко Ю.І., Ганзя Р.С., Пономар В.А.] // V Міжнар. Наук.-техн. конф. “Захист інформації і безпека інформаційних систем” : Праці Наук.-техн. конф., 02–03 червня 2016 р. Львів : Нац. ун-т “Львівська політехніка”, 2016. С. 110-111.

279. ETSI Quantum safe cryptography and security / White Paper №8, 2015. Режим доступу: https://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf.

280. NIST Post-Quantum Cryptography Standardization. National Institute of Standards and Technology Internal, Report 8105 [Електронний ресурс] // NIST- Режим доступу: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

281. Gorbenko I., Ponomar V. Examining a possibility to use and the benefits of post-quantum algorithms dependent on the conditions of their application // Eastern-European Journal of Enterprise Technologies. 2017. Vol. 2 NO 9 (86). P.21–32. Available at: http://journals.uran.ua/

282. EUF-CMA and SUF-CMA. [Електронний ресурс]. Режим доступу: https://blog.cryptographyengineering.com/euf-cma-and-suf-cma/

283. IT-Grundschutz Catalogues. [Електронний ресурс]. Режим доступу: https://www.bsi.bund.de/EN/Topics/ITGrundschutz/ITGrundschutzCatalogues/itgrundschutzcatalogues_node.html

284. Efficient FPGA Implementations of LowMC and Picnic / [Daniel Kales, Sebastian Ramacher, Christian Rechberger, Roman Walch, Mario Werner]// Режим доступу: https://eprint.iacr.org/2019/1368.pdf.

285. The Picnic Signature Scheme. Design Document / [Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Greg Zaverucha] // November 29, 2017. Version 1.0. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-1-submissions.

286. The Picnic Signature Scheme Design Document / [Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, Greg Zaverucha] // March 30, 2019. Version 2.0. Режим доступу: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions.

287. NIST submissions. Picnic. Picnic-FS. Picnic-UR. Non-interactive Proof of Knowledge. Електронний ресурс. Режим доступу: https://pqc-wiki.fau.edu/w/Special:DatabaseHome.

288. Itai Dinur. The Picnic Post-Quantum Signature Scheme and its Security nalysis. Режим доступу: https://www.cs.technion.ac.il/~biham/Workshops/Cryptoday/2018/Slides/cryptoday-2018-itai-dinur-picnic.pdf.

289. CRYSTALS-Dilithium. Algorithm Specifications and Supporting Documentation. Access mode: https://pq-crystals.org/dilithium/data/dilithium-specification-round2.pdf

290. Відео-конференція NIST. Режим доступу: https://icmconference.org/?page_id=14324

291. Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of CheckingCryptographic Protocols for Faults (Extended Abstract). InEUROCRYPT, Lecture Notesin Computer Science, pages 37–51. Springer, 1997.

292. Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In CRYPTO, Lecture Notes in Computer Science, pages 513–525. Springer, 1997.

293. The sorcerer’s apprentice guide to fault attacks / H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan // Proceedings of the IEEE, vol. 94, no. 2, pp. 370–382, 2006.

294. Flipping bits in memorywithout accessing them: An experimental study of dram disturbance errors / [Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu] // SIGARCH Comput. Archit. News, vol. 42,no. 3, pp. 361–372, Jun. 2014. Access mode: http://doi.acm.org/10.1145/2678373.2665726

295. Leon Groot Bruinderink and Peter Pessl. Differential Fault Attacks on Deterministic Lattice Signatures // Access mode: https://eprint.iacr.org/2018/355.pdf

296. Vadim Lyubashevsky, Chris Peikert and Oded Regev. A Toolkit for Ring-LWE Cryptography / EUROCRYPT, volume 7881 of Lecture Notes in Computer Science, pages 35–54. Springer, 2013.

