Міністерство інфраструктури України
Державна служба зв'язку
Одеська національна академія зв'язку ім. О. С. Попова

**М. В. Захарченко, О. В. Онацький,
Л. Г. Йона, Т. М. Шинкарчук**

АСИМЕТРИЧНІ МЕТОДИ ШИФРУВАННЯ В ТЕЛЕКОМУНІКАЦІЯХ

Рекомендовано Міністерством освіти і науки, молоді та спорту України як навчальний посібник для студентів вищих навчальних закладів

**Захист інформації
в телекомунікаційних системах та мережах**

**Модуль 2 – Криптографічні методи захисту інформації
в телекомунікаційних системах та мережах**

Навчальний посібник
за напрямами підготовки студентів
6.050903 – Телекомунікації
6.170101 – Безпека інформаційних і комунікаційних систем
6.170102 – Системи технічного захисту інформації
6.050901 – Радіотехніка
6.090504 – Мережі та системи поштового зв'язку

Одеса 2011

УДК 004.056.55:621.39(075)
ББК 32.882
А90

Гриф надано Міністерством освіти і науки
молоді та спорту України
(лист № 1/11-8780 від 22.09.2011 р.)

Р е ц е н з е н т и :

Кошевий В. М., д.т.н., професор, Одеська національна морська академія, завідуючий кафедри морського радіозв'язку;

Баранов П. Ю., д.т.н., професор, директор Інституту радіоелектроніки та телекомунікацій, Одеського національного політехнічного університету.

Захарченко М. В. Асиметричні методи шифрування в телекомунікаціях: навч. посіб. / М. В. Захарченко, О. В. Онацький, Л. Г. Йона, Т. М. Шинкарчук. – Одеса: ОНАЗ ім. О. С. Попова, 2011. – 184 с. – (Криптографічні методи захисту інформації в телекомунікаційних системах та мережах: модуль 2 з дисципліни „Захист інформації в телекомунікаційних системах та мережах").

ISBN 978-966-7598-71-6

Розглянуто математичні основи теорії чисел та основні вимоги до геш-функцій, забезпечуючих мінімізацію мережного трафіку та надлишковість відкритого тексту при криптографічному перетворенні, проведено аналіз сучасних способів організації секретного зв'язку без попереднього обміну ключами алгоритму електронно-цифрового підпису.

Теоретичний матеріал відповідає навчальним програмам „Захист інформації в телекомунікаційних системах та мережах", „Криптографія та стеганографія" і супроводжується достатньою кількістю наведених типових прикладів, контрольних запитань та задач, забезпечуючих самоперевірку засвоєння матеріалу.

Розраховано для студентів-бакалаврів за напрямами: телекомунікації; безпека інформаційних і комунікаційних систем; системи технічного захисту інформації; радіотехніка; мережі та системи поштового зв'язку.

ISBN 978-966-7598-71-6

ЗМІСТ

ПЕРЕДМОВА

Дисципліну Н3.16 – захист інформації в телекомунікаційних системах та мережах – включено до навчального плану освітньо-професійної програми підготовки бакалаврів з напряму 6.050903 телекомунікації.

Мета навчальної дисципліни – формування у студентів базових знань з проблеми захисту інформаційних ресурсів у системах телекомунікацій і мережах від порушення її конфіденційності, цілісності та доступності.

Курс базується переважно на дисциплінах: Н2.01 – вища математика; Н3.01 – теорія електричних кіл та сигналів; Н3.03 – основи схемотехніки; Н3.04 – обчислювальна техніка та мікропроцесори; Н3.02 – теорія електричного зв'язку; Н3.14 – системи документального електрозв'язку; Н3.11 – телекомунікаційні та інформаційні мережі, з яких студенти повинні знати: теорію ймовірності та математичної статистики; елементи дискретної математики та комбінаторики; види первинних сигналів електрозв'язку та їхній математичний опис; амплітудно-частотні та фазові спектри; частотні характеристики електричних кіл; аналогові та цифрові компоненти; засади побудови мікропроцесорних систем; основні положення теорії інформації; основні характеристики ліній передавання; характеристики і загальні засади функціонування систем комутації та систем передавання електрозв'язку.

Дисципліна складається з двох модулів:

модуль 1 – Архітектура систем захисту інформації: основи законодавчої та нормативно-правової бази України; системний підхід до розв'язування проблеми захисту інформації; загрози щодо інформації в телекомунікаційних системах; критерії інформаційної безпеки (лекцій – 16 год.; практичних занять – 8 год.; лабораторних робіт – 8 год.; самостійна робота – 16 год.; всього – 48 год.);

модуль 2 – Криптографічні методи захисту інформації в телекомунікаційних системах та мережах: типи та класифікація алгоритмів шифрування; загальні засади побудови симетричних та асиметричних криптосистем; електронний цифровий підпис, вирішення проблем автентифікації в телекомунікаційних системах (лекцій – 16 год.; практичних занять – 8 год.; лабораторних робіт – 8 год.; самостійна робота – 24 год.; всього – 56 год.).

Структура модуля 2

№	Тематика та зміст лекційного курсу	Літера-тура
1	Захист інформації при передаванні каналами зв'язку. Термінологія. Типи та класифікація алгоритмів шифрування. Шифри заміни та переставляння. Засада шифрування за методом Вермана. Умови стійкості шифрів	Л1, Л2, Л3, Л4
2	Криптосистеми з відкритим ключем. Поняття односпрямованої функції, односпрямованої функції з секретом. Модель криптосистеми з відкритим ключем. Криптоалгоритм Меркле–Хеллмана	Л1, Л2, Л3, Л4
3	Система Idempotent Elements. Шифр Шаміра. Стандарти асиметричного шифрування RSA, Рабіна, Ель–Гамаля	Л1, Л2, Л3, Л4
4	Стійкість RSA. Односпрямовані геш-функції. Алгоритм цифрового підпису RSA. Засади керування ключовою системою. Генерування, зберігання і розподіл ключів. Алгоритм Діффі–Хеллмана	Л1, Л2, Л3, Л4
5	Загальні засади побудови симетричних криптосистем. Математичні операції, використані в симетричних крипто-системах. Основи архітектури сучасних симетричних крипто-систем. Блокові алгоритми шифрування. Шифри на базі мережі Фейстеля	Л1, Л2, Л3, Л4
6	Криптосистема DES. Схема алгоритму шифрування DES. Режими роботи алгоритму DES	Л1, Л2, Л3, Л4
7	Стандарт шифрування ГОСТ 28147–89. Схема алгоритму шифрування за ГОСТом 28147–89. Режими роботи стандарту шифрування за ГОСТом 28147–89	Л1, Л2, Л3, Л4
8	Стандарт шифрування IDEA. Реалізація мережі Фейстеля у стандарті шифрування IDEA	Л1, Л2, Л3, Л4

Література

1 **Захарченко Н. В.** Развитие криптографии и ее место в современном обществе. Ч. 1. Классические методы шифрования: учеб. пособ. / Н. В. Захарченко, Л. Г. Йона, Ю. В. Щербина, А. В. Онацкий. – Одесса: ОНАС им. А. С. Попова, 2003. – 95 с.

2 **Кисель В. А.** Основы криптографии: учеб. пособ. / В. А. Кисель, Н. В. Захарченко. – Одесса: УГАС им. А. С. Попова, 1997. – 48 с.

3 **Защита информации** в системах телекоммуникации: учеб. пособ.; под ред. В. Л. Банкета. – Одесса: УГАС им. А. С. Попова, 1997. – 96 с.

4 **Горохов С. М.** Сучасні криптографічні системи: навч. посіб. / С. М. Горохов, Л. Г. Йона, О. В. Онацький; за ред. М. В. Захарченка. – Одеса: ОНАЗ ім. О. С. Попова, 2007. – 152 с.

Перелік практичних занять модуля 2

Теми занять	Кількість годин
Елементи теорії чисел. Розв'язування задач на криптоалгоритм Меркле–Хеллмана та систему Idempotent Elements	2
Розв'язування завдань на алгоритм Шаміра, RSA, Рабіна, Ель–Гамаля, Діффі–Хеллмана	2
Вивчення криптосистеми DES. Режими роботи криптосистеми DES	2
Вивчення криптосистеми за ГОСТом 28147–89. Режими роботи за ГОСТом 28147–89	2

Перелік лабораторних занять модуля 2

Найменування лабораторних робіт	Кількість годин
Дослідження шифру „Подвійний квадрат"	2
Дослідження асиметричного алгоритму шифрування RSA	2
Дослідження алгоритму генерації ключа методом Діффі–Хеллмана	2
Дослідження алгоритму створювання цифрового підпису на підставі RSA	2

Перелік знань та вмінь, яких має набути студент протягом вивчення матеріалу

Вміти використовувати нормативно-правову базу України у сфері захисту інформації, положення, інструкції, технічну документацію і рекомендувати заходи щодо перекривання можливих каналів втрати інформації в телекомунікаційних системах та мережах.

Під керівництвом провідного фахівця виконувати обчислення потрібних параметрів систем технічного захисту інформації в системах та мережах зв'язку.

ВСТУП

Невпинно зростає різноманіття і складність проблем інформаційної безпеки, які виникають в процесі розвитку інформаційних технологій. Сучасні вирішення багатьох проблем захисту інформації неможливо уявити без використовування криптографічних методів.

Серед численних проблем забезпечення інформаційної безпеки, вирішуваних за допомогою криптографічних методів та засобів, завдання забезпечення цілісності та вірогідності передаваної інформації є на сьогодні одним з найактуальніших. З урахуванням сучасних вимог щодо інформаційно-телекомунікаційних систем – це завдання все частіше перетворюється на серйозну проблему. Надто актуальною вона є у фінансовій сфері, оскільки задля функціонування електронної платіжної системи неодмінною умовою є зберігання цілісності та вірогідності всіх документів.

Асиметричну криптографію винайдено в семидесятих роках минулого сторіччя. Останніми трьома десятиріччями вона набула широкого розвитку і посіла майже таке саме місце, що й блочне симетричне шифрування. Асиметричне шифрування, або шифрування на відкритому ключі, ґрунтується на докорінно відмінній ідеології, і впевнено посіло власну нішу серед систем захисту інформації.

Концепцію криптографії з відкритим ключем було запропоновано Уїтфілдом Діффі (Whitfield Diffie)[1] та Мартіном Хеллманом (Martin Hellman)[2] і, незалежно, Ральфом Меркле (Ralph Merkle). Ще в сорокових роках минулого століття Клод Шеннон (Claude Shannon)[3] запропонував будувати шифр у такий спосіб, щоби завдання щодо його зламу було еквівалентне до певної математичної задачі, яка потребує недосяжного для сучасних комп'ютерів обсягу обчислень. Вперше ідею К. Шеннона представили Діффі та Хеллман на Національній комп'ютерній конференції (National Computer Conference) у Нью-Йорку 1976 року, і кілька місяців потому у пресі з'явилася їхня головна робота „New Directions in Cryptography" (Нові напрями у криптографії). Ця робота не лише істотно змінила криптографію, але й призвела до появи та бурхливого розвитку нових напрямів у математиці. Вона поклала початок криптографії з відкритим ключем та теорії криптографічних протоколів.

---

[1] **Уїтфілд Діффі** (народ. 1944 р. у США). 1965 року закінчив Массачусетський технологічний інститут. До початку 70–х років минулого століття, пропрацювавши у кількох місцях (заслужений інженер компанії Sun Microsystems), став одним з кількох цілковито незалежних експертів з безпеки, вільним криптографом.

[2] **Мартін Хеллман** (народ. 1945 р. у США). У 70-х роках минулого століття став професором Стенфордського університету в Каліфорнії. 1974 року познайомився з У. Діффі, і разом вони розпочали вивчення проблеми розподілу ключів, а через певний час до них долучився **Ральф Меркле**. 1976 року Діффі, Хеллман та Меркле здійснили переворот у світі криптографії.

[3] **Клод Елвуд Шеннон** (1916–2001) – американський математик та електротехнік, один з творців математичної теорії інформації, значною мірою визначив результатами своїх досліджень розвиток загальної теорії дискретних автоматів. Клода Шеннона називають „батьком теорії інформації". Як говорив сам Шеннон, робота в області криптографії підштовхнула його до створення теорії інформації.

На сьогодні асиметричне шифрування застосовується для ідентифікації та автентифікації користувачів, захисту каналів передавання даних від нав'язування помилкових даних, захисту електронних документів від копіювання та підробки.

Один з нових напрямів криптографії з відкритими ключами – системи на еліптичних кривих. Еліптичні криві давно вивчалися в математиці, але їхнє використовування у криптографічних цілях було вперше запропоновано Кобліцем (Need Koblitz) та Міллером (Victor Miller) в 1985 року. З 1998 року використовування еліптичних кривих для розв'язку криптографічних завдань, таких як цифровий підпис, було закріплено в стандартах США ANSI X9.62 та FIPS 186–2, а 2001 року стандарт ГОСТ Р 34.10–2001 було ухвалено в Росії. 2002 року на Україні ухвалено стандарт цифрового підпису, який ґрунтується на еліптичних кривих ДСТУ 4145–2002, а 2003 року – закони „Про електронний цифровий підпис" та „Про електронні документи та електронний документообіг" (додатки Б, В).

Основна перевага криптосистем на еліптичних кривих полягає в тому, що, порівняно із „звичайними" криптосистемами, вони забезпечують набагато вищу стійкість за рівної трудомісткості або, навпаки, істотно меншу трудомісткість за рівної стійкості. Це пояснюється тим, що для обчислення обернених функцій на еліптичних кривих відомі лише алгоритми з експоненційним зростанням трудомісткості, тоді як для звичайних систем запропоновано субекспоненційні методи. Як наслідок, рівень стійкості, досяжний, скажімо, в RSA при використанні 1024-бітових модулів, у системах на еліптичних кривих зреалізовується при розмірі модуля 160 біт, що забезпечує більш просту як програмну, так і апаратну реалізацію.

Криптографічні алгоритми з відкритим ключем використовують математичний апарат з теорії чисел. Ми розглянемо необхідний мінімум з цієї теорії: класичні теореми Ферма, Ейлера, Уїлсона, алгоритм Евкліда й ряд інших означень та теорем. Читачі, знайомі з теорією чисел, можуть безпосередньо перейти до розділу 2.

1 ЕЛЕМЕНТИ ТЕОРІЇ ЧИСЕЛ

Теорія чисел займається вивченням властивостей цілих чисел. *Цілими* називаються не лише числа натурального ряду 1, 2, 3, ... (додатні цілі), але також нуль та від'ємні цілі –1, –2, –3, ...

Далі вживатиметься позначення множини {..., –3, –2, –1, 0, 1, 2, 3, ...} цілих чисел літерою *Z*. Множини цілих чисел {1, 2, 3, ...} називаються *множинами натуральних чисел* і стандартно позначаються літерою *N*.

Багато цілих чисел можна подавати як добуток менших чисел. Важливі характеристики й відношення цілих чисел може бути здобуто на підставі аналізу їхньої структури з огляду на складові множники.

1.1 Подільність

Означення Ціле число a є *кратне* до числа b, якщо $a = bm$ для певного цілого числа m. Ненульове ціле число b ділить ціле число a, що позначається як $b \mid a$, якщо a є кратне до b. Ціле число b, що ділить ціле число a, називається *дільником* числа a.

За означенням, $9 \mid 27$, оскільки $27 = 9 \cdot 3$, але 5 не ділить 12, тому що не існує цілого числа m такого, щоб $12 = 5m$. Цілі числа 1, 2, 3, 4, 6 та 12, й лише вони, є додатними дільниками числа 12. Цілі числа –1, –2, –3, –4, –6 та –12 також є дільниками числа 12.

Теорема 1 (алгоритм подільності) Для додатних цілих чисел a і b існують єдині невід'ємні числа q і r, де $0 \le r < b$ такі, що $a = bq + r$. Такі цілі числа r та q називаються, відповідно, *остачею* і *часткою* від ділення a на b.

Якщо $a < b$, тоді q має дорівнювати 0, щоб виконувалося $a = bq + r$, де $0 \le r < b$ і $q \ge 0$. Наприклад, для $a = 4$ і $b = 7$ алгоритм подільності надає $q = 0$ й $r = a = 4$, так що $4 = 7 \cdot 0 + 4$.

Через єдність q і r, якщо можна здобути q і r у якийсь інший спосіб, причому $a = bq + r$, $0 \le r < b$ і $q \ge 0$, то ці q і r мають збігатися з тими, які існують згідно з теоремою 1.

Оскільки кожний додатний дільник додатного числа n не може бути більшим за саме це число, усі додатні дільники числа n можна віднайти, перебираючи усі цілі числа від 1 до n і перевіряючи, чи не ділять вони n. Так, приміром, ми визначили, що додатним дільником числа 12 є числа 1, 2, 3, 4, 6 і 12. Так само можна довести, що додатними дільниками числа 90 є 1, 2, 3, 5, 6, 9, 10, 15, 18, 30, 45 і 90.

Перевірка доводить, що числа 1, 2, 3 і 6 є дільниками як 12, так і 90. Числа 1, 2, 3 і 6 називаються *спільними дільниками* чисел 12 і 90. Більш того, 6 є найбільший з цих спільних дільників. Звернімо увагу на те, що спільні дільники

1, 2, 3 і 6 є також дільниками найбільшого спільного дільника 6. У контексті найбільших спільних дільників розглядаються лише додатні дільники.

Означення Додатне ціле число d називається *спільним дільником* чисел a і b, якщо $d \mid a$ і $d \mid b$.

Означення Додатне ціле число d називається *найбільшим спільним дільником* чисел a і b, якщо

1) $d \mid a$ і $d \mid b$;

2) з $c \mid a$ і $c \mid b$ виходить $c \mid d$.

Найбільший спільний дільник (НСД) чисел a і b позначається через НСД (a, b).

Теорема 2 Якщо d і c – найбільші спільні дільники цілих чисел a і b, тоді $c = d$; інакше кажучи, існує єдиний спільний дільник для цілих додатних чисел a і b.

Теорема 3 Найбільший спільний дільник додатних цілих чисел a і b існує. Такий найбільший спільний дільник може бути записаний у формі

$$\text{НСД}\ (a, b) = u \cdot a + v \cdot b$$

для певних цілих чисел u і v. Окрім того, найбільший спільний дільник – це найменше додатне ціле число такої форми.

Теорема 4 Якщо $a = bq + c$, то НСД (a, b) = НСД (b, c); інакше кажучи, кожен дільник a і b є дільником b і c й навпаки.

Теорема 5 Якщо a, b і c – цілі числа, НСД (a, b) = 1 і $a \mid bc$, то $a \mid c$.

Означення Якщо найбільший спільний дільник a і b дорівнює 1, то числа a і b називаються *взаємно простими.*

Наприклад, числа 6, 10, 15 через (6, 10, 15) = 1 – взаємно прості. Числа 8, 13, 21 через (8, 13) = (8, 21) = (13, 21) = 1 – прості парами. Числа 15 і 27 не є взаємно простими.

Безпосередньо з означення виходить, що, якщо a і b – взаємно прості числа, то існують цілі числа u і v такі, що $au + bv = 1$.

Зазначимо, що якщо a і b – додатні цілі числа, то ab є кратне як до a, так і до b. Якщо розглядати безліч усіх чисел, які є кратні до a і b, то, згідно із засадою повного впорядкування, існує найменше кратне чисел a і b. Якщо c – найменше кратне чисел a і b, а d – інше кратне чисел a і b, то $c \mid d$. Інакше кажучи, існують q та r такі, що $d = qc + r$, де $r < c$. Оскільки як a, так і b ділять числа d і c, вони також ділять r. Але тоді r було б кратним до a і b, яке є менше за c, що призводить до суперечності. Отже, доходимо таких означень.

Означення Додатне ціле число m називається *спільним кратним* цілих чисел a і b, якщо $a \mid m$ і $b \mid m$.

Означення Додатне ціле число m називається *найменшим спільним кратним* цілих чисел a і b, якщо

1) $a \mid m$ і $b \mid m$;

2) з $a \mid n$ і $b \mid n$ виходить $m \mid n$.

Найменше спільне кратне (НСК) з чисел a і b позначатимемо НСК (a, b).

Вправи

1 Знайдіть додатні дільники кожного з таких чисел:

а) 54; б) 63; в) 72; г) 73; д) 74.

2 Для додатних цілих чисел a і b знайдіть невід'ємні цілі числа q і r, де $0 \le r < b$ такі, що $a = bq + r$:

а) $a = 54$, $b = 27$; б) $a = 47$, $b = 47$; в) $a = 93$, $b = 17$; г) $a = 43$, $b = 8$.

3 Для додатних цілих чисел a і b знайдіть НСД (a, b), НСК (a, b), якщо їх визначено:

а) $a = 54$, $b = 27$; б) $a = 12$, $b = 16$; в) $a = 33$, $b = 1$; г) $a = 6$, $b = 15$.

4 Доведіть, що для взаємно простих чисел a і b та певного числа n існують цілі числа x і y такі, що $ax + by = n$.

1.2 Алгоритм Евкліда

Один із способів обчислення найбільшого спільного дільника двох чисел – використання алгоритму Евкліда[4].

Теорема 6 (алгоритм Евкліда) Нехай маємо два числа – a і b; $a \ge 0$, $b \ge 0$; вважаємо, що $a > b$. Знаходимо ряд рівнянь:

$$
\begin{aligned}
a &= b\,q_1 + r_1 && 0 \le r_1 < b \\
b &= r_1\,q_2 + r_2 && 0 \le r_2 < r_1 \\
r_1 &= r_2\,q_3 + r_3 && 0 \le r_3 < r_2 \\
r_2 &= r_3\,q_4 + r_4 && 0 \le r_4 < r_3 \\
&\cdots\cdots\cdots && \cdots\cdots\cdots \\
r_{n-3} &= r_{n-2}\,q_{n-1} + r_{n-1} && 0 \le r_{n-1} < r_{n-2} \\
r_{n-2} &= r_{n-1}\,q_n + r_n && 0 \le r_n < r_{n-1} \\
r_{n-1} &= r_n\,q_{n+1} && r_{n+1} = 0
\end{aligned}
$$

який завершується, коли знаходимо певне $r_{n+1} = 0$. Тоді r_n – найбільший спільний дільник чисел a і b.

[4] **Евклід** (біля 300 р. до н. е.) – давньогрецький математик. Основна робота Евкліда називається „Початки", що містить тринадцять книг: VII–IX книги присвячено теорії чисел. Алгоритм обчислення найбільшого спільного дільника двох чисел викладено у IX книзі.

Останнє є неминуче, тому що ряд $b, r_1, r_2, \ldots$ як ряд спадних цілих не може містити понад b додатних. Маємо: $b > r_1 > r_2 > \ldots > r_n \geq 0$, отже, процес обірветься що найбільш через b кроків.

Приклад 1.1 Нехай $a = 525$, $b = 231$; знайти НСД. Використаємо алгоритм Евкліда:

$$525 = 231 \cdot 2 + 63;$$
$$231 = 63 \cdot 3 + 42;$$
$$63 = 42 \cdot 1 + 21;$$
$$42 = 21 \cdot 2.$$

Здобудемо останню додатну остачу $r_3 = 21$. Отже, НСД (525, 231) = 21.

Приклад 1.2 Нехай $a = 1234$, $b = 54$; знайти НСД.

$$1234 = 54 \cdot 22 + 46;$$
$$54 = 46 \cdot 1 + 8;$$
$$46 = 8 \cdot 5 + 6;$$
$$8 = 6 \cdot 1 + 2;$$
$$6 = 2 \cdot 3.$$

Остання ненульова остача дорівнює 2, тому НСД (1234, 54) = 2.

За допомогою алгоритму Евкліда можна знаходити числа u і v із Z (теорема 3) такі, що $r_n = au + bv$ = НСД (a, b). Насправді, з ланцюга рівностей маємо

$$r_n = r_{n-2} - r_{n-1} q_n = r_{n-2} - (r_{n-3} - r_{n-2} q_{n-1}) q_n = \ldots = au + bv = \text{НСД } (a, b)$$

(слідуємо за ланцюгом рівностей знизу нагору, вилучаючи з кожної наступної рівності остачу і підставляючи її до виразу, який здобули ще до цього моменту).

Приклад 1.3 Подати НСД (85, 34) у формі $85u + 34v$.

$$85 = 34 \cdot 2 + 17;$$
$$34 = 17 \cdot 2 + 0.$$

Отже, НСД (85, 34) = 17 і НСД (85, 34) = $17 = 85 \cdot 1 + 34(-2)$.

Приклад 1.4 Подати НСД (252, 580) у формі $252u + 580v$.

$$580 = 252 \cdot 2 + 76;$$
$$252 = 76 \cdot 3 + 24;$$
$$76 = 24 \cdot 3 + 4;$$
$$24 = 4 \cdot 6 + 0.$$

Обернене підставляння дає

$$4 = 76 - 24 \cdot 3 = 76 - [252 - 76 \cdot 3] \cdot 3 = 76 \cdot 10 + 252(-3) =$$
$$= [580 - 252 \cdot 2] \cdot 10 + 252(-3) = 580 \cdot 10 + 252(-23).$$

Приклад 1.5 Подати НСД (252, 576) у формі $252u + 576v$.

$$576 = 252 \cdot 2 + 72;$$
$$252 = 72 \cdot 3 + 36.$$

Після зворотного підставляння здобуваємо

$$36 = 252 - 72 \cdot 3 = 252 - (576 - 252 \cdot 2)\, 3 = 252 \cdot 7 + 576(-3).$$

Теорема 7 Нехай задано цілі числа a і b, які не дорівнюють нулеві, тоді числа $a / \text{НСД}(a, b)$ і $b / \text{НСД}(a, b)$ є взаємно простими, тобто

$$\text{НСД}\left(\frac{a}{\text{НСД}(a, b)}, \frac{b}{\text{НСД}(a, b)}\right) = 1.$$

Використання найбільшого спільного дільника стає дуже корисним при знаходженні розв'язків рівнянь форми $ax + by = c$.

Означення Діофантовим рівнянням першого ступеня називається рівняння форми $ax + by = c$ з цілими коефіцієнтами a, b, c, розв'язуване на безлічі цілих чисел.

Теорема 8 Рівняння $ax + by = c$, де a, b і c – цілі числа, має цілочисловий розв'язок (тобто існують цілі числа x і y такі, що $ax + by = c$) тоді й лише тоді, коли c ділиться на НСД (a, b). Якщо c ділиться на НСД (a, b), то розв'язок $ax + by = c$ має форму

$$x_0 = \frac{uc}{\text{НСД}(a, b)};\quad y_0 = \frac{vc}{\text{НСД}(a, b)},$$

де u і v – будь-які розв'язки рівняння НСД $(a, b) = au + bv$.

Приклад 1.6 Знайти розв'язок рівняння $85x + 34y = 51$.

Здобуваємо НСД(85, 34) = 17 і $85 \cdot 1 + 34(-2) = 17$. Тоді розв'язок матиме форму

$$x_0 = \frac{uc}{\text{НСД}(a, b)} = \frac{1 \cdot 51}{17} = 3;$$

$$y_0 = \frac{vc}{\text{НСД}(a, b)} = \frac{(-2) \cdot 51}{17} = -6.$$

Для перевірки обчислюємо

$$85 \cdot 3 + 34(-6) = 255 + (-204) = 51.$$

Інший спосіб побудови розв'язку полягає у безпосередньому використанні рівняння $au + bv$ = НСД (a, b). Оскільки $u = 1,\ v = -2$, то

$$a(1) + b(-2) = 17.$$

Помноживши рівняння на 3, дістанемо

$$a(3) + b(-6) = 51.$$

Зауважимо, що за $x = 5,\ y = -11$

$$85 \cdot 5 + 34(-11) = 425 + (-374) = 51.$$

Доходимо висновку, що може існувати понад одного розв'язку.

Приклад 1.7 Знайти розв'язок рівняння $252x + 580y = 20$.

Дістаємо НСД $(252, 580) = 4$ і $252(-23) + 580 \cdot 10 = 4$. Помноживши кожний доданок на 5, матимемо

$$252(-115) + 580 \cdot 50 = 20.$$

Отже, $x = -115,\ y = 50$ є розв'язком.

Теорема 9 Якщо a і b – ненульові цілі числа і (x_0, y_0) – розв'язок рівняння $ax + by = c$, тоді кожний інший розв'язок (x, y) має форму

$$x = x_0 + \frac{b}{d}t; \qquad y = y_0 - \frac{b}{d}t,$$

де t – довільне ціле число, а d = НСД (a, b).

Повертаючись до прикладів, можна записати спільний розв'язок рівняння:

$85x + 34y = 51$ має форму $x = 3 + 2t$ і $y = -6 - 5t$;
$252x + 580y = 20$ має форму $x = -115 + 145t$ і $y = 50 - 63t$.

Розглянемо приклади використовування ланцюгових дробів для розв'язування простих діофантових рівнянь та порівнянь першого ступеня.

Припустімо, що $\frac{P_k}{Q_k}$ – останній відповідний дріб у поданні ланцюгового дробу раціонального числа $\frac{a}{b}$, де НСД $(a, b) = 1$. Тоді $a = P_k,\ b = Q_k$. Використавши відомі рекурентні співвідношення

$$P_k = q_k P_{k-1} + P_{k-2};$$
$$Q_k = q_k Q_{k-1} + Q_{k-2};$$
$$P_{-1} = 1;\ P_0 = q_1;$$
$$Q_{-1} = 0;\ Q_0 = 1,$$

дістанемо один з розв'язків діофантова рівняння $ax - by = 1$:

$$x_0 = (-1)^{k-1} Q_{k-1};\ \ y_0 = (-1)^{k-1} P_{k-1}.$$

Решта розв'язків матимуть форму

$$x = (-1)^{k-1} Q_{k-1} + bt,\ \ y = (-1)^{k-1} P_{k-1} + at,\ \ t \in Z.$$

Приклад 1.8 Розв'язати діофантово рівняння $31x - 23y = 11$.

Оскільки 11 ділиться на НСД (31, 23) = 1, розв'язок існує. Заповнюємо таблицю:

k	–1	0	1	2	3
q_k	–	1	2	1	7
P_k	1	1	3	4	31
Q_k	0	1	2	3	23

Отже, $k = 3$, $\frac{P_2}{Q_2} = \frac{4}{3}$. Здобуваємо розв'язок:

$$x = (-1)^2 \cdot 11 \cdot 3 + 23t = 33 + 23t;$$
$$y = (-1)^2 \cdot 11 \cdot 4 + 31t = 44 + 31t,$$

де $t \in Z$.

Перевірка:

$$31(33 + 23t) - 23(44 + 31t) =$$
$$= 31 \cdot 33 + 31 \cdot 23t - 23 \cdot 44 - 23 \cdot 31t = 31 \cdot 33 - 23 \cdot 44 = 11.$$

Приклад 1.9 Розв'язати діофантово рівняння $655x - 115y = 700$.

Оскільки 700 ділиться на НСД (655, 115) = 5, розв'язок існує. Розділимо ліву та праву частини рівняння на 5, дістанемо $131x - 23y = 140$. Заповнюємо таблицю:

k	–1	0	1	2	3	4
q_k	–	5	1	2	3	2
P_k	1	5	6	17	57	131
Q_k	0	1	1	3	10	23

Отже, $k = 4$, $\frac{P_3}{Q_3} = \frac{57}{10}$. Знаходимо розв'язок:

$$x = (-1)^3 \cdot 140 \cdot 10 + 23t = -1400 + 23t;$$
$$y = (-1)^3 \cdot 140 \cdot 57 + 131t = -7980 + 131t,$$

де $t \in Z$.

Перевірка:

$$131(-1400 + 23t) - 23(-7980 + 131t) =$$
$$= 131(-1400) + 131 \cdot 23t + 23 \cdot 7980 - 23 \cdot 131t = 131(-1400) + 23 \cdot 7980 = 140.$$

Вправи

1 Задано алгоритм ділення $a = bq + r$. Знайдіть q і r для поданих нижче значень a і b:

а) $a = 75,\ b = 8$; б) $a = 102,\ b = 5$; в) $a = 81,\ b = 9$; г) $a = 76,\ b = 25$.

2 Знайдіть найбільший спільний дільник для таких пар чисел:

а) $a = 621,\ b = 437$; б) $a = 822,\ b = 436$; в) $a = 289,\ b = 377$.

3 Знайдіть найменше спільне кратне для пар чисел із вправи 2.

4 Для поданих нижче пар чисел знайдіть u і v такі, що $au + bv = $ НСД (a, b):

а) $a = 83,\ b = 17$; б) $a = 361,\ b = 418$; в) $a = 216,\ b = 324$.

5 Доведіть, що для цілих чисел a і b, які не дорівнюють нулеві, $a \,/\, $НСД (a, b) і $b \,/\, $НСД (a, b) є взаємно прості, тобто

$$\text{НСД}\left(\frac{a}{\text{НСД}(a, b)}, \frac{b}{\text{НСД}(a, b)} \right) = 1.$$

6 Розв'язати діофантово рівняння за допомогою відповідних дробів:

а) $43x - 111y = 87$; б) $39x - 111y = 89$; в) $41x - 111y = 87$.

1.3 Прості числа

Означення Ціле число, яке є більше за 1, називається *простим*, якщо воно не має додатних дільників, окрім 1 і самого себе. Додатне ціле число, більше за 1, називається *складеним*, якщо воно не є простим.

На сьогодні складено таблиці усіх простих чисел, які не перевищують 50 мільйонів, далі відомі лише окремі їхні представники. У додатку А подано таблицю простих чисел, що не перевищують 2200. У криптографії використовують прості числа (понад 512 біт).

Серед перших 10-ти додатних цілих чисел є чотири простих числа: 2, 3, 5 і 7. Цілі числа $4 = 2 \cdot 2,\ 6 = 2 \cdot 3,\ 8 = 2 \cdot 4,\ 9 = 3 \cdot 3$ і $10 = 2 \cdot 5$ є складеними. Отже, якщо $n = r\,s$, де $1 < r < n$ і $1 < s < n$, то n – складене число. За означенням, ціле число 1 не є ані простим, ані складеним. Число 2 – єдине парне просте число. Визначити, чи є невелике ціле число простим, намагаючись поділити його на менші прості числа, порівняно легко, оскільки кількість можливих варіантів є невелика. Однак питання про те, чи є простим велике ціле число, може стати дуже складним. Подана нижче теорема засвідчує, що існує нескінченно багато простих чисел.

Теорема 10 Існує нескінченно багато простих чисел.

Доведення. Припустімо, що існує лише скінченна кількість простих чисел, приміром $p_1, p_2, ..., p_k$. Розглянемо ціле число $(p_1p_2...p_k) + 1$. Припустімо, що p_r – певне просте число і $p_r \mid ((p_1p_2...p_k) + 1)$. Але тоді $p_r \mid (p_1p_2...p_k)$, звідки випливає, що $p_r \mid 1$, а це призводить до суперечності, оскільки $p_r > 1$. Отже, $(p_1p_2...p_k) + 1$ – просте число, що, в свою чергу, також є суперечністю, оскільки цього числа немає серед зазначеної скінченної сукупності простих чисел. Отже, наше припущення стосовно того, що існує скінченна кількість простих чисел, є помилкове, тому доходимо висновку, що простих чисел має бути нескінченно багато.

Оскільки розкладання цілих чисел на прості множники є дуже важливим завданням, треба мати швидкий і простий спосіб визначання того, чи є певне додатне ціле число простим чи складеним. Подана нижче теорема доводить, що для перевірки простоти числа треба визначити лише певні з його можливих дільників.

Теорема 11 Якщо додатне ціле число n є складеним, тоді n має простий дільник p такий, що $p^2 \le n$.

Доведення. Нехай p – найменший простий дільник числа n. Якщо n розкладається на множники r та s, тоді $p \le r$ та $p \le s$. Отже, $p^2 \le rs = n$.

Приміром, щоб визначити, чи є $n = 521$ простим, слід розглянути лише прості числа, які є менше або дорівнюють 22, тому що $22^2 = 484$, а $23^2 = 529$. Прості числа, які є менше або дорівнюють 22, – це 2, 3, 5, 7, 11, 13, 17 й 19. Перевіряючи кожне з них, визначаємо, що жодне з них не ділить 521. Тому 521 є простим числом згідно з попередньою теоремою.

Як зазначає наступна теорема, прості числа утворюють безліч певного роду будівельних блоків для цілих чисел.

Теорема 12 Кожне додатне ціле число або дорівнює 1, або є просте, або може бути записане як добуток простих чисел.

Ціле число 37 – просте. Ціле число $1554985071 = 3 \cdot 3 \cdot 4463 \cdot 38713$ – добуток чотирьох простих чисел, два з яких збігаються.

Теорема 13 (основна теорема арифметики) Кожне додатне ціле число більше за 1 є або простим, або може бути записане у формі добутку простих чисел, причому цей добуток є єдиний з точністю до порядку співмножників.

Наприклад, $n = 39616304 = 2 \cdot 13 \cdot 7 \cdot 2 \cdot 23 \cdot 13 \cdot 2 \cdot 13 \cdot 2 \cdot 7 =$
$= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 7 \cdot 7 \cdot 13 \cdot 13 \cdot 13 \cdot 23$

становить собою два розкладання на множники числа n, однак у кожному з добутків одне й те саме просте число використано однакову кількість разів.

Відмінним є лише порядок записування простих чисел. Фактично маємо 12600 різноманітних способів розкладання на множники числа n з використанням 10-ти простих співмножників, однак кожне таке розкладання містить рівно чотири двійки, дві сімки, три множники, дорівнювані 13, і один, дорівнюваний 23. Зазвичай прості множники групують, використовуючи експонентний запис. Наприклад,

$$n = 2^4 \cdot 7^2 \cdot 13^3 \cdot 23^1.$$

Теорема 14 Кожне додатне ціле число більше за 1, може бути записане в єдиний спосіб з точністю до порядка у формі $q_1^{k(1)} q_1^{k(2)} \dots q_1^{k(n)}$, де $k(1)$, $k(2)$, ..., $k(n)$ – додатні цілі числа.

Тепер зрозуміло, чому 1 не входить до множини простих чисел. Інакше теорема про єдиність розкладання на прості множники була б помилковою. Якщо розкладання на прості множники є відоме, то прості числа, що формують розкладання на прості множники кожного дільника цього числа, утворюють підмножину відповідної множини для дільника.