297. Number "Not Used" Once – Practical fault attack on pqm4 implementations of NIST candidates / [Prasanna Ravi1, Debapriya Basu Roy, Shivam Bhasin, Anupam Chattopadhyay, and Debdeep Mukhopadhyay] // Access mode: https://eprint.iacr.org/2018/211.pdf

298. Christopher Ambrose, Joppe W. Bos, Björn Fay, Marc Joye, Manfred Lochter, and Bruce Murray. Differential Attacks on Deterministic Signatures // CT-RSA, volume 10808 of LNCS, pages 339–353. Springer, 2018.

299. CRYSTALS-Dilithium. Submission to the NIST Post-Quantum Cryptography Standardization [NIS], 2017/[Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler and Damien Stehlé] // Access mode: https://pq-crystals.org/dilithium.

300. Hermann Seuschek, Johann Heyszl, and Fabrizio De Santis. A Cautionary Note: Side-Channel Leakage Implications of Deterministic Signature Schemes // CS2@HiPEAC, pages 7–12. ACM, 2016.

301. Breaking Ed25519 in WolfSSL/[Niels Samwel, Lejla Batina, Guido Bertoni, Joan Daemen, and Ruggero Susella] // CT-RSA, volume 10808 of Lecture Notes in Computer Science, pages 1–20. Springer, 2018.

302. James Howe, Ayesha Khalidy, Marco Martinoli, Francesco Regazzoniz and Elisabeth Oswald. Fault Attack Countermeasures for Error Samplers // Lattice-Based Cryptography. Access mode: https://eprint.iacr.org/2019/206.pdf

303. PQC Standardization Process: Third Round Candidate Announcement. July 22, 2020. [Electronic resource]. Access mode: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

304. NIST Post-Quantum Cryptography – A Hardware Evaluation Study / [Kanad Basu, Deepraj Soni, Mohammed Nabeel, Ramesh Karri] // Access mode: https://eprint.iacr.org/2019/047.pdf

305. K. Gaj. Comprehensive comparison of hardware performance of fourteen round 2 SHA-3 candidates with 512-bit outputs using field programmable gate arrays / K. Gaj, E. Homsirikamol, M. Rogawski // 2nd SHA-3 Candidate Conference, Santa Barbara, August, pp. 23–24, 2010.

306. Report on the development of the advanced encryption standard (AES) / [J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback] // 2001. Access mode: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4863838/.

307. K. Aoki. Fast implementations of AES candidates / K. Aoki, H. Lipmaa // AES Candidate Conference, pp. 106–120, 2000.

308. Crystals-Dilithium: Digital signatures from module lattices / [L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehle] // Access mode: https://pq-crystals.org/dilithium/index.shtml, 2018.

309. From 5-pass MQ-based identification to MQ-based signatures / [Ming-Shing Chen, A. Hulsing, J. Rijneveld, S. Samardjiska, P. Schwabe] // Access mode: https://eprint.iacr.org/2016/708.pdf.

310. The Lattice-Based Digital Signature Scheme qTesla / [Erdem Alkim, Paulo S., L. M. Barreto, Nina Bindel, Juliane Kramer, Patrick Longa, Jefferson E. Ricardini] // Access mode: https://eprint.iacr.org/2019/085.pdf.

311. Reinier Broker. Constructing supersingular elliptic curves. J. Comb. Number Theory, (3): pp. 269–273, 2009.

312. Steven D. Galbraith. Constructing isogenies between elliptic curves over Finite Fields. LMS J. Comput. Math,. 2: pp. 118–138 (electronic), 1999.

313. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, PQCrypto, volume 7071 of Lecture Notes in Computer Science, pp. 19–34. Springer, 2011.

314. Jacques Velu. Isogenies entre courbes elliptiques. – C. R. Acad. Sci. Paris Ser. A-B,– 273: A238 – A241, 1971.

315. David Jao, Vladimir Soukharev. Isogeny-Based Quantum-Resistant Undeniable Signatures. PQCrypto 2014: pp. 160-179.

316. O. Ahmadi, and R. Granger. On isogeny classes of Edwards curves over nite elds, J. Number Theory, 132 (6), pp. 1337-1358, (2011).