Теорема 15 Якщо $a = p_1^{a(1)} p_2^{a(2)} p_3^{a(3)} \dots p_k^{a(k)}$ і $b = p_1^{b(1)} p_2^{b(2)} p_3^{b(3)} \dots p_k^{b(k)}$, де p_i – прості числа, які ділять a, чи b, певні показники ступеня можуть дорівнювати 0.

Нехай $m(i) = \min[a(i), b(i)]$ та $M(i) = \max[a(i), b(i)]$ для $0 \le i \le k$. Тоді

$$\text{НСД}\ (a, b) = p_1^{m(1)} p_2^{m(2)} p_3^{m(3)} \dots p_k^{m(k)}$$

та

$$\text{НСК}\ (a, b) = p_1^{M(1)} p_2^{M(2)} p_3^{M(3)} \dots p_k^{M(k)}.$$

Наприклад, $a = 195000$ і $b = 10435750$. Розкладання на прості множники чисел a і b має форму $a = 2^3 \cdot 3^1 \cdot 5^4 \cdot 13^1$ і $b = 2^1 \cdot 5^3 \cdot 13^3 \cdot 19^1$. Дістаємо

$$\begin{aligned}\text{НСД}\ (195000, 10435750) &= 2^{\min(3,1)} 3^{\min(1,0)} 5^{\min(4,3)} 13^{\min(1,3)} 19^{\min(0,1)} = \\ &= 2^1 \cdot 3^0 \cdot 5^3 \cdot 13^1 \cdot 19^0 = 2^1 \cdot 5^3 \cdot 13^1 = 3250;\\ \text{НСК}\ (195000, 10435750) &= 2^{\max(3,1)} 3^{\max(1,0)} 5^{\max(4,3)} 13^{\max(1,3)} 19^{\max(0,1)} = \\ &= 2^3 \cdot 3^1 \cdot 5^4 \cdot 13^3 \cdot 19^1 = 626145000.\end{aligned}$$

Теорема 16 Якщо a і b – додатні цілі числа, тоді

$$\text{НСД}\ (a, b) \cdot \text{НСК}\ (a, b) = ab.$$

Приклад 1.10 Знайти НСК (91, 203).

Спочатку визначимо НСД (91, 203), скориставшись алгоритмом Евкліда, а потім розділимо на нього добуток чисел 91 і 203. Оскільки НСД (91, 203) = 7, дістаємо

$$\text{НСК}\ (91, 203) = \frac{91 \cdot 203}{7} = 2639.$$

Вправи

1 Розкладіть кожне з поданих нижче цілих чисел на прості множники:

а) 728; б) 1599; в) 4899; г) 131; д) 523.

2 Скористайтесь теоремами даного підрозділу для знаходження НСД та НСК поданих нижче чисел:

а) $a = 162,\ b = 12$; б) $a = 71,\ b = 23$; в) $a = 72,\ b = 30$; г) $a = 75,\ b = 99$.

3 Якщо a і b – прості числа, чи означає це, що $a^2 + b^2$ – просте число?

4 Два простих числа a і b називаються *числами-близнюками*, якщо різниця між ними дорівнює 2, тобто $a + 2 = b$. Приміром, 3 і 5 є числами-близнюками. Знайдіть три інші пари чисел-близнюків.

5 Чи є середнє арифметичне двох простих чисел-близнюків простим числом?

1.4 Метод вилучення множників Ферма

Наступна теорема є основою алгоритму розкладання на прості множники, який називається *методом вилучення множників Ферма*. Метод використовується для визначення того, чи є число простим.

Теорема 17 Ціле непарне число $n > 1$ не є простим тоді й лише тоді, коли існують невід'ємні цілі числа p і q такі, що $n = p^2 - q^2$.

Значить, якщо n можна подати як різницю квадратів двох невід'ємних цілих чисел, скажімо, $n = p^2 - q^2$, тоді $n = (p - q)(p + q)$. Оскільки $p - q > 1$, то також $p + q > 1$ й n не є простим числом.

І, навпаки, якщо $n = rs$, де $r \geq s > 1$, тоді n можна подати як $[(r + s)/2]^2 - -[(r - s)/2]^2$. Оскільки n є непарне, r та s також є непарними, отже, $r + s$ та $r - s$ – парні числа. Вважаючи, що $p = (r + s)/2$ і $q = (r - s)/2$, знаходимо, що p та q – цілі невід'ємні числа і $p - q = s > 1$. Коли $n = 1$, припустімо, що $p = 1$, а $q = 0$.

Цей метод полягає у спробі знайти цілі числа p та q такі, що $n = p^2 - q^2$, або, що є еквівалентне, $p^2 = n + q^2$, або $q^2 = p^2 - n$. Якщо використовується перше рівняння, вважаємо $q = 1, 2, \ldots$ доти, поки $n + q^2$ не стане повним квадратом. Якщо до значення $q = (n - 1)/2$ повного квадрата не досягнуто, розглянемо ситуацію, коли $q = (n - 1)/2$, що дає $n + q^2 = [(n + 1)/2]^2$ та призводить до розкладання n на множники. Оскільки q має форму $(r - s)/2$, де r і s – дільники n, то очевидно, що q не може перевищити $(n - 1)/2$. Отже, якщо до значення $q = (r - 1)/2$ повного квадрата не досягнуто, число n є простим.

При використанні другого рівняння, тобто $q^2 = p^2 - n$, візьмемо в якості m найменше ціле число таке, що $m^2 > n$, і послідовно вважатимемо $p = m, m + 1, \ldots$ доти, поки $p^2 - n$ не стане повним квадратом. Як і раніш, q не може перевищити

$(n-1)/2$; отже, якщо до значення $p=(n+1)/2$ повного квадрата не матимемо, число n є простим. Перевага використання другого співвідношення полягає в тому, що перевірці на повний квадрат підлягає менша кількість чисел.

Приклад 1.11 Розглянемо вживання запису $p^2 = n + q^2$ для перевірки, чи є простим число $n = 527$.

Розглянемо $q = 1, 2, ..., (n-1)/2$.

q	$n + q^2$
1	$527 + 1 = 528$
2	$527 + 4 = 531$
3	$527 + 9 = 536$
4	$527 + 16 = 543$
5	$527 + 25 = 552$
6	$527 + 36 = 563$
7	$527 + 49 = 576 = 24^2$

Отже, $n = 527$ є складеним і його дільники легко обчислюються:

$$527 = 24^2 - 7^2 = (24-7)(24+7) = 17 \cdot 31.$$

Вправа

Скориставшись методом вилучення множників Ферма, визначити, чи є подані нижче числа простими:

а) 1001; б) 1349; в) 4851; г) 1079; д) 8051; е) 7931; ж) 567.

1.5 Порівняння

Означення Нехай $a, b \in Z$, $p \in N$. Вважають, що число a є порівнянне з b за модулем p, якщо $a - b$ при діленні p дають однакові остачі.

Запис має форму

$$a \equiv b \pmod{p}.$$

Число a є порівнянне з b за модулем p тоді й лише тоді, коли $a - b$ ділиться на p без остачі:

$$\frac{a-b}{p} = k.$$

Очевидно, це, в свою чергу, має місце тоді й лише тоді, коли знайдеться таке ціле число k, що

$$a = b + pk.$$

Порівнянність a з b за модулем p означає, що a і b становлять собою один і той самий елемент у кільці Z_p.

Відношенню $a \equiv b \pmod{p}$, при якому порівнянні між собою числа вважаються у певному розумінні рівні, не відмінні одне від одного, можна подати наочну інтерпретацію, подану на рис. 1.1, використовуючи періодичність, властиву розподілу порівнянних між собою чисел у натуральному ряді.

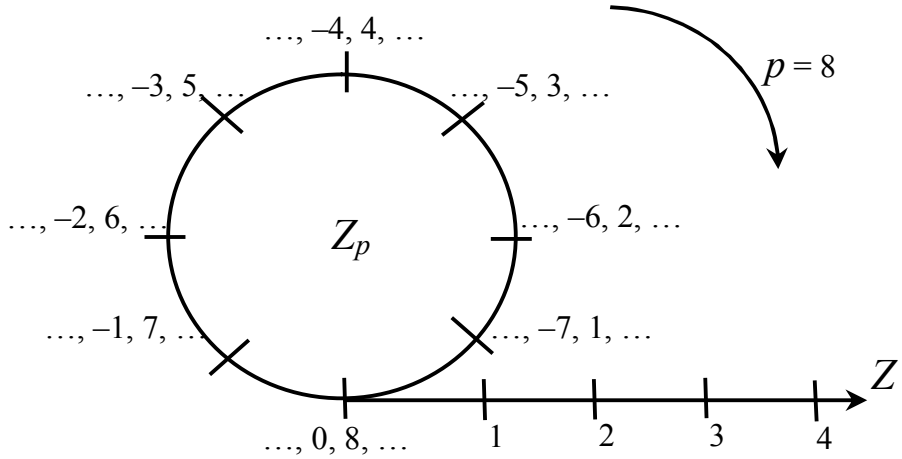


Рисунок 1.1

Приклад 1.12

$17 \equiv 2 \pmod{5}$, оскільки $17 = 2 + 5 \cdot 3$;
$-22 \equiv 1 \pmod{23}$, оскільки $-22 = 1 - 23 \cdot 1$;
$14 \equiv 0 \pmod{7}$, оскільки $14 = 0 + 7 \cdot 2$;
$-5 \pmod 3 \equiv -2 \pmod 3 \equiv 1 \pmod 3$.

Якщо $p = mn$, то m і n – будь-які цілі додатні числа, які можна записати у формі $a \equiv b \pmod{m}$ і $a \equiv b \pmod{n}$.

Приклад 1.13 $16 \equiv 1 \pmod{15}$, слідує $16 \equiv 1 \pmod 3$ та $16 \equiv 1 \pmod 5$.

Модульна арифметика багато в чому є подібна до звичайної арифметики. Приміром, вона так само є комутативна, асоціативна й дистрибутивна. Окрім того, наведення кожного проміжного результату за модулем p надає такий самий результат, що й виконання всього обчислення з подальшим приведенням кінцевого результату за модулем p:

$$(a + b) \bmod p \equiv [(a \bmod p) + (b \bmod p)] \bmod p;$$
$$(a - b) \bmod p \equiv [(a \bmod p) - (b \bmod p)] \bmod p;$$
$$(ab) \bmod p \equiv [(a \bmod p)(b \bmod p)] \bmod p;$$
$$[a(b + c)] \bmod p \equiv [(ab) \bmod p + (ac) \bmod p] \bmod p.$$

З правила множення виходить правило піднесення до степеня:

$$a^r \equiv (a^{mn}) \bmod p \equiv (a^m \bmod p)^n \bmod p,$$

де $r = mn$.

Приклад 1.14 Обчислити
$3^{50} \pmod 8 \equiv [3^2 \pmod 8)]^{25} \bmod 8 \equiv [9 \pmod 8)]^{25} \bmod 8 \equiv 1^{25} \pmod 8 \equiv 1$.

Означення Нехай p – додатне ціле число. Множина всіх класів еквівалентності за модулем p позначається Z_p і називається *множиною класів вираховувань за модулем p*.

Класи вираховувань за модулем p є новими об'єктами. Вони є класами еквівалентності. Елементи кожного класу еквівалентності є порівнянні між собою за модулем p. Наприклад, нехай $p = 3$. Маємо три класи еквівалентності за модулем 3, тому множина

$$Z_3 = \{[0], [1], [2]\}$$

складається з трьох елементів. Елементи Z_3 – класи еквівалентності й, отже, множини. Ці три множини містять 0, 1 та 2. У кожному з цих класів еквівалентності всі елементи є порівнянні між собою за модулем 3, тобто $a \equiv b \pmod 3$ тоді й лише тоді, коли a і b належать до одного й того самого класу еквівалентності – класу вираховувань. Отже,

$$[0] = \{..., –6, –3, 0, 3, 6, 9, ...\};$$
$$[1] = \{..., –8, –5, –2, 1, 4, 7, ...\};$$
$$[2] = \{..., –7, –4, –1, 2, 5, 8, ...\}.$$

Означення Нехай Z_p – множина класів вираховувань за модулем p. Для кожного заданого цілого числа m існує ціле число r таке, що $0 \le r \le p – 1$ і $[m] = [r]$ або $m \equiv r \pmod p$. При цьому вважають, що $[[m]]_p = r$.

Приклад 1.15 Для $p = 5$ дістаємо $Z_5 = \{[0], [1], [2], [3], [4]\} = \{[r]: 0 \le r \le 4\}$.

На множині Z_p можна визначити операції складання та множення. Якщо $[a]$ – клас вираховувань за модулем p, який містить a, і $[b]$ – клас вираховувань за модулем p, який містить b, то складання і множення визначимо за співвідношеннями

$$[a] \oplus [b] = [a + b] = [[a + b]]_p;$$
$$[a] \otimes [b] = [ab] = [[ab]]_p,$$

де складання і множення в центрі та праворуч здійснюється між цілими числами, а складання і множення ліворуч виконується між класами еквівалентності.

Приклад 1.16 Для $p = 5$ дістаємо $Z_5 = \{[0], [1], [2], [3], [4]\}$.

$[2] \oplus [4] = [2 + 4] = [6] = [1]$, оскільки $6 \equiv 1 \pmod 5$;
$[2] \otimes [4] = [2 \cdot 4] = [8] = [3]$, оскільки $8 \equiv 3 \pmod 5$.

Обчислюючи суми й добутки, можна створювати таблиці „сум" та „добутків" для класів вираховувань за модулем 5:

$[a] \oplus [b]$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[1]	[2]	[3]	[4]
[1]	[1]	[2]	[3]	[4]	[0]
[2]	[2]	[3]	[4]	[0]	[1]
[3]	[3]	[4]	[0]	[1]	[2]
[4]	[4]	[0]	[1]	[2]	[3]

$[a] \otimes [b]$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]
[2]	[0]	[2]	[4]	[1]	[3]
[3]	[0]	[3]	[1]	[4]	[2]
[4]	[0]	[4]	[3]	[2]	[1]

Означення Якщо $b \equiv r \pmod{p}$ для додатного цілого числа p, то стверджують, що r є *вираховування* числа b за модулем p.

Повна система вираховувань за модулем p є множина $S = \{r_1, r_2, ..., r_p\}$, де перетинання множини S з кожним класом вираховувань за модулем p містить одне ціле число, тобто S містить одного й лише одного представника з кожного такого класу вираховувань. Повна система вираховувань $\{0, 1, 2, ..., (p-1)\}$ називається *первинною системою вираховувань.* Якщо b – ціле число, $b \equiv r \pmod{p}$ та $0 \le r \le p-1$, то це єдине первинне вираховування за модулем p позначається через $r = [[b]]_p$. *Зведена система вираховувань* за модулем p є підмножина повної системи вираховувань, що складається лише з тих цілих чисел, які є взаємно простими з p, тобто $\{r: r \in S$ та НСД $(r, p) = 1\}$.

Повну систему вираховувань дістаємо шляхом вибору одного цілого числа з кожного класу вираховувань [0], [1], ..., $[p-1]$ множини Z_p. Наприклад, для $p = 6$ $\{24, 7, –58, 15, 40, 113\}$ – повна система вираховувань за модулем 6, оскільки

$$\begin{aligned}
24 &\equiv 0 \bmod 6, && \text{тому} && 24 \in [0];\\
7 &\equiv 1 \bmod 6, && \text{тому} && 7 \in [1];\\
-58 &\equiv 2 \bmod 6, && \text{тому} && -58 \in [2];\\
15 &\equiv 3 \bmod 6, && \text{тому} && 15 \in [3];\\
40 &\equiv 4 \bmod 6, && \text{тому} && 40 \in [4];\\
113 &\equiv 5 \bmod 6, && \text{тому} && 113 \in [5].
\end{aligned}$$

Очевидно, що $\{0, 1, 2, 3, 4, 5\}$ є повною (й до того ж первинною) системою вираховувань за модулем 6. Згідно з означенням, $[[24]]_6 = 0$ і $[[–58]]_6 = 2$.

Теорема 18 Якщо p – додатне ціле число, $\{r_1, r_2, ..., r_p\}$ – повна система вираховувань за модулем p і a – певне ціле число, то $a \equiv r_k \pmod{p}$ для одного й лише одного k, $1 \le k \le p$.

Елементи повної системи вираховувань {24, 7, –58, 15, 40, 113} та первинної (повної) системи вираховувань {0, 1, 2, 3, 4, 5} за модулем 6, які є взаємно простими з $p = 6$, містять в собі, відповідно, множини {7, 113} та {1, 5}. Тому обидві останні множини є зведеними системами вираховувань за модулем 6. При цьому стверджують, що множина {1, 5} є первинною зведеною системою вираховувань за модулем 6.

Теорема 19

а) якщо $a \equiv b \pmod p$ і $c \equiv d \pmod p$, тоді $a + c \equiv (b + d) \bmod p$ і $ac \equiv bd \pmod p$;

б) якщо $ac \equiv bc \pmod p$ і НСД $(c, p) = 1$, тоді $a \equiv b \pmod p$;

в) якщо $a \equiv b \pmod p$, тоді $a^m \equiv b^m \pmod p$ для всіх цілих додатних чисел m;

г) якщо $a \equiv b \pmod{mp}$, тоді $a \equiv b \pmod m$ і $a \equiv b \pmod p$;

д) для $c \neq 0$ співвідношення $ac \equiv bc \pmod p$ має місце тоді й лише тоді, коли $a \equiv b\left(\bmod \dfrac{p}{\text{НСД}(c, p)}\right)$;

е) якщо $a \equiv b \pmod m$, $a \equiv b \pmod p$ і НСД $(m, p) = 1$, тоді $a \equiv b \pmod{mp}$.

Теорема 20 Порівняння $ax \equiv c \pmod p$ має розв'язком ціле число x тоді й лише тоді, коли НСД $(a, p) \mid c$. Всі цілочислові розв'язки мають форму

$$x = x_0 + \frac{tp}{\text{НСД}(a, p)},$$

де t – яке завгодно ціле число, а для x_0 існує таке y_0, що (x_0, y_0) є розв'язком рівняння $ax + py = c$.

Теорема 21 Якщо НСД $(a, p) \mid c$, тоді $ax \equiv c \pmod p$ має скінченну кількість різноманітних розв'язків за модулем p. Ці розв'язки мають форму

$$x_0 + \frac{tp}{\text{НСД}(a, p)} \quad \text{за модулем} \quad p = \left[\left[x_0 + \frac{tp}{\text{НСД}(a, p)}\right]\right]_p$$

для $t = 1, 2, 3, \ldots,$ НСД (a, p), де для x_0 існує таке y_0, що (x_0, y_0) є розв'язком рівняння $ax + py = c$.

Приклад 1.17 Розв'язати рівняння $35x \equiv 14 \pmod{84}$.

Знаходимо НСД $(35, 84) = 7$ й перевіряємо $7 \mid 14$, тоді рівняння має сім різних розв'язків за модулем 84, які мають форму

$$x_0 + \frac{84t}{7} = x_0 + 12t\,.$$

Для $t = 1, 2, 3, \ldots, 7$ та (x_0, y_0) розв'язком є $35x + 84y = 14$, що є рівнозначне до

$$5x + 12y = 2.$$

Перевірка зазначає як розв'язок $x_0 = -2$ та $y_0 = 1$. Сім різних розв'язків за модулем 84 мають форму

t	$x_0 + 12t$
1	$-2 + 12 \cdot 1 = 10$
2	$-2 + 12 \cdot 2 = 22$
3	$-2 + 12 \cdot 3 = 34$
4	$-2 + 12 \cdot 4 = 46$
5	$-2 + 12 \cdot 5 = 58$
6	$-2 + 12 \cdot 6 = 70$
7	$-2 + 12 \cdot 7 = 82$

Коли НСД $(a, p) = 1$, існує єдиний розв'язок порівняння $ax \equiv c \pmod{p}$.

Приклад 1.18 Розв'язати рівняння $6x \equiv 7 \pmod{55}$.

Знаходимо НСД $(6, 55) = 1$ і, очевидно, $1 \mid 7$. Тому існує лише один розв'язок за модулем 55, який має форму

$$x_0 + \frac{tp}{\text{НСД}(a, p)} = x_0 + \frac{1 \cdot 55}{1} = x_0 + 55,$$

де (x_0, y_0) є розв'язком рівняння $ax + py = c$ або $6x + 55y = 7$.

Для знаходження x_0 та y_0 почнемо перебирання з повертанням за алгоритмом Евкліда:

$$55 = 6 \cdot 9 + 1;$$
$$6 = 1 \cdot 6 + 0.$$

$6(-9) + 55 \cdot 1 = \text{НСД}(6, 55) = 1$. Перемножуємо кожен доданок на 7 і одержуємо

$$6(-63) + 55 \cdot 7 = 7.$$

Отже, $x_0 = -63$ та $x = -63 + 55 = -8$, тоді $x = -8 + 55 = 47$.

Приклад 1.19 Розв'язати рівняння $623x \equiv -406 \pmod{84}$.

Число 623 є більше за модуль рівняння 84, а –406 – є від'ємним. Оскільки ми відшукуємо розв'язок за модулем 84, обираємо цілі числа в діапазоні 0, 1, 2, ..., 83, оскільки вони є можливими залишками при діленні на 84 і простими представниками класів еквівалентності, породжених порівнянністю за модулем 84. Використовуючи алгоритм ділення, одержуємо

$$623 = 84 \cdot 7 + 35, \quad \text{або} \quad 623 \equiv 35 \pmod{84};$$
$$-406 = 84(-5) + 14, \text{або } -406 \equiv 14 \pmod{84}.$$

Отже, порівняння

$$35x \equiv 14 \pmod{84}$$

є рівнозначне до початкового $623x \equiv -406 \pmod{84}$.

Розв'язок рівняння $35x \equiv 14 \pmod{84}$ вже було знайдено (див. прикл. 1.17).

Означення Якщо $ax \equiv b \pmod{p}$, тоді

$$x \equiv (-1)^{k-1} P_{k-1}\, b \pmod{p},$$

де $\frac{p}{a} = \frac{P_k}{Q_k}$ – k-тий відповідний дріб.

Приклад 1.20 Розв'язати рівняння $18x \equiv 11 \pmod{23}$, використовуючи відповідні дроби.

Використовуємо алгоритм Евкліда для знаходження НСД (23, 18):

$$23 = 18 \cdot 1 + 5;$$
$$18 = 5 \cdot 3 + 3;$$
$$5 = 3 \cdot 1 + 2;$$
$$3 = 2 \cdot 1 + 1;$$
$$2 = 1 \cdot 2\ .$$

Отже, $k = 5$; $q_1 = 1$; $q_2 = 3$; $q_3 = 1$; $q_4 = 1$; $q_5 = 2$.

Знаходимо $P_1 = 1$; $P_2 = 3 \cdot 1 + 1 = 4$; $P_3 = 1 \cdot 4 + 1 = 5$; $P_4 = 1 \cdot 5 + 4 = 9$. Тоді

$$x \equiv (-1)^{5-1} \cdot 9 \cdot 11 \pmod{23} \equiv 99 \pmod{23} \equiv 7.$$

Означення Числа a та x є *зворотними* за модулем p, якщо

$$ax \equiv 1 \pmod{p}.$$

Наприклад, $3 \cdot 4 \equiv 1 \pmod{11}$; $2 \cdot 8 \equiv 1 \pmod{15}$; $17 \cdot 3 \equiv 1 \pmod{25}$.

Означення Число x є *дискретним логарифмом* числа b, якщо виконується рівність

$$a^x \equiv b \pmod{p}.$$

Приклад 1.21 Знайти x, якщо $2^x \equiv 6 \pmod{11}$.

Обчислимо послідовність степенів 2 за модулем 11:

$2^1 \pmod{11} \equiv 2$; $2^2 \pmod{11} \equiv 4$; $2^3 \pmod{11} \equiv 8$; $2^4 \pmod{11} \equiv 5$;

$2^5 \pmod{11} \equiv 10$; $2^6 \pmod{11} \equiv 9$; $2^7 \pmod{11} \equiv 7$; $2^8 \pmod{11} \equiv 3$;

$2^9 \pmod{11} \equiv 6$; $2^{10} \pmod{11} \equiv 1$.

Далі послідовність повторюється. Отже, $x = 9$.

Означення Число a називається *квадратом* (квадратичним вирахуванням) за модулем p, якщо існує число x та виконується рівність

$$x^2 \equiv a \pmod{p}.$$

Наприклад, число 5 є квадратом за модулем 11, оскільки рівняння $x^2 \equiv 5 \pmod{11}$ має розв'язок, якщо $x = 4$.

Визначимо деякі властивості квадратів:

1) нехай p – просте число і $l = (p - 1)/2$, тоді, якщо a – квадрат за модулем p, то $a^l \equiv 1 \pmod{p}$; якщо a не є квадратом за модулем p, то $a^l \equiv -1 \pmod{p}$;

2) нехай $n = pq$. Число a є квадратом за модулем n, коли a – квадрат за модулем p і a – квадрат за модулем q;

3) нехай $n = pq$, де p і q – прості числа. Вилучимо з ряду чисел від 1 до n ті, які діляться на p і на q. Розділимо ті числа, що залишилися, – $(p - 1)(q - 1)$ – на чотири групи: квадрати за модулем n, квадрати за модулем p, квадрати за модулем q та числа, що не є квадратами за згаданими модулями. Числа, що не є квадратами за модулем p та q, називаються *псевдоквадратами*.

Приклад 1.22 Нехай $p = 5$, $q = 7$, тоді $n = 35$ і $(5 - 1)(7 - 1) = 24$.

Розглянемо всі числа від 1 до 35, вилучимо з них 5, 10, 15, 20, 25, 30, 35, 7, 14, 21, 28, які діляться на 5 і 7. Частина, що залишилася, розпадається на чотири групи по шість елементів у кожній:

квадрати за модулем 35	1, 4, 9, 11, 16, 29;
квадрати лише за модулем 5	6, 19, 24, 26, 31, 34;
квадрати лише за модулем 7	2, 8, 18, 22, 23, 32;
псевдоквадрати	3, 12, 13, 17, 27, 33.

Вправи

1 Обчислити:

а) $(12 \cdot 30) \bmod 9$; б) $(16 \cdot 18) \bmod 17$; в) $343 \pmod{19}$; г) $2401 \pmod{19}$.

2 Обчислити:

а) $4^{10} \pmod{14}$; б) $3^{15} \pmod{10}$; в) $11^7 \pmod{13}$; г) $7^{17} \pmod{8}$.

3 Знайти квадрати за модулями 5, 11, 55 та псевдоквадрати, якщо $p = 5$, $q = 11$.

4 Знайти розв'язок поданих нижче порівнянь:

а) $4x \equiv 3 \pmod{7}$; б) $17x \equiv 3 \pmod{15}$; в) $20x \equiv 8 \pmod{33}$;
г) $24x \equiv 6 \pmod{81}$; д) $91x \equiv 26 \pmod{169}$; е) $23x \equiv 1 \pmod{36}$.

5 Знайти x:

а) $6^x \equiv 5 \pmod{13}$; б) $3^x \equiv 5 \pmod{16}$; в) $2^x \equiv 2 \pmod{5}$; г) $9^x \equiv 3 \pmod{23}$.

6 Довести, що, якщо a – ціле непарне число, то $a^2 \equiv 1 \pmod{8}$.

1.6 Символи Лежандра та Якобі

Символ Лежандра, позначається $L(a, p)$; є визначуваний, якщо a – будь-яке ціле число $a \in Z$, а p – просте число, $p > 2$. Символ Лежандра дорівнює 0, 1 або –1:

$L(a, p) = 0$, якщо a ділиться на p;
$L(a, p) = 1$, якщо a – квадратичне вираховування за модулем p;
$L(a, p) = -1$, якщо a – квадратичне невираховування за модулем p.

Значення $L(a, p)$ можна обчислити за формулою

$$L(a, p) = (a^{(p-1)/2}) \bmod p.$$

Символ Якобі, позначається $J(a, n)$, є узагальненням символу Лежандра на складені модулі. Символ Якобі є визначуваний для кожного цілого значення a і кожного непарного цілого значення n:

якщо n – просте число, тоді $J(a, n) = 0$, якщо a ділиться на n;

якщо n – просте число, тоді $J(a, n) = 1$, якщо a – квадратичне вираховування за модулем n;

якщо n – просте число, тоді $J(a, n) = -1$, якщо a – квадратичне невираховування за модулем n;

якщо n – складене число, тоді $J(a, n) = J(a, p_1)\ldots J(a, p_m)$, де $p_1\ldots p_m$ – розкладання n на прості множники.

Поданий нижче алгоритм дозволяє рекурсивно обчислити символ Якобі:

$J(0, n) = 0$; $J(1, n) = 1$;
$J(ab, n) = J(a, n)\, J(b, n)$;
$J(a, n) = J((a \bmod n), n)$;
$J(a, b_1b_2) = J(a, b_1)\, J(a, b_2)$;
$J(2, n) = 1$, якщо $(n^2 - 1)/8$ парне і -1, якщо інакше.

Символ Якобі не можна використовувати для визначення того, чи є a квадратичним вираховуванням за модулем n (окрім випадку, коли n – просте число). Якщо $J(a, n) = 1$ і n – складене число, то твердження, що a є квадратичним вираховуванням за модулем n, необов'язково є правильне. Наприклад:

$$J(7, 143) = J(7, 11)\, J(7, 13) = 1,$$

проте не існує таких цілих чисел x, що $x^2 \equiv 7 \pmod{143}$.

Вправа

Обчислити:

а) $L(7, 13)$; б) $J(2, 25)$; в) $J(7, 17)$; г) $J(6, 143)$.

1.7 Китайська теорема про залишки

Розглянемо системи порівнянь:

$$\begin{aligned} x &\equiv a_1 \pmod{p_1};\\ x &\equiv a_2 \pmod{p_2};\\ &\vdots\\ x &\equiv a_n \pmod{p_n}, \end{aligned}$$

де числа p_i – попарно взаємно прості. Тобто потрібно знайти ціле число x, яке при діленні на p_i дає залишок a_i, якщо НСД $(p_i, p_j) = 1$ за $i \neq j$.

Ще з давнини люди розглядали системи порівнянь і успішно їх розв'язували. Дуже часто ставилися завдання на усний розрахунок, на зразок такого. Уявіть, що група мавп намагається розкласти на окремі купки купу кокосових горіхів. Якщо мавпи розкладуть горіхи купками по п'ять штук, то залишиться чотири горіхи. Якщо розкладуть купками по чотири, залишиться три горіхи. Купки по сім горіхів дадуть залишок два. Купки по дев'ять горіхів – залишок шість. Яка є мінімально можлива кількість горіхів?

Якщо x – можлива кількість горіхів у купці, тоді наявність чотирьох в залишку при розкладанні у купки по п'ять штук можна подати як

$$x \equiv 4 \pmod 5.$$

Аналогічно, інші умови мають форму

$$x \equiv 3 \pmod 4;$$
$$x \equiv 2 \pmod 7;$$
$$x \equiv 6 \pmod 9.$$

Найменше ціле додатне число x, що задовольняє чотирьом порівнянням, і є шуканим розв'язком. Розв'язок таких завдань подає подана нижче теорема.

Теорема 22 (китайська теорема про залишки) Нехай $p_1, p_2, \ldots, p_n$ – попарно взаємно прості числа, тобто НСД $(p_i, p_j) = 1$ для всіх i та j, менших чи рівних n, де $i \neq j$. Тоді система рівнянь

$$\begin{aligned} x &\equiv a_1 \pmod{p_1}; \\ x &\equiv a_2 \pmod{p_2}; \\ &\vdots \\ x &\equiv a_n \pmod{p_n}, \end{aligned}$$

має розв'язок, єдиний за модулем, який дорівнює цілому числу $p_1 p_2 \ldots p_n$. Потім, якщо

$$M_j = \frac{\prod_{i=1}^{n} p_i}{p_j}$$

і z_j – розв'язок порівняння $M_j z_j \equiv a_j \pmod{p_j}$ для кожного j, розв'язок має форму

$$x = \left[\left[\sum_{j=1}^{n} M_j z_j\right]\right]_{p_1 p_2 \cdots p_n}.$$

Доведення. Нехай x визначено згідно з теоремою. Тоді за кожного k, $1 \le k \le n$,

$$x = \left[\left[\sum_{j=1}^{n} M_j z_j\right]\right]_{p_1 p_2 \cdots p_n},$$

отже

$$x \equiv \sum_{j=1}^{n} M_j z_j \left(\mod \prod_{i=1}^{n} p_i\right) \equiv \sum_{j=1}^{n} M_j z_j \left(\mod p_k\right) \equiv M_k z_k \left(\mod p_k\right) \equiv a_k \left(\mod p_k\right),$$

тому x задовольняє n порівнянням, $x \equiv a_k \left(\mod p_k\right)$ за $1 \le k \le n$. Якщо x' також задовольняє n порівнянням, тоді

$$x - x' \equiv 0 (\mod p_i) \quad \text{за } 1 \le i \le n.$$

Оскільки НСД $(p_i, p_j) = 1$ за $i \ne j$, дістаємо

$$x \equiv x' \left(\mod \prod_{i=1}^{n} p_i\right),$$

тобто розв'язок x є єдиний за модулем $\prod_{i=1}^{n} p_i$.

Приклад 1.23 Знайти розв'язок системи порівнянь

$$x \equiv 1 \pmod 4;$$
$$x \equiv 7 \pmod{11}.$$

Оскільки числа 4 та 11 є взаємно прості, існує ціле число, а саме 10, таке, що $4 \cdot 10 \equiv 7 \pmod{11}$, й існує ціле число 3 таке, що $11 \cdot 3 \equiv 1 \pmod 4$. Отже, $4 \cdot 10 + 11 \cdot 3 = 73$, яке є порівнянне з 29 за модулем 44 і задовольняє обом наведеним вище порівнянням.

Приклад 1.24 Знайдемо відповідь на запитання щодо мавп та горіхів, розв'язавши систему порівнянь

$$x \equiv 4 \pmod 5;$$
$$x \equiv 3 \pmod 4;$$
$$x \equiv 2 \pmod 7;$$
$$x \equiv 6 \pmod 9.$$

Маємо

$$M_1 = 4 \cdot 7 \cdot 9 = 252;$$
$$M_2 = 5 \cdot 7 \cdot 9 = 315;$$
$$M_3 = 5 \cdot 4 \cdot 9 = 180;$$
$$M_4 = 5 \cdot 4 \cdot 7 = 140.$$

Оскільки числа 5 та 252 – взаємно прості, існує ціле число z_1 таке, що $252\, z_1 \equiv 4 \pmod 5$ чи $2\, z_1 \equiv 4 \pmod 5$, чи $z_1 \equiv 2 \pmod 5$. Отже, z_1 може дорівнювати 2.

Оскільки числа 4 та 315 – взаємно прості, існує ціле число z_2 таке, що $315\, z_2 \equiv 3 \pmod 4$ чи $3\, z_2 \equiv 3 \pmod 4$. Отже, z_2 може дорівнювати 1.

Оскільки числа 7 та 180 – взаємно прості, існує ціле число z_3 таке, що $180\, z_3 \equiv 2 \pmod 7$ чи $5\, z_3 \equiv 2 \pmod 7$. Отже, z_3 може дорівнювати 6.

Оскільки числа 9 та 140 – взаємно прості, існує ціле число z_4 таке, що $140\, z_4 \equiv 6 \pmod 9$ чи $5\, z_4 \equiv 6 \pmod 9$. Отже, z_4 може дорівнювати 3.

Дістаємо

$$x \equiv (2 \cdot 252 + 1 \cdot 315 + 6 \cdot 180 + 3 \cdot 140) \mod (5 \cdot 4 \cdot 7 \cdot 9),$$

чи $x \equiv 2319 \pmod{1260}$ та $x = 1059$ – найменший додатний цілочисловий розв'язок.

Вправа

Розв'язати системи порівнянь:

а)	$x \equiv 9 \pmod{12}$; $x \equiv 6 \pmod{25}$.	б)	$x \equiv 3 \pmod 4$; $x \equiv 5 \pmod 9$.	в)	$x \equiv 2 \pmod{13}$; $x \equiv 5 \pmod{21}$.
г)	$x \equiv 5 \pmod 7$; $x \equiv 12 \pmod{15}$; $x \equiv 18 \pmod{22}$.	д)	$x \equiv 7 \pmod{17}$; $x \equiv 9 \pmod{13}$; $x \equiv 3 \pmod{12}$.	е)	$x \equiv 5 \pmod 9$; $x \equiv 3 \pmod{11}$; $x \equiv 4 \pmod 5$.

1.8 Функція Ейлера

Нехай $p = n_1^{\alpha_1} n_2^{\alpha_2} \ldots n_k^{\alpha_k}$ – розкладання на прості множники числа p. Кожен додатний дільник числа p дорівнює 1, або ділиться на p_i при певному i, і кожне ціле число, яке є взаємно простим з p, не має жодного зі згаданих чисел в якості дільника. Деякі властивості числа p залежать від кількості цілих чисел s, $1 \le s \le p$, що не містять жодного з n_i в якості дільника.

Означення Нехай $\varphi(p)$ – кількість додатних цілих чисел, які є менші за p, і взаємно простих з p, тобто $\varphi(p)$ – кількість зведених вираховувань за модулем p. Функція φ називається *тотієнт-функцією Ейлера*[5], чи *функцією Ейлера*.

Наприклад:

$\varphi(1) = 1$;	$\varphi(5) = 4$;	$\varphi(9) = 6$;
$\varphi(2) = 1$;	$\varphi(6) = 2$;	$\varphi(10) = 4$;
$\varphi(3) = 2$;	$\varphi(7) = 6$;	$\varphi(11) = 10$;
$\varphi(4) = 2$;	$\varphi(8) = 4$;	$\varphi(12) = 4$.