317. Post-Quantum Cryptography – Project Overview (2016) / Electronic resource. Access mode: https://csrc.nist.gov/projects/post-quantum-cryptography

318. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8240 / [Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, Yi-Kai Liu]// Electronic resource. Access mode: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf

319. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NISTIR 8309 / [Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, Yi-Kai Liu] // Electronic resource. Access mode: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

320. National Institute of Standards and Technology Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process / Electronic resource. Access mode: https://csrc.nist.gov/CSRC/media/Projects/Post-QuantumCryptogra-phy/documents/call-for-proposals-final-dec-2016.pdf

321. Algorithms of asymmetric encryptation and encapsulation of keys of post-quantum period of 5-7 levels of stability and their applications / [Gorbenko I.D., Kachko O.G, Alekseychuk A.N., Kuznetsov O.O., Gorbenko Yu.I., Onoprienko V.V., Yesina M.V., Candii S.O.] // Radiotekhnika. 2019. Is. 198. P. 5-18. DOI:10.30837/rt.2019.3.198.01.

322. Gorbenko I.D., Kachko O.G., Esina M.V. General statements and analysis of the end-to-end encryption algorithm NTRU Prime IIT Ukraine // Radiotekhnika. Kharkov : KNURE, 2018. Is. 193. P. 5-16.

323. Calculation of general parameters for NTRU Prime Ukraine of 6-7 levels of stability / [Gorbenko I. D., Alekseychuk A.N., Kachko O.G., Yesina M.V.,Stelnik I.V., Kandy S.O., Bobukh V. A., Ponomar V.A.] // Telecommunications and Radio Engineering. 2019. Vol. 78, Is. 4. P. 327-340. DOI: 10.1615/TelecomRadEng.v78.i4.40.

324. Methods of building general parameters and keys for NTRU Prime Ukraine of 5th–7th levels of stability. Product form /[Gorbenko I.D., Kachko O.G., Gorbenko Yu.I., Stelnik I.V., Kandyi S.O., Yesina M.V.] // Telecommunications and Radio Engineering. 2019. Vol. 78, Is. 7 P. 579-594. DOI: 10.1615/TelecomRadEng.v78.i7.30.98.

325. Asymmetric encryption algorithm optimization based on using NTRU Prime mathematics / [Kachko O.G, Gorbenko Yu.I., Yesina M.V, Akolzina O.] // Radiotekhnika. 2017. Issue 191. P. 5-10.

326. DSTU 8961:2019 Information Technology. Cryptographic information protection. Asymmetric encryption and key encapsulation algorithms.

327. DSTU 7564:2014 Information Technology. Cryptographic information protection. Hashing function.

328. DSTU 8845:2019 Information Technology. Cryptographic information protection. Symmetric flow transformation algorithm.

329. Choosing Parameters for NTRUEncrypt / [J. Horstein, J.Pipher, J.Schanck, J.Silverman, W. Whyte, Z. Zhang] // https://eprint.iacr.org/2015/708.pdf

330. Kulibaba V.A. Processes and Methods for Selectins System-Wide Parameters and Analysis of Resistance Against Third-Party Channel Attacks for the Key Encapsulation Mechanism DSTU 8961:2019.- Radiotekhnika. 2021. Issue 205. P. 71-78.

331. Nick Howgrave Graham NTRU Cryptosystems Technical Report. Report #4, Version 2. A Meet-In-TheMiddle Attack on an NTRU Private key / Nick Howgrave Graham, Joseph H. Silverman, William Whyte.- NTRU Cryptosystems, 5 Burlington Woods, Burlington, MA 02144.

332. Bos J.W., Costello C., Ducas L. [et all]. Frodo: take of the ring! Practical, quantum-secure key exchange from LWE // Proc. of ACM CCS 16, ACM Press, Okt. 2006, P. 1006-1018.

333. Alkim E., Ducas L., Poeppelmann T., Schwabe P. Post-quantum key exchange – a new hope / http://cryptojedi.org/papers/#newhope, 2016.

334. Ducas L., Lepoint T., Lyubachevsky V. [et all] CRYSTALS – Delithium: digital sygnatures from module lattices / https://cryptojedi.org/papers/delithium- 20170617.pdf.

335. Kiltz E., Lyubachevsky V., Schaffner C. A concrete treatment of Fiat-Naor signatures in the quantum random-oracle model // Cryptology ePrint Archive, Report 2017/916, http://eprint.iacr.org/2017/916.

ПРЕДМЕТНИЙ ПОКАЖЧИК

Bra вектор, 71
Ket вектор, 71
NTRU-решітка, 305, 491
HB-криптографія, 317
CB-криптографія, 318
LB-криптографія, 320
MQ-криптографія, 321
EUF - CMA безпека, 310, 510, 524
SUF- CMA безпека, 370, 512

А
Алгоритм Дойча, 51, 230
Апорія (парадокс) Зенона, 65
Алгоритм Шора, 233, 255
Алгоритм Гровера, 237, 259
Алгоритм RAINBOW, 298, 385
Атаки на RAINBOW, 299, 441
Алгоритм Ванга, 304, 309
Алгоритм ЦП CRYSTALS-DILITHIUM, 363, 393, 500, 527
Алгоритм ЕП FALCON, 363, 495
Алгоритм PICNIC, 385, 519
Атака Decision-LWE, 371
Атака Search-LWE, 371
Атаки на основі решіток, 341, 373
Алгоритм Бабаї, 373
Алгоритм блочної редукції Коркіна – Золотарьова (BKZ), 374, 502
Алгоритм реалізації Primal Attack, 377
Алгоритм реалізації Dual Attack, 378
Алгоритм інкапсуляції ключів Classic MCELIECE, 443
Арифметика Монтгомери, 394, 491
Алгоритм цифрового підпису Picnic, 520
Алгоритми на основі алгебраїчних решіток (qTesla та Crystal-Dilitium), 373, 535
Алгоритм на основі багатовимірної криптографії (MQDSS), 535
Алгоритми PQC, 362, 541

Б
Блоха сфера, 14, 102
Белла стан, 48, 73
Бозони, 75
Бозе-Ейнштейнівський конденсат, 148, 167
Блочні шифри LowMC, 522
Багатокористувацький EUF-CMA, 523

В
Вектор, 13
Відстань Хеммінга, 155
Випадкова модель оракула, 448, 522

Г
Гамільтоніан, 15, 88, 125
Гільбертовий простір,17,
Гейзенберга невизначеність, 87, 166, 213
Гейт Адамара, 106

Д
Декогеренція, 72, 135, 162, 173
Дерева Мерклі, 427
Дерево сигнатур Мерклі, 429
Двійковий код Гоппи, 442

Е
Ермітово-пов’язаний вектор, 13, 32
Ентропія, 58, 363
Ентропія фон-Неймана, 59
Ефект Комптона, 68
Ефект Керра, 134
Ефект Рамана, 149
ЕПР-парадокс, 47, 160, 219
Екзистенційна підробка, 352, 360
ЕЦП на основі кодів, 423

З
Зверхущільнене кодування, 54
Задачі SIS та LWE, 372, 484, 518
Засліплююча ізогенія, 556

І
Ізогенії еліптичних кривих, 317, 326, 554
Ізогенії на кривих Едвардса, 552
Ін'єктивна функція, 431