[5] Функцію φ, розглядувану нами, названо на честь **Леонарда Ейлера** (1707–1783), перу якого належить найбільша кількість математичних робіт. Творча спадщина Ейлера становить понад 75 вагомих томів. Йому належать відкриття практично в усіх галузях математики. Лише в теорії чисел він має понад 140 оригінальних робіт, уведення цілої ряду малих теорем Ферма. Він вважається засновником топології, а також цілих розділів математичного аналізу. Престижну премію Паризької Академії наук, що присуджується один раз на два роки, Ейлер отримував 12 разів. Численну кількість з нині існуючих систем математичних позначень упроваджено Ейлером.

Кожне додатне ціле число p може бути подано за допомогою додатних цілих чисел, що не перевершують і взаємно простих з кожним дільником числа p. Приміром, $6 = 2 \cdot 3$ має чотири дільники: 1, 2, 3 та 6.

$$\varphi(1) + \varphi(2) + \varphi(3) + \varphi(6) = 1 + 1 + 2 + 2 = 6.$$

Цю властивість сформульовано у поданій нижче теоремі.

Теорема 23 (Гаусса) Якщо p – додатне ціле число, то

$$\sum_{d|p} \varphi(d) = p,$$

де дільники d є додатними дільниками числа p.

Приклад 1.25 Нехай $p = 12$. Дільниками 12 є 1, 2, 3, 4, 6 та 12. Значення функції Ейлера є

$$\varphi(1) + \varphi(2) + \varphi(3) + \varphi(4) + \varphi(6) + \varphi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12.$$

Для ілюстрації вище наведеної теореми 23 в окремому випадку $d = 1, 2, 3, 4, 6$ та 12 визначаємо, що відповідні значення n/d є рівні, отже $n/d = 12, 6, 4, 3, 2$ та 1, тобто дві згадані суми є рівні.

Тепер перейдемо до способів обчислення $\varphi(p)$ для кожного цілого додатного числа p. Розв'язати зазначене завдання допоможуть три подані нижче теореми.

Теорема 24 Якщо числа m та n – взаємно прості, тоді

$$\varphi(mn) = \varphi(m)\varphi(n).$$

Наприклад, нехай $m = 8$ та $n = 15$. Тоді $\varphi(8) = 4$, оскільки лише 1, 3, 5 та 7 – додатні цілі числа, які є менше за 8 і взаємно прості з 8. Також $\varphi(15) = 8$, оскільки лише 1, 2, 4, 7, 8, 11, 13 та 14 – додатні цілі числа, які є менше за 15 та взаємно прості з 15. Отже,

$$\varphi(120) = \varphi(8)\,\varphi(15) = 32,$$

що можна перевірити безпосередньо. Відповідно до твердження теореми 24, вважають, що φ – мультиплікативна відносно взаємно простих множників. Тепер зазначимо, в який спосіб слід обчислювати $\varphi(p)$, коли p становить собою степінь єдиного простого числа.

Теорема 25 Якщо p – просте число, то $\varphi(p^k) = p^k - p^{k-1}$.

Приклад 1.26 Обчислити функцію Ейлера

$$\varphi(49) = \varphi(7^2) = 7^2 - 7^{2-1} = 49 - 7 = 42.$$

Наслідок. Ціле додатне число p є простим тоді й лише тоді, коли $\varphi(p) = p - 1$.

Теорема 26 Якщо p – ціле додатне число з розкладанням на прості множники форми

$$p = n_1^{\alpha_1} n_2^{\alpha_2} \ldots n_t^{\alpha_t},$$

тоді

$$\varphi(p) = \prod_{i=1}^{t} \left[n_i^{\alpha_{i-1}} (n_i - 1) \right] = p \prod_{i=1}^{t} \left(1 - \frac{1}{n_i} \right),$$

де n_i – всі прості числа, які є дільниками числа p.

Приклад 1.27 Обчислити функцію Ейлера:

$$\varphi(5) = 5 - 1 = 4;$$

$$\varphi(30) = \varphi(2 \cdot 3 \cdot 5) = (2-1)(3-1)(5-1) = 8;$$

$$\varphi(40) = 40\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right) = 40\left(\frac{1}{2}\right)\left(\frac{4}{5}\right) = 16;$$

$$\varphi(60) = 60\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{5}\right) = 60\left(\frac{1}{2}\right)\left(\frac{2}{3}\right)\left(\frac{4}{5}\right) = 16;$$

$$\varphi(405) = \varphi(81)\,\varphi(5) = 54 \cdot 4 = 216.$$

Нехай $p = 39616304 = 2^4 \cdot 7^2 \cdot 13^3 \cdot 23^1$, тоді

$$\varphi(39616304) = 2^3(2-1)7^1(7-1)13^2(13-1)23^0(23-1) =$$
$$= 8 \cdot 1 \cdot 7 \cdot 6 \cdot 169 \cdot 12 \cdot 1 \cdot 22 = 14990976.$$

Теорема 27 Якщо ціле число p є більше за 2, тоді $\varphi(p)$ – парне.

Теорема 28 (Уїлсона) Ціле додатне число p є простим тоді й лише тоді, коли $(p-1)! \equiv -1 \pmod{p}$.

Наприклад, нехай $p = 5$. Тоді $(5-1)! = 4! = 24 \equiv -1 \pmod 5$. Зазначимо, що в теоремі йдеться про те, що добуток $(p-1)!$ не може бути порівнянним з -1, якщо число p не є простим. За допомогою теореми можна перевіряти простоту числа p, встановлюючи, чи правильне є порівняння $(p-1)! \equiv -1 \pmod{p}$. Проте такий критерій не використовується для великих значень p, оскільки обчислення $(p-1)! \pmod{p}$ практично є недоцільне.

Означення Нехай $ax \equiv b \bmod p$, де $p > 1$; НСД $(a, p) = 1$, тоді розв'язок

$$x \equiv (ba^{\varphi(p)-1}) \bmod p.$$

Приклад 1.28 Розв'язати порівняння $6x \equiv 7 \pmod{55}$.

Віднаходимо НСД $(6, 55) = 1$, тоді розв'язок становитиме

$$x \equiv (7 \cdot 6^{\varphi(55)-1}) \bmod 55.$$

Віднаходимо $\varphi(55) = \varphi(11 \cdot 5) = (11-1)(5-1) = 40$, тоді

$$x \equiv \left(7 \cdot 6^{39}\right) \bmod 55 \equiv \left(7 \cdot 6^{32} \cdot 6^4 \cdot 6^2 \cdot 6\right) \bmod 55 \equiv \left(7 \cdot 36 \cdot 31 \cdot 36 \cdot 6\right) \bmod 55 \equiv 47.$$

Вправи

1 Обчислити значення функції Ейлера $\varphi(p)$ для $p = 46, 96, 1823, 2025, 2231$.

2 Розв'язати порівняння $23x \equiv 1 \pmod{36}$.

3 Довести, що, якщо число p – просте і $p > 2$, тоді $(p-2)! \equiv 1 \pmod{p}$.

4 Побудувати таблицю значень $\varphi(p)$ за $13 \le p \le 50$.

1.9 Порядок цілого числа

Теорема 29 (Ейлера) Якщо p – ціле додатне число і НСД $(a, p) = 1$, тоді $a^{\varphi(p)} \equiv 1 \pmod{p}$.

Ціле число a називають *первинним коренем* за модулем p. Наприклад, коли $a = 3$, $p = 4$, маємо $\varphi(4) = 2$, отже $3^2 = 9 \equiv 1 \pmod 4$.

Якщо в теоремі 29 p – просте число, то кожне ціле додатне число, яке є менше за p, є взаємно простим з p, отже $\varphi(p) = p - 1$. Тобто, як окремий випадок, доцільна є подана нижче теорема.

Теорема 30 (мала теорема Ферма) Якщо p – просте число, то для кожного такого цілого числа a, що $0 < a < p$, маємо $a^{p-1} \equiv 1 \pmod{p}$.

Наприклад, якщо $p = 7$, то $p - 1 = 6$. У такому разі шостий степінь кожного цілого додатного числа, яке є менше за $p = 7$, має бути порівнянний з 1 за модулем 7:

$$1^6 = 1 \equiv 1 \pmod 7;$$
$$2^6 = 64 \equiv 1 \pmod 7;$$
$$3^6 = 729 \equiv 1 \pmod 7;$$
$$4^6 = 4096 \equiv 1 \pmod 7;$$
$$5^6 = 15625 \equiv 1 \pmod 7;$$
$$6^6 = 46656 \equiv 1 \pmod 7;$$
$$7^6 = 117649 \equiv 0 \pmod 7.$$

Твердження, обернене щодо малої теореми Ферма, є помилкове. Наприклад, $3^{90} \equiv 1 \pmod{91}$, проте $91 = 7 \cdot 13$ – складене число. З іншого боку, якщо p – ціле додатне число і $0 < a < p$ таке, що $a^{p-1} \not\equiv 1 \pmod{p}$, тоді p не може бути простим. Тобто мала теорема Ферма містить частковий критерій простоти числа, оскільки з її допомогою можна довести, що ціле додатне число не є простим без визначення нетривіального дільника числа p. Складені додатні числа n є такі, що $a^{n-1} \equiv 1 \pmod{n}$ для певного a, $1 < a < n$, певною мірою є

схожі з простими числами; з цієї причини такого роду складене число n називається *псевдопростим* числом за основою a. Отже, число $n = 91$ – псевдопросте за основою $a = 3$. Проте, якщо обрати $a = 2$, то дістанемо $2^{90} \not\equiv 1 \pmod{91}$, тобто число $n = 91$ не є псевдопростим за основою 2. Отже, 91 – псевдопросте число за основою 3, але не є псевдопростим за основою 2.

Теорема 31 Якщо p і q – прості числа, причому $p \neq q$ та k – довільне ціле число, то

$$a^{k\varphi(pq)+1} \bmod (pq) \equiv a.$$

Приклад 1.29 Візьмімо $p = 5$, $q = 7$. Тоді $pq = 35$, а функція Ейлера – $\varphi(35) = 4 \cdot 6 = 24$. Розглянемо випадок $k = 2$, тобто підноситимемо числа до степеня $2 \cdot 24 + 1 = 49$. Дістанемо $9^{49} \bmod 35 \equiv 9$, $23^{49} \bmod 35 \equiv 23$.

Твердження. Нехай p і q – два різних простих числа ($p \neq q$). Тоді

$$\varphi(pq) = (p - 1)(q - 1).$$

Доведення. В ряді 1, 2, ..., $pq - 1$ невзаємно простими з pq будуть числа

$$p, 2p, 3p, ..., (q - 1)p$$

та

$$q, 2q, 3q, ..., (p - 1)q.$$

Взагалі таких чисел буде $(p - 1) + (q - 1)$. Отже, кількість чисел, взаємно простих з pq, дорівнюватиме $pq - 1 - (p - 1) - (q - 1) = pq - q - p + 1 = (p - 1)(q - 1)$.

Означення Нехай p – ціле додатне число і a – ціле число таке, що НСД $(a, p) = 1$. *Порядком* числа a за модулем p називається найменше ціле додатне число k таке, що $a^k \equiv 1 \pmod{p}$. Це число позначається через $\operatorname{ord}_n a$.

Теорема 32 Нехай p – ціле додатне число, НСД $(a, p) = 1$ та $k = \operatorname{ord}_p a$. Тоді

а) якщо $a^m \equiv 1 \pmod{p}$, де m – ціле додатне число, тоді $k \mid m$;

б) $k \mid \varphi(p)$;

в) для цілих r та s, $a^r \equiv a^s \pmod{p}$ тоді й лише тоді, коли $r \equiv s \pmod{k}$;

г) жодні два з цілих чисел $a, a^2, a^3, ..., a^k$ не є порівнянними за модулем k;

д) якщо m – ціле додатне число, то порядок числа a^m за модулем p дорівнює $\dfrac{k}{\text{НСД}(k, m)}$;

е) порядок числа a^m за модулем p дорівнює k тоді й лише тоді, коли числа m та k – взаємно прості.

Приклад 1.30 Нехай $p = 14 = 2 \cdot 7$, отже $\varphi(14) = (2 - 1)(7 - 1) = 6$. Первинна приведена система вираховувань для $p = 14$ є множина $\{1, 3, 5, 9, 11, 13\}$.

m	$[[a^m]]_p$	m	$[[a^m]]_p$
1	5	8	11
2	11	9	13
3	13	10	9
4	9	11	3
5	3	12	1
6	1	13	5
7	5	14	11

Розглянемо подану вище таблицю вираховувань для степенів числа $a = 5$, з якої бачимо, що після $m = 6$ іде повторення однієї й тієї самої схеми. Отже, $k = \text{ord}_{14}\, 5 = 6$. Для $m = 12$, $a^m = 5^{12} \equiv 1 \pmod{14}$ та $k \mid m$, що узгоджується з теоремою 32 (а). Також $\text{ord}_{14}\, 5 \mid \varphi(14)$, оскільки $6 \mid 6$ [теорема 32 (б)].

Крім того, $2 \equiv 8 \equiv 14 \pmod 6$ і $5^2 \equiv 5^8 \equiv 5^{14} \equiv 11 \pmod{14}$ [теорема 32 (в)]. Згідно з таблицею, жодні два з чисел 5^1, 5^2, 5^3, 5^4, 5^5 та 5^6 не є порівнянними за модулем 14. Оскільки $\text{ord}_p\, b \mid \varphi(p)$ для кожного цілого числа b та $\varphi(p) = 6$ для $p = 14$, порядок кожного b у $\{1, 3, 5, 9, 11, 13\}$ можна легко обчислити, як це було вчинено для $a = 5$.

Порядок числа b за модулем 14 має вигляд

b	$\text{ord}_p\, b$
1	1
3	6
5	6
9	3
11	3
13	2

Якщо $m = 4$, то $5^4 \equiv 9 \pmod{14}$, але $\text{ord}_{14}\, 5 / \text{НСД}(\text{ord}_{14}\, 5, 4) = 6 / \text{НСД}(6, 4) = 3$. Згідно з таблицею порядків, $\text{ord}_{14}\, 5^4 = 3$ [теорема 32 (д)].

Лише $b = 3$ та $b = 5$ мають порядок за модулем 14. Показниками степеня m у таблиці, наведеній вище, що вони визначають значення a^m, порівнянне з числом 3 або з числом 5, є $m = 1, 5, 7, 11$ та 13. Це лише такі значення m, які є взаємно простими з числом $p = 14$ [теорема 32(е)].

Теорема 33 Якщо $\text{НСД}(a, p) = \text{НСД}(b, p) = 1$ та $\text{ord}_p\, a$ є взаємно простим з $\text{ord}_p\, b$, тоді $\text{ord}_p\,(ab) = (\text{ord}_p\, a)(\text{ord}_p\, b)$.

Приклад 1.31 Якщо $p = 11$, то всі наведені вираховування є взаємно простими з p. Таблиця порядків за модулем 11 має форму

Вираховування	Порядок	Вираховування	Порядок
1	1	6	10
2	10	7	10
3	5	8	10
4	5	9	5
5	5	10	2

Якщо $a = 3$ та $b = 10$, тоді $ab = 30 \equiv 8 \pmod{11}$. Отже, $\operatorname{ord}_{11}(ab) = \operatorname{ord}_{11} 30 = \operatorname{ord}_{11} 8 = 10 = (\operatorname{ord}_{11} 3)(\operatorname{ord}_{11} 10)$, тобто НСД (3, 11) = НСД (10, 11) = 1, $\operatorname{ord}_{11} 3 = 5$ та $\operatorname{ord}_{11} 10 = 2$ – взаємно прості. Зауважимо, що якщо $a = 3$ та $c = 7$, тоді $\operatorname{ord}_{11} 3 = 5$ не є взаємно простим з $\operatorname{ord}_{11} 7 = 10$. У цьому разі $\operatorname{ord}_{11}(ac) = \operatorname{ord}_{11} 21 = \operatorname{ord}_{11} 10 = 2 \neq (\operatorname{ord}_{11} 3)(\operatorname{ord}_{11} 7) = 50$.

Результати, здобуті в теоремах 32 та 33, зумовлюють формулювання критерію, який називається *критерієм простоти числа Лукаса*.

Теорема 34 (Лукаса) Якщо p – ціле додатне число й існує таке ціле число a, що

$$a^{p-1} \equiv 1 \pmod{p}$$

та

$$a^{p-1/n} \not\equiv 1 \pmod{p}$$

для кожного простого числа n, яке ділить $p - 1$, тоді p – просте число.

Для того щоб використовувати критерій Лукаса для перевірки p, треба вміти розкладати на множники число $p - 1$, що саме собою може становити труднощі. Більш того, потрібно знаходити відповідне a. Ціле число a, введене в теоремі 34, називається *примітивним коренем* числа p. Використовуючи критерій з числом $a = 7$, можна довести, що *число Мерсена*[6] $p = 2^{31} - 1$ є простим, оскільки $p - 1 = 2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331$.

Якщо НСД $(a, p) = 1$ і число p – просте, теорема Ферма стверджує, що $a^{p-1} \equiv 1 \pmod{p}$. Її узагальнення, теорема Эйлера, для будь-якого додатного числа p дає $a^{\varphi(p)} = 1 \pmod{p}$. Окрім цих і деяких інших випадків, обчислення a^e за модулем p чи, більш точно, обчислення $[[a^e]]_p$, тобто залишку від ділення a^e на p, для великого значення e може становити значні складнощі, оскільки саме обчислення a^e у таких випадках і ділення його на p практично є недоцільні. Знайти значення $[[a^e]]_p$ можна у такий спосіб.

Для $e = [b_m\, b_{m-1} \ldots b_1\, b_0]$ (двійковий запис числа e) розпочнімо з

$$p_m = [[a]]_p.$$

Потім для $k = m - 1, m - 2, \ldots, 2, 1$ та 0 обчислюємо

[6] Число Мерсена – число вигляду $M_n = 2^n - 1$, де n – натуральне число. Названо на честь французького математика **Марена Мерсена** (1588–1648), одного із засновників Паризької Академії наук, товариша Декарта і Ферма. Числа Мерсена відіграють важливу роль в теорії чисел, криптографії та генераторах псевдовипадкових чисел.

$$p_k = \begin{cases} \left[\left[p_{k+1}^2 \right]\right]_p & \text{за} \quad b_k = 0; \\ \left[\left[p_{k+1}^2 a \right]\right]_p & \text{за} \quad b_k = 1. \end{cases}$$

Остаточним результатом є $p_0 = [[a^e]]_p$. Більш докладно, розпочинаючи з $p_m = [[a]]_p$, одержимо наступний добуток p_k, підносячи до квадрата попередній добуток і зводячи одержане за модулем p за $b_k = 0$; підносячи до квадрата попередній добуток, множачи його на a і зводячи одержане за модулем p за $b_k = 1$.

Приклад 1.32 Обчислити $[[3^{103}]]_{41}$.

Оскільки $103 = 2^6 + 2^5 + 2^2 + 2^1 + 1 = 1100111$, тоді дістаємо

k	b_k	$p_k = \left[\left[p_{k+1}^2 a^{b_k} \right]\right]_p$
6	1	$3 \pmod{41} \equiv 3$
5	1	$(3^2 \cdot 3) \bmod 41 \equiv 27$
4	0	$27^2 \pmod{41} \equiv 729 \pmod{41} \equiv 32$
3	0	$32^2 \pmod{41} \equiv 1024 \pmod{41} \equiv 40$
2	1	$(40^2 \cdot 3) \bmod 41 \equiv 4800 \pmod{41} \equiv 3$
1	1	$(3^2 \cdot 3) \bmod 41 \equiv 27$
0	1	$(27^2 \cdot 3) \bmod 41 \equiv 2187 \pmod{41} \equiv 14$

Тому $[[3^{103}]]_{41} = 14$. Використовуючи порівнянність за модулем 41, дістаємо

$$3^{10} = (3^5 \bmod 41)^2 \bmod 41 \equiv 38^2 \pmod{41} \equiv 9;$$
$$3^{50} = (3^{10} \bmod 41)^5 \bmod 41 \equiv 9^5 \pmod{41} \equiv 9;$$
$$3^{103} = (3^{50} \cdot 3^{50} \cdot 3^3) \bmod 41 \equiv (9 \cdot 9 \cdot 27) \bmod 41 \equiv 14.$$

Вправи

1 Обчислити:

а) $[[37]]_4$; б) $[[149]]_{27}$; в) $[[8!]]_6$; г) $[[48]]_{23}$; д) $[[3^{275}]]_{100}$.

2 Визначити $\operatorname{ord}_n a$ для $1 \le a \le n - 1$ за умов

а) $n = 9$; б) $n = 20$; в) $n = 27$.

3 Довести за допомогою критерію Лукаса, що подані нижче числа є простими:

а) $p = 37$; б) $p = 199$; в) $p = 547$.

1.10 Обчислення у скінченних полях

Поле F є множина, на якій визначено операції складання і множення, що задовольняють вимогам: асоціативності, комутативності, дистрибутивності, існування адитивного нуля та мультиплікативної одиниці, адитивних

обернених та мультиплікативних обернених для всіх елементів, за винятком нуля. Скінченне поле $F(p)$ зі скінченним числом p елементів відіграє важливу роль у криптографії. У загальному випадку число елементів $p = q^n$, де q – певне просте число і $n \geq 1$. Такі скінченні поля називають *полями Галуа*[7] і позначають $GF(q^n)$ чи $GF(q)$ за $n = 1$. Багато криптосистем базуються на полях Галуа $GF(q)$, де q – велике просте число.

Якщо q – просте число, тоді число $a \in [1,\ q-1]$ є взаємно простим з q й тому зворотний елемент a^{-1} має єдине значення. Тим самим однозначно визначається операція ділення.

Позначимо через $GF^*(q)$ множину всіх ненульових елементів поля $GF(q)$. Певний елемент g з $GF^*(q)$ називається *твірним*, або *елементом, що породжує* $GF^*(q)$, якщо для усіх a з $GF^*(q)$, знайдеться таке ціле x, що $g^x \equiv a \pmod q$. Всього маємо $\varphi(q-1)$ твірних елементів g. Число x називається *дискретним логарифмом елемента a за модулем q*. Обчислення дискретних логарифмів – саме складно розв'язуване завдання, як і розкладання на множники.

Приклад 1.33 Поле Галуа $GF(5)$ має елементи $0, 1, 2, 3, 4$ і описується таблицями складання та множення:

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Ще один тип поля Галуа, використовуваний у криптографії, ґрунтується на арифметиці за модулем багаточленів степеня n, чиї коефіцієнти – цілі числа за модулем q, де q – просте. Вони мають елементи, які описуються багаточленами степеня не вище за $n-1$ у формі

$$a(X) = a_{n-1}X^{n-1} + \ldots + a_1 X + a_0.$$

Кожен елемент $a(X)$ є вираховуванням за модулем $P(X)$, де $P(X)$ – незвідний багаточлен степеня n (тобто $P(X)$ не можна розкласти на співмножники – багаточлени меншого за n степеня).

Арифметичні дії над коефіцієнтами a_i виконуються за модулем q, а найвищий степінь X дорівнює $n-1$, оскільки виконується зведення за модулем

[7] **Галуа Еварист** (1811–1832) – французький математик початку XIX століття. Праці з теорії алгебричних рівнянь поклали початок розвитку сучасної алгебри. З ідеями Галуа пов'язані такі її найважливіші поняття, як група, поле й ін. Наукова спадщина Галуа – невелика кількість дуже стисло написаних робіт, через новизну ідей незрозумілих за життя Галуа. Роботи, що залишилися після передчасної смерті Галуа, опубліковано 1846 року.

багаточлена $P(X)$, який має старший степінь n.

Особливий інтерес становлять поля $GF(2^n)$. Тут коефіцієнтами a_i є 0 та 1. Тому багаточлен $a(X)$ степеня не вище за $n-1$ можна подати як вектор з n двійкових цифр: $a_{n-1}a_{n-2}\ldots a_1a_0$.

Кожен з n-бітових векторів відповідає конкретному елементові поля $GF(2^n)$. Наприклад, поле Галуа $GF(2^3)$ має елементи:

Багаточлени	Двійкова форма
0	000
1	001
x	010
$x+1$	011
x^2	100
x^2+1	101
x^2+x	110
x^2+x+1	111

Організація обчислень в полях Галуа передбачає знання певних властивостей багаточленів та їхніх коренів у двійковому полі $GF(2)$. Стисло подамо деякі з них.

Властивість 1: ненульові елементи поля $GF(2^n)$ є коренями узагальненого багаточлена $X^{2^n-1}+1$.

Властивість 2: кожен багаточлен $P(X)$ степеня n, незвідний над полем $GF(2)$, є дільником двочлена $X^{2^n-1}+1$, і кожен дільник двочлена $X^{2^n-1}+1$, що не зводиться над полем $GF(2)$, має степінь, що дорівнює n і менше.

Властивість 3: всі елементи поля $GF(2^n)$ можна здобути як сукупність залишків від ділення $100\ldots00$ на багаточлен $P(X)$, який не зводиться, котра входить до розкладання двочлена $X^{2^n-1}+1$. Ці залишки – корені двочлена $X^{2^n-1}+1$, тобто перетворюють його на нуль. Кількість залишків дорівнює 2^n-1.

Властивість 4: в полі $GF(2^n)$ існує примітивний елемент α, такий, що кожен ненульовий елемент поля $GF(2^n)$ може бути подано як певний степінь α, тобто мультиплікативна група $GF(2^n)$ є циклічною.

Приклад 1.34 Визначення елементів α_i поля $GF(2^4)$. Відповідно до властивості 1, ненульові елементи поля $GF(2^4)$ є коренями узагальненого двочлена $X^{2^4-1}+1=X^{15}+1$. Двочлен $X^{15}+1$ можна подати у вигляді добутку незвідних багаточленів-співмножників:

$$X^{15}+1=P(X^1)P(X^2)P_1(X^4)P_2(X^4)P_3(X^4),$$

де $P\left(X^1\right)=X+1;\ \ P\left(X^2\right)=X^2+X+1;\ P_1\left(X^4\right)=X^4+X+1;\ P_2\left(X^4\right)=X^4+X^3+1;$ $P_3\left(X^4\right)=X^4+X^3+X^2+X+1.$

Відповідно до властивості 3, обчислимо елементи α_i поля $GF\left(2^4\right)$ як сукупність залишків від ділення $100\ldots00$ на незвідний багаточлен $P_1\left(X^4\right)=X^4+X+1.$

Ділять на $P_1\left(X^4\right)=X^4+X+1\leftrightarrow 10011$ одиницю із зростаючою кількістю нулів, тобто ділять одночлени X^j, де $j=0,1,2,3,\ldots$, на багаточлен X^4+X+1. Степені одночленів X^0, X^1, X^2, X^3 є менші за степені багаточлена $P_1\left(X^4\right)$, тому перші чотири залишки від ділення на $P_1\left(X^4\right)$ дорівнюють тим, які ділять, тобто одночленам X^0, X^1, X^2, X^3. Для одночлена $X^4\leftrightarrow 10000$ здобуваємо залишок:

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline & 0\ 0\ 1\ 1 \end{array}\ \Bigg|\begin{array}{l} 1\ 0\ 0\ 1\ 1 \\ \hline \\ \leftrightarrow X^4 \end{array}$$

Для одночлена $X^5\leftrightarrow 100000$ здобуваємо залишок

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline & 0\ 1\ 1\ 0 \end{array}\ \Bigg|\begin{array}{l} 1\ 0\ 0\ 1\ 1 \\ \hline \\ \leftrightarrow X^5 \end{array}$$

Схему обчислення залишків подано нижче:

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^4\leftrightarrow & 0\ 0\ 1\ 1\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^7\leftrightarrow & 1\ 0\ 1\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^8\leftrightarrow & 0\ 1\ 0\ 1\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{10}\leftrightarrow & 0\ 1\ 1\ 1\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{12}\leftrightarrow & 1\ 1\ 1\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{13}\leftrightarrow & 1\ 1\ 0\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{14}\leftrightarrow & 1\ 0\ 0\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{15}=X^0\leftrightarrow & 0\ 0\ 0\ 1 \end{array}$$

Обчислені залишки і нульові елементи $\alpha_0 - \alpha_{14}$ поля Галуа $GF(2^4)$ зведено в таблицю

X^i	Залишок	α_i
X^0	0001	α_0
X^1	0010	α_1
X^2	0100	α_2
X^3	1000	α_3
X^4	0011	α_4
X^5	0110	α_5
X^6	1100	α_6
X^7	1011	α_7
X^8	0101	α_8
X^9	1010	α_9
X^{10}	0111	α_{10}
X^{11}	1110	α_{11}
X^{12}	1111	α_{12}
X^{13}	1101	α_{13}
X^{14}	1001	α_{14}

Поле Галуа $GF(2^4)$ побудовано як поле багаточленів з коефіцієнтами 0 та 1 за модулем незвідного багаточлена $P_1(X^4) = X^4 + X + 1 \leftrightarrow 10011$.

В полі Галуа $GF(2^n)$ визначено чотири алгебричні операції. Операції складання та віднімання виконуються як операції порозрядного складання за модулем 2; операція множення елементів поля виконується як множення відповідних звідних багаточленів за модулем незвідного багаточлена $P(X)$, тобто багаточлена, за модулем якого побудовано елементи поля $GF(2^n)$.

Приклад 1.35 $\alpha_5 = 0110$, $\alpha_6 = 1100$, $\alpha_5 + \alpha_6 = 1010$, оскільки

$$\begin{array}{cccc} 0 & 1 & 1 & 0 \\ \oplus\ 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \end{array}$$

Щоб виконати ділення елемента b на елемент a у полі $GF(2^n)$ за модулем $P(X)$, спочатку знаходять обернений елемент $a^{-1} (\mathrm{mod}\, P(X))$, а потім обчислюють

$$(ba^{-1}) \bmod P(X).$$

Кожен двійковий вектор довжиною n, окрім 0, є взаємно простим з незвідним багаточленом $P(X)$, незалежно від значення $P(X)$. Тому кількість вираховувань, взаємно простих з $P(X)$, дорівнює $\varphi(P(X)) = 2^n - 1$ (розширення

функції Ейлера для багаточленів). Тому

$$a^{-1} \equiv a^{\varphi(P(X))-1} \left(\mathrm{mod}\, P(X)\right) \equiv a^{2^n-2} \left(\mathrm{mod}\, P(X)\right).$$

Приклад 1.36 $\alpha_{14} = 1001$, $\alpha_{14}\alpha_{14} = \alpha_{14}^2 = \alpha_{13}$ за $\mathrm{mod}\, P_1\left(X^4\right) \leftrightarrow 10011$.

$$\begin{array}{rcccc} & & \times & 1 & 0 & 0 & 1 \\ & & & 1 & 0 & 0 & 1 \\ \hline & & & 1 & 0 & 0 & 1 \\ \oplus & & 1 & 0 & 0 & 1 & \\ \hline & & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$\begin{array}{rccccccc|cccccc} \oplus & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ & 1 & 0 & 0 & 1 & 1 & & & \multicolumn{5}{l}{\hrulefill} \\ \cline{1-8} \alpha_{13} \leftrightarrow & & & & 1 & 1 & 0 & 1 & & & & & \end{array}$$

Приклад 1.37 Нехай $a = 100$ та $P(X) = 1011$ в полі $GF(2^3)$.

$a^{-1} \equiv 100^{2^3-2} \left(\mathrm{mod}\, 1011\right) \equiv 100^6 \bmod (1011) \equiv (100^2 \cdot 100^4) \bmod 1011.$

$100^2 \left(\mathrm{mod}\, 1011\right) \equiv 10000 \oplus 10110 = 110$

$$\begin{array}{rccccc|cccc} \oplus & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & & \multicolumn{4}{l}{\hrulefill} \\ \cline{1-5} & & & 1 & 1 & 0 & & & & \end{array}$$

чи

$100^4 \left(\mathrm{mod}\, 1011\right) \equiv 110^2 \left(\mathrm{mod}\, 1011\right) \equiv 010$

чи

$$\begin{array}{rccccc} & & \times & 1 & 1 & 0 \\ & & & 1 & 1 & 0 \\ \hline & & & 0 & 0 & 0 \\ \oplus & & 1 & 1 & 0 & \\ & 1 & 1 & 0 & & \\ \hline & 1 & 0 & 1 & 0 & 0 \end{array} \qquad \begin{array}{rccccc|cccc} \oplus & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & & \multicolumn{4}{l}{\hrulefill} \\ \cline{1-5} & & & 0 & 1 & 0 & & & & \end{array}$$

$100^2 \cdot 100^4 \left(\mathrm{mod}\, 1011\right) \equiv 110 \cdot 010 \left(\mathrm{mod}\, 1011\right) \equiv 1100 \left(\mathrm{mod}\, 1011\right) \equiv 111$

чи

$$\begin{array}{rccccc|cccc} \oplus & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & \multicolumn{4}{l}{\hrulefill} \\ \cline{1-5} & & 1 & 1 & 1 & & & & & \end{array}$$

Отже, $a^{-1} = 111$.

Перевірка: $a = 100$, $a^{-1} = 111$, $P(X) = 1011$, $a\, a^{-1} = 110 \cdot 100 = 11100$.

$$
\begin{array}{rccccc|cccc}
 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
\oplus & 1 & 0 & 1 & 1 & & & & & \\
\hline
 & & 1 & 0 & 1 & 0 & & & & \\
\oplus & & 1 & 0 & 1 & 1 & & & & \\
\hline
 & & & 0 & 0 & 1 & & & &
\end{array}
$$

тобто $(a\, a^{-1}) \bmod 1011 \equiv 1.$

Переваги обчислень у полі $GF(2^n)$:

1) всі елементи поля Галуа мають скінченний розмір, ділення елементів не має жодних помилок щодо округлення;

2) складання й віднімання елементів поля $GF(2^n)$ не потребує ділення на модуль;

3) алгоритми обчислень в полі $GF(2^n)$ припускають паралельну реалізацію;

4) для поля $GF(2^n)$ зазвичай застосовують як модуль тричлен $P(X^n) = X^n + X + 1.$

Довгий рядок нулів поміж коефіцієнтами за X^n та X забезпечує простішу реалізацію швидкого множення (зі зведенням за модулем). Тричлен $P(X^n)$ має бути незвідним і примітивним. Тричлен $P(X^n) = X^n + X + 1$ є примітивним для таких значень n ($n < 1000$):

$$1, 3, 4, 6, 9, 15, 22, 28, 30, 46, 60, 63, 127, 153, 172, 303, 471, 532, 865, 900\ .$$

2 КРИПТОСИСТЕМИ З ВІДКРИТИМ КЛЮЧЕМ

2.1 Односпрямовані функції

В роботі „New Directions in Cryptography" Діффі та Хеллман запропонували принципово новий спосіб організації секретного зв'язку без попереднього обміну ключами, названий шифруванням з відкритим ключем. При такому способі для зашифровування та розшифровування використовуються різні ключі і знання одного з них не дає практичної можливості визначити другий. Внаслідок цього ключ зашифровування може бути відкритим без втрати стійкості шифру і лише ключ розшифровування має триматися одержувачем у секреті, тому криптосистеми з відкритим ключем називають *асиметричними* (*несиметричними*) криптосистемами.

Базовим поняттям криптографії з відкритим ключем є поняття односпрямованої функції (one-way function). За заданим аргументом $x \in X$ нескладно обчислити значення цієї функції $F(x)$, тоді як визначити x з $F(x)$ є надто складно, тобто немає алгоритму для розв'язування цього завдання з поліноміальним часом роботи. Теоретично x за відомим значенням $F(x)$ можна знайти завжди, перевіряючи почергово всі можливі значення x доти, поки відповідне значення $F(x)$ не збігатиметься із заданим. Проте практично за значної розмірності множини X такий підхід неможливо здійснити.

Односпрямованою називається функція $F(x)$: $X \to Y$, $x \in X$, яка має дві властивості:

– існує поліноміальний алгоритм обчислення значень $y = F(x)$;

– не існує поліноміального алгоритму інвертування функції $F(x) = y$.

Поліноміальним називатимемо алгоритм, виконання якого завершується понад за $p(n)$ кроків, де n – розмір вхідного завдання, який зазвичай вимірюється кількістю символів тексту, що описує це завдання.

Зауважимо, що до сьогодні не доведено існування односпрямованих функцій. Використовування їх за підґрунтя асиметричних алгоритмів шифрування є припустиме лише до того моменту, поки не буде знайдено ефективні алгоритми, які виконували б пошук односпрямованих функцій за поліноміальний час.

Прикладом кандидата на назву односпрямованої функції є модульне піднесення до степеня, тобто функція $F(x) \equiv a^x \bmod p$, де a – примітивний елемент поля $GF(p)$; p – велике просте число. Те, що ця функція може бути ефективно обчислена навіть за розрядності параметрів у кілька сотень знаків, можна довести на прикладі: a^{25} можна обчислити за допомогою шести операцій множення (за множення вважається і піднесення до квадрата). Число 25 у двійковій системі обчислення записується як 11001, тобто $25 = 2^4 + 2^3 + 2^0$. Тому

$$a^{25} \pmod p \equiv (a^{16}a^8a) \bmod p \equiv ((((a^2a)^2)^2)^2a) \bmod p.$$

Завдання обчислення функції, оберненої до модульного піднесення до степеня, називають *завданням дискретного логарифмування*. До сьогодні не відомо жодного ефективного алгоритму обчислення дискретних логарифмів великих чисел.

Односпрямована функція в якості функції зашифровування є непридатна, оскільки, якщо $F(x)$ – зашифроване повідомлення x, то ніхто, враховуючи законного одержувача, не зможе відновити x. Обійти цю проблему можна за допомогою односпрямованої функції з секретом (one-way trapdoor function). Наприклад, функція E_k: $X \to Y$, яка має обернену функцію D_k: $Y \to X$, проте визначити обернену функцію лише за E_k без знання секрету k є неможливо.

Односпрямованою функцією з секретом k називають функцію E_k: $X \to Y$, залежну від параметра k, яка має такі властивості:

– за кожного k існує поліноміальний алгоритм обчислення значень $E_k(x)$;

– за невідомого k не існує поліноміального алгоритму інвертування E_k;

– за відомого k існує поліноміальний алгоритм інвертування E_k.

Функцію E_k можна використовувати для зашифровування інформації, а обернену до неї функцію D_k – для розшифровування, оскільки за всіх $x \in X$ є справедлива рівність

$$D_k\,(E_k(x)) = x.$$

При цьому мається на увазі, що той, хто знає, як зашифровувати інформацію, зовсім не обов’язково має знати, як розшифровувати її. Так само як і у випадку з односпрямованою функцією, питання щодо існування односпрямованих функцій з секретом є відкрите.