К
Кубіт, 13,100
Кронекерів добуток матриць, 45
Квантова телепортація, 58, 145, 225
Квантовий гейт $\widehat{NOT}$, 103
Квантовий гейт Адамара, 106
Квантовий гейт Тоффолі, 110, 204
Квантовий алгоритм пошуку Гровера, 119, 205, 261
Квантовий ефект Зенона, 160, 171
Квантове перетворення Фур'є, 240, 258
Код Стена, 163
Квантовий логічний вентиль, 173
Кудит, 175
Квантовий захист інформації, 175, 184
Код Гоппи, 318, 405
Код Голея, 414
Криптосистема Нідеррайтера, 195, 319, 411
Криптосистема Мак-Еліса, 195, 406, 450
Keccak – сімейство sponge-функцій, 545
Криптокодові системи, 195
Криптографічна живучість, 329
Криптографія на основі функцій хешування, 195, 335
Кваліфікація порушника, 252
Колізія, 299, 479
Квантові гроші Стівена Віснера, 203
Квантова інформаційна безпека, 195
Кодові криптосистеми, 422

Л
Логічний елемент NOT, 19
Логічний елемент Z, 21
Локальна мережа Бенеша, 454
Логічний елемент Адамара H, 23
Логічний елемент Y, 24
Логічний елемент $\boldsymbol{\frac{\pi}{8}}$, 25
Логічний елемент S, 27
Логічний елемент Ф, 29
Логарифмічний підпис, 429

М
Матриця щільності, 59
Матриці Паулі, 98
Моделі порушника, 254, 327, 343, 368, 516
Моделі загроз, 327, 368, 517
Метод Мінранка, 439
Модель випадкового оракула (ROM), 366, 371, 444
Модель квантового випадкового оракула (QROM), 367
Модель випадкового оракула із квантовим доступом (ВОКД), 334
Модель безпеки KDS-CMA, 514
Модульне навчання з помилками (MLWE), 527
Модульне коротке ціле рішення (MSIS), 527

Н
Нерівності Белла, 97
Навчання з помилками (LWE), 371
Національний стандарт ДСТУ 8961:2019, 557

О
Оператор Гамільтона, 88, 124
Оператор клонування, 114
Оракул, 262, 287
Оператор дифузії, 263
Одностороння функція з таємним входом, 445

П
Підпис Лампорта (L-OTS), 346
Підпис Вінтерніца, 347
Постійна Планка, 68
Переплутані стани (базисні стани Белла), 73, 96, 151, 217
Принцип Паулі, 75
Постійна Рідберга, 83
Принцип додатковості Бора, 93
Принцип суперпозиції квантової системи, 215
Протокол BB84, 178
Протокол Екерта, 180
Протокол B92, 182
Протокол E91, 192
Поляризація фотонів, 141, 192
Постквантова криптографія, 195
Підпис з публічним ключем, 195
Паралельні обчислення, 114
Початковий стан (нонс), 373, 517, 531
Перетворення Монтгомери, 393
Поліном Oil-Vinegar, 437
Постквантовий ЕП Dilithium на алгебраїчній решітці, 471, 500

Р
Решіткові системи, 195, 321, 326
Редукція базису решітки, 321
Розширений алгоритм Евкліда, 242, 492
Рівні стійкості та безпеки алгоритмів PQC, 544

С
Спін, 13, 74, 84, 97, 222
Середовище Керра, 134
Синдром помилки, 157, 419
Синдром декодування, 318
Селективна підробка, 337, 338
Стандарти “Вершина 1” та “Вершина 2”, 490, 505, 508
Сімейство блочних шифрів LowMC, 522
Структура Rijndael, 428
Стійкість "Купини - 256", 428
Схема Rainbow, 436

Т
Теорема про заборону клонування невідомого стану, 175
Теорема Белла, 222
Теорія ігор, 342, 513
Трійка поліноміальних Алгоритмів $S = (K, S, V)$, 513

У
Уявний експеримент Галілея, 66

Ф
Фотон, 13, 68
Ферміони, 75
Фактор Ерміта, 375
Функція розширюваного виводу SHAKE 256, 391, 527
Формула Вейєрштрасса, 549
Формула Велу, 549

Х
Хвильова функція, 11, 72

Підготовка до друку ТОВ «Тропеа».
Формат 70х100 1/16.
Замовлення № 059/25.
Віддруковано згідно з наданим оригінал-макетом
ТОВ «Тропеа».
Наклад 130 прим.