Для практичної криптографії знайдено кілька функцій – кандидатів на назву односпрямованої функції з секретом. Для них другу властивість не доведено, проте відомо, що завдання інвертування є еквівалентне до розв’язування складної математичної задачі.

Вживання односпрямованої функції з секретом у криптографії дозволяє:

– організовувати обмін шифрованими повідомленнями з використовуванням лише відкритих каналів зв’язку, тобто відмовитися від секретних каналів зв’язку для попереднього обміну ключами;

– включати до завдання розкриття шифру складне математичне завдання і тим самим підвищувати обґрунтованість стійкості шифру;

– розв’язувати нові криптографічні завдання, відмінні від шифрування (електронний цифровий підпис тощо).

Стійкість більшості сучасних асиметричних алгоритмів базується на двох математичних проблемах, які на даному етапі є складнообчислюваними:

– дискретне логарифмування в скінченних полях;

– факторизація великих чисел.

Оскільки сьогодні не існує ефективних алгоритмів розв'язування згаданих проблем, або їхній розв'язок потребує залучення потужних обчислювальних ресурсів, або часових витрат, ці математичні завдання набули широкого використовування в побудові асиметричних алгоритмів.

2.2 Модель криптосистеми з відкритим ключем

Асиметричні криптосистеми передбачають наявність двох ключів: відкритого, призначеного для зашифровування передаваного повідомлення, і закритого, за допомогою якого одержувач розшифровує прийняту криптограму.

Несекретний ключ може передаватися відкритим каналом. Його знання не надає зловмисникові можливості дістати доступ до інформації, що міститься у повідомленні.

На рис. 2.1 подано структурну схему криптосистеми з відкритим ключем.

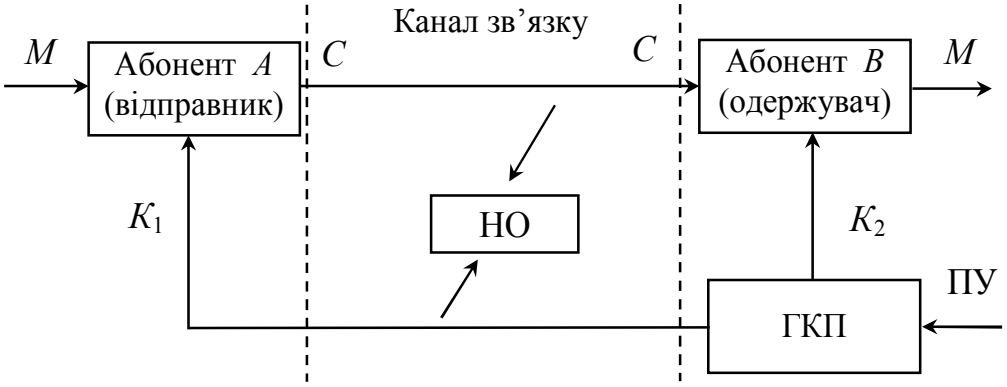


Рисунок 2.1 – Структурна схема криптосистеми з відкритим ключем

Генератор ключової пари (ГКП) видає пару ключів (K_1, K_2) залежно від початкових умов (ПУ), відомих лише одержувачеві повідомлення. Відкритий ключ K_1 передається відправникові незахищеним каналом зв'язку. Відправник зашифровує повідомлення M, використовуючи ключ K_1. Шифртекст C передається незахищеним каналом зв'язку одержувачеві.

Одержувач розшифровує криптограму (відновлюючи вихідне повідомлення), використовуючи секретний ключ K_2.

Несанкціонована особа (НО) має доступ до незахищених каналів і тому може перехопити криптограму C і відкритий ключ K_1. Більш того, вона може володіти алгоритмом шифрування. Єдине, чим вона не володіє, – це ключем K_2.

Найбільш відомі системи з відкритим ключем:

– ранцева криптосистема Меркле–Хеллмана (Knapsack Cryptosystem);

– криптосистема RSA;

– криптосистема Ель–Гамаля (El Gamal Cryptosystem);

– криптосистема Діффі–Хеллмана (Diffie–Hellman);

– криптосистема, базована на властивостях еліптичних кривих (Elliptic Curve Cryptosystem);

– електронно-цифровий підпис DSS (Digital Signature Standard).

2.3 Криптоалгоритм Меркле–Хеллмана

Криптосистему було розроблено Ральфом Меркле та Мартіном Хеллманом 1978 року [10]. Вона стала першим алгоритмом шифрування з відкритим ключем широкого призначення. Криптосистема Меркле–Хеллмана належить до ранцевих алгоритмів. Спочатку алгоритм забезпечував лише шифрування повідомлень, але пізніше Аді Шамір змодифікував відповідну криптосистему для підтримки засобів цифрового підпису. Безпека ранцевих алгоритмів базується на відомому математичному завданні про укладання ранця (рюкзака). У підґрунті алгоритму лежить ідея шифрування повідомлення на підставі розв'язку серії завдань стосовно укладання ранця.

Нехай задано послідовність елементів a_i, належних до множини A, котра задовольняє умові, відповідно до якої кожен черговий елемент має вагу, що перевищує сумарну вагу усієї решти елементів, тобто

$$a_i = \sum_{j=1}^{i-1} a_j .$$

Таку послідовність називаються *швидкозростаючою*.

Розглянемо задачу. Визначити підмножину елементів S, що складається з елементів a_i, які належать до множини A, сума яких дорівнює T, за умови, що виконується вимога $a_i = \sum_{j=1}^{i-1} a_j$.

$$\sum_{i \in S} a_j = T .$$

Приклад 2.1 Нехай множину A задано набором елементів a_i, що задовольняють умові швидкозростаючої послідовності

1, 4, 7, 13, 27, 55, 150, 310, 623.

Потрібно визначити ті елементи цієї множини, сума яких становить 1002.

Аналізуватимемо елементи множини A, розпочинаючи з останнього.

Число 623 < 1002 є елементом підмножини S, оскільки без нього скласти дану підмножину не можна.

Число 310 також належить до множини S, оскільки

$$1002 – 623 = 379 > 310$$

і без нього також не можна дістати число 1002.

Число 150 не належить до підмножини S, оскільки

$$1002 – (623 + 310) = 69 < 150.$$

Число 55 належить до підмножини S, оскільки

$$1002 – (623 + 310) = 69 > 55.$$

Число 27 не належить до підмножини S, оскільки

$$1002 - (623 + 310 + 55) = 14 < 27.$$

Число 13 належить до підмножини S, оскільки

$$1002 - (623 + 310 + 55) = 14 > 13.$$

Число 7 не належить до підмножини S, оскільки

$$1002 - (623 + 310 + 55 + 13) = 1 < 7.$$

Число 4 не належить до підмножини S, оскільки

$$1002 - (623 + 310 + 55 + 13) = 1 < 4.$$

І, врешті, число 1 належить до підмножини S, оскільки

$$1002 - (623 + 310 + 55 + 13) = 1 = 1.$$

Отже, підмножина S містить числа

$$\{1, 13, 55, 310, 623\}.$$

Задача обчислення суми підмножини S для швидкозростаючої послідовності вважається за нескладну.

Узагальнено її можна сформулювати в такий спосіб.

Нехай $A = (a_1, a_2, \ldots, a_n)$ і $B = (b_1, b_2, \ldots, b_n)$ – два вектори. Їхній скалярний добуток

$$C = A \cdot B = \sum_{i=1}^{n} a_i b_i .$$

Знаходження C за відомими A і B не пов'язано з жодними складнощами, тоді як завдання відновлення B за відомими C і A належить до розряду складнообчислюваних. Зазвичай, елементи b_i вектора B у криптографічних алгоритмах набувають значень $\{1, 0\}$. У цьому векторі одиниці розташовуються на позиціях, відповідних до елементів, належних до множини S.

Сутність криптографічного алгоритму Меркле–Хеллмана, який ґрунтується на задачі „про наповнення ранця" зводиться до такого.

Алгоритм Меркле–Хеллмана. Одержувач інформації обирає початковий вектор

$$W = (w_1, w_2, \ldots, w_n),$$

елементи якого w_i, де $i = 1, 2, \ldots, n$, задовольняють вимозі швидкозростаючої послідовності.

Потім одержувач обирає просте число p, значення якого є більше за суму w_i, і певне число r (необов'язково просте), що задовольняє умові $r \le p - 1$, НСД $(p, r) = 1$. Після цього він формує відкритий ключ $K = (k_1, k_2, \ldots, k_n)$ за правилом

$$k_i \equiv (w_i r) \bmod p, \qquad i = 1, 2, \ldots, n.$$

Значення елементів ключа передаються відправникові повідомлення відкритим каналом зв'язку. Значення початкового вектора *w*, а також чисел *p* та *r* тримаються одержувачем повідомлення в секреті.

Відправник розбиває шифроване повідомлення *M* на блоки розміром по *n* символів.

$$M = (m_1, m_2, \ldots, m_n),$$

де $m_i = \{1, 0\}$, $i = 1, 2, \ldots, n$.

Використовуючи ключ *K*, відправник зашифровує повідомлення згідно з правилом

$$C = K \cdot M = \sum_{i=1}^{n} k_i m_i .$$

Потім шифртекст передається відправником відкритим каналом одержувачеві.

Одержувач розв'язує рівняння

$$re \equiv 1 \pmod{p}$$

відносно *e*, інакше кажучи, він знаходить обернене значення для числа *r* за модулем *p*, після чого, перетворює одержану криптограму за правилом

$$C' \equiv (Ce) \bmod p.$$

C' можна привести до форми

$$C' = (Ce) \bmod p \equiv \left(\sum_{i=1}^{n} k_i \, m_i e \right) \bmod p \equiv \left(\sum_{i=1}^{n} m_i \, w_i re \right) \bmod p \equiv \sum_{i=1}^{n} w_i m_i .$$

Далі розв'язуємо завдання щодо визначення суми підмножини.

Приклад 2.2 Нехай $n = 5$. Одержувач обирає початковий вектор

$$W = (2, 4, 7, 15, 29).$$

Для обраного набору чисел виконується вимога швидкозростаючої послідовності. Одержувач обирає $p = 59$ і $r = 40$. З урахуванням обраних величин формується ключ:

$$k_1 \equiv (2 \cdot 40) \bmod 59 \equiv 21;$$
$$k_2 \equiv (4 \cdot 40) \bmod 59 \equiv 42;$$
$$k_3 \equiv (7 \cdot 40) \bmod 59 \equiv 44;$$
$$k_4 \equiv (15 \cdot 40) \bmod 59 \equiv 10;$$
$$k_5 \equiv (29 \cdot 40) \bmod 59 \equiv 39.$$

Потім відкритий ключ $K = (21, 42, 44, 10, 39)$ передається відправникові незахищеним каналом.

Відправник передає повідомлення

$$M = (1, 1, 0, 1, 1),$$

попередньо зашифрувавши його за правилом

$$C = 21 \cdot 1 + 42 \cdot 1 + 44 \cdot 0 + 10 \cdot 1 + 39 \cdot 1 = 112.$$

Попередньо розв'язавши рівняння

$$40e \equiv 1 \pmod{59}$$

$$e \equiv \left(40^{\varphi(59)-1}\right) \bmod 59 \equiv 40^{57} \pmod{59} \equiv (40^{32} \cdot 40^{16} \cdot 40^{8} \cdot 40) \bmod 59 \equiv$$

$$\equiv 15(29(41 \cdot 40 \bmod 59) \bmod 59) \bmod 59 \equiv (15(29 \cdot 47) \bmod 59) \bmod 59 \equiv$$

$$\equiv (15 \cdot 6) \bmod 59 \equiv 31$$

і визначивши, що його значення дорівнює 31, одержувач обчислює значення

$$C' \equiv (112 \cdot 31) \bmod 59 \equiv 50.$$

Віднайшовши значення C', одержувач розв'язує задачу про вміст ранця, аналізуючи елементи початкового вектора, розпочинаючи з останнього:

$50 > 29$	за $m_5 = 1$;
$50 - 29 = 21 > 15$	за $m_4 = 1$;
$21 - 15 = 6 < 7$	за $m_3 = 0$;
$6 > 4$	за $m_2 = 1$;
$6 - 4 = 2 = 2$	за $m_1 = 1$.

Отже, одержане повідомлення має форму $M = (1, 1, 0, 1, 1)$.

Вправа

Зашифрувати повідомлення M й розшифрувати криптограму C за допомогою алгоритму Меркле–Хеллмана.

№ варіанта	n	p	r	W	M
1	5	89	57	2, 7, 10, 21, 43	1, 0, 0, 1, 1
2	6	197	75	3, 7, 11, 24, 47, 101	1, 0, 0, 1, 1, 0
3	7	397	75	3, 7, 11, 24, 47, 101, 195	1, 0, 0, 1, 1, 0, 1
4	5	97	27	2, 7, 11, 21, 43	1, 0, 1, 1, 1
5	6	173	92	2, 7, 10, 21, 43, 85	1, 1, 0, 1, 1, 1
6	7	367	75	3, 7, 11, 22, 45, 91, 185	1, 1, 0, 1, 1, 0, 0
7	5	79	37	2, 5, 9, 19, 41	1, 1, 0, 1, 1
8	6	157	45	3, 5, 9, 19, 37, 77	1, 1, 0, 1, 1, 0
9	7	293	23	2, 5, 9, 17, 37, 71, 145	1, 0, 1, 0, 1, 0, 1
10	5	149	27	7, 9, 17, 35, 71	1, 1, 0, 1, 1
11	6	223	25	5, 7, 13, 27, 55, 111	1, 1, 0, 1, 1, 1
12	7	227	35	2, 3, 7, 13, 27, 53, 107	1, 1, 1, 0, 0, 1, 1
13	7	397	25	3, 7, 11, 24, 47, 101, 195	1, 0, 0, 1, 1, 0, 1

2.4 Система Idempotent Elements

Система Idempotent Elements (IE) є модифікацією системи Меркле–Хеллмана. Одержувач обирає початковий вектор $W = (p_1, p_2, \ldots, p_n)$, де $p_1, p_2, \ldots, p_n$ – різні прості числа, і обчислює n рівнозначних елементів $e_1, e_2, \ldots, e_n$ (idempotent elements):

$$\begin{array}{llll} e_1 \equiv 1 \pmod{p_1}, & e_1 \equiv 0 \pmod{p_2}, & \ldots, & e_1 \equiv 0 \pmod{p_n}; \\ e_2 \equiv 0 \pmod{p_1}, & e_2 \equiv 1 \pmod{p_2}, & \ldots, & e_2 \equiv 0 \pmod{p_n}; \\ \ldots & \ldots & \ldots & \ldots \\ e_n \equiv 0 \pmod{p_1}, & e_n \equiv 0 \pmod{p_2}, & \ldots, & e_n \equiv 1 \pmod{p_n}. \end{array}$$

Одержувач обирає просте число q таке, що $q > \sum_{i=1}^{n} e_i$, і довільне число r. Формується відкритий ключ $K = (k_1, k_2, \ldots, k_n)$ згідно з правилом

$$k_i \equiv (e_i r) \bmod q, \qquad i = 1, 2, \ldots, n.$$

Відправник довільно розбиває повідомлення $M = (m_1, m_2, \ldots, m_n)$ на два, M_1 і M_2, у такий спосіб, що кожен елемент m_i міститься одноразово в M_1 чи в M_2. Зашифровується повідомлення згідно з формулою

$$C = \left| \sum_{M_1} k_i m_i - \sum_{M_2} k_i m_i \right|.$$

Щоб розшифрувати повідомлення, одержувач розв'язує рівняння $re \equiv 1 \pmod{q}$ і обчислює C' та C'' за формулами

$$C' \equiv (Ce) \bmod q;$$
$$C'' = N - C',$$

де $N = \prod_{i=1}^{n} p_i$. Повідомлення приховано у числах C' та C'', тому одержувач наводить їх у векторній формі:

$$C' = (C' \pmod{p_1},\ C' \pmod{p_2},\ \ldots,\ C' \pmod{p_n});$$
$$C'' = (C'' \pmod{p_1},\ C'' \pmod{p_2},\ \ldots,\ C'' \pmod{p_n}).$$

Шукане повідомлення одержується з векторів C' та C'', елементи яких становлять $1 \pmod{p_i}$, $-1 \pmod{p_i}$ або $0 \pmod{p_i}$, заміною $1 \pmod{p_i}$ і $-1 \pmod{p_i}$ на 1 і $0 \pmod{p_i}$ на 0.

Приклад 2.3

Нехай $n = 5$, $M = (1, 1, 0, 1, 1)$ й $W = (2, 3, 5, 7, 11)$.

Елементи e_1, e_2, e_3, e_4, e_5 визначаються з рівнянь (згідно з теоремою про „китайський залишок"):

$e_1 \equiv 1 \pmod 2,\ e_1 \equiv 0 \pmod 3,\ e_1 \equiv 0 \pmod 5,\ e_1 \equiv 0 \pmod 7,\ e_1 \equiv 0 \pmod{11}$	$e_1 = 1155$
$e_2 \equiv 0 \pmod 2,\ e_2 \equiv 1 \pmod 3,\ e_2 \equiv 0 \pmod 5,\ e_2 \equiv 0 \pmod 7,\ e_2 \equiv 0 \pmod{11}$	$e_2 = 1540$
$e_3 \equiv 0 \pmod 2,\ e_3 \equiv 0 \pmod 3,\ e_3 \equiv 1 \pmod 5,\ e_3 \equiv 0 \pmod 7,\ e_3 \equiv 0 \pmod{11}$	$e_3 = 1386$
$e_4 \equiv 0 \pmod 2,\ e_4 \equiv 0 \pmod 3,\ e_4 \equiv 0 \pmod 5,\ e_4 \equiv 1 \pmod 7,\ e_4 \equiv 0 \pmod{11}$	$e_4 = 330$
$e_5 \equiv 0 \pmod 2,\ e_5 \equiv 0 \pmod 3,\ e_5 \equiv 0 \pmod 5,\ e_5 \equiv 0 \pmod 7,\ e_5 \equiv 1 \pmod{11}$	$e_5 = 210$

Обираємо просте число q за умови $q > \sum_{i=1}^{5} e_i$:

$$\sum_{i=1}^{5} e_i = 1155 + 1540 + 1386 + 330 + 210 = 4621.$$

Обираємо просте $q = 4649$ і довільне число $r = 3475$, за допомогою яких обчислюємо елементи відкритого ключа:

$$k_1 \equiv (1155 \cdot 3475) \bmod 4649 \equiv 1538;$$
$$k_2 \equiv (1540 \cdot 3475) \bmod 4649 \equiv 501;$$
$$k_3 \equiv (1386 \cdot 3475) \bmod 4649 \equiv 4635;$$
$$k_4 \equiv (330 \cdot 3475) \bmod 4649 \equiv 3096;$$
$$k_5 \equiv (210 \cdot 3475) \bmod 4649 \equiv 4506.$$

Розкладемо M на M_1 та M_2: $M_1 = (m_1, m_3) = (1, 0)$ та $M_2 = (m_2, m_4, m_5) = (1, 1, 1)$. Зашифровуємо повідомлення

$$C = \left|\left(m_1k_1 + m_3k_3\right) - \left(m_2k_2 + m_4k_4 + m_5k_5\right)\right| = \left|k_1 - \left(k_2 + k_4 + k_5\right)\right| =$$
$$= \left|1538 - \left(501 + 3096 + 4506\right)\right| = \left|1538 - 8103\right| = 6565.$$

Для розшифровування повідомлення розв'яжемо рівняння $3475e \equiv 1 \pmod{4649}$ і віднайдемо $e = 4550$.

Обчислюємо

$$C' \equiv (6565 \cdot 4550) \bmod 4649 \equiv 925;$$
$$N = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310;$$
$$C'' = 2310 - 925 = 1385.$$

Наведемо у векторній формі:

C' = (925 (mod 2), 925 (mod 3), 925 (mod 5), 925 (mod 7), 925 (mod 11)) =
= (1, 1, 0, 1, 1);

C'' = (1385 (mod 2), 1385 (mod 3), 1385 (mod 5), 1385 (mod 7), 1385 (mod 11)) =
= (1, –1, 0, –1, –1).

Після заміни в C'' –1 на 1 матимемо вихідне повідомлення $M = (1, 1, 0, 1, 1)$.

Практична реалізація. Для послідовності з п'яти елементів розв'язати задачу з ранцевим алгоритмом нескладно. Придатні для практичного використовування ранці мають містити не менше 250 елементів. Довжина кожного елемента швидкозростаючої послідовності має перебувати у діапазоні поміж 200 і 400 біт, а довжина модуля має становити від 100 до 200 біт. Для набуття цих значень у практичних реалізаціях використовують генератори випадкових послідовностей.

Розкривати подібні ранці „в лоб" – лише гаяти час. Якщо навіть комп'ютер буде здатен перевіряти мільйон варіантів за секунду, перевірка всіх можливих варіантів ранця потребуватиме понад 10^{46} років [8].

Стійкість ранцевого методу. Криптосистему, базовану на задачі про укладання ранця, зламали не мільйони комп'ютерів, а пара криптографів. Шамір та Циппел [11] виявили слабкі місця у перетворюванні, що дозволило відновити швидкозростаючу послідовність ранця з нормальної. Після зламу схеми Мерклі–Хеллмана було запропоновано безліч інших систем на базі алгоритму укладання ранця (Graham–Shamir, Lu–Lee, Goodman–McAuley, Niemi та інші), але всі вони були проаналізовані і зламані з використанням одних і тих самих криптографічних методів.

Вправа

Зашифрувати повідомлення M й розшифрувати криптограму C за допомогою системи Idempotent Elements:

а) $n = 4$, $W = (5, 7, 11, 13)$, $M = (1, 1, 0, 1)$;
б) $n = 5$, $W = (2, 5, 7, 13, 17)$, $M = (1, 0, 0, 1, 0)$;
в) $n = 5$, $W = (2, 7, 11, 13, 17)$, $M = (1, 0, 0, 1, 0)$;
г) $n = 7$, $W = (2, 3, 5, 7, 11, 13, 17)$, $M = (1, 1, 1, 1, 0, 1, 0)$.

2.5 Алгоритм Шаміра

Шифр, запропонований Аді Шаміром (Adi Shamir)[8], дозволяє організовувати обмін секретними повідомленнями відкритою лінією зв'язку для осіб, які не мають захищених каналів та секретних ключів.

Припустімо, два абоненти A та B сполучені лінією зв'язку (рис. 2.2).

Абонент A хоче передати повідомлення M абонентові B у такий спосіб, щоб ніхто не дізнався про його зміст. Абонент A обирає випадкове велике просте число p і відкрито передає його абонентові B. Потім A обирає два числа – k_a та q_a такі, що

$$k_a q_a \equiv 1 \bmod (p - 1).$$

[8] **Аді Шамір** (народ. 1952 р., Ізраїль) – вчений у галузі теорії обчислювальних систем, лауреат премії Т'юринга. Разом з **Рівестом** і **Адлеманом** розробив криптографічний алгоритм з відкритим ключем RSA, а також зробив потужний внесок у розвиток диференційного криптоаналізу.

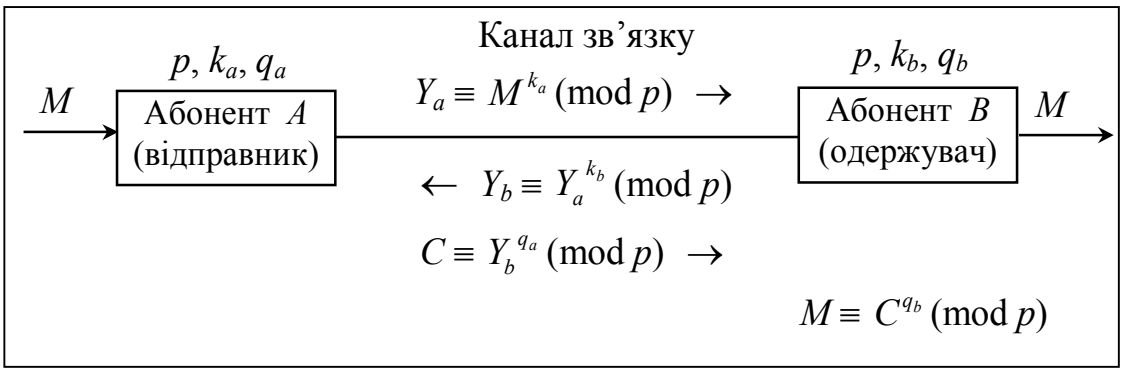


Рисунок 2.2 – Алгоритм Шаміра

Числа k_a та q_a є секретними. Абонент B теж обирає два секретні числа – k_b та q_b такі, що

$$k_b q_b \equiv 1 \bmod (p-1).$$

Після вибору чисел абонент A передає своє повідомлення M, використовуючи триступінчастий протокол. Якщо $M < p$ (M розглядається як число), то повідомлення M передається одразу; якщо $M > p$, то повідомлення подається у формі $M = m_1, m_2, ..., m_i$, де усі $m_i < p$, і потім передаються послідовно $m_1, m_2, ..., m_i$. При цьому для шифрування кожного m_i оптимальніше буде обирати випадково нові пари (k_a, q_a) та (k_b, q_b) – інакше надійність системи знижуватиметься. Сьогодні такий шифр використовується здебільшого для передавання чисел, наприклад секретних ключів, значення яких є менше за p. Тому ми розглядатимемо лише випадок $M < p$. Подамо опис протоколу.

1) Абонент A обчислює число $Y_a \equiv M^{k_a} (\mathrm{mod}\, p)$ і відкритою лінією зв'язку надсилає абонентові B.

2) Абонент B, отримавши Y_a, обчислює число $Y_b \equiv Y_a^{k_b} (\mathrm{mod}\, p)$ і відкритою лінією зв'язку надсилає абонентові A.

3) Сторона A обчислює число $C \equiv Y_b^{q_a} (\mathrm{mod}\, p)$ й передає його стороні B.

4) Сторона B, отримавши число C, обчислює повідомлення

$$M \equiv C^{q_b} (\mathrm{mod}\, p).$$

Справедливість останнього пункту твердження випливає з ланцюжка рівності

$$M \equiv C^{q_b} (\mathrm{mod}\, p) \equiv \left(Y_b^{q_a}\right)^{q_b} \bmod p \equiv \left(Y_a^{k_b}\right)^{q_a q_b} \bmod p \equiv \left(M^{k_a}\right)^{k_b q_a q_b} \bmod p \equiv$$
$$\equiv M^{k_a k_b q_a q_b} (\mathrm{mod}\, p) \equiv M^{(k_a q_a k_b q_b) \bmod (p-1)} (\mathrm{mod}\, p) \equiv M.$$

Приклад 2.4 Нехай абонент A хоче передати абонентові B повідомлення $M = 10$. Сторона A обирає $p = 23$, $k_a = 7$ (НСД (7, 22) = 1) й обчислює q_a:

$$q_a \equiv \left(7^{\varphi(22)-1}\right) \bmod (23-1) \equiv 7^9 \left(\bmod 22\right) \equiv \left(7^6 \cdot 7^2 \cdot 7\right) \bmod 22 \equiv \left(15 \cdot 5 \cdot 7\right) \bmod 22 \equiv 19.$$

Аналогічно, сторона *B* обирає параметри $k_b = 5$ (НСД (5, 22) = 1) й обчислює q_b:

$$q_b \equiv \left(5^{\varphi(22)-1}\right) \bmod (23-1) \equiv 5^9 \pmod{22} \equiv \left(5^6 \cdot 5^2 \cdot 5\right) \bmod 22 \equiv \left(5 \cdot 3 \cdot 5\right) \bmod 22 \equiv 9.$$

Переходимо до протоколу Шаміра:

1) $Y_a \equiv 10^7 \pmod{23} \equiv \left(10^4 \cdot 10^2 \cdot 10\right) \bmod 23 \equiv \left(18 \cdot 8 \cdot 10\right) \bmod 23 \equiv 14;$
2) $Y_b \equiv 14^5 \pmod{23} \equiv \left(14^2 \cdot 14^2 \cdot 14\right) \bmod 23 \equiv \left(12 \cdot 12 \cdot 14\right) \bmod 23 \equiv 15;$
3) $C \equiv 15^{19} \pmod{23} \equiv \left(15^{16} \cdot 15^2 \cdot 15\right) \bmod 23 \equiv \left(16 \cdot 18 \cdot 15\right) \bmod 23 \equiv 19;$
4) $M \equiv 19^9 \pmod{23} \equiv \left(19^6 \cdot 19^2 \cdot 19\right) \bmod 23 \equiv \left(2 \cdot 16 \cdot 19\right) \bmod 23 \equiv 10.$

Отже, абонент *B* отримав передаване повідомлення $M = 10$.

Шифр Шаміра цілковито розв'язує завдання обміну повідомленнями, закритими для прочитання, у разі, коли абоненти можуть користуватися лише відкритими лініями зв'язку. Проте при цьому повідомлення пересилається тричі від одного абонента до іншого, що є недоліком.

Вправа

Для шифру Шаміра із заданими параметрами p, k_a, k_b знайти параметри, яких не вистачає, і описати процес передавання повідомлення M від A до B.

№ варіанта	p	k_a	k_b	M	№ варіанта	p	k_a	k_b	M
1	19	5	7	14	16	41	9	3	8
2	23	15	7	16	17	43	15	11	16
3	19	11	5	10	18	47	13	7	11
4	23	9	3	17	19	37	17	23	12
5	29	5	9	15	20	23	7	19	15
6	17	15	7	11	21	61	13	17	23
7	31	11	13	19	22	53	7	19	11
8	17	9	3	23	23	59	17	19	13
9	29	5	11	4	24	67	15	13	19
10	31	17	7	11	25	71	11	19	24
11	29	11	5	10	26	29	5	17	4
12	23	9	3	17	27	31	11	7	19
13	37	5	7	24	28	41	13	17	11
14	31	19	7	16	29	59	11	13	17
15	19	11	5	6	30	53	17	23	16

2.6 Стандарт асиметричного шифрування RSA

Невдовзі після появи ранцевого алгоритму Меркле–Хеллмана було створено перший повноцінний алгоритм з відкритим ключем, який можна використовувати і для шифрування, і для створення цифрових підписів, алгоритм RSA. Алгоритм RSA названо на честь трьох винахідників –

Рональда Рівеста[9], Аді Шаміра і Леонарда Адлемана[10]. Його було запропоновано 1978 року [12, 13]. Розробникам цього алгоритму вдалося ефективно втілити ідею односпрямованих функцій з секретом. Стійкість RSA базується на складності факторизації великих цілих чисел. Відкритий і закритий ключі є функціями двох великих простих чисел розрядністю 100…200 десяткових цифр і навіть більше. Відновлення відкритого тексту за шифртекстом та відкритим ключем є рівнозначне до розкладання числа на два великі прості множники. Багато років алгоритм RSA протистоїть численним спробам криптографічного розкриття. Криптоаналіз ані доводить, ані спростовує безпеку алгоритму RSA, тим самим обґрунтовуючи міру довіри щодо алгоритму. 1993 року алгоритм RSA було ухвалено за стандарт PKCS # 1: RSA Encryption Standard.

Алгоритм RSA. У довільний спосіб обираються два великі прості числа p та q. Обчислюється добуток $n = pq$.

Обчислюється функція Ейлера: $\varphi(n) = (p - 1)(q - 1)$.

Довільно обирається просте число e – ключ зашифровування, яке задовольняє умовам $e < \varphi(n)$; НСД $(e, \varphi(n)) = 1$.

Обчислюється число d – ключ розшифровування, яке є оберненим до числа e, тобто

$$ed \equiv 1 \pmod{\varphi(n)}.$$

Пару чисел (e, n) робимо відкритим ключем і розміщуємо у загальнодоступному довіднику, а числа p, q тримаються у секреті, d – секретний ключ.

При шифруванні повідомлення M спочатку розкладаємо на цифрові блоки, чиї розміри є менше за n, тобто якщо p та q є 100-розрядними простими числами, то n міститиме близько 200 розрядів і кожен блок повідомлення m_i повинен мати близько 200 розрядів у довжину. Зашифроване повідомлення C складатиметься з блоків c_i такої самої довжини. Формула зашифровування буде мати вигляд

$$C \equiv M^e \pmod{n}.$$

Розшифровування забезпечується операцією піднесення до степеня d за модулем n одержаного шифртексту C:

$$M \equiv C^d \pmod{n},$$

оскільки

$$C^d \pmod{n} \equiv M^{ed} \pmod{n} \equiv (M^{k(p-1)(q-1)+1}) \bmod n \equiv (MM^{k(p-1)(q-1)}) \bmod (pq) \equiv M.$$

Все вищевикладене зведено до табл. 2.1.

---

[9] **Рональд Рівест** (народ. 1947 р., США) – фахівець з криптографії. Рівест – один з авторів алгоритму RSA. Він винайшов такі симетричні алгоритми шифрування як RC2, RC4, RC5 і брав участь у розробці RC6. Літери RC означають „шифр Рівеста” (*Rivest Cipher*) або, неформально, „код Рона” (*Ron's Code*). Окрім RC, він автор геш-функцій MD2, MD4…MD6.

[10] **Леонард Адлеман** (народ. 1945 р., США) – американський вчений-теоретик у галузі комп'ютерних наук. Відомий як співавтор системи шифрування RSA.

Таблиця 2.1 – Алгоритм RSA

Відкритий ключ n – добуток двох простих чисел p та q (p та q мають триматися в секреті); e – ключ зашифровування, число взаємно просте з функцією Ейлера НСД $(e, \varphi(n)) = 1$ і $e < \varphi(n)$
Закритий ключ $d \equiv e^{-1} \pmod{\varphi(n)}$ – ключ розшифровування
Зашифровування $C \equiv M^e \pmod{n}$
Розшифровування $M \equiv C^d \pmod{n}$

Приклад 2.5 Маємо $p = 11$, $q = 5$, $M = 15$.

Обчислюємо $n = 11 \cdot 5 = 55$.

Визначаємо функцію Ейлера: $\varphi(55) = (11–1)(5–1) = 40$.

Обираємо ключ зашифровування $e = 7$, який задовольняє умовам $7 < 40$; НСД $(7, 40) = 1$.

Визначаємо d – ключ розшифровування з рівняння

$$7d \equiv 1 \pmod{40}.$$

Розглянемо кілька способів знаходження d.

Спосіб 1 Для розв'язання рівняння $7d \equiv 1 \pmod{40}$ використовуємо алгоритм Евкліда:

$$40 = 7 \cdot 5 + 5;$$
$$7 = 5 \cdot 1 + 2;$$
$$5 = 2 \cdot 2 + 1;$$
$$2 = 1 \cdot 2 + 0.$$

Обернене підставляння дає

$$1 = 5 – 2 \cdot 2 = 5 – (7 – 5 \cdot 1)2 = 5 \cdot 3 + 7(–2) =$$
$$= (40 – 7 \cdot 5)3 + 7(–2) = 40 \cdot 3 + 7(–17).$$

Оскільки $–17 \equiv 23 \pmod{40}$, то $d = 23$.

Спосіб 2 Вираз $7d \equiv 1 \pmod{40}$ подамо у формі $7d + 40k = 1$.

Для розв'язання діофантова рівняння використовуємо алгоритм Евкліда:

$$\begin{array}{l} 40 = 7 \cdot 5 + 5 \\ 7 = 5 \cdot 1 + 2 \\ \underline{5 = 2 \cdot 2 + 1} \\ 2 = 1 \cdot 2 + 0. \end{array} \quad \uparrow$$

Складаємо ряд l_i коефіцієнтів у зворотному порядку обчислень (зазначено стрілкою). Останній рядок не враховується. Дістаємо

$$l_1 = 2,\ l_2 = 1,\ l_3 = 5.$$

Складаємо новий ряд h_i коефіцієнтів, перше значення якого завжди становить одиницю: $h_1 = 1$; друге значення ряду h_i дорівнює першому значенню ряду l_1: $h_2 = l_1$. Останні значення ряду h_i обчислюємо за формулою

$$h_i = l_{i-1}h_{i-1} + h_{i-2} \text{ за } i \geq 3.$$

Останні два значення ряду h_i визначають значення коефіцієнтів k та d'. Дістаємо

$$h_1 = 1,\ h_2 = 2,\ h_3 = 3,\ h_4 = 17.$$

Матимемо $d' = -17,\ k = 3$.

$$7(-17) + 40 \cdot 3 = 1.$$

Оскільки $-17 \equiv 23 \pmod{40}$, тоді $d = 23$.

$$7 \cdot 23 \equiv 1 \pmod{40}.$$

Спосіб 3 Для віднаходження d скористаємося формулою: якщо $ax \equiv b \pmod{p}$, тоді $x \equiv \left(ba^{\varphi(p)-1}\right) \bmod p$.

Підставимо значення

$$d \equiv \left(7^{\varphi(40)-1}\right) \bmod 40 \equiv 7^{15} \left(\bmod 40\right) \equiv \left(7^8 \cdot 7^4 \cdot 7^2 \cdot 7\right) \bmod 40 \equiv \left(1 \cdot 1 \cdot 9 \cdot 7\right) \bmod 40 \equiv 23.$$

Зашифровуємо повідомлення M:

$$C \equiv 15^7 \pmod{55} \equiv (15^6 \cdot 15) \bmod 55.$$

Знаходимо

$$15^2 \pmod{55} \equiv 225 \pmod{55} \equiv 5;$$
$$15^6 \pmod{55} \equiv (15^2 \bmod 55)^3 \bmod 55 \equiv 5^3 \pmod{55} \equiv 15.$$

Тоді

$$C \equiv (15 \cdot 15) \bmod 55 \equiv 5.$$

Розшифровуємо повідомлення C:

$$M \equiv 5^{23} \pmod{55} \equiv (5^{16} \cdot 5^4 \cdot 5^2 \cdot 5) \bmod 55 \equiv (5 \cdot 20 \cdot 25 \cdot 5) \bmod 55 \equiv 15.$$

Приклад 2.6 Зашифруємо абревіатуру RSA. Для цього літери R, S та A закодуємо п’ятивимірними двійковими векторами, скориставшись двійковим записом їхніх порядкових номерів у англійській абетці: R = 18 = 10010; S = 19 = 10011; A = 1 = 00001. Тепер подамо це повідомлення у формі послідовності чисел, що містяться в інтервалі $\overline{0,526}$. Дістанемо

$$\text{RSA} = (100101001), (100001) = (M_1 = 297, M_2 = 33).$$

Нехай $p = 17$ і $q = 31$, обчислюємо $n = 17 \cdot 31 = 527$, $\varphi(n) = (17 - 1)(31 - 1) = 480$. Обираємо $e = 7$ та визначаємо $d = 343$. Зашифровуємо повідомлення M_1 та M_2:

$$C_1 \equiv 297^7 \pmod{527} \equiv 474;$$
$$C_2 \equiv 33^7 \pmod{527} \equiv 407.$$

У результаті отримуємо шифртекст

$$C = (C_1 = 474, C_2 = 407).$$

Розшифровуємо повідомлення C:

$$M_1 \equiv 474^{343} \pmod{527} \equiv 297;$$
$$M_2 \equiv 407^{343} \pmod{527} \equiv 33.$$

Повернувшись до літерного запису, після розшифровування отримуємо RSA.

Приклад 2.7 Припустімо, що ми вирішили використовувати 8-бітовий код ASCII для символів з розміром блока, який дорівнює символові чи літері. Нам потрібно, щоб: $n \geq 2^8 = 256$. Нехай $p = 41$ та $q = 73$, отже, $n = 41 \cdot 73 = 2993$; $\varphi(2993) = 40 \cdot 72 = 2880$. Нехай $e = 217$, отже, НСД (217, 2880) = 1. Використавши алгоритм Евкліда для розв'язання рівняння $217d \equiv 1 \pmod{2880}$, дістанемо $d = 1513$. У такому разі слово MONEY, зашифроване з використанням RSA-методу з ключем $(e, n) = (217, 2993)$, матиме форму

i	Блок	M_i	C_i
1	M	77	1537
2	O	79	79
3	N	78	1246
4	E	69	1529
5	Y	89	235

Як бачимо з прикладу, маємо число $M_2 = C_2 = 79$, яке не піддається шифруванню за алгоритмом RSA, що є недоліком. Кількість чисел від 1 до n, які не піддаються шифруванню, визначається за формулою

$$N = (1 + \text{НСД}(d - 1, p - 1))(1 + \text{НСД}(d - 1, q - 1)).$$

2.7 Стійкість RSA

Стійкість алгоритму RSA залежить цілковито від трудомісткості розв'язку проблеми розкладання на множники великих чисел. З технічної точки зору це не є коректно. Твердження, що безпека RSA залежить від проблеми розкладання на множники великих чисел, є гіпотетичним. Ніхто й ніколи не довів математично, що для відновлення M при C та e треба розкласти n на множники. Зрозуміло, що може бути знайдено зовсім інший спосіб криптоаналізу RSA. Але якщо цей новий спосіб дозволить криптоаналітикові

розкрити d, його також може бути використано для розкладання на множники великих чисел.

Розглянемо „лобовий метод" зламу системи RSA, який полягає у знаходженні числа d, мультиплікативно оберненого до e за модулем $\varphi(n)$. Це нескладно зробити, якщо є відомі числа p та q. Отже, розв'язавши завдання щодо розкладання на множники цілого числа n, можна дешифрувати систему RSA. Для того, щоб ускладнити завдання розкладання n на прості множники, числа p та q мають обиратися у випадковий спосіб і мати чималі значення. Окрім того, числа p та q не мають бути „надто близькими" одне до одного. Але такий „лобовий" злам є менш ефективний, аніж навіть спроба розкласти n на множники.

Розглянемо можливість використовування близькості значень p та q. Без обмеження наближеності можна вважати, що $p > q$. Для величин $x = (p+q)/2$ та $y = (p-q)/2$ є справедливе відношення $x^2 - y^2 = n$. Для того, щоб знайти розкладання n на прості множники, достатньо буде обрати цілі числа x та y. Перебираючи в порядку зростання варіанти $x > \sqrt{n}$, легко знайти розв'язок, оскільки $x = (p+q)/2$ буде близьким до $\sqrt{n}$ за умови, що p й q є близькі. У результаті дістаємо: $p = x + y$, $q = x - y$.

Приклад 2.8 Нехай $n = pq = 851$. Скористаємося описаним вище способом, щоб розкласти n на прості множники p та q. Оскільки $\sqrt{851} \approx 29{,}17$, обираємо $x = 30$, обчислюємо $30^2 - 851 = 49$ і дістаємо розв'язок: $x = 30$; $y = 7$. Звідси $p = 30 + 7 = 37$; $q = 30 - 7 = 23$.

Атака при використовуванні спільного модуля. Є можлива реалізація RSA, при якій всім користувачам надається однаковий модуль n, але кожному передається окреме значення показників степеня e та d. На жаль, така реалізація працювати не буде. Найбільш очевидна проблема полягає в наступному.

Нехай одне й те саме повідомлення колись зашифровувалося різними показниками (з тим самим модулем) і ці два показники – взаємно прості числа (як це зазвичай і буває). Тоді відкритий текст може бути розкрито навіть за відсутності певних відомостей про один з ключів розшифровування [14].

Нехай M – відкритий текст повідомлення, e_1 та e_2 – два ключі зашифровування, n – спільний модуль. Шифртексти повідомлень:

$$C_1 \equiv M^{e_1} \pmod{n};$$
$$C_2 \equiv M^{e_2} \pmod{n}.$$

Криптоаналітикові відомі n, e_1, e_2, C_1 та C_2. У який же спосіб він може дізнатися про M? Оскільки e_1 та e_2 – взаємно прості числа, то за допомогою розширеного алгоритму Евкліда можна знайти r та s, для яких: $re_1 + se_2 = 1$.

Вважаючи s за від'ємне (тут r або s має бути від'ємним; припустімо, що від'ємним буде s), знову скористаємося розширеним алгоритмом для обчислення C_2^{-1}. Матимемо

$$\left(C_2^{-1}\right)^{-s} C_1^r \equiv M \left(\mathrm{mod}\, n\right).$$

Існують дві інші, більш тонкі форми розкриття систем такого типу. Одна використовує ймовірнісний метод для розкладання n на множники. Інша є детермінованим алгоритмом обчислення певного секретного ключа без розкладання модуля на множники [15].

Приклад 2.9 Нехай $n = 49$, $e_1 = 23$, $e_2 = 11$, $C_1 = 29$ і $C_2 = 8$. Оскільки e_1 та e_2 – взаємно прості числа НСД (11, 23) = 1, то з рівняння $23r + 11s = 1$ за допомогою розширеного алгоритму Евкліда знаходимо $r = 1$, $s = –2$ (можна $r = 12$, $s = –25$) і визначаємо $C_2^{-1} = 43$.

Обчислюємо M з виразу $M \equiv (43^2 \cdot 29^1) \bmod 49 \equiv 15$. Можна перевірити правильність визначення M:

$$C_1 \equiv 15^{23} \pmod{49} \equiv 29; \;\; C_2 \equiv 15^{11} \pmod{49} \equiv 8.$$

Приклад 2.10 Нехай $n = 2993$, $e_1 = 217$, $e_2 = 197$, $C_1 = 235$ і $C_2 = 1200$. Оскільки e_1 і e_2 – взаємно прості числа, то з рівняння $217r + 197s = 1$ знаходимо $r = 69$, $s = –76$. Визначаємо $C_2^{-1} = 2352$.

Обчислюємо M з виразу $M \equiv (2352^{76} \cdot 235^{69}) \bmod 2993 \equiv 89$. Перевіримо правильність визначення M:

$$C_1 \equiv 89^{217} \pmod{2993} \equiv 235; \;\; C_2 \equiv 89^{197} \pmod{2993} \equiv 1200.$$

Метод безключового читання RSA. Криптоаналітикові відомі відкритий ключ (e, n) та шифртекст C. Він добирає число i, для якого виконується співвідношення $C^{e^i} \pmod{n} \equiv C$. Далі просто провадить i разів дешифрування на відкритому ключі перехопленого шифртексту (це виглядає як $(((C^e)^e)^e\ldots)^e \bmod n \equiv C^{e^i} \pmod{n}$). Знайшовши таке i, криптоаналітик обчислює $C^{e^{i-1}} \pmod{n}$ (тобто $i – 1$ разів повторює операцію дешифрування) – це значення і є відкритий текст M.

Приклад 2.11 Нехай $n = 49$, $e = 23$, $C = 29$. Обчислюємо:

$$29^{23} \pmod{49} \equiv 8;$$
$$29^{23^2} \pmod{49} \equiv 29^{529} \pmod{49} \equiv 15;$$
$$29^{23^3} \pmod{49} \equiv 29^{12167} \pmod{49} \equiv 29 = C.$$

Отже, $M = 15$. Перевіримо: $C \equiv 15^{23} \pmod{49} \equiv 29$.

Злам RSA на основі дібраного шифртексту. Є методи зламу, призначені для розкриття реалізацій алгоритмів RSA. Вони розкривають не власне алгоритм, а протокол, що використовує його. Важливо розуміти, що само собою використовування RSA не гарантує безпеки. Річ ще й у деталях реалізації. Розглянемо такий сценарій, корисно уявити собі Алісу, Боба та Єву, трьох вигаданих осіб, що стали персонажами при обговорюванні питань криптографії.

Сценарій. Єва, котра підслуховувала лінії зв'язку Аліси, перехопила повідомлення C, зашифроване за допомогою RSA відкритим ключем Аліси. Єва хоче прочитати повідомлення M. У математичній формі їй потрібно дізнатись про M, для якого $M \equiv C^d \pmod{n}$. Для розкриття M вона спочатку обирає перше випадкове число x, яке є менше за n і дістає відкритий ключ Аліси e. Потім вона обчислює:

$$Y \equiv x^e \pmod{n};$$
$$Z \equiv (YC) \bmod n;$$
$$t \equiv x^{-1} \pmod{n}.$$

Єва просить Алісу підписати Z її закритим ключем d (Аліса має підписати повідомлення, а не його геш-значення). Аліса надсилає Єві

$$u \equiv Z^d \pmod{n},$$

після чого Єва обчислює:

$$(tu) \bmod n \equiv (x^{-1} Z^d) \bmod n \equiv (x^{-1} Y^d C^d) \bmod n \equiv C^d \pmod{n} \equiv M.$$

І Єва відкриває M.

Приклад 2.12 Припустімо, що шифртекст $C = 1537$ зашифровувався з використанням RSA-методу з ключем $(e, n) = (217, 2993)$ (див. приклад в 2.7).

Криптоаналітик обирає число $x = 207$ і обчислює:

$$Y \equiv 207^{217} \pmod{2993} \equiv 1594;$$
$$Z \equiv (1594 \cdot 1537) \bmod 2993 \equiv 1704;$$
$$t \equiv 207^{-1} \pmod{2993} \equiv 882.$$

Потім криптоаналітик просить підписати повідомлення Z закритим ключем d і як наслідок має $u = 974$.

Криптоаналітик відкриває M:

$$M \equiv (882 \cdot 974) \bmod 2993 \equiv 77.$$

На підставі розглянутих атак можна зробити такі обмеження для алгоритму RSA:

– знання однієї пари секретного/відкритого показників для певного модуля дозволяє криптоаналітикові розкласти модуль на множники;

– знання однієї пари секретного/відкритого показників для певного модуля дозволяє криптоаналітикові обчислити інші пари показників, не розкладаючи модуль на множники;

– у протоколах мереж зв'язку, які застосовують RSA, не повинен використовуватися спільний модуль;

– секретні показники мають бути великими числами;

– недостатньо використовувати стійкий криптографічний алгоритм: мають бути безпечними вся криптосистема та криптографічний протокол.

Алгоритм RSA є стандартом де-факто, визнаним майже у всьому світі. Організація ISO розробила стандарт цифрового підпису, який ґрунтується на RSA; RSA служить інформаційним доповненням стандарту ISO 9796 [16]. Багато компаній використовують алгоритм PKCS, створений компанією RSA Data Security Inc. Алгоритм RSA було запатентовано у США. Термін дієздатності закінчився 20 вересня 2000 року.

Вправа

Зашифрувати повідомлення $M = 15$ та розшифрувати криптограму за допомогою алгоритму RSA.

№ варіанта	p	q	e	№ варіанта	p	q	e
1	7	23	9, 13, 26	16	7	29	9, 11, 26
2	5	11	16, 33, 35	17	5	11	16, 27, 35
3	5	13	24, 35, 36	18	5	13	24, 37, 36
4	5	17	16, 48, 49	19	5	17	16, 48, 27
5	5	19	18, 32, 49	20	5	19	18, 32, 47
6	7	11	25, 33, 37	21	7	11	23, 25, 33
7	7	13	23, 33, 38	22	7	13	23, 33, 39
8	7	17	18, 35, 36	23	7	17	18, 37, 54
9	7	19	15, 31, 48	24	7	19	15, 25, 48
10	11	13	18, 26, 113	25	11	13	18, 26, 47
11	11	17	36, 88, 131	26	11	17	36, 88, 37
12	11	19	36, 103, 123	27	11	19	36, 53, 123
13	13	17	68, 92, 133	28	13	17	68, 92, 35
14	13	19	46, 59, 96	29	13	19	46, 133, 96
15	17	19	33, 68, 131	30	17	19	33, 68, 241
Примітка. Треба обрати лише одне число e з трьох запропонованих.							

2.8 Алгоритм Рабіна

Алгоритм Рабіна [17] є модифікацією алгоритму RSA. Безпека алгоритму Рабіна базується на складності пошуку квадратного кореня за модулем складеного числа.

Обираються два простих числа – p і q, – порівнянних з 3 mod 4. Ці прості числа є закритим ключем, а їхній добуток $n = pq$ – відкритим ключем:

$$p \equiv 3 \pmod 4 \equiv -1 \pmod 4;$$
$$q \equiv 3 \pmod 4 \equiv -1 \pmod 4.$$

Вважатимемо, що e є фіксоване і завжди становить 2, тоді шифртекст повідомлення M обчислюється як

$$C \equiv M^2 \pmod n.$$

Розшифрування криптограми C. Введемо допоміжні величини x та y:

$$x \equiv C^k \pmod p;$$
$$y \equiv C^l \pmod q,$$

де $4k = p + 1;\ 4l = q + 1$.

Для x^2 та y^2 одержуємо

$$x^2 \equiv C^{2k} \pmod p \equiv \left[\left(M^2\right)^{\frac{p+1}{4}} \right]^2 \bmod p \equiv M^{p+1} \pmod p \equiv (M^{p-1}M^2) \bmod p \equiv$$

$$\equiv M^2 \pmod p;$$

$$y^2 \equiv C^{2l} \pmod q \equiv \left[\left(M^2\right)^{\frac{q+1}{4}} \right]^2 \bmod q \equiv M^2 \pmod q.$$

Одержуємо чотири системи рівнянь для M_1, M_2, M_3, M_4:

$$\begin{cases} M_1 \equiv x \pmod p; \\ M_1 \equiv y \pmod q; \end{cases} \quad \begin{cases} M_2 \equiv x \pmod p; \\ M_2 \equiv -y \pmod q; \end{cases} \quad \begin{cases} M_3 \equiv -x \pmod p; \\ M_3 \equiv y \pmod q; \end{cases} \quad \begin{cases} M_4 \equiv -x \pmod p; \\ M_4 \equiv -y \pmod q. \end{cases}$$

Одним з чотирьох результатів M_1, M_2, M_3 та M_4 є повідомлення M. Якщо повідомлення написане словами, обрати правильне M нескладно. З іншого боку, якщо повідомлення є потоком випадкових бітів (призначених для генерування ключів цифрового підпису), тоді визначити, яке саме M є правильним, завдання складне. Одним із способів розв'язати цю проблему є додавання до повідомлення відомого заголовка, яке виконується перед зашифровуванням.

Приклад 2.13 Дано $p = 3,\ q = 11,\ M = 8$.

Перевіряємо виконання умов:

$$3 \equiv 3 \pmod 4 \equiv -1 \pmod 4;$$
$$11 \equiv 3 \pmod 4 \equiv -1 \pmod 4.$$

Визначаємо $n = 3 \cdot 11 = 33$.

Зашифровуємо повідомлення $C \equiv 8^2 \pmod{33} \equiv 31$.

Одержувач знаходить $k = (3 + 1) / 4 = 1,\ l = (11 + 1) / 4 = 3$ й обчислює x та y:

$$x \equiv 31^1 \pmod 3 \equiv 1;$$
$$y \equiv 31^3 \pmod{11} \equiv 3.$$

Складаємо чотири системи рівнянь і визначаємо M_1, M_2, M_3 та M_4:

$$\begin{cases} M_1 \equiv 1 (\mathrm{mod} 3); \\ M_1 \equiv 3 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_2 \equiv 1 (\mathrm{mod} 3); \\ M_2 \equiv -3 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_3 \equiv -1 (\mathrm{mod} 3); \\ M_3 \equiv 3 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_4 \equiv -1 (\mathrm{mod} 3); \\ M_4 \equiv -3 (\mathrm{mod} 11). \end{cases}$$

$$\Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow$$

$$M_1 = 25 \qquad M_2 = 19 \qquad M_3 = 14 \qquad \boldsymbol{M_4 = 8} \text{ (!)}$$

Кожне з чисел – 25, 19, 14, 8 – могло б бути первинним повідомленням, оскільки

$$25^2 \equiv 19^2 \equiv 14^2 \equiv 8^2 \equiv 31 \text{ (mod 33)}.$$

Приклад 2.14 Дано $p = 19$, $q = 11$, $M = 16$.

Перевіряємо виконання умов:

$$19 \equiv 3 \text{ (mod 4)} \equiv -1 \text{ (mod 4)};$$
$$11 \equiv 3 \text{ (mod 4)} \equiv -1 \text{ (mod 4)}.$$

Визначаємо $n = 19 \cdot 11 = 209$.

Зашифровуємо повідомлення $C \equiv 16^2 \text{ (mod 209)} \equiv 47$.

Одержувач знаходить $k = (19 + 1) / 4 = 5$, $l = (11 + 1) / 4 = 3$ і обчислює x та y:

$$x \equiv 47^5 \text{ (mod 19)} \equiv 16;$$
$$y \equiv 47^3 \text{ (mod 11)} \equiv 5.$$

Складаємо чотири системи рівнянь і визначаємо M_1, M_2, M_3 та M_4:

$$\begin{cases} M_1 \equiv 16 (\mathrm{mod} 19); \\ M_1 \equiv 5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_2 \equiv 16 (\mathrm{mod} 19); \\ M_2 \equiv -5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_3 \equiv -16 (\mathrm{mod} 19); \\ M_3 \equiv 5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_4 \equiv -16 (\mathrm{mod} 19); \\ M_4 \equiv -5 (\mathrm{mod} 11). \end{cases}$$

$$\Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow$$

$$\boldsymbol{M_1 = 16} \text{ (!)} \qquad M_2 = 149 \qquad M_3 = 60 \qquad M_4 = 193$$

Кожне з чисел – 16, 149, 60, 193 – могло б бути первинним повідомленням, оскільки

$$16^2 \equiv 149^2 \equiv 60^2 \equiv 193^2 \equiv 47 \text{ (mod 209)}.$$

Вправа

Зашифрувати повідомлення та розшифрувати криптограму за допомогою алгоритму Рабіна. Згідно з варіантом обрати прості числа p та q. В якості вихідного повідомлення M взяти дату народження (наприклад, 31 січня – вихідне повідомлення $M = 3101$).

№ варіанта	p	q	№ варіанта	p	q
1	11, 13, 17	137, 173, 191	16	13, 71, 173	17, 97, 107
2	13, 23, 29	193, 191, 181	17	29, 79, 113	37, 73, 103
3	29, 31, 37	173, 157, 167	18	41, 83, 157	13, 97, 199
4	43, 53, 101	89, 149, 163	19	11, 61, 149	17, 53, 191
5	97, 47, 73	149, 197, 151	20	19, 73, 101	29, 37, 179
6	61, 59, 181	61, 29, 139	21	23, 101, 193	73, 167, 97
7	89, 53, 67	137, 29, 131	22	59, 37, 109	29, 113, 107
8	197, 71, 181	127, 97, 101	23	67, 113, 149	41, 127, 89
9	79, 41, 149	53, 97, 107	24	73, 71, 149	157, 131, 173
10	17, 83, 197	29, 61, 103	25	79, 13, 17	137, 149, 139
11	31, 73, 197	37, 149, 163	26	73, 83, 89	197, 173, 151
12	43, 61, 157	41, 89, 151	27	41, 61, 47	53, 163, 181
13	13, 47, 181	17, 61, 139	28	109, 173, 43	61, 167, 97
14	29, 59, 193	37, 101, 131	29	17, 31, 181	137, 73, 179
15	41, 67, 109	53, 89, 127	30	89, 23, 109	109, 29, 191

2.9 Алгоритм Вільямса

Хью Вільямс [18] оптимізував алгоритм Рабіна, внісши до нього зміни, котрі усувають неоднозначність приймання.

Обираються два прості числа p і q такі, щоб $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$ і $n = pq$. Крім того, використовується невелике ціле число s, для якого $J(s, n) = -1$, де $J(s, n)$ – символ Якобі.

Числа s та n передаються відправнику незахищеним каналом. Секретним ключем є

$$k = \frac{1}{2}\left[\frac{1}{4}(p-1)(q-1)+1\right].$$

Для зашифровування повідомлення M обчислюється C_1 таке, що $J(M,n) = (-1)^{C_1}$, і визначаються проміжні повідомлення

$$M' \equiv (s^{C_1} M) \bmod n;$$
$$C_2 \equiv M' \pmod 2.$$

Криптограма обчислюється так само, як і в алгоритмі Рабіна:

$$C \equiv (M')^2 \bmod n.$$

Одержувачеві передається три числа – C, C_1, C_2.

Для розшифровування C одержувач обчислює M'':

$$\pm M'' \equiv C^k \pmod n;$$
$$M \equiv (s^{-C_1} M'') \bmod n.$$

Правильний знак M'' визначає C_2.

Приклад 2.15 Дано $p = 7$, $q = 11$, $M = 8$. Оберемо $s = 2$, для якого $J(2, 77) = -1$.

Зашифроване повідомлення:

$$J(8, 77) = -1, \text{ отже } C_1 = 1;$$
$$M' \equiv (2 \cdot 8) \bmod 77 \equiv 16;$$
$$C_2 \equiv 16 \pmod 2 \equiv 0;$$
$$C \equiv 16^2 \pmod{77} \equiv 25.$$

Розшифровування повідомлення:

$$k = \frac{1}{2}\left[\frac{1}{4}(7-1)(11-1)+1\right] = 8;$$
$$M'' \equiv 25^8 \pmod{77} \equiv 16;$$
$$M \equiv (2^{-1} \cdot 16) \bmod 77 \equiv 8.$$

2.10 Алгоритм Ель–Гамаля

Цей алгоритм є альтернативою алгоритму RSA і при рівному значенні ключа забезпечує таку саму криптостійкість. Безпека алгоритму Ель–Гамаля [19, 20] базується на складності обчислювання дискретних логарифмів.

Учасники інформаційного процесу обирають просте число p і ціле число q, який є первинним коренем за модулем p (рис. 2.3).

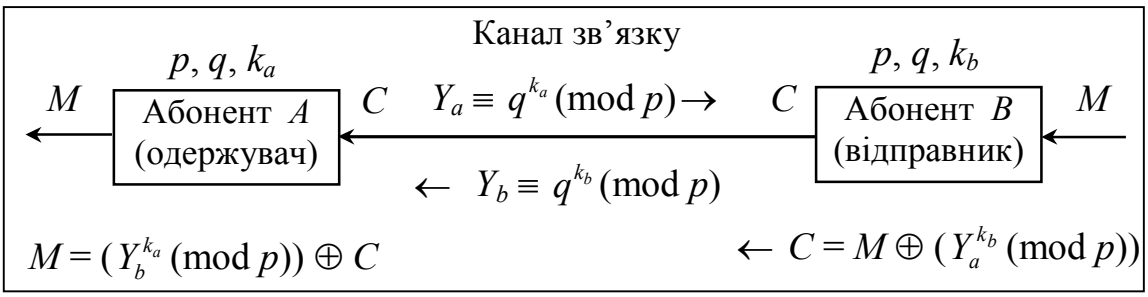


Рисунок 2.3 – Алгоритм Ель–Гамаля

Сторона A генерує секретний ключ $k_a < p$ і обчислює відкритий ключ

$$Y_a \equiv q^{k_a} \pmod p.$$

Сторона B обирає число $k_b < p$ і за його допомогою зашифровує передаване повідомлення M у такий спосіб:

$$Y_b \equiv q^{k_b} \pmod p \text{ і } C = M \oplus (Y_a^{k_b} \pmod p).$$

Величина M є послідовністю двійкових символів, які передаються до каналу зв'язку. Величина $Y_a^{k_b} \pmod p$ перед підсумовуванням перетворюється на послідовність двійкових символів.

Сторона A, отримавши повідомлення у формі Y_b та C, відновлює його:

$$M = (Y_b^{k_a} \pmod p) \oplus C.$$

Алгоритм Ель–Гамаля – перший криптографічний алгоритм з відкритим ключем, який використовується для шифрування повідомлень і цифрових підписів, вживання якого не обмежено патентами США.

Приклад 2.16 Нехай $p = 11$, $q = 3$, $k_a = 7$, $k_b = 4$, $M = 6$.

Відкритий ключ, що надсилається стороні B, становить

$$Y_a \equiv 3^7 \pmod{11} \equiv 2187 \pmod{11} \equiv 9.$$

Повідомлення, зашифроване на стороні B, має вигляд

$$Y_b \equiv 3^4 \pmod{11} \equiv 81 \pmod{11} \equiv 4;$$
$$Y_a^{k_b} \equiv 9^4 \pmod{11} \equiv 5;$$
$$C = 110 \oplus 101 = 011.$$

Сторона A, отримавши зашифроване повідомлення у формі Y_b та C, розшифровує його:

$$Y_b^{k_a} \equiv 4^7 \pmod{11} \equiv 5;$$
$$M = 011 \oplus 101 = 110.$$

Вправа

Зашифрувати повідомлення $M = 14$ та розшифрувати криптограму за допомогою алгоритму Ель–Гамаля.

№ варіанта	p	q	k_a	k_b	№ варіанта	p	q	k_a	k_b
1	23	11	14	21	16	23	14	17	16
2	29	11	14	21	17	29	14	17	16
3	31	11	14	21	18	31	14	17	16
4	37	11	14	21	19	37	14	17	16
5	43	11	14	21	20	43	14	17	16
6	23	12	15	19	21	23	15	18	13
7	29	12	15	19	22	29	15	18	13
8	31	12	15	19	23	31	15	18	13
9	37	12	15	19	24	37	15	18	13
10	43	12	15	19	25	43	15	18	13
11	23	13	16	18	26	23	16	21	12
12	29	13	16	18	27	29	16	21	12
13	31	13	16	18	28	31	16	21	12
14	37	13	16	18	29	37	16	21	12
15	43	13	16	18	30	43	16	21	12

2.11 Алгоритм Діффі–Хеллмана

Діффі й Хеллман запропонували 1976 року [21] алгоритм для створення криптографічних систем з відкритим ключем, який так само, як і алгоритм Ель–Гамаля, базується на складності обчислення дискретного логарифма. Алгоритм Діффі–Хеллмана може бути використано задля розподілу ключів (генерування секретного ключа), але його не можна використовувати для шифрування повідомлення.

Згідно з цим алгоритмом, учасники інформаційного процесу A та B домовляються щодо значення великого простого числа p і простого дискретного кореня цього числа a (рис. 2.4).

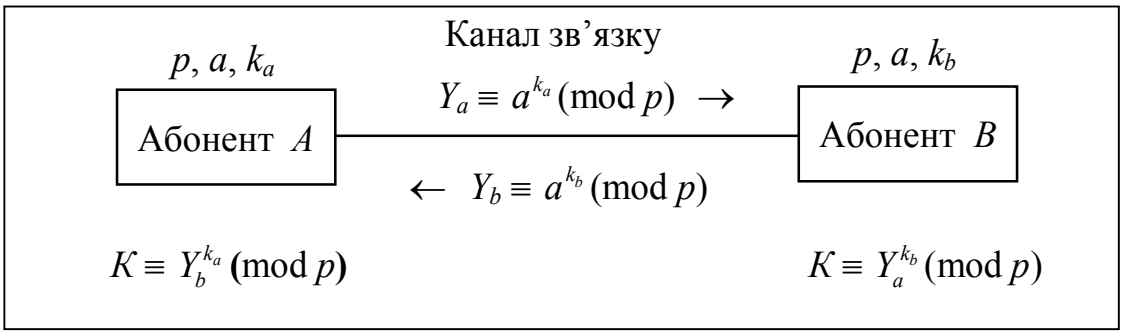


Рисунок 2.4 – Обмін ключами з використанням алгоритму Діффі–Хеллмана

Сторона A обирає випадкове число k_a, а сторона B – випадкове число k_b у такий спосіб, щоби виконувалися умови

$$1 < k_a < p - 1 \quad \text{та} \quad 1 < k_b < p - 1.$$

Числа k_a та k_b тримаються сторонами A та B в секреті.

Сторона A формує відкритий ключ за правилом

$$Y_a \equiv a^{k_a} (\mathrm{mod}\, p).$$

Аналогічно сторона B формує відкритий ключ за правилом

$$Y_b \equiv a^{k_b} (\mathrm{mod}\, p).$$

Після обміну несекретними ключами Y_a та Y_b сторони обчислюють значення секретного числа K:

$$K \equiv Y_a^{k_b} (\mathrm{mod}\, p) \equiv a^{k_a k_b} (\mathrm{mod}\, p);$$
$$K \equiv Y_b^{k_a} (\mathrm{mod}\, p) \equiv a^{k_b k_a} (\mathrm{mod}\, p).$$

Здобуте число K для ймовірного зловмисника є секретним, оскільки розв'язання рівнянь Y_a та Y_b для великих чисел є неможливе.

Алгоритм Діффі–Хеллмана можна поширити на випадок з трьома і більше учасниками [8].

Приклад 2.17 Нехай $p = 13$, $a = 7$, $k_a = 3$, $k_b = 4$.

Відкритий ключ, який надсилає сторона A, є

$$Y_a \equiv 7^3 \pmod{13} \equiv 343 \pmod{13} \equiv 5.$$

Відкритий ключ, який надсилає сторона B, є

$$Y_b \equiv 7^4 \pmod{13} \equiv 2401 \pmod{13} \equiv 9.$$

Секретне число, обчислюване обома сторонами, є

$$K \equiv 5^4 \pmod{13} \equiv 625 \pmod{13} \equiv 1;$$
$$K \equiv 9^3 \pmod{13} \equiv 729 \pmod{13} \equiv 1.$$

Приклад 2.18 Нехай $p = 199$, $a = 50$, $k_a = 13$, $k_b = 27$.

Відкритий ключ, надісланий стороною A, є

$$Y_a \equiv 50^{13} \pmod{199} \equiv (50^8 \cdot 50^4 \cdot 50) \bmod 199 \equiv (49 \cdot 7 \cdot 50) \bmod 199 \equiv 36.$$

Відкритий ключ, надісланий стороною B, є

$$Y_b \equiv 50^{27} \pmod{199} \equiv (50^{16} \cdot 50^8 \cdot 50^3) \bmod 199 \equiv (13 \cdot 49 \cdot 28) \bmod 199 \equiv 125.$$

Секретне число, обчислюване обома сторонами, є

$$K \equiv 36^{27} \pmod{199} \equiv (36^{16} \cdot 36^8 \cdot 36^3) \bmod 199 \equiv (115 \cdot 151 \cdot 90) \bmod 199 \equiv 103;$$
$$K \equiv 125^{13} \pmod{199} \equiv (125^8 \cdot 125^4 \cdot 125) \bmod 199 \equiv (63 \cdot 62 \cdot 125) \bmod 199 \equiv 103.$$

Алгоритм обміну ключами Діффі–Хеллмана запатентовано у США та Канаді. Ліцензію на цей патент разом з іншими патентами в області криптографії з відкритим ключем отримала група Public Key Partners (PKP). Термін дієздатності патенту США закінчився 1997 року. Алгоритм Діффі–Хеллмана також працює в комутативних кільцях. Шмулі та Кевін МакКерлі запропонували варіант алгоритму, в якому модуль є складеним числом. Міллер і Ніл Кобліц розширили цей алгоритм для використання з еліптичними кривими. Ель–Гамаль використовував основоположну ідею алгоритму Діффі–Хеллмана для розроблення алгоритму шифрування та цифрового підпису.

Алгоритм Діффі–Хеллмана також працює в полі Галуа. У ряді реалізаціях використовується цей підхід, тому що обчислення виконуються набагато швидше, але важливо ретельно вибирати поле досить велике, щоб забезпечити потрібну стійкість. Криптографічна стійкість алгоритму Діффі–Хеллмана, заснована на передбачуваній складності проблеми дискретного логарифмування (discrete logarithm problem). Однак, хоча вміння вирішувати проблему дискретного логарифмування дозволить зламати алгоритм Діффі–Хеллмана, зворотне твердження ще не доведене. Необхідно відзначити, що алгоритм Діффі–Хеллмана працює тільки на лініях зв'язку, надійно захищених від модифікації. Однак, у тих випадках, коли в каналі можлива модифікація даних, з'являється можливість атаки „людина посередині" (main-in-the-middle attack).

Вправа

Завдання на генерування ключа методом Діффі–Хеллмана. Число $p = 101$.

№ варіанта	a	k_a	k_b	№ варіанта	a	k_a	k_b
1	50	11	12	16	70	11	12
2	50	13	14	17	70	13	14
3	50	15	16	18	70	15	16
4	50	17	18	19	70	17	18
5	50	19	20	20	70	19	20
6	40	11	12	21	55	11	12
7	40	13	14	22	55	13	14
8	40	15	16	23	55	15	16
9	40	17	18	24	55	17	18
10	40	19	20	25	55	19	20
11	60	11	12	26	45	11	12
12	60	13	14	27	45	13	14
13	60	15	16	28	45	15	16
14	60	17	18	29	45	17	18
15	60	19	20	30	45	19	20

2.12 Криптосистема на еліптичних кривих

Загальні положення

Криптосистеми на *еліптичних кривих* [22] належать до класу криптосистем з відкритим ключем. Їхня безпека базується здебільшого на складності розв'язування задачі дискретного логарифмування у групі точок еліптичної кривої над скінченним полем. Цим зумовлено їхню потужну криптостійкість порівняно з іншими алгоритмами. Існують стійкі криптоалгоритми на еліптичних кривих, базовані на труднощах розкладання великих цілих чисел, коли еліптична крива задається над скінченним кільцем за складеним модулем, але вони зустрічаються дуже рідко. Проте слід зауважити, що криптостійкість є відносним поняттям, пов'язаним з поняттям найоптимальнішого відомого алгоритму зламу системи.

Еліптичні криві – математичний об'єкт, який може бути визначено над яким завгодно полем. У криптографії зазвичай використовуються скінченні поля. Для точок на еліптичній кривій вводиться операція складання, яка відіграє ту саму роль, що й операція множення у криптосистемах RSA та Ель–Гамаля.

Іншою перевагою криптосистем на еліптичних кривих є висока швидкість опрацьовування інформації. Але й тут не все так просто. Зрозуміло, що, маючи потужну криптостійкість, криптосистеми на еліптичних кривих дозволяють використовувати ключ меншої довжини. Проте, прийнятна для роботи в мережах, швидкість обчислень досягається лише при використанні спеціалізованих обчислювачів (це цілком природно для криптосистем з відкритим ключем) та полів спеціальних характеристик.

Криптосистеми на еліптичних кривих, як і інші криптосистеми з відкритим ключем, недоцільно застосовувати для шифрування великих обсягів даних. Проте, їх можна ефективно використовувати для систем цифрового підпису та ключового обміну. З 1998 року використовування еліптичних кривих для розв'язування криптографічних завдань, таких як цифровий підпис, було закріплено у стандартах США ANSI X9.62 та FIPS 186–2, а 2001 року аналогічний стандарт ГОСТ Р34.10–2001 було ухвалено в Росії. В Україні ухвалений стандарт цифрового підпису базується на еліптичних кривих ДСТУ 4145–2002 (див. додаток Е). Зазначимо, що безпека таких систем цифрового підпису спирається не лише на стійкість алгоритму на еліптичних кривих, але й на стійкість використовуваної геш-функції.

Численні дослідження засвідчили, що криптосистеми на підставі еліптичних кривих перевершують інші системи з відкритим ключем за двома важливими параметрами: мірою захищеності з розрахунку на кожен біт ключа та за швидкодією програмної і апаратної реалізації. Це пояснюється тим, що для обчислення обернених функцій на еліптичних кривих відомі лише алгоритми з експоненційним зростанням трудомісткості, тоді як для звичайних систем запропоновано субекспоненційні методи. Як наслідок, той рівень стійкості, який досягається, скажімо, в RSA при використанні 1024-бітових модулів, в системах на еліптичних кривих зреалізовується при розмірі модуля 160 біт, що забезпечує простішу як програмну, так і апаратну реалізацію.

Детальне вивчення еліптичних кривих потребує знань вищої алгебри, більше алгебричної геометрії. Проте далі матеріал викладатиметься при можливості без залучення складних конструкцій алгебри і в обсязі, достатньому для розуміння принципів побудови та функціонування відповідних криптосистем. Детальніше викладення теорії еліптичних кривих та їхнього використання у криптографії може бути знайдене, наприклад, в роботах [22, 23, 24].

Група точок еліптичної кривої

Еліптичні криві (ECC – Elliptic curve cryptography) – це не є еліпси. Вони так називаються лише тому, що описуються кубічними рівняннями, подібними до тих, які використовуються для обчислювання кривої еліпса. Взагалі кубічні рівняння для еліптичних кривих мають форму

$$y^2 + axy + by = x^3 + cx^2 + dx + e,$$

де a, b, c, d та e є дійсними числами, які задовольняють певним простим умовам. Означення еліптичної кривої включає також певний елемент, який позначається O й називається *невласним елементом* (а також *нескінченним,* або *нульовим елементом*). Такі рівняння називаються *кубічними*, або *рівняннями третього порядку*, оскільки в них найвищий показник степеня становить 3.

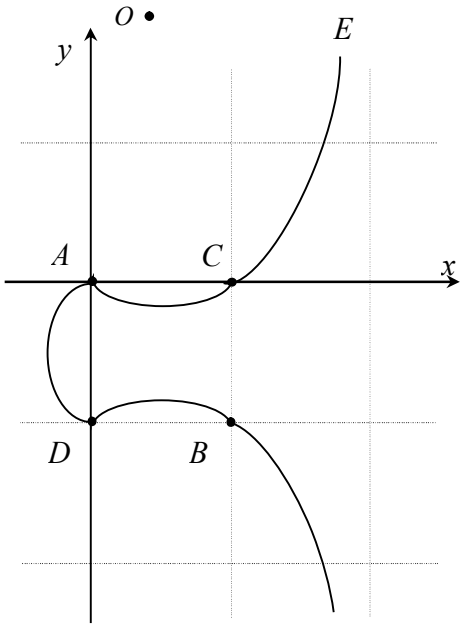


Рисунок 2.5 – Група з п’яти точок еліптичної кривої E, O – нескінченно віддалена точка

Розглянемо еліптичну криву E (рис. 2.5), яка відповідає рівнянню $y^2 + y = x^3 - x^2$. На цій кривій лежать лише чотири точки, координати яких є цілими числами. Це точки $A(0, 0)$, $B(1, -1)$, $C(1, 0)$, $D(0, -1)$.

Для визначення операції складання на групі точок еліптичної кривої вважатимемо, що:

– на площині існує нескінченно віддалена точка $O \in E$, в якій збігаються всі вертикальні прямі;

– дотична до кривої перетинає точку дотику P двічі.

Тепер можна сформулювати правила складання точок $P, Q \in E$ (рис. 2.6):

– проведемо пряму лінію через точки P та Q, знайдемо третю точку S перетинання цієї прямої з кривою E;

– знайдемо через точку S вертикальну пряму до перетинання з кривою E у точці T;

– шукана сума є $P + Q = T$.

Застосувавши ці правила до групи точок $G = \{A, B, C, D, O\}$, знайдемо (рис. 2.7) $A + A = B$, $A + B = C$, $A + C = D$, $A + D = O$, або $2A = B$, $3A = C$, $4A = D$, $5A = O$, $6A = A$.

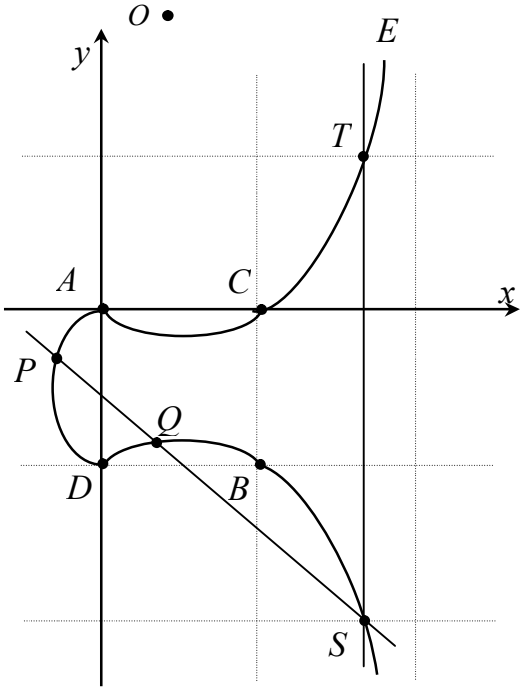


Рисунок 2.6 – Складання точок на еліптичній кривій $P + Q = T$

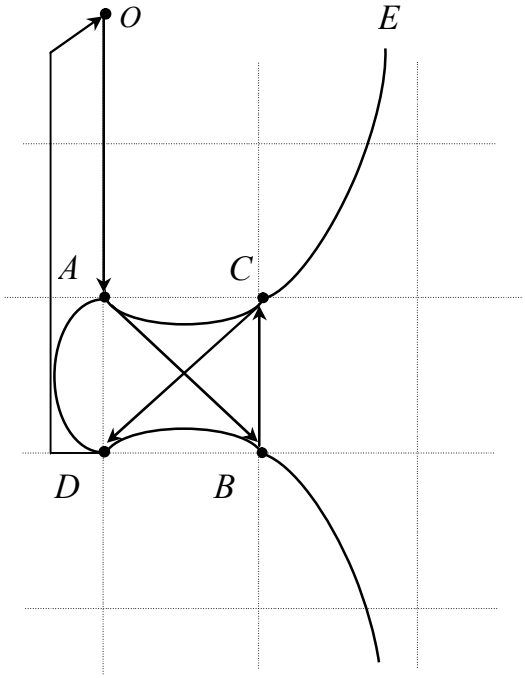


Рисунок 2.7 – Адаптивна абелева група $\{A, B, C, D, O\}$ на еліптичній кривій E

Для кожної з точок $P, Q \in G$ справедливе відношення $P + Q = Q + P$. Для кожної точки $P \in G$ справедливе є $P + O = P$; інакше кажучи, точка O – адитивний одиничний елемент групи G.

Еліптична крива над полем GF(p)

У реальних криптосистемах використовується рівняння $y^2 \equiv (x^3 + ax + b) \bmod p$, де $a, b \in GF(p)$; $(4a^3 + 27b^2) \bmod p \neq 0$; $p > 3$ – просте. Група $E(GF(p))$ складається з усіх точок (x, y); $x, y \in GF(p)$, які задовольняють рівнянню, і нескінченно віддаленої точки O.

Множина $E_p(a, b)$ складається з усіх точок (x, y), $x \geq 0$, $p > y$, які задовольняють рівнянню $y^2 \equiv (x^3 + ax + b) \bmod p$, й точки в нескінченності O. Кількість точок в $E_p(a, b)$ позначатимемо $\#E_p(a, b)$. Ця величина має важливе значення для криптографічних додатків еліптичних кривих.

Визначену над точками з $E(GF(p))$ операцію складання алгебрично може бути описано в такий спосіб:

1) $P + O = O + P = P$;

2) якщо $P = (x, y)$, тоді $P + (x, -y) = O$. Точка $(x, -y)$ є від'ємним значенням точки P і позначається $-P$. Зазначимо, що $(x, -y)$ лежить на еліптичній кривій і належить до $E_p(a, b)$. Наприклад, у разі $E_{23}(1, 1)$ для $P = (13, 7)$ матимемо $-P = (13, -7)$. Але $-7 \bmod 23 \equiv 16$, отже $-P = (13, 16)$;

3) якщо $P = (x_1, y_1)$ і $Q = (x_2, y_2)$, то $P + Q = (x_3, y_3)$ визначається згідно з правилами

$$x_3 \equiv (\lambda^2 - x_1 - x_2) \bmod p;$$
$$y_3 \equiv (\lambda\,(x_1 - x_3) - y_1) \bmod p,$$

де

$$\lambda \equiv \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} (\bmod p), & \text{за } P \neq Q; \\ \dfrac{3x_1^2 + a}{2y_1} (\bmod p), & \text{за } P = Q. \end{cases}$$

Число λ – кутовий коефіцієнт січної, проведеної через точки $P = (x_1, y_1)$ та $Q = (x_2, y_2)$. За $P = Q$ січна перетворюється на дотичну, чим і пояснюється наявність двох формул для обчислення λ.

Приклад 2.19 Розглянемо криву

$$E_7\,(2, 6):\ y^2 \equiv (x^3 + 2x + 6) \bmod 7.$$

Перевіримо умову:

$$(4 \cdot 2^3 + 27 \cdot 6^2) \bmod 7 \equiv 3 \neq 0\ .$$

Отже, дана крива є несингулярна. Знайдемо певну (випадкову) точку в $E_7(2, 6)$. Нехай $x = 5$, тоді

$$y^2 \equiv (5^3 + 2 \cdot 5 + 6) \bmod 7 \equiv (125 + 10 + 6) \bmod 7 \equiv 1 \pmod 7$$

і $y \equiv 1 \pmod 7$ або $y \equiv -1 \equiv 6 \pmod 7$. Маємо одразу дві точки: (5, 1) та (5, 6).

Знайдемо ще пару точок шляхом обчислення композиції. Спочатку знайдемо 2(5, 1):

$$\lambda = \frac{3 \cdot 5^2 + 2}{2 \cdot 1} = \frac{0}{2} \equiv 0 \pmod 7;$$
$$x_3 = 0 - 2 \cdot 5 \equiv 4 \pmod 7;$$
$$y_3 = 0(5 - 4) - 1 \equiv 6 \pmod 7.$$

Одержали $2(5, 1) = (4, 6)$ (можна переконатися, що відшукану точку розташовано на кривій, підставивши її координати до рівняння). Знайдемо ще одну точку $3(5, 1) = (5, 1) + (4, 6)$:

$$\lambda = \frac{6 - 1}{4 - 5} = -\frac{5}{1} \equiv 2 \pmod 7;$$
$$x_3 = 2^2 - 5 - 4 \equiv 2 \pmod 7;$$
$$y_3 = 2(5 - 2) - 1 \equiv 5 \pmod 7.$$

Одержали $3(5, 1) = (2, 5)$.

Отже, знайдено чотири точки. Для криптографічного використовування кривої важливо знати, скільки всього точок містить множина $E_7(2, 6)$.

Приклад 2.20 Нехай $p = 23$.

Розглянемо еліптичну криву E: $y^2 = x^3 + x + 1$. $E_{23}(1, 1)$ складається з точки O, а також з точок (0, 1); (0, 22); (1, 7); (1, 16); (3, 10); (3, 13); (4, 0); (5, 4); (5, 19); (6, 4); (6, 19); (7, 11); (7, 12); (9, 7); (9, 16); (11, 3); (11, 20); (12, 4); (12, 19); (13, 7); (13, 16); (17, 3); (17, 20); (18, 3); (18, 20); (19, 5); (19, 18).

Нехай $P = (3, 10)$ і $Q = (9, 7)$. Знайдемо $P + Q$ і $2P$. Нехай $P + Q = (x_3, y_3)$, тоді

$$\lambda = \frac{7 - 10}{9 - 3} = -\frac{1}{2} \equiv 11 \pmod{23};$$
$$x_3 = 121 - 3 - 9 = 109 \equiv 17 \pmod{23} \equiv -6 \pmod{23};$$
$$y_3 = 11(3 + 6) - 10 = 89 \equiv 20 \pmod{23}.$$

Отже, $P + Q = (17, 20)$. Знайдемо $2P = P + P = (x_3, y_3)$. Тоді

$$\lambda = \frac{3 \cdot 9 + 1}{20} = \frac{1}{4} \equiv 6 \pmod{23};$$
$$x_3 = 36 - 6 = 30 \equiv 7 \pmod{23};$$
$$y_3 = 6(3 - 7) - 10 = -34 \equiv 12 \pmod{23}.$$

Отже, $2P = (7, 12)$.

Приклад 2.21 Нехай $p = 5$, $a = b = 1$, $4a^3 + 27b^2 = 4 + 27 \equiv 31 \pmod 5 \neq 0$.

Розглянемо еліптичну криву $y^2 = x^3 + ax + b$. $E_5(1, 1)$ складається з точок: $P = (0, 1)$; $2P = (4, 2)$; $3P = (2, 1)$; $4P = (3, 4)$; $5P = (3, 1)$; $6P = (2, 4)$; $7P = (4, 3)$; $8P = (0, 4)$; $9P = O$, при цьому $10P = P$ (рис. 2.8).

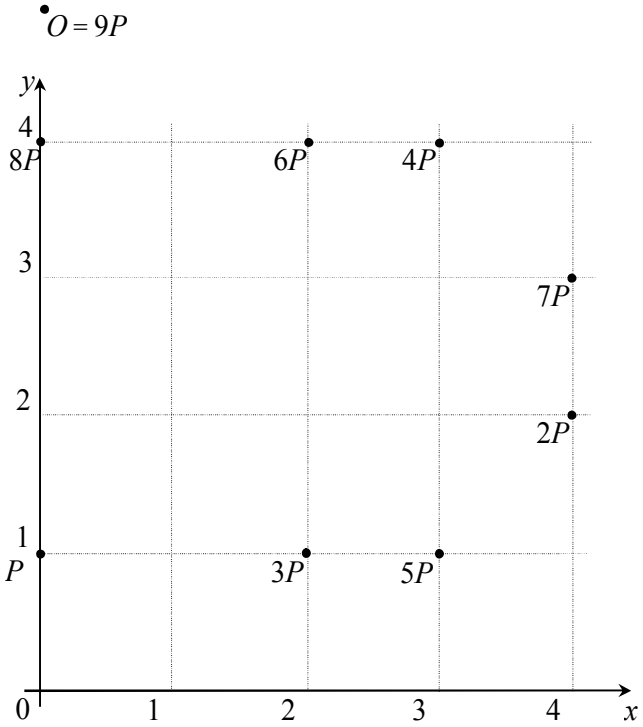


Рисунок 2.8 – Адитивна група $E(Z_5)$

Вибір параметрів кривої

Розглянемо основні рекомендації щодо вибору параметрів еліптичної кривої, призначеної для розв'язання криптографічних завдань, а саме щодо вибору коефіцієнтів a, b й модуля p. Фактично критерієм вибору є неможливість здійснення певного роду атак, пропонованих для певних класів кривих. Рекомендації, викладені нижче, виходять із стратегії вибору випадкової кривої. Ця стратегія вважається за найбільш надійну з точки зору забезпечення стійкості результатів криптосистеми. Альтернативний підхід, тут не розглядуваний, полягає у систематичному конструюванні кривої із заданими властивостями, що зазвичай стає ефективнішим з обчислювальної точки зору. Для реалізації цього підходу запропоновано спеціальні методи, але здобуті криві фактично обираються з відносно невеликого класу і викликають підозри щодо наявності певних специфічних властивостей, які можуть надати можливість з часом відшукати алгоритми для їхнього зламу.

Опишемо процес формування випадкової кривої.

1) Обираємо довільно просте число p. Бітова довжина числа p, $t = \lfloor \log p \rfloor + 1$, має бути такою, щоб унеможливити вживання загальних методів знаходження логарифмів на кривій, що мають трудомісткість $T(2^{t/2})$. Величина

$t = 128$ біт (чотири машинні слова на 32-бітових комп'ютерах) сьогодні є недостатня, оскільки існують повідомлення про злам відповідних кривих. Інше міркування базується на тому, що шифр на еліптичній кривій має бути не менш стійким, ніж блоковий шифр AES (Advanced Encryption Standard). Вважається, що стійкість AES забезпечується повною довжиною його ключа, яка становить 128, 196 та 256 біт. Оскільки стійкість шифру на еліптичній кривій визначається величиною $t/2$, довжина модулів еліптичних кривих має становити відповідно 256, 392 та 512 біт.

2) Обираємо випадкові числа a та b такі, що $a, b \pmod p \neq 0$ і $(4a^3 + 27b^2) \bmod p \neq 0$. Звернімо увагу на те, що при обчисленні композиції точок параметр b ніде не фігурує. Тому для підвищення ефективності підрахунку інколи рекомендують випадково обирати лише b, а за a приймати невелике ціле число. Приміром, стандарт США FIPS 186–2 передбачає використовування кривих з параметром $a = -3$, що спрощує обчислення.

3) Визначаємо кількість точок на кривій $n = \#E_p(a, b)$ (це є найтрудомісткіший етап опису процесу). Важливо, щоб n мало великого простого дільника q, а найоптимальніше, саме було б простим числом: $n = q$. Якщо n розкладається на невеликі множники, то в $E_p(a, b)$ існує багато невеликих підмножин з власними генераторами і алгоритм Поліга–Хеллмана [25] швидко обчислює логарифм на кривій через логарифми в цих невеликих підмножинах. Якщо пошук кривої з $n = q$ потребує надто багато часу, то можна припустити $n = hq$, де h – невелике число. Знову підкреслимо, що стійкість криптосистеми на еліптичній кривій визначається не модулем p, а кількістю елементів q у підмножині точок кривої. Але якщо множник h – невелике число, то q є величиною того самого порядку, що й p. Якщо n не відповідає вимогам, то слід повернутися до кроку 2.

4) Перевіряємо, чи виконуються нерівності $(p^k - 1) \bmod q \neq 0$ для всіх k, $0 < k < 32$. Якщо ні, то повертаємося до кроку 2. Ця перевірка запобігає можливості MOV-атаки (названої за прізвищами її авторів Menezes, Okamoto, Vanstone), а також дозволяє вилучити з розгляду так звані суперсингулярні криві та криві з $\#E_p(a, b) = p - 1$ [26, 27]. Метод MOV і згадані особливі типи кривих дозволяють звести завдання обчислення логарифма на кривій до простіших задач.

5) Перевіряємо, чи виконується нерівність $q \neq p$. Якщо ні, то повертаємося до кроку 2. Річ у тім, що для кривих з $q = p$, названих аномальними, існують ефективні методи обчислювання логарифмів [26, 27].

6) На даному кроці відповідну для криптографічних додатків криву здобуто. Маємо параметри p, a, b, кількість точок n і розмір підмножини точок q. Зазвичай ще потрібно знайти точку G – генератор цієї підмножини. Якщо $q = n$, то кожна точка (окрім O) є генератором. Якщо $q < n$, то обираємо

випадкові точки G', поки не дістанемо $G = [n/q]G' \neq O$. Щоб знайти випадкову точку на кривій, обираємо випадкове число $x < p$, обчислюємо $e \equiv (x^3 + ax + b) \bmod p$ і робимо спробу обчислити квадратний корінь $y \equiv \sqrt{e} \pmod p$. Якщо корінь існує, то дістанемо точку (x, y), інакше – обираємо інше число x. Алгоритми обчислювання квадратного кореня за модулем простого числа подано в роботі [25].

Завдання, яке має розв'язувати криптоаналітик, використовуючи криптосистеми на базі еліптичних рівнянь, називається *завданням дискретного логарифмування на еліптичній кривій* і формулюється в такий спосіб. Задано точки P та Q на еліптичній кривій порядку n, де n – кількість точок на кривій. Треба визначити єдину точку x таку, що $P = xQ$.

Розглянемо використання еліптичних кривих у криптографії.

Обмін ключами за схемою Діффі–Хеллмана

Обмін ключами з використанням еліптичних кривих може бути здійснено у такий спосіб. Спочатку обираються просте число p і параметри a та b для еліптичної кривої. Це задає еліптичну групу точок $E_p(a, b)$. Потім в $E_p(a, b)$ обирається *генерувальна точка* $G = (x, y)$. При обиранні G є важливо, щоб найменше значення n, при якому $nG = O$, було б надто великим простим числом. Параметри $E_p(a, b)$ та G криптосистеми – це параметри, відомі усім учасникам. Обмін ключами між користувачами A і B можна провести за поданою нижче схемою.

1) Сторона A обирає ціле число k_a, яке є менше за n. Це число буде власним ключем учасника A. Потім учасник A генерує відкритий ключ $Y_a = k_aG$. Відкритий ключ є певною точкою з $E_p(a, b)$.

2) Так само учасник B обирає власний ключ k_b і обчислює відкритий ключ $Y_b = k_bG$.

3) Учасник A генерує секретний ключ $K = k_aY_b$, а учасник B – секретний ключ $K = k_bY_a$.

Обидві подані формули дають однаковий результат, оскільки

$$k_aY_b = k_a(k_bG) = k_b(k_aG) = k_bY_a.$$

Приклад 2.22 Нехай $p = 211$, $G = (2, 2)$, $E_p(0, -4)$, що відповідає кривій $y^2 = x^3 - 4$. Можна підрахувати, що $241G = O$. Власним ключем користувача A є $k_a = 121$, тому відкритим ключем A буде

$$Y_a = 121(2, 2) = (115, 48).$$

Власним ключем користувача B є $k_b = 203$, тому відкритим ключем B буде

$$Y_b = 203(2, 2) = (130, 203).$$

Спільний секретний ключ

$$K = 121(130, 203) = 203(115, 48) = (161, 69).$$

Зверніть увагу на те, що спільний секретний ключ є парою чисел. Якщо цей ключ передбачається використовувати як сеансовий ключ для традиційного шифрування, то з цієї пари чисел треба генерувати одне відповідне значення. Можна, наприклад, використовувати просто координату x чи певну просту функцію від x.

У літературі можна знайти аналіз кількох підходів щодо зашифровування/розшифровування, які передбачають використання еліптичних кривих. Розглянемо найпростіший з цих підходів.

Першим завданням у згаданій системі є зашифровування відкритого тексту повідомлення M, яке надсилатиметься у вигляді значення (x, y) для точки P_M. Тут точка P_M містить зашифрований текст, який згодом розшифровуватиметься.

Користувач A обирає власний ключ k_a і генерує відкритий ключ Y_a. Щоб зашифрувати й надіслати повідомлення P_M користувачеві B, користувач A обирає довільне додатне ціле число r і обчислює зашифрований текст C_M, який складається з пари точок $C_M = (rG, P_M + rY_b)$.

Зазначимо, що сторона A використовує відкритий ключ Y_b сторони B. Для того, щоб розшифрувати цей шифртекст, сторона B помножує першу точку в парі на секретний ключ B і віднімає результат з другої точки:

$$P_M + rY_b - k_b\,(rG) = P_M + r(k_bG) - k_b\,(rG) = P_M.$$

Користувач A замаскував повідомлення P_M за допомогою додавання до нього rY_b. Нікому, окрім цього користувача, невідоме значення r, тому, хоча Y_b й є відкритим ключем, ніхто не зможе усунути маску rY_b. Проте користувач A розмістив у повідомленні й „підказку”, якої вистачить, щоб усунути маску тому, хто має власний ключ k_b. Криптоаналітикові для відновлення повідомлення доведеться обчислити r за наведеними G та rG, що становить собою складне завдання.

Приклад 2.23 Розглянемо випадок $p = 751$, $G = (0, 376)$, $E_p(-1, 188)$, що відповідає кривій $y^2 = x^3 - x + 188$.

Припустімо, що користувач A збирається доправити користувачеві B повідомлення, кодоване еліптичною точкою $P_M = (562, 201)$. Для цього користувач A обирає довільне число $r = 386$ і відшукує відкритий ключ користувача $B - Y_b = (201, 5)$. Обчислює $386(0, 376) = (676, 558)$ та $(562, 201) + 386(201, 5) = (385, 328)$. Отже, користувач A повинен мати повідомлення $\{(676, 558), (385, 328)\}$.

Безпека, гарантована криптографічним підходом на базі еліптичних кривих, залежить від того, наскільки складним для розв’язання буде завдання щодо визначення r за даними rP та P. Це завдання зазвичай називають

проблемою логарифмування на еліптичній кривій. Найбільш швидким з відомих сьогодні методів логарифмування на еліптичній кривій є так званий ρ-метод Полларда (Pollard) [28].

Протокол Мессі–Омури

Протокол Мессі–Омури дозволяє передавати повідомлення абонента *A* абонентові *B* відкритим каналом зв'язку без попередньої передачі будь-якої ключової інформації (рис. 2.9).

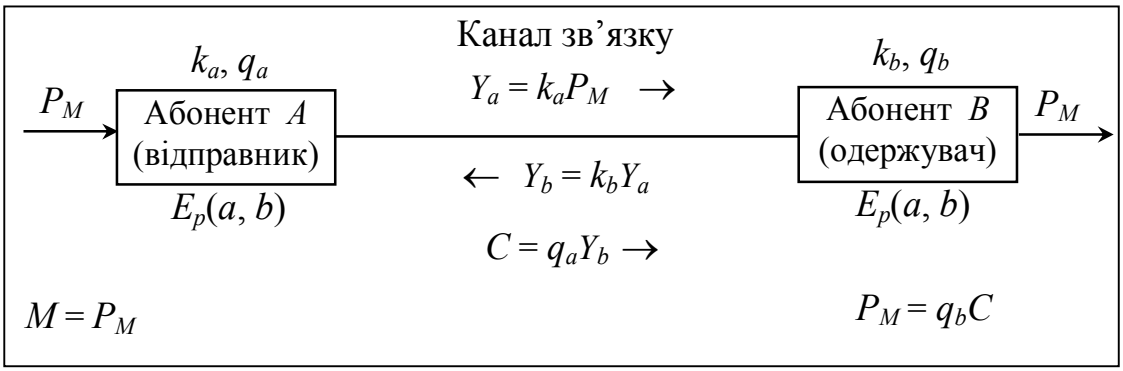


Рисунок 2.9 – Протокол Мессі–Омури

Нехай $E_p(a, b)$ – еліптична крива, відома учасникам інформаційного процесу. Абонент *A* обирає ключ зашифровування k_a взаємно простий з *n*, за якого $nG = O$, і обчислює обернене число:

$$q_a \equiv \left(k_a^{-1}\right) \bmod n.$$

Аналогічно, абонент *B* обирає число k_b взаємно просте з *n* і обчислює q_b, тобто створює власний ключ:

$$q_b \equiv \left(k_b^{-1}\right) \bmod n.$$

Абонент *A* розміщує власне повідомлення *M* у певній точці P_M високого порядку еліптичної кривої і, помноживши її на власне секретне значення k_a, відшукує точку

$$Y_a = k_a P_M.$$

Зміст повідомлення Y_a абонент *A* надсилає абонентові *B*. Абонент *B* обчислює

$$Y_b = k_b Y_a$$

і надсилає результат Y_b абонентові *A*, який знімає власний „замок", обчислюючи

$$C = q_a Y_b,$$

і повертає здобуту точку *C* абонентові *B*.

Абонент *B* розшифровує повідомлення P_M, тобто помножує отриману від абонента *A* точку *C* на власний секретний ключ:

$$P_M = q_b C.$$

Дійсно, із урахуванням комутативності та асоціативності операції групи,

$$P_M = q_b C = (q_a q_b) Y_b = (q_a q_b k_b) Y_a = (q_a q_b k_b k_a) P_M = (k_a q_a)(k_b q_b) P_M = P_M.$$

Повідомлення M, „вкладене” в точку P_M, може бути використано як ключ симетричної криптосистеми. Зауважимо, що в даному разі не потрібна публікація жодної інформації про параметри протоколу, окрім власне еліптичної кривої. Платнею за це є потреба триразового передавання відкритими каналами.

Приклад 2.24 Нехай $E_p(0, -4)$, $G = (2, 2)$, $241G = O$, $p = 211$, що відповідає кривій $y^2 = x^3 - 4$. Припустімо, що користувач A збирається доправити користувачеві B повідомлення, кодоване еліптичною точкою $P_M = (208, 179)$. Для цього користувач A обирає число $k_a = 5$ й визначає

$$q_a \equiv 5^{-1} (\mathrm{mod}\, 241) \equiv \left(5^{\varphi(241)-1}\right) \mathrm{mod}\, 241 \equiv 5^{239} \left(\mathrm{mod}\, 241\right) \equiv 193.$$

Аналогічно, сторона B обирає число $k_b = 7$ й обчислює

$$q_b \equiv 7^{-1} (\mathrm{mod} 241) \equiv \left(7^{\varphi(241)-1}\right) \mathrm{mod} 241 \equiv 7^{239} \left(\mathrm{mod} 241\right) \equiv 69$$

Згідно з протоколом визначаємо:

$$Y_a = 5(208, 179) = (150, 85);$$
$$Y_b = 7(150, 85) = (156, 201);$$
$$C = 193(156, 201) = (120, 180);$$
$$P_M = 69(120, 180) = (208, 179).$$

Отже, абонент B отримав передане повідомлення $P_M = (208, 179)$.

Шифр Ель–Гамаля на еліптичній кривій

Для користувачів обираються загальна еліптична крива $E_p(a, b)$ й точка G на ній такі, що G, $2G$, $3G$, ..., nG – різні точки і $nG = O$ для певного простого числа n.

Кожен користувач мережі обирає число k, $0 < k < n$, яке зберігає як власний секретний ключ, і обчислює точку на кривій $Y = kG$, яка слугуватиме за його відкритий ключ. Параметри кривої та перелік відкритих ключів передаються всім користувачам мережі.

Припустімо, що користувач A хоче передати повідомлення користувачеві B. Вважатимемо, що повідомлення подано у формі числа $M < p$. Користувач A виконує такі дії:

1) обирає випадкове число r, $0 < r < n$;
2) обчислює $R = rG$, $P = rY_b = (x, y)$;
3) зашифровує $C \equiv (Mx) \bmod p$;
4) надсилає користувачеві B шифртекст (R, C).

Користувач B після отримання шифртексту (R, C) виконує такі дії:

1) обчислює $Q = k_bR = (x, y)$;

2) розшифровує $M \equiv (Cx^{-1}) \bmod p$.

Подамо обґрунтування протоколу. Для цього достатньо буде зазначити, що

$$k_bR = k_b(rG) = r(k_bG) = rY_b,$$

тобто $Q = P$.

Координата x точки Q залишається секретною для криптоаналітика, оскільки йому невідоме число r. Криптоаналітик може спробувати обчислити r з точки P, але для цього йому потрібно розв'язати проблему дискретного логарифмування на кривій, що вважається за неможливе.

Найбільш імовірним варіантом використання розглянутого алгоритму буде передавання в якості числа M секретного ключа для блокового чи потокового шифру. В цьому разі доречно обирати параметри кривої у такий спосіб, щоб $\log n$ приблизно удвічі перевищував довжину ключа шифру.

Приклад 2.25 Нехай $p = 211$, $G = (2, 2)$, $E_p(0, -4)$, що відповідає кривій $y^2 = x^3 - 4$. Можна підрахувати, що $241G = O$, тоді обираємо $r = 43$. Власним ключем користувача B є $k_b = 91$, тому відкритим ключем B буде

$$Y_b = 91(2, 2) = (206, 121).$$

Користувач A хоче передати повідомлення $M = 25$ користувачеві B. Користувач A обчислює:

1) $R = 43(2, 2) = (124, 119)$;

2) $P = 43(206, 121) = (142, 15)$;

3) $C \equiv (25 \cdot 142) \bmod 211 \equiv 174$;

4) надсилає користувачеві B шифртекст $\{(124, 119), 174)\}$.

Користувач B обчислює:

1) $Q = 91(124, 119) = (142, 15)$;

2) $M \equiv (174 \cdot 142^{-1}) \bmod 211 \equiv (174 \cdot 159) \bmod 211 \equiv 25$.

Вправи

1 Для еліптичної кривої з параметрами $p = 7$, $a = 2$, $b = 6$ обчислити композиції точок $2(2, 2)$; $2(4, 6)$; $(1, 3) + (1, 4)$; $(3, 5) + (5, 1)$.

2 Зашифрувати повідомлення M та розшифрувати криптограму з використанням алгоритму Ель–Гамаля на еліптичній кривій:

а) $p = 211$, $G = (2, 2)$, $E_p(0, -4)$, $r = 2$, $k_b = 2$, $M = 2$;

б) $p = 751$, $G = (0, 376)$, $E_p(-1, 188)$, $r = 2$, $k_b = 2$, $M = 2$.

3 ГЕШ-ФУНКЦІЇ

3.1 Односпрямовані геш-функції

У всьому різноманітті проблем забезпечення інформаційної безпеки, що розв'язуються за допомогою криптографічних методів та засобів, завдання забезпечення цілісності та вірогідності переданої інформації є на сьогодні одним з найголовніших. З урахуванням сучасних вимог щодо інформаційно-телекомунікаційних систем – це завдання все частіше і частіше перетворюється на серйозну проблему. Надто актуальна вона є у фінансовій сфері, оскільки задля надійного функціонування платіжної системи необхідною умовою є зберігання цілісності та вірогідності усіх документів.

Невід'ємною частиною електронно-цифрового підпису є використовування геш-функцій. *Геш-функцією* (англ. hash – подрібнювати, змішувати) називається перетворення h, яка перетворює інформаційну послідовність M довільної довжини на послідовність фіксованої довжини $h(M)$, називану *геш-кодом*. Окрім того, геш-функції широко застосовуються і для розв'язування багатьох інших питань, пов'язаних із забезпеченням захисту потоків даних, наприклад для гешування паролів користувачів з метою подальшого їхнього шифрування та зберігання у базі даних. Цей метод застосовується в ОС Windows NT (використовується геш-функція MD4 разом з DES). Функція гешування може служити за криптографічну контрольну суму – код виявлення змін (MDC – Manipulation Detection Code) або для перевірки цілісності повідомлення (MIC – Message Integrity Check).

Однією з найважливіших характеристик геш-функцій, що зумовили їхнє широке впровадження у практику, виявилася здатність отримувати з відкритого тексту великої довжини (наприклад в геш-функції SHA максимальна довжина відкритого тексту обмежена 2^{64} бітами) геш-коду набагато меншою довжиною (у російському стандарті ГОСТ Р 34.11–94 довжина геш-коду становить 256 біт, західні геш-функції мають переважно геш-код довжиною 160...180 біт), що в певних випадках дозволяє дуже ефективно скоротити мережний трафік. Застосовування геш-функцій надає можливість усувати надлишковість відкритого тексту, що при подальшому криптографічному перетворенні геш-коду відкритого тексту позитивно позначається на криптографічних властивостях зашифрованого повідомлення.

До функції $h(M)$ ставляться такі вимоги:

– результат роботи геш-функцій має залежати від усіх двійкових символів вихідного повідомлення, а також від їхнього взаємного розташування;

– геш-функція має бути стійкою в розумінні звертання;

– геш-функція має бути стійкою в розумінні виявлення колізій.

Область використання геш-функцій:

– захист паролів при їхньому передаванні та зберіганні;

– формування контрольних кодів MDC;

– отримування стисненого зразка повідомлення перед формуванням електронного підпису;

– оперативний контроль перебігу програм.

Існує три методи побудови геш-функцій:

– на базі певної складно обчислюваної математичної задачі;

– на базі алгоритмів блокового шифрування;

– розроблення з нуля.

Кожен із зазначених методів має власні переваги й недоліки, однак, найбільш поширеними сьогодні є останні два. Це пов'язано з тим, що при побудові геш-функцій з нуля, виникає можливість враховувати таку їхню властивість, як ефективна програмна реалізація. Широке застосовування геш-функцій, побудованих на базі алгоритмів блокового шифрування, є наслідком ретельного опрацювання питання стійкості багатьох з існуючих алгоритмів.

Найбільш відомі алгоритми отримування геш-образів повідомлень – MD5, SHA, RIPEMD, ГОСТ Р 34.11–94, TIGER, HAVAL.

MD5 – представник сімейства алгоритмів обчислення геш-функцій MD (Message Digest Algorithm), запропонованого Р. Рівестом [29]; розроблено 1991 р.; перетворює інформаційну послідовність довільної довжини на геш-образ розрядністю 128 біт.

RIPEMD – розроблено в межах європейського проекту RIPE (Race Integrity Primitives Evaluation) Європейського співтовариства; є модифікацією алгоритму MD4; перетворює інформаційну послідовність довільної довжини на геш-образ розрядністю 128 (RIPEMD-128) або 160 біт (RIPEMD-160) [30].

TIGER – розроблено Р. Андерсоном та Е. Біхемом; призначений для реалізації на 64-розрядних комп'ютерах; перетворює інформаційну послідовність довільної довжини на геш-образ розрядністю 192 біти.

HAVAL – односпрямована геш-функція змінної довжини. Функція HAVAL є модифікацією функції MD5. Алгоритм HAVAL опрацьовує повідомлення блоками розміром у 1024 розряди, що є удвічі більше, аніж в алгоритмі MD5. У HAVAL використовується вісім 32-розрядних змінних зчеплення, тобто удвічі більше, аніж в алгоритмі MD5, і змінна кількість раундів опрацювання – від трьох до п'яти (на кожному раунді виконується 16 кроків). Функція HAVAL може видавати геш-значення обсягом у 128, 160, 192, 224 або 256 розрядів [30, 36].

Розглянемо два приклади практичної реалізації геш-функцій: SHA, побудованої з нуля, і ГОСТ Р 34.11–94 – на базі блочного алгоритму шифрування ГОСТ 28147–89.

3.2 Алгоритм стійкого гешування SHA

Алгоритм Secure Hash Algorithm (SHA – алгоритм стійкого гешування) є частиною стандарту SHS (Secure Hash Standard), прийнятого 1993 року Національним інститутом стандартів і технологій США (NIST), Агентством національної безпеки (АНБ) США [31].

Розглянемо версію алгоритму SHA-1, в якому здійснюється перетворення інформаційної послідовності довільної довжини на геш-образ розрядністю 160 біт, названий *згорткою повідомлення* (Message Digest).

Робота алгоритму розпочинається з того, що вхідна послідовність ділиться на блоки по 512 біт. Перед тим, як розбити її, потрібно, щоб довжини утворених блоків у бітовому подаванні дорівнювали 512 біт. Для цього до згаданої послідовності приписуються одиниця й потрібна кількість нулів, щоб її довжина стала на 64 біти менша за число, кратне до 512. Потім до послідовності приписується 64-бітове подання довжини вхідної послідовності. Нехай після доповнення здобута інформаційна послідовність матиме форму

$$M = m_1, m_2, \ldots, m_i, \ldots, m_n;\ \ i = \overline{1, n};\ \ |m_i| = 512.$$

Далі зініціалізовується п’ять 32-розрядних змінних:

$$A = 67452301\text{h};\ \ B = \text{EFCDAB89h};\ \ C = 98\text{BADCFEh};$$
$$D = 10325476\text{h};\ \ E = \text{C3D2E1F0h},$$

при цьому стартовий вектор гешування (синхронадсилання) є результат конкатенації цих змінних, тобто

$$\text{SHA}_0 = (A, B, C, D, E).$$

На вхід i-того циклу перетворення SHA_i надходить i-тий блок інформаційної послідовності та результат роботи попереднього циклу SHA_{i-1}, тобто

$$\text{SHA}_i = h(m_i, \text{SHA}_{i-1}).$$

Основний цикл, що чиниться над одним 512-бітовим блоком, складається з чотирьох раундів, кожен з яких включає по 20 операцій. Кожна операція становить собою набір нелінійних функцій від трьох змінних (B, C і D) та операцій циклічного зсуву й підсумовування. Ці функції мають таку форму:

1-й раунд	$f_1(B, C, D) = (B \wedge C) \vee (\overline{B} \wedge D)$	за $0 \le t \le 19$;
2-й раунд	$f_2(B, C, D) = B \oplus C \oplus D$	за $20 \le t \le 39$;
3-й раунд	$f_3(B, C, D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$	за $40 \le t \le 59$;
4-й раунд	$f_4(B, C, D) = B \oplus C \oplus D$	за $60 \le t \le 79$.

Очевидно, що використовуються лише три функції. Для $0 \le t \le 19$ функція є умовним поданням: якщо B, то C, інакше – D. Для $20 \le t \le 39$ і $60 \le t \le 79$ функція дає біт парності. Для $40 \le t \le 59$ функція є істинною, коли є істинні не

менше двох її аргументів.

Для кожного раунду визначається одна константа:

$$K_1 = \text{5A827999h} = [2^{30}\sqrt{2}\,] \quad \text{при } 0 \le t \le 19;$$
$$K_2 = \text{6ED9EBA1h} = [2^{30}\sqrt{3}\,] \quad \text{при } 20 \le t \le 39;$$
$$K_3 = \text{8F1BBCDCh} = [2^{30}\sqrt{5}\,] \quad \text{при } 40 \le t \le 59;$$
$$K_4 = \text{CA62ClD6h} = [2^{30}\sqrt{10}\,] \quad \text{при } 60 \le t \le 79,$$

де t – номер операції ($0 \le t \le 79$); $[2^{30}\sqrt{*}\,]$ – ціла частина числа.

Логіку циклу подано на рис. 3.1.

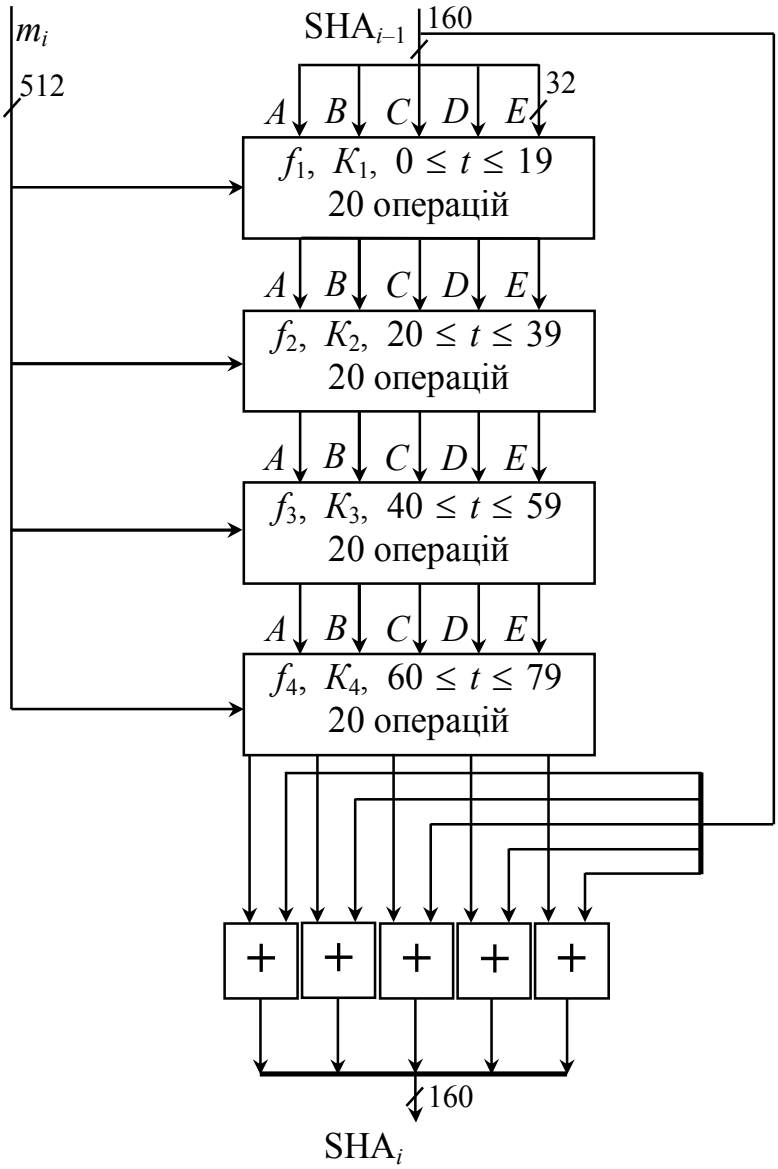


Рисунок 3.1 – Опрацювання одного 512-бітового блока в SHA-1

Всі чотири раунди мають подібну структуру, але в кожному застосовується власна логічна функція. У кожному раунді на вхід подається поточний 512-бітовий блок m_i і 160-бітове значення буфера *ABCDE*. Окрім того, на кожному кроці використовується додавана до поточного значення константа K_t.

Вихідне значення четвертого раунду (80-й крок) додається до вхідного значення першого раунду, внаслідок чого виходить SHA_i. Складання виконується за модулем 2^{32}. Після опрацювання всіх 512-бітових блоків на виході матимемо 160-бітовий профіль повідомлення.

Розглянемо логіку будь-якої з 80-ти операцій опрацювання одного 512-бітового блока. Кожна операція має форму (рис. 3.2)

$$A, B, C, D, E \leftarrow (E + f_t(B, C, D) + \mathrm{Rol}^5(A) + W_t + K_t), A, \mathrm{Rol}^{30}(B), C, D,$$

де $\mathrm{Rol}^5(A)$ – циклічний зсув ліворуч 32-бітового аргументу A на 5 біт;

W_t – 32-бітове слово, вилучене з поточного 512-бітового блока введення;

$\mathrm{Rol}^{30}(B)$ – циклічний зсув ліворуч 32-бітового аргументу B на 30 біт;

+ – додавання за модулем 2^{32}.

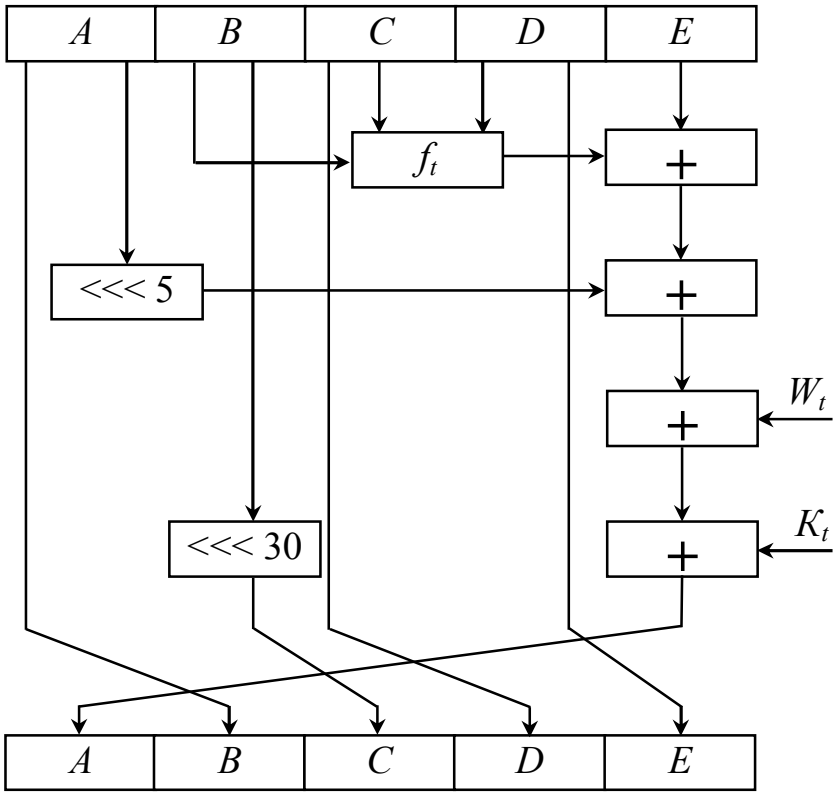


Рисунок 3.2 – Схема однієї операції SHA-1

Кожна з функцій f_t отримує на вході три 32-бітових слова і видає на виході одне 32-бітове слово. Значення всіх функцій подано в табл. 3.1.

Залишилося зазначити, у який спосіб з 512-бітового блока повідомлення m_i вилучаються 32-бітові значення слів W_t. Відповідну схему подано на рис. 3.3. Перші 16 значень W_t є безпосередньо 16-ма словами поточного блока. Решта значень відшукуються за формулою

$$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1 \quad \text{для} \quad 16 \le t \le 79.$$

Таблиця 3.1 – Значення логічних функцій SHA-1

B	C	D	$f_{0\ldots19}$	$f_{20\ldots39}$	$f_{40\ldots59}$	$f_{60\ldots79}$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

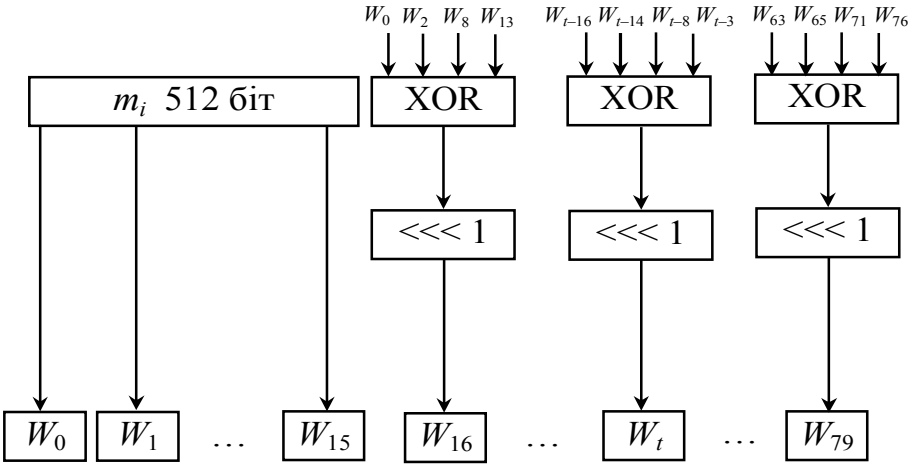


Рисунок 3.3 – Створення послідовності з 80-ти слів для опрацювання одного блока SHA-1

Отже, на перших 16-ти кроках опрацювання значення W_t дорівнює відповідному слову в блоці повідомлення. Для решти 64-х кроків значення W_t є наслідком циклічного зсуву ліворуч на один біт результату зв'язування операцією XOR чотирьох попередніх значень W_t. У SHA-1 16 слів блока розширюються до 80-ти слів для використання з функцією стиснення. Це породжує надто велику надлишковість і взаємозв'язок блоків стисненого повідомлення, однак ускладнює завдання знаходження блоків повідомлення, які породжують однаковий висновок функції стиснення.

Алгоритм SHA нагадує алгоритм MD4, але відрізняється від останнього наявністю розширеного перетворення, додатковим раундом опрацювання, оптимізованим лавинним ефектом та обчисленням 160-розрядного геш-значення. Сьогодні існують версії алгоритму SHA-256 та SHA-512.

3.3 Функція гешування за ГОСТом Р 34.11–94

Ідея використовувати алгоритм блочного шифрування для побудови надійних схем гешування виглядає природною. Однак при такому підході виникають дві проблеми. По-перше, розмір блока більшості блокових шифрів є

недостатній для того, щоб геш-функція була стійка проти методу на базі парадокса „дня народження” (додаток Г). По-друге, запропонований метод потребує задавання певного ключа, на якому відбувається шифрування. Надалі цей ключ слід тримати в секреті, бо зловмисник, знаючи його і геш-значення, може виконати процедуру у зворотному напрямку. Наступним кроком у розвитку ідеї використовувати блочний шифр для гешування є підхід, при якому черговий блок тексту подається як ключ, а геш-значення попереднього кроку – в якості вхідного блока. Вихід блочного алгоритму шифрування є поточним геш-значенням. Існує безліч модифікацій цього методу, в тому числі геш-функції, вихід яких є удвічі довший за блок. В ряду модифікацій проміжне геш-значення підсумовується покоординатно за модулем 2 з блоком тексту. В даному випадку мається на увазі, що розмір ключа і блока у шифрі збігаються. У літературі зустрічаються 12 різних схем гешування для випадку, коли розмір ключа і блока у шифрі збігаються [8]:

$$
\begin{aligned}
H_i &= E_{H_{i-1}}(M_i) \oplus M_i;\\
H_i &= E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1};\\
H_i &= E_{H_{i-1}}(M_i) \oplus M_i \oplus H_{i-1};\\
H_i &= E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i;\\
H_i &= E_{M_i}(H_{i-1}) \oplus H_{i-1};\\
H_i &= E_{M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1};\\
H_i &= E_{M_i}(H_{i-1}) \oplus M_i \oplus H_{i-1};\\
H_i &= E_{M_i}(H_{i-1} \oplus M_i) \oplus H_{i-1};\\
H_i &= E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i;\\
H_i &= E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1};\\
H_i &= E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1};\\
H_i &= E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i,
\end{aligned}
$$

де $E_k(M)$ означає результат застосування алгоритму блокового шифрування з ключем k до блока. В усіх подібних схемах вважають $H_0 = I_{\text{п}}$, де $I_{\text{п}}$ – початкове значення.

Стійкість подібних схем залежить від криптографічних та інших властивостей алгоритмів блокового шифрування, що лежать у їхньому підґрунті. Зокрема, навіть якщо алгоритм шифрування є стійким, деякі із запропонованих схем можуть бути схильні до колізій. До подібних ефектів можуть призводити такі властивості алгоритму шифрування, як комплементарність (шифрування інвертованого відкритого тексту на інвертованому ключі призводить до інвертованого шифртексту), наявність слабких і напівслабких ключів і т. п.

У підґрунті функції гешування за ГОСТом Р 34.11–94 [32, 44] лежить алгоритм блокового шифрування ГОСТу 28147–89 [33]. Функція перетворює інформаційну послідовність довільної довжини на геш-образ розрядністю 256 біт.

Нехай M – вхідна інформаційна послідовність довжиною $|M|$. Послідовність розбивається на 256-розрядні блоки. Останній неповний блок доповнюється до потрібного розміру. Додаються два 256-розрядних блоки, що містять код довжини послідовності (L) і контрольну суму (I). Кожен блок m_i утвореної розширеної послідовності Ext(m) розглядається як результат конкатенації чотирьох 64-розрядних двійкових наборів:

$$m_i = (A_i, B_i, C_i, D_i).$$

Тоді процес обчислення геш-образу $h(M)$ може бути описано як

$$\text{GOST}_i = h(m_i, \text{GOST}_{i-1}),$$

де GOST_i – результат i-го циклу перетворення; GOST_0 – 256-розрядний стартовий вектор гешування, на вибір якого обмежень не накладається.

Процедура обчислення функції h складається з послідовності кроків:

крок 1:

– зініціалізовуються змінні L (поточне значення довжини опрацьованої частини вхідної послідовності) та I (значення контрольної суми);

– якщо довжина вхідної неопрацьованої послідовності є більшою за 256 ($|M| > 256$), алгоритм переходить до кроку 3, в іншому разі проводяться наступні дії.

крок 2:

– $L \equiv (L + |M|) \bmod 2^{256}$;

– $I = I \oplus \text{Ext}(M)$; $\text{Ext}(M) = (0^{256-|M|}, M)$, де $0^{256-|M|}$ – послідовність бітових нулів довжиною $256 - |M|$, тобто на цьому етапі обчислюється поточне значення контрольної суми;

– GOST = h(Ext(M), GOST) – обчислюється значення функції гешування h від аргументів, що становлять собою гешований блок і початковий вектор гешування GOST;

– GOST = $h(L, \text{GOST})$;

– GOST = $h(I, \text{GOST})$ – кінцевий результат геш-функції $h(M) = \text{GOST}$;

– кінець роботи алгоритму.

крок 3:

– подається інформаційна послідовність M у формі $M = (M_L, M_R)$, $|M_R| = 256$;

– GOST = $h(M_R, \text{GOST})$;

– $L \equiv (L + 256) \bmod 2^{256}$;

– $I = I \oplus M_R$;

– $M = M_L$;

– перехід до кроку 2.

Алгоритм обчислення GOST_i містить три кроки:

– генерування чотирьох 256-розрядних ключів k_{A_i}, k_{B_i}, k_{C_i}, k_{D_i} для зашифровування в режимі простої заміни 64-розрядних частин *i*-го блока;

– зашифровування частин блока m_i з використовуванням алгоритму за ГОСТом 28147–89;

– змішування результатів зашифровування.

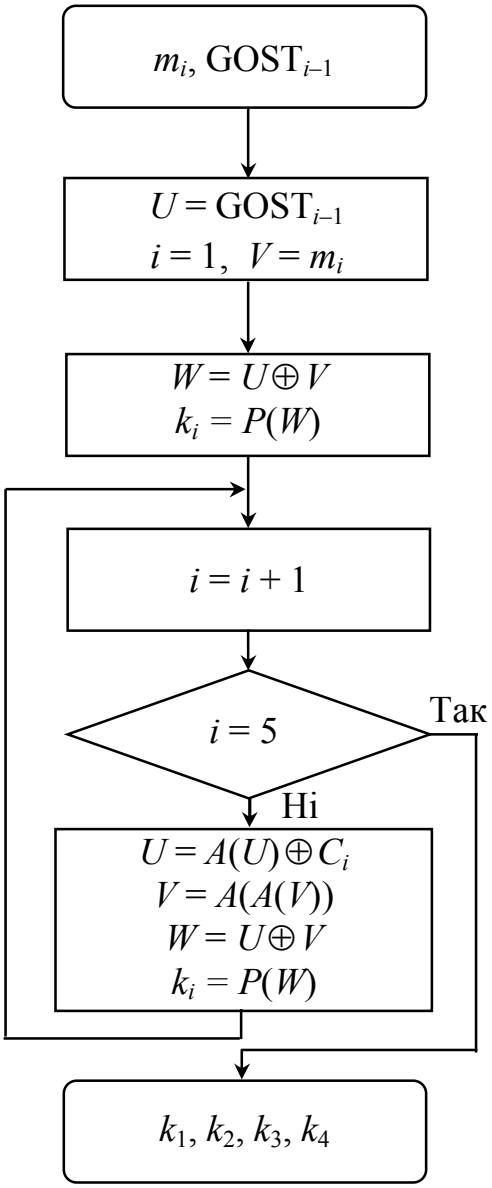


Рисунок 3.4 – Алгоритм генерування секретних ключів за ГОСТом Р 34.11–94

Генерування секретних ключів. При генеруванні секретних ключів використовуються певні *h* та *M* (вхідна послідовність, подана у двійковому вигляді) і започатковано константи

$$C_2 = C_4 = 0^{256}$$

та

$$C_3 = 1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 (0^8 1^8)^4 (1^8 0^8)^4,$$

де a^k – двійкова послідовність з *k* бінарних знаків a – {0, 1}. Задано два перетворення для 256-розрядних блоків:

$$X = (A, B, C, D) = (b_{32}, \ldots, b_2, b_1),$$

де b_i – байти блока;

$$A(X) = (C \oplus D, A, B, C);$$
$$P(X) = (b_{\varphi(32)}, \ldots, b_{\varphi(2)}, b_{\varphi(1)}),$$

де $\varphi(i + 1 + 4(k - 1)) = 8i + k$, $0 \le i \le 3$, $1 \le k \le 8$.

Функція *P* становить собою перестановлення над підблоками довжиною 8 біт вихідної двійкової послідовності довжиною 256 біт. Функція $A(X)$ перетворює цю послідовність шляхом поділу на чотири підблоки по 64 біти кожен. Алгоритм генерування секретних ключів подано на рис. 3.4.

Шифрувальне перетворення. На наступному кроці частини блока піддаються зашифровуванню на здобутих ключах:

$$E(m_i) = E(E_{k_{A_i}}(A_i), E_{k_{B_i}}(B_i), E_{k_{C_i}}(C_i), E_{k_{D_i}}(D_i)).$$

Перемішуюча функція. Результат циклу перетворення формується після кроку перемішування. Задано перетворення 256-розрядного блока

$$X = (w_{16}, \ldots, w_2, w_1),$$

де w_i – 16-розрядні слова блока:

$$T(X) = (w_1 \oplus w_2 \oplus w_3 \oplus w_4 \oplus w_{13} \oplus w_{16}, w_{16}, \ldots, w_3, w_2).$$

Результат циклу перетворення є

$$\text{GOST}_i = h(m_i, \text{GOST}_{i-1}) = T^{61}(\text{GOST}_{i-1} \oplus T(m_i \oplus T^{12}(E(m_i)))),$$

де степінь функції T позначає, скільки разів вона застосовується до бітової послідовності.

Криптографічна стійкість певної геш-функції базується на стійкості застосованого в ній блочного алгоритму шифрування, використаного в режимі простої заміни.

При практичному використанні геш-функції мають виконуватися такі вимоги:

– алгоритм повинен мати високу швидкість опрацювання інформації (це є надто актуально для банківських програм, де потрібна надзвичайна оперативність опрацювання інформації);

– геш-функція повинна бути стійкою проти атаки методом „грубої сили”;

– програмна реалізація геш-функції повинна бути оптимізована для використовування на сучасній апаратно-програмній базі.

Цим вимогам має задовольняти як сам алгоритм продукування геш-значення, так і гешувальна функція. Приклад обчислення геш-функції за ГОСТом 34.11–94 подано у додатку Ж.

У сучасних умовах алгоритмічного підвищення швидкості продукування геш-значення може бути досягнуто за рахунок застосування простого перетворення, яке переводить одне повідомлення на інше за допомогою елементарної операції, наприклад, вилучення довільного блока повідомлення. Подібними перетвореннями можна також описати залежність між двома практично не відмінними один від одного повідомленнями. Такий тип повідомлень дуже часто зустрічається у банківській справі, приміром, з метою заповнення бланків платіжних доручень. Звідси випливає, що для збільшення швидкості опрацювання необхідно, щоб алгоритм продукування геш-значення містив також алгоритм обчислення геш-значення одного повідомлення з геш-значення іншого повідомлення, яке виходить з початкового за допомогою елементарного перетворення.

3.4 Стійкість геш-функцій

З точки зору криптографічної стійкості, важливою властивістю геш-функцій є відсутність колізій, тобто неможливість знайти такі значення $x \neq y$, щоб $h(x) = h(y)$. У криптографічних додатках важливим поняттям є *криптографічно стійка геш-функція*, для якої не існує ефективного алгоритму знаходження значень $x \neq y$, де б виконувалася умова $h(x) = h(y)$ (функція, стійка

в сильному розумінні), або не існує ефективного алгоритму знаходження колізії за заданого x такого $y \neq x$, що $h(x) = h(y)$. Р. Андерсон засвідчив [34], що відсутність колізій не дозволяє робити висновки щодо практичної стійкості геш-функцій. Інакше кажучи, дана вимога носить формальний характер. Практично значущою є відсутність у геш-функцій кореляції. Вільною від кореляції називається геш-функція, у якої неможливо знайти пари таких значень $x \neq y$, що вага Хеммінга двійкового вектора $h(x)$ xor $h(y)$ буде меншою за вагу Хеммінга стосовно бінарного вектора $h(M)$ для певного малого M. Свобода від кореляції, з точки зору криптографічної стійкості, є значно потужнішою властивістю геш-функцій, аніж свобода від колізій.

4 ЕЛЕКТРОННО-ЦИФРОВИЙ ПІДПИС

4.1 Загальні положення

Найбільш важливою сферою застосовування криптографії з відкритим ключем є цифрові підписи. Протягом багатьох століть при діловому листуванні, укладання контрактів та оформленні будь-яких інших важливих паперів, підпис відповідальної особи чи виконавця був неодмінною умовою визнання його статусу або беззаперечним свідченням його важливості. Подібний акт переслідував дві цілі:

– гарантування автентичності листа шляхом звірення підпису з наявним зразком;

– гарантування авторства документа (з юридичної точки зору).

Виконання цих вимог ґрунтується на таких властивостях підпису:

– підпис є автентичний, тобто за його допомогою одержувачеві документа можна довести, що він належить власникові (на практиці це визначається графологічною експертизою);

– підпис є непідробний, тобто є доказом, що лише та людина, чий автограф стоїть на документі, могла підписати цей документ і ніхто інший не зміг би цього зробити;

– підпис неможливо перенести, тобто він є частиною документа і тому перенести його на інший документ неможливо;

– документ з підписом є незмінним, тобто після підписання його неможливо змінити, залишивши даний факт непоміченим;

– підпис є беззаперечний, тобто особа, яка підписала документ, у разі визнання експертизою, що саме вона засвідчила цей документ, не може заперечити факт підписання;

– кожна особа, котра має зразок підпису, може переконатися в тому, що даний документ підписано власником підпису.

З переходом до безпаперових способів передавання та зберігання даних, а також з розвитком систем електронного переказування грошових коштів, у основі яких – електронний аналог паперового платіжного доручення, проблема віртуального підтвердження автентичності документа набула особливої гостроти. Розвиток будь-яких подібних систем тепер неможливо уявити без існування електронних підписів під електронними документами. Проте застосовування і широке розповсюдження *електронно-цифрових підписів* (ЕЦП) спричинило цілий ряд правових проблем. Приміром, ЕЦП може застосовуватися на основі домовленостей всередині певної групи користувачів системи передавання даних і, відповідно до домовленості всередині цієї групи, повинен мати юридичну силу. Але чи буде електронний підпис мати доказову силу в суді, наприклад, за оскарження факту передавання платіжного доручення? Так, тому що 2003 року в Україні прийнято закони: „Про

електронні документи та електронний документообіг", „Про електронний цифровий підпис" (додатки Б, В).

Хоча ЕЦП зберіг практично всі головні властивості звичайного підпису, певні особливості реалізації електронного автографа роблять його окремим класом підписів. Тому юридичні, правові та методологічні аспекти застосовування ЕЦП мають враховувати його специфіку.

Існує кілька методів побудови схем ЕЦП, а саме:

1) Шифрування *електронного документа* (ЕД) на основі симетричних алгоритмів. Така схема передбачає наявність у системі третьої особи (арбітра), що користується довірою учасників обміну підписаними у подібний спосіб електронними документами. Взаємодія користувачів даної системи здійснюється за таким алгоритмом:

– учасник A зашифровує повідомлення на власному секретному ключі k_A, зміст якого узгоджено з арбітром, потім зашифроване повідомлення передається арбітрові із зазначенням адресата даного повідомлення (інформація, що ідентифікує адресата, передається також у зашифрованому вигляді);

– арбітр розшифровує отримане повідомлення на ключі k_A, проводить необхідні перевірки і потім зашифровує на секретному ключі учасника B (k_B). Далі зашифроване повідомлення надсилається учасникові B разом з інформацією про те, що воно надійшло від учасника A;

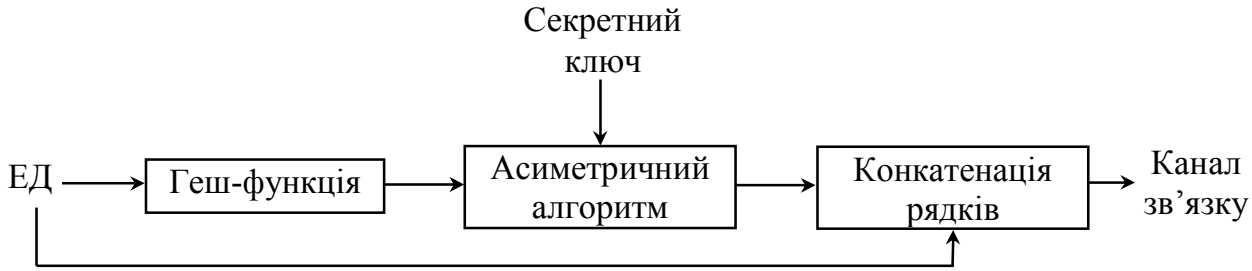
– учасник B розшифровує це повідомлення й переконується в тому, що відправником є учасник A.

За авторизацію документа у наведеній схемі вважатиметься сам факт зашифровування ЕД секретним ключем і передавання зашифрованого ЕД арбітра. Основною перевагою цієї схеми є наявність третьої сторони, що виключає будь-які суперечні питання між учасниками інформаційного обміну, тобто в даному разі не потрібно додаткової системи арбітражу ЕЦП. Недоліком схеми є наявність третьої сторони і використовування симетричних алгоритмів шифрування.

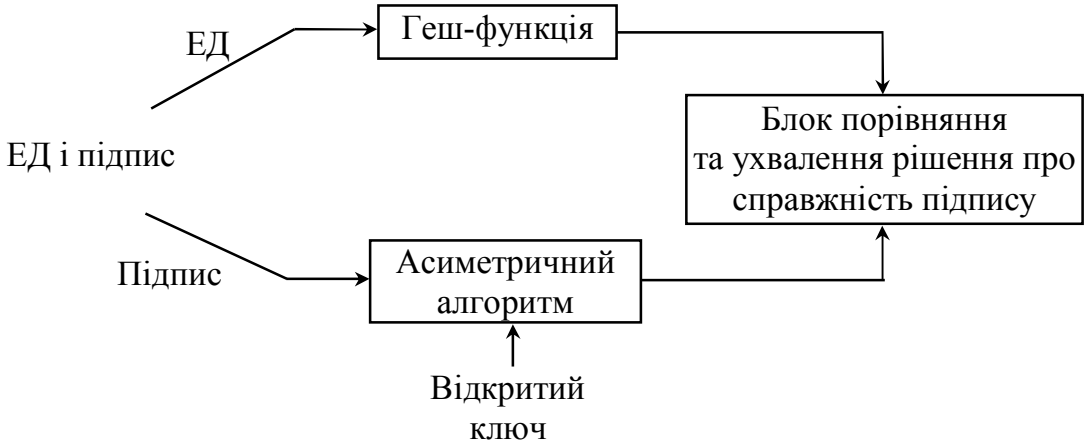
2) Шифрування ЕД з використовуванням асиметричних алгоритмів шифрування. Фактом підписання документа в такій схемі є зашифровування документа на секретному ключі його відправника. Ця схема теж використовується дуже рідко внаслідок того, що довжина ЕД може виявитися критичною. Застосовування асиметричних алгоритмів для зашифровування повідомлень великої довжини є неефективне з точки зору швидкісних характеристик. У цьому разі не потрібно наявності третьої сторони, хоча вона може виступати в ролі сертифікаційного органу відкритих ключів користувачів.

3) Розвитком попередньої ідеї стала найбільш розповсюджена схема ЕЦП, а саме: зашифровування остаточного результату опрацювання ЕД геш-

функцією за допомогою асиметричного алгоритму. Структурну схему такого варіанта побудови ЕЦП подано на рис. 4.1.



а) передавальна сторона



б) приймальна сторона

Рисунок 4.1 – Структурна схема побудови ЕЦП

Генерування підпису відбувається в такий спосіб:

1) Учасник *A* обчислює геш-код від ЕД. Отриманий геш-код проходить процедуру перетворення з використанням власного секретного ключа, після чого отримане значення (що і є ЕЦП) разом з ЕД надсилається учасникові *B*.

2) Учасник *B* повинен отримати ЕД з ЕЦП та сертифікований відкритий ключ користувача *A*, а потім провести розшифровування на ньому ЕЦП, сам ЕД підлягає операції гешування, після чого результати порівнюються і, якщо вони співпадають, ЕЦП визнається за справжній, в іншому разі – за помилковий.

Стійкість даного типу ЕЦП базується на стійкості асиметричних алгоритмів шифрування і застосовуваних геш-функцій.

Крім розглянутих, існують „екзотичні" варіанти побудови схем ЕЦП (груповий підпис, беззаперечний підпис, довірений підпис і т. п.). Появу цих різновидів зумовлено різноманіттям завдань, розв'язуваних за допомогою електронних технологій передавання та опрацювання ЕД.

Взагалі підписаний ЕД виглядає як пара, що складається з бінарних рядків (M, S), де M становить собою ЕД, а S – підпис, розв'язок рівняння $E_k(S) = M$, де E_k є функцією з секретом.

У зв’язку з вищевикладеним означенням ЕЦП, можна відокремити такі його властивості:

– є непідробним, оскільки розв’язати рівняння $E_k(S) = M$ може лише власник секрету k;

– однозначно ідентифікує автора, тобто людину, котра підписала цей документ;

– верифікація підпису здійснюється на основі знання функції E_k;

– є непереносним на інший ЕД; виняток становить випадок, коли для використаної геш-функції виявлено колізії;

– ЕД з ЕЦП може передаватися відкритими каналами, оскільки будь-яка зміна ЕД призведе до того, що процедура перевірки ЕЦП виявить цей факт.

4.2 Алгоритм цифрового підпису RSA

Технологія застосовування електронного цифрового підпису припускає наявність мережі абонентів, які надсилають один одному електронні документи. У цій ситуації для формування електронного підпису кожного абонента використовують окрему пару ключів – K_1 і K_2. Секретний ключ K_2 відомий лише користувачеві, а його ідентифікаційний номер ID і ключ K_1 розміщують у загальнодоступному для інших абонентів мережі каталозі. Це дозволяє будь-якому абонентові мережі перевіряти істинність цифрового підпису документів, отримуваних від її власника. Значення ідентифікаційного номера використовується в певних алгоритмах формування сигнатури.

Найбільш поширеною системою формування електронного підпису є система, в основі якої лежить алгоритм RSA. Узагальнену схему формування й перевірки цифрового підпису RSA [35] подано на рис. 4.2.

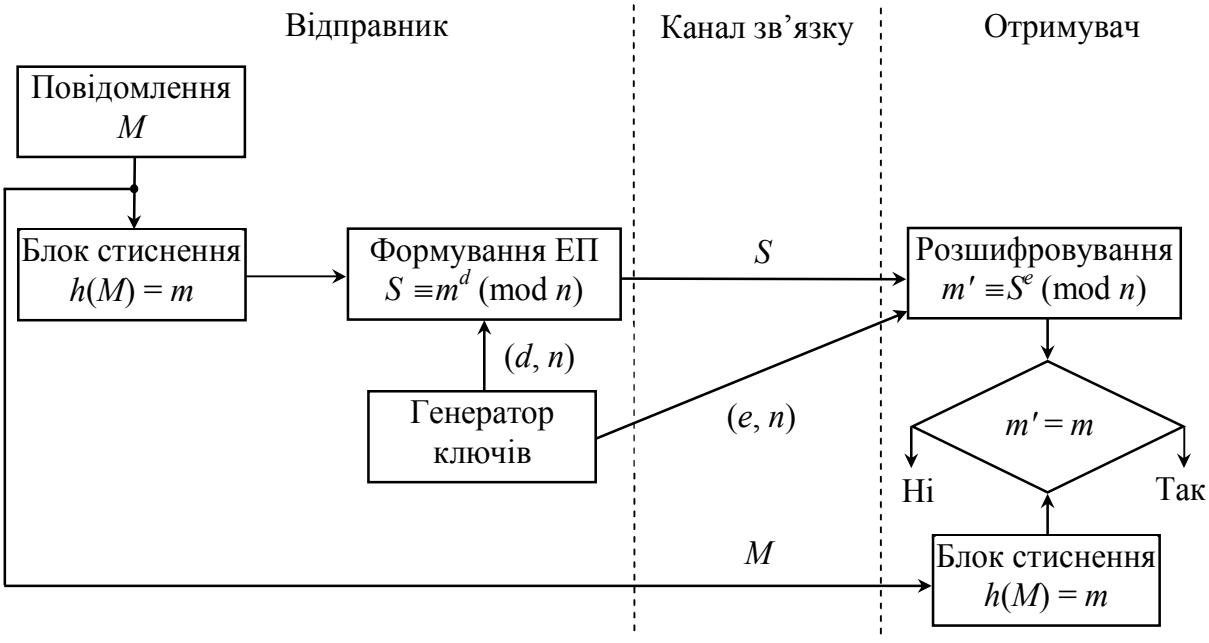


Рисунок 4.2 – Узагальнена схема цифрового підпису RSA

Обираються великі прості числа p і q, обчислюється число $n = p\,q$, функція Ейлера $\varphi(n) = (p-1)(q-1)$ і число $e < \varphi(n)$, взаємно просте з $\varphi(n)$ (відкритий ключ $K_1 = e$). Врешті, обчислюється число d (секретний ключ $K_2 = d$), взаємно обернене з e. У відкритому каталозі розміщують ключ (e, n), а ключ d зберігається у автора документа.

Припустімо, що відправник хоче підписати повідомлення M перед його надсиланням. При цьому передбачається, що сам текст документа шифрувати не потрібно. Спочатку повідомлення M стискають за допомогою геш-функції h в ціле число m:

$$h(M) = m.$$

Потім відправник зашифровує m відомим лише йому значенням d:

$$S \equiv m^d \pmod{n}.$$

Пара чисел (M, S) передається адресатові як електронний документ M, підписаний електронним підписом S.

Адресат, отримавши підписаний документ (M, S), обчислює значення m у два різних способи. По-перше, він відновлює геш-значення m', застосовуючи криптографічне перетворення підпису S з використанням відкритого ключа e:

$$m' \equiv S^e \pmod{n}.$$

По-друге, одержувач відшукує результат гешування отриманого повідомлення M за допомогою такої самої геш-функції h:

$$h(M) = m.$$

Якщо обидва значення співпадають $m' = m$, тобто дотримується рівність

$$S^e \pmod{n} = h(M),$$

то одержувач визнає пару (M, S) за справжнє значення документа.

В якості геш-функції у схемі підпису RSA використовують функції сімейства MD. Відкритий ключ e називають *ідентифікатором*.

Приклад 4.1 Дано $p = 5$ і $q = 11$, тоді

$$n = 11 \cdot 5 = 55;$$
$$\varphi(55) = (11-1)(5-1) = 40.$$

Нехай задано ключі $e = 3$ і $d = 27$. У відкритому каталозі розміщують значення (3, 55), а ключ d зберігається у автора документа.

Відправник хоче підписати повідомлення M, для якого значення геш-функції дорівнює 13:

$$h(M) = 13.$$

У цьому разі відправник обчислює

$$S \equiv 13^{27} \pmod{55} \equiv 7$$

і формує підписане повідомлення

$$(M, 7).$$

Одержувач, отримавши підписане повідомлення, обчислює значення геш-функції

$$h(M) = 13$$

та

$$m' \equiv 7^3 \pmod{55} \equiv 13.$$

Значення m' та геш-функції $h(M)$ співпали, тобто підпис є справжній.

Недоліки алгоритму цифрового підпису RSA. При обчисленні модуля n, ключів e і d для системи цифрового підпису RSA доводиться перевіряти велику кількість додаткових умов. Недотримання якоїсь з цих умов робить можливою фальсифікацію цифрового підпису з боку того, хто виявить таке недотримання. При підписанні важливих документів у жодному разі не можна допускати такої можливості навіть теоретично.

Цифровий підпис RSA є вразливий щодо так званої мультиплікативної атаки. Інакше кажучи, алгоритм цифрового підпису RSA дозволяє криптоаналітикові без знання секретного ключа d сформувати підписи під тими документами, в яких результат гешування можна обчислити як добуток результатів гешування вже підписаних документів.

Приклад 4.2 Припустімо, що криптоаналітик може сконструювати три повідомлення – M_1, M_2 і M_3, які мають геш-значення

$$m_1 = h(M_1); \;\; m_2 = h(M_2); \;\; m_3 = h(M_3),$$

причому

$$m_3 \equiv (m_1 \, m_2) \bmod n.$$

Припустімо також, що для двох повідомлень M_1 і M_2 отримано справжні підписи

$$S_1 \equiv m_1^{\,d} \pmod{n}$$

та

$$S_2 \equiv m_2^{\,d} \pmod{n}.$$

Тоді криптоаналітик може легко обчислити підпис S_3 для документа M_3, навіть не знаючи секретного ключа d:

$$S_3 \equiv (S_1 \, S_2) \bmod n.$$

Дійсно,

$$(S_1\, S_2) \bmod n \equiv (m_1{}^d\, m_2{}^d) \bmod n \equiv (m_1 m_2)^d \bmod n \equiv m_3{}^d \,(\bmod n) \equiv S_3.$$

Атака на підпис RSA в схемі з нотаріусом. Є електронний нотаріус, який підписує документи, що проходять через нього (односпрямовані геш-функції не використовуються, нотаріус шифрує власним закритим ключем всі повідомлення). M – певний відкритий текст, який нотаріус не бажає підписувати. Зловмисник знає відкритий ключ (e, n) нотаріуса і хоче підписати документ M.

Зловмисник обирає певне випадкове число x, взаємно просте з n, і обчислює

$$y \equiv x^e \,(\bmod n),$$

потім здобуває значення $M' \equiv (y\, M) \bmod n$ і передає його на підпис нотаріусові. Нотаріус підписує (адже це вже не текст M)

$$M'{}^d \,(\bmod n) \equiv S.$$

Зловмисник одержує

$$S \equiv M'{}^d \,(\bmod n) \equiv (y\, M)^d \bmod n = ((x^e)^d\, M^d) \bmod n = (x\, M^d) \bmod n,$$

й, отже, розв'язує рівність

$$M^d \equiv (S\, x^{-1}) \bmod n,$$

значення якої і є підписом тексту M.

Приклад 4.3 Нехай $n = 2993$; $e = 217$; $M = 77$ – певний відкритий текст, який нотаріус не бажає підписувати. Зловмисник обирає випадкове число $x = 23$ і обчислює

$$y \equiv 23^{217} \,(\bmod 2993) \equiv 2505;$$
$$M' \equiv (2505 \cdot 77) \bmod 2993 \equiv 1333.$$

Значення $M' = 1333$ він передає на підпис нотаріусові і отримує $S = 2793$. Потім зловмисник розв'язує рівність

$$M^d = S \equiv (2793 \cdot 23^{-1}) \bmod 2993$$

і знаходить $S = 1683$, що і є підписом $M = 77$.

Вправи

У всіх завданнях припускатимемо, що $h(M) = m$ для всіх значень m.

1 Побудувати підпис RSA для повідомлення m при поданих нижче параметрах користувача.

№ варіанта	p	q	d	m	№ варіанта	p	q	d	m
1	5	11	27	7	11	19	7	91	17
2	5	13	29	10	12	17	7	77	15
3	7	11	43	5	13	29	5	69	14
4	7	13	29	15	14	23	7	41	16
5	11	23	67	8	15	11	19	47	18
6	13	29	197	3	16	17	11	59	19
7	17	23	145	9	17	23	17	145	11
8	19	29	89	7	18	17	29	39	10
9	23	5	51	13	19	13	23	97	9
10	29	7	95	12	20	29	13	263	8

2 Для зазначених відкритих ключів користувача RSA перевірити справжність підписаних повідомлень.

№ варіанта	n	e	Повідомлення	№ варіанта	n	e	Повідомлення
1	55	3	(22, 15), (7, 28), (6, 36)	11	87	11	(9, 29), (13, 34), (6, 51)
2	65	5	(6, 42), (13, 41), (10, 30)	12	86	7	(79, 17), (7, 38), (85, 45)
3	77	7	(13, 41), (11, 28), (15, 26)	13	94	5	(69, 23), (7, 47), (39, 91)
4	91	5	(15, 71), (11, 46), (16, 74)	14	82	3	(53, 19), (2, 30), (34, 58)
5	95	11	(82, 43), (93, 82), (11, 33)	15	74	19	(57, 17), (17, 7), (51, 23)
6	85	13	(5, 51), (1, 81), (62, 27)	16	62	17	(29, 23), (19, 9), (57, 37)
7	35	17	(26, 31), (2, 29), (8, 43)	17	58	13	(52, 23), (37, 47), (20, 74)
8	57	19	(3, 13), (15, 55), (22, 79)	18	93	11	(12, 21), (9, 29), (25, 67)
9	69	17	(14, 11), (62, 26), (64, 94)	19	54	7	(15, 17), (11, 47), (29, 33)
10	81	13	(32, 22), (64, 37), (51, 59)	20	46	19	(25, 13), (31, 29), (3, 37)

4.3 Електронний підпис на базі шифру Ель–Гамаля

Нехай відправник збирається підписати документ M. Він обирає велике просте число p і число g. Ці числа передаються або зберігаються у відкритому вигляді і можуть бути спільними для цілої групи користувачів. Відправник обирає випадкове число k – секретний ключ, $1 < k < p - 1$, і обчислює

$$Y \equiv g^k \pmod{p}.$$

Число Y подає в якості відкритого ключа.

Опишемо послідовність дій для побудови підпису. Спочатку обчислюється значення геш-функції $h(M) = m$ і обирається випадкове число x таке, що $x < p - 1$, та взаємно просте з $p - 1$, і обчислюються числа

$$r \equiv g^x \pmod{p};$$
$$u \equiv (m - k\,r) \bmod (p - 1);$$
$$s \equiv (x^{-1}\,u) \bmod (p - 1).$$

Формується підписане повідомлення (M, r, s).

Одержувач насамперед обчислює значення геш-функції $h(M) = m$ і потім перевіряє підпис, використовуючи рівність

$$(Y^r r^s) \bmod p = g^m \pmod p.$$

Якщо рівність виконується, то підпис є справжній.

Приклад 4.4 Нехай $p = 23$; $g = 5$; $k = 7$; $h(M) = 3$; $x = 5$.

Відправник обчислює відкритий ключ

$$Y \equiv 5^7 \pmod{23} \equiv 17.$$

Переходить до обчислення підпису:

$$r \equiv 5^5 \pmod{23} \equiv 20;$$
$$u \equiv (3 - 7 \cdot 20) \bmod (23 - 1) \equiv 17;$$
$$s \equiv (5^{-1} \cdot 17) \bmod (23 - 1) \equiv 21.$$

Формується підписане повідомлення у вигляді (M, 20, 21), яке передається одержувачеві.

Одержувач перевіряє справжність підпису. Спочатку він обчислює значення геш-функції $h(M) = 3$, а потім

$$(17^{20} \cdot 20^{21}) \bmod 23 \equiv 10;$$
$$5^3 \pmod{23} \equiv 10.$$

Одержувач робить висновок, що підпис є справжній.

Вправи

У всіх завданнях припускатимемо, що $h(M) = m$ для всіх значень m.

1 Абоненти певної мережі застосовують підпис Ель–Гамаля зі спільними параметрами $p = 23$, $g = 5$. Для зазначених секретних параметрів абонентів знайти відкритий ключ Y і побудувати підпис для повідомлення m.

№ варіанта	k	x	m	№ варіанта	k	x	m
1	11	3	15	11	12	5	7
2	10	15	5	12	13	17	13
3	17	13	8	13	10	5	8
4	18	7	5	14	17	13	15
5	13	19	15	15	19	9	15
6	11	17	10	16	11	9	5
7	15	17	11	17	21	3	20
8	17	19	12	18	20	13	17
9	14	9	25	19	18	17	10
10	16	13	5	20	17	19	13

2 Для зазначених відкритих ключів Y користувачів системи Ель–Гамаля зі спільними параметрами $p = 23$, $g = 5$ перевірити справжність підписаних повідомлень.

№ варіанта	Y	Повідомлення
1	1	(15; 20, 3), (15; 10, 5), (15; 19, 3)
2	10	(5; 19, 17), (7; 17, 8), (6; 17, 8);
3	3	(3; 17, 12), (2; 17, 12), (8; 21, 11)
4	18	(5; 17, 1), (5; 11, 3), (5; 17, 10)
5	11	(15; 7, 1), (10; 15, 3), (15; 7, 16)
6	21	(11; 8, 19), (13; 15, 13), (13; 15, 10)
7	17	(3; 21, 20), (3; 20, 21), (3; 17, 21)
8	15	(15; 7, 11), (13; 7, 6), (17; 7, 19)
9	13	(22; 11, 17), (25; 12, 15), (25; 11, 25)
10	7	(15; 21, 20), (15; 11, 20), (17; 13, 21)

4.4 Стандарт цифрового підпису DSS

Алгоритм цифрового підпису DSA

Національний інститут стандартів і технологій США опублікував федеральний стандарт опрацювання інформації FIPS PUB 186, відомий також як DSS (Digital Signature Standard – стандарт цифрового підпису) [37]. Стандарт DSS базується на алгоритмі гешування SHA і становить нову технологію використання цифрового підпису – алгоритм DSA (Digital Signature Algorithm – алгоритм цифрового підпису). Алгоритм DSA є „класичним” прикладом схеми ЕЦП на базі використовування геш-функції та асиметричного алгоритму шифрування. Стандарт DSS було запропоновано 1991 року, а його відкоригована версія – 1993 року, у відповідь на що виникли сумніви щодо безпеки відповідної схеми. 1996 року до нього було внесено незначні корективи.

У даному стандарті підпис становить собою два надвеликих цілих числа, отримані у відповідності до процедур і параметрів, окреслених в DSS. Стійкість системи в цілому ґрунтується на складності знаходження дискретних логарифмів у скінченних полях.

У стандарті DSS використовується алгоритм, призначений забезпечити лише функцію цифрового підпису. На відміну від RSA, даний алгоритм не може бути використано для шифрування чи обміну ключами.

Алгоритм цифрового підпису DSA створено з урахуванням складнощів обчислення дискретних логарифмів і спирається на схеми, запропоновані Ель–Гамалем та Шнорром.

В алгоритмі використовуються три параметри, які є відкритими і передбачаються відомими групі користувачів. Обирається просте число p довжиною між бітами 512...1024 і q – 160-бітове простий дільник числа $(p-1)$. Нарешті обирається число g форми $h^{(p-1)/q} \pmod p$, де h є цілим числом між 1 та $(p-1)$, з тим обмеженням, що g має бути більше за 1. Алгоритм схематично подано в табл. 4.1.

Таблиця 4.1 – Алгоритм цифрового підпису DSA

Глобальні компоненти відкритого ключа: p – просте число, $2^{L-1} < p < 2^L$, де $512 \leqq L \leqq 1024$ і L кратне до 64, тобто має довжину між бітами 512...1024 з кроком 64 біти; q – простий дільник $(p-1)$, де $2^{159} < q < 2^{160}$, тобто довжиною 160 бітів; $g \equiv h^{(p-1)/q} \pmod p$, де h є яким завгодно цілим числом таким, що $1 < h < (p-1)$ і $h^{(p-1)/q} \pmod p > 1$
Особистий ключ користувача k – випадкове чи псевдовипадкове число, $0 < k < q$
Відкритий ключ користувача $Y \equiv g^k \pmod p$
Секретний номер повідомлення користувача x – випадкове чи псевдовипадкове число, $0 < x < q$
Створення підпису: $r \equiv (g^x \bmod p) \bmod q$; $s \equiv [x^{-1}(h(M) + k\,r)] \bmod q$; якщо $r = 0$ або $s = 0$, обираємо інше x. *Підпис*: (r, s)
Перевірка підпису (верифікація): $w \equiv (s'^{-1}) \bmod q$; $u_1 \equiv [h(M')w] \bmod q$; $u_2 \equiv (r'\,w) \bmod q$; $v \equiv [(g^{u_1} Y^{u_2}) \bmod p] \bmod q$. *Перевірка*: $v = r'$

Параметри p, q і g публікуються для всіх користувачів системи ЕД з ЕЦП. Знаючи ці числа, кожен користувач обирає особистий ключ та генерує відкритий ключ. Особистий ключ k повинен бути числом від 1 до $(q-1)$ і обиратися у випадковий чи псевдовипадковий спосіб (тримається в секреті). Відкритий ключ обчислюється на базі особистого ключа за формулою $Y \equiv g^k \pmod p$. Обчислити Y за наявним значенням k відносно просто. Однак при тому, що у значенні відкритого ключа завдання визначання значення k за значенням Y є важким, оскільки для цього треба обчислити дискретний логарифм Y за основою g та за модулем p.

Створення підпису (рис. 4.3). Для генерації ЕЦП користувач обчислює дві величини – r і s, які є функціями компонентів відкритого ключа (p, q, g), особистого ключа користувача k, геш-коду повідомлення $h(M)$ (геш-код M обчислюється з використанням алгоритму SHA-1) і певного цілого числа x, яке має обиратися у випадковий чи псевдовипадковий спосіб і бути єдиним для кожного виконання підпису.

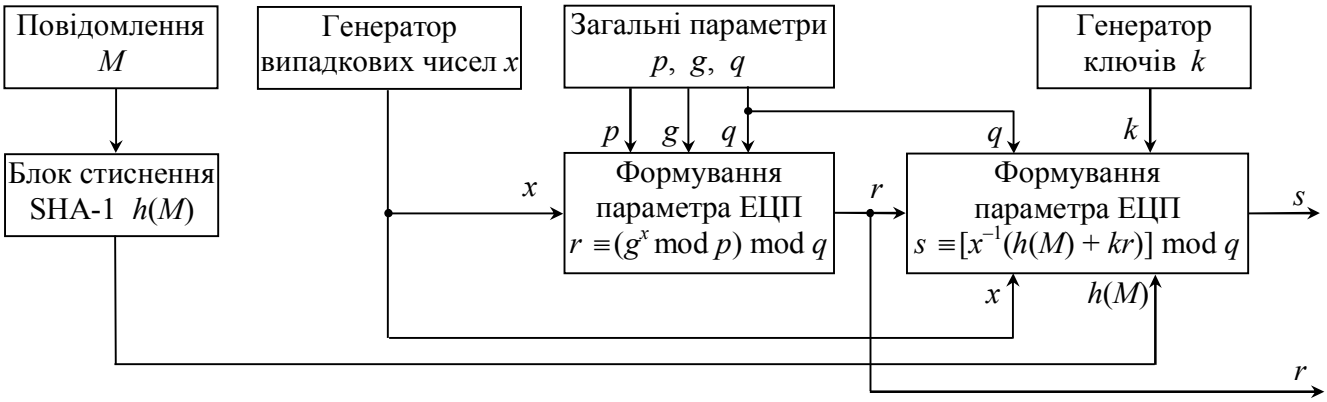


Рисунок 4.3 – Створення підпису DSA

Значення r та s є ЕЦП повідомлення M і передаються разом з ним відкритими каналами зв'язку.

Перевірка підпису (рис. 4.4). Нехай прийнято повідомлення M' і його значення r' та s'.

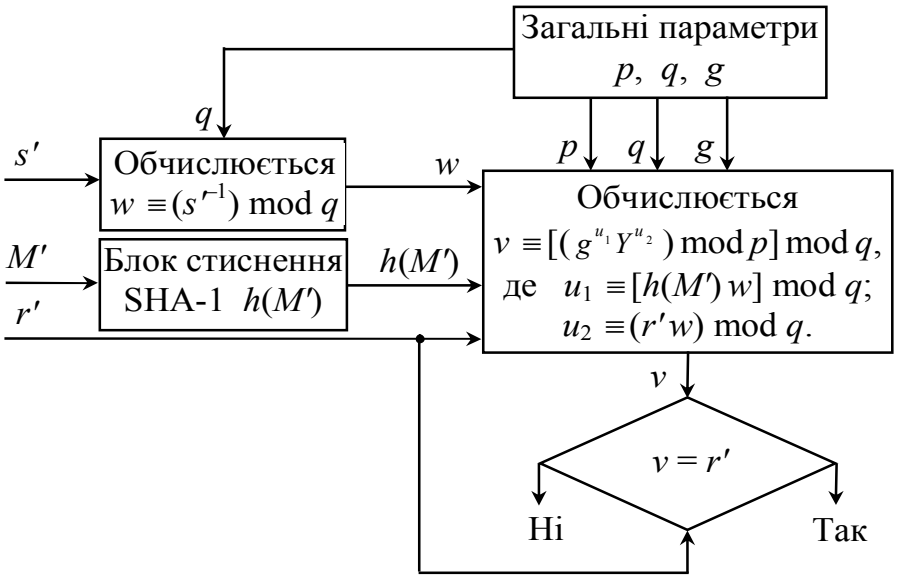


Рисунок 4.4 – Перевірка підпису DSA

Одержувач перевіряє виконання умов $0 < r' < q$ та $0 < s' < q$; якщо хоча б одну з них порушено, підпис скасовується. Потім обчислюється величина v, яка є функцією компонентів відкритого ключа, відкритого ключа відправника і геш-коду або надісланого повідомлення. Якщо ця величина відповідає компоненту r' підпису, то підпис підтверджується. Слід звернути увагу на те,

що перевірка наприкінці здійснюється зі значенням r', яке не залежить від повідомлення. Обґрунтування рівності $v = r$ в алгоритмі цифрового підпису DSA подано в додатку Д.

Приклад 4.5 Нехай $p = 211$; $q = 7$; $g = 144$; $k = 2$; $x = 3$; $h(M) = 15$. Відкритий ключ користувача

$$Y \equiv 144^2 \pmod{211} \equiv 58.$$

Створення підпису:

$$r \equiv [144^3 \pmod{211}] \bmod 7 \equiv 123 \pmod 7 \equiv 4;$$
$$s \equiv [3^{-1}(15 + 2 \cdot 4)] \bmod 7 \equiv (3^{-1} \cdot 23) \bmod 7 \equiv 3.$$

Підпис: (4, 3). Перевірка підпису:

$$w \equiv (3^{-1}) \bmod 7 \equiv 5;$$
$$u_1 \equiv (15 \cdot 5) \bmod 7 \equiv 5;$$
$$u_2 \equiv (4 \cdot 5) \bmod 7 \equiv 6;$$
$$v \equiv [(144^5 \cdot 58^6) \bmod 211] \bmod 7 \equiv 123 \pmod 7 \equiv 4.$$

$v = r = 4$ – підпис є справжній.

Стійкість DSA

Криптографічна стійкість схеми DSA проти атак методом „грубої сили” в першу чергу залежить від обсягу параметрів p і q (у даному разі 512 та 160 біт). Відповідно криптостійкість проти атаки методом „грубої сили” на параметр p становитиме 2^{160}. А успішна атака на параметр q можлива лише в тому разі, якщо зловмисник зможе обчислювати дискретні логарифми в полях Галуа $GF(2^{512})$ з кількістю попередніх обчислень пропорційно до

$$L(p) = \mathrm{e}^{\sqrt{\ln p \ln\ln p}}.$$

Однією з теоретично можливих атак на схему DSA є компрометація параметра x. Для кожного підпису потрібно нове значення x, яке має бути обране у випадковий спосіб. Якщо зловмисник розкриє значення x, яке вживалося при підписанні повідомлення (таке є можливе, якщо буде виявлено певні слабкості у процедурі генерування x), секретний ключ k може бути відтворено. Інший можливий варіант: два підписи було згенеровано на одному значенні x. У цьому разі зловмисник так само буде у змозі відновити k. Отже, одним з чинників, що підвищують безпеку використовування схем ЕЦП, є наявність „доброго” генератора випадкових чисел.

Вправи

1 Абоненти корпоративної мережі застосовують електронний цифровий підпис стандарту DSA із загальними параметрами $p = 251$; $q = 25$; $h(M) = 17$.

Для зазначених секретних параметрів абонентів знайти відкритий ключ Y і побудувати підпис для повідомлення m (виконати перевірку підпису).

№ варіанта	k	x	g	№ варіанта	k	x	g
1	8	3	20	11	12	8	243
2	11	2	64	12	19	8	211
3	9	7	149	13	24	3	241
4	15	6	219	14	16	2	94
5	13	8	20	15	20	8	125
6	11	3	149	16	18	8	20
7	8	7	219	17	23	13	243
8	7	6	20	18	17	11	113
9	6	9	64	19	21	17	101
10	13	6	249	20	18	19	93

2 Абоненти корпоративної мережі застосовують електронний цифровий підпис стандарту DSA. Для заданого відкритого ключа Y і відомих параметрів $p = 271$; $q = 27$ перевірити дійсність підписаних повідомлень.

№ варіанта	g	Y	$h(M)$	Повідомлення
1	211	77	17	(5, 3); (21, 19); (23, 18)
2	106	28	17	(7, 3); (13, 8); (17, 10)
3	217	160	13	(5, 7); (16, 7); (16, 23)
4	140	160	13	(5, 23); (5, 5); (13, 8)
5	114	169	17	(5, 9); (18, 4); (5, 25)
6	258	125	13	(9, 7); (17, 1); (17, 4)
7	5	126	13	(2, 13); (2, 5); (22, 19)
8	238	144	17	(7, 23); (21, 19); (23, 8)
9	238	140	13	(2, 3); (25, 9); (2, 10)
10	206	125	17	(15, 3); (1, 13); (18, 11)
11	126	160	17	(2, 13); (22, 11); (22, 7)
12	156	144	17	(17, 14); (18, 5); (3, 17)
13	217	28	17	(2, 7); (14, 3); (18, 19)
14	211	258	17	(17, 3); (17, 16); (7, 1)
15	88	106	17	(5, 3); (23, 23); (2, 17)
16	248	178	17	(16, 3); (6, 19); (16, 14)

3 Абоненти корпоративної мережі застосовують електронний цифровий підпис стандарту DSA із загальним параметром $p = 251$. Знайти відкритий ключ Y і побудувати підпис для повідомлення $m = h(M)$ (виконати перевірку підпису).

№ варіанта	k	x	g	q	$h(M)$
1	18	8	20	25	67
2	11	3	20	25	59
3	15	6	21	125	93
4	13	9	20	25	47
5	12	9	21	125	91
6	24	9	26	125	43
7	16	7	20	25	83
8	20	7	21	125	51
9	17	11	29	25	53
10	23	13	23	125	87

4.5 Стандарт електронного підпису за ГОСТом Р 34.10–94

У ГОСТі Р 34.10–94 [42] цифровий підпис становить собою два великих цілих числа. Загальнодоступні параметри схеми ЕЦП p, q і g мають задовольняти таким умовам:

– p – просте число, $2^{509} < p < 2^{512}$ або $2^{1020} < p < 2^{1024}$;
– q – простий дільник $(p-1)$ і $2^{254} < q < 2^{256}$;
– g: $g^q \pmod p \equiv 1$ і $1 < g < p-1$.

Секретний ключ користувача k обирається випадково і має задовольняти нерівності $0 < k < q$. Відкритий ключ користувача обчислюється відповідно до рівності $Y \equiv g^k \pmod p$.

Генерування ЕЦП. Процедура створення підпису повідомлення M складається з таких кроків:

– обчислюється геш-код повідомлення M: $h(M) = m$ (геш-функція, яка використовується в даному стандарті відповідно до ГОСТу Р 34.11–94). Якщо $h(M) \pmod p \equiv 0$, то $h(M)$ присвоюється значення $0\ldots0_{255}1$;

– випадково обирається значення x (аналогічно до DSA), що задовольняє умові $0 < x < q$;

– обчислюється значення $r \equiv (g^x \bmod p) \bmod q$; якщо $r = 0$, слід повернутися до попереднього етапу і дібрати інше значення x;

– обчислюється значення $s \equiv [k\,r + x\,h(M)] \bmod q$; якщо $s = 0$, то слід дібрати інше значення x. В іншому разі підписом повідомлення M є числа r і s.

Перевірка ЕЦП. Процедура перевірки ЕЦП складається з послідовності дій:

– перевіряється виконання умов $0 < s < q$ та $0 < r < q$; якщо хоча б одна з цих умов не виконується, підпис анулюється;

– обчислюється геш-код отриманого повідомлення $h(M) = m$; якщо $h(M) \pmod p \equiv 0$, то бітове подання $h(M)$ дорівнює $0\ldots0_{255}1$;

– обчислюються значення: $z_1 \equiv [s\,h(M)^{-1}] \bmod q$; $z_2 \equiv [-r\,h(M)^{-1}] \bmod q$;

– обчислюється значення $v \equiv [g^{z_1} Y^{z_2} \pmod p] \bmod q$;

– перевіряється рівність $r = v$; якщо вона виконується, підпис підтверджується, тобто в даному випадку вважається, що повідомлення підписано певним відправником і в процесі передавання його цілісність не було порушено.

З 2001 року замість ГОСТ Р 34.10–94 використовується новий російський стандарт ГОСТ Р 34.10–2001, що описує алгоритми формування та перевірки електронного цифрового підпису (повна назва: „ГОСТ Р 34.10–2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи”) [43].

Приклад 4.6 Нехай $p = 67$; $q = 11$; $g = 25$; $k = 6$; $x = 8$; $h(M) = 3$. Відкритий ключ користувача

$$Y \equiv 25^6 \pmod{67} \equiv 62.$$

Створення підпису для повідомлення M:

$$r \equiv [25^8 \pmod{67}] \bmod 11 \equiv 24 \pmod{11} \equiv 2;$$
$$s \equiv (6 \cdot 2 + 8 \cdot 3) \bmod 11 \equiv 36 \pmod{11} \equiv 3.$$

Отримуємо підписане повідомлення: (M, 2, 3).

Тепер виконаємо перевірку підпису. Якщо повідомлення не змінено, то $h(M) = 3$. Обчислимо

$$z_1 \equiv (3 \cdot 3^{-1}) \bmod 11 \equiv 12 \pmod{11} \equiv 1;$$
$$z_2 \equiv (-2 \cdot 3^{-1}) \bmod 11 \equiv 3;$$
$$v \equiv [(25^1 \cdot 62^3) \bmod 67] \bmod 11 \equiv 24 \pmod{11} \equiv 2.$$

Оскільки $v = r = 2$, підпис є справжній.

Вправи

У всіх завданнях припускатимемо, що $h(M) = m$ для всіх значень m.

1 Для зазначених відкритих ключів Y користувачів ГОСТа Р 34.10–94 із загальнодоступними параметрами $q = 11$, $p = 67$, $g = 25$ перевірити справжність підписаних повідомлень.

№ варіанта	Y	Повідомлення	№ варіанта	Y	Повідомлення
1	14	(10; 4, 5), (10; 7, 5), (10; 3, 8)	9	22	(6; 9, 5), (8; 8, 3), (7; 4, 1)
2	24	(1; 3, 5), (1; 4, 3), (1; 4, 5)	10	64	(10; 7, 3), (7; 7, 10), (8; 7, 5)
3	40	(7; 7, 4), (7; 9, 2), (5; 9, 2)	11	14	(10; 4, 5), (10; 7, 5), (10; 3, 8)
4	22	(6; 9, 5), (8; 8, 3), (7; 4, 1)	12	24	(1; 3, 5), (1; 4, 3), (1; 4, 5)
5	64	(10; 7, 3), (7; 7, 10), (8; 7, 5)	13	40	(7; 7, 4), (7; 9, 2), (5; 9, 2)
6	14	(10; 4, 5), (10; 7, 5), (10; 3, 8)	14	22	(6; 9, 5), (8; 8, 3), (7; 4, 1)
7	24	(1; 3, 5), (1; 4, 3), (1; 4, 5)	15	64	(10; 7, 3), (7; 7, 10), (8; 7, 5)
8	40	(7; 7, 4), (7; 9, 2), (5; 9, 2)	16	14	(10; 4, 5), (10; 7, 5), (10; 3, 8)

2 Група користувачів ГОСТу Р 34.10–94 має у своєму розпорядженні загальнодоступні параметри $q = 11$, $p = 67$, $g = 25$. Обчислити відкритий ключ Y і побудувати підпис для повідомлення m при таких секретних параметрах.

№ варіанта	k	x	m	№ варіанта	k	x	m
1	3	1	13	11	5	7	7
2	8	3	4	12	2	9	19
3	5	9	5	13	4	7	21
4	2	7	6	14	5	6	10
5	4	9	14	15	7	5	28
6	5	3	9	16	9	3	25
7	7	9	15	17	8	3	21
8	9	7	16	18	6	9	31
9	10	3	8	19	7	8	27
10	8	5	17	20	10	9	24

4.6 Алгоритм електронного підпису ECDSA

Алгоритм ECDSA (Elliptic Curve Digest Signature Algorithm) прийнято в якості стандартів ANSI X9.62 та IEEE P1363 [38, 39, 40]. ECDSA – система з відкритим ключем для створення цифрового підпису, аналогічна за своєю будовою до DSA, але визначуваний не над полем цілих чисел, а в групі точок еліптичної кривої.

Алгоритм генерування ключів:

1) обираємо еліптичну криву E, визначену на Z_p. Кількість точок в $E(Z_p)$ повинно ділитися на велике ціле n;

2) обираємо точку $P \in E(Z_p)$ порядку n;

3) обираємо довільне число $d \in [1, n-1]$;

4) обчислюємо $Q = dP$;

5) секретним ключем оголошуємо d, відкритим – (E, P, n, Q).

Алгоритм формування підпису під повідомленням M:

1) обираємо випадкове число $k \in [1, n-1]$;

2) обчислюємо $kP = (x_1, y_1)$ та $r \equiv x_1 \pmod n$. Якщо $r \neq 0$, переходимо до кроку 3, в противному разі – повертаємося до кроку 1;

3) обчислюємо $k^{-1} \pmod n$;

4) обчислюємо $s \equiv [k^{-1}(h(M) + dr)] \bmod n$. Якщо $s \neq 0$, переходимо до кроку 5, в противному разі – повертаємося до кроку 1;

5) підписом під повідомленням M є пара цілих чисел (r, s).

Примітки:

1) В якості геш-функції $h(M)$ на кроці 4 в стандартах ANSI X9.62 і IEEE P1363 використовується SHA-1.

2) За $r = 0$ результат обчислення s не залежить від секретного ключа d.

3) За $s = 0$ потрібного для перевірки підпису числа $s^{-1} (\mathrm{mod}\ n)$ не існує.

Алгоритм перевірки підпису:

1) якщо r і s – цілі числа, належать до інтервалу $[1, n - 1]$, переходимо до кроку 2, в противному разі – результат перевірки є негативний (підпис анулюється);

2) обчислюємо $w \equiv s^{-1} (\mathrm{mod}\ n)$ та $h(M)$;

3) обчислюємо $u_1 \equiv [h(M)w] \bmod n$ та $u_2 \equiv (r\,w) \bmod n$;

4) обчислюємо $u_1 P + u_2 Q = (x_0, y_0)$ та $v \equiv x_0 \ (\mathrm{mod}\ n)$;

5) підпис є справжній за $v = r$.

Приклад 4.7 Нехай $p = 211$; $n = 241$; $h(M) = 17$; $P = (2, 2)$; $E_p(0, -4)$, що відповідає кривій $y^2 = x^3 - 4$.

Генерування ключів

Обираємо будь-яке число $d = 7$ і обчислюємо $Q = 7(2, 2) = (179, 199)$.

Розглянемо детальніше вирахування одержаного результату.

1) Складемо дві точки (2, 2) + (2, 2):

$$\text{л} \equiv \left(\frac{3 \cdot 2^2 + 0}{2 \cdot 2} \right) \bmod 211 \equiv 3;$$

$$x_3 \equiv (3^2 - 2 - 2) \bmod 211 \equiv 5;$$

$$y_3 \equiv [3\ (2 - 5) - 2] \bmod 211 \equiv -11\ (\mathrm{mod}\ 211) \equiv 200.$$

Таким чином, $(2, 2) + (2, 2) = 2(2, 2) = (5, 200)$.

2) Складемо дві точки (5, 200) + (5, 200):

$$\text{л} \equiv \left(\frac{3 \cdot 5^2 + 0}{2 \cdot 200} \right) \bmod 211 \equiv \frac{75}{400} \bmod 211 \equiv \frac{3}{16} \bmod 211 \equiv 198;$$

$$x_3 \equiv (198^2 - 5 - 5) \bmod 211 \equiv 159;$$

$$y_3 \equiv [198\ (5 - 159) - 200] \bmod 211 \equiv -97\ (\mathrm{mod}\ 211) \equiv 114.$$

Отже, $(5, 200) + (5, 200) = 4(2, 2) = (159, 114)$.

3) Складемо дві точки (159, 114) + (5, 200):

$$\text{л} \equiv \left(\frac{200 - 114}{5 - 159} \right) \bmod 211 \equiv -\frac{86}{154} \bmod 211 \equiv -17\ (\mathrm{mod}\ 211) \equiv 194;$$

$$x_3 \equiv (194^2 - 159 - 5) \bmod 211 \equiv 125;$$

$$y_3 \equiv [194\ (159 - 125) - 114] \bmod 211 \equiv 152.$$

Отже, $(159, 114) + (5, 200) = 6(2, 2) = (125, 152)$.

4) Складемо дві точки (125, 152) + (2, 2):

$$\text{л} \equiv \left(\frac{2 - 152}{2 - 125} \right) \bmod 211 \equiv \frac{150}{123} \bmod 211 \equiv \left(150 \cdot 199\right) \bmod 211 \equiv 99;$$

$$x_3 \equiv (99^2 - 125 - 2) \bmod 211 \equiv 179;$$
$$y_3 \equiv [99\,(125 - 179) - 152] \bmod 211 \equiv -12 \pmod{211} \equiv 199.$$

Отже, $(125, 152) + (2, 2) = 7(2, 2) = (179, 199)$.
Секретним ключем є $d = 7$, відкритим – $[E_{211}(0, -4), (2, 2), 241, (179, 199)]$.

Формування підпису під повідомленням M:

1) обираємо випадкове число $k = 23$;

2) обчислюємо $23(2, 2) = (87, 50)$ и $r \equiv 87 \pmod{241} \equiv 87$;

3) обчислюємо $23^{-1} \pmod{241} \equiv 21$;

4) обчислюємо $s \equiv [21\,(17 + 7 \cdot 87)] \bmod 241 \equiv 132$;

5) підписом під повідомленням M є пара цілих чисел (87, 132).

Перевірка підпису:

1) числа $r = 87$ та $s = 132$ – цілі числа, що належать до інтервалу [1, 240];

2) обчислюємо $w \equiv 132^{-1} \pmod{241} \equiv 42$;

3) обчислюємо

$$u_1 \equiv (17 \cdot 42) \bmod 241 \equiv 232;$$
$$u_2 \equiv (87 \cdot 42) \bmod 241 \equiv 39;$$

4) обчислюємо

$$232\,(2, 2) + 39\,(179, 199) = (111, 66) + (136, 11) = (87, 50);$$
$$v \equiv 87 \pmod{241} \equiv 87;$$

5) підпис є справжній, оскільки $v = r = 87$.

4.7 Класифікація атак на схеми електронного підпису

Стійкість схеми електронного підпису залежить від стійкості використовуваних криптоалгоритмів та геш-функцій і визначається стосовно пари загроза–атака.

Наведемо класифікацію атак на схеми електронного підпису:

– *атака на основі відомого відкритого ключа* (key-only attack) – найслабша з атак, практично завжди доступна для зловмисника;

– *атака на основі відомих підписаних повідомлень* (known-message attack) – у розпорядженні зловмисника є певне (поліноміальне від k) число пар (M, S), де M – певне повідомлення, а S – припустимий підпис для нього, при цьому зловмисник не може впливати на вибір M;

– *проста атака з вибором підписаних повідомлень* (generic chosen-message attack) – зловмисник має можливість обрати певну кількість підписаних повідомлень, при цьому відкритий ключ він отримує після такого вибору;

– *спрямована атака з вибором повідомлень* (directed chosen-message attack) – обираючи підписані повідомлення, зловмисник знає відкритий ключ;

– *адаптивна атака з вибором повідомлень* (adaptive chosen-message attack) – зловмисник знає відкритий ключ; вибір кожного наступного підписаного повідомлення він може робити на основі знання припустимого підпису попереднього обраного повідомлення.

Кожна атака спрямована на досягнення певної мети. Можна відокремити такі різновиди загроз для схем електронного підпису (у порядку зростання потужності):

– *екзистенційна підробка* (existential forgery) – створення зловмисником підпису для якого-небудь, можливо безглуздого, повідомлення m', яке відрізняється від перехопленого;

– *селективна підробка* (selective forgery) – створення підпису для обраного повідомлення;

– *універсальна підробка* (universal forgery) – знаходження ефективного алгоритму формування підпису, функціонально еквівалентного до S;

– *повне розкриття* (total break) – обчислення секретного ключа, можливо відмінного від k^{secret}, який відповідає відкритому ключу k^{public}, що надає можливість формувати підписи для будь-яких повідомлень.

Найбільш надійними є схеми, стійкі проти найслабкішої із загроз на базі найпотужнішої з атак, тобто проти екзистенційної підробки на базі атаки з вибором підписаних повідомлень. Справедливе є твердження: *схеми електронного підпису, стійкі проти екзистенційної підробки на базі атаки з вибором підписаних повідомлень, існують тоді й тільки тоді, коли існують односпрямовані функції* [1, 28].

4.8 Особливі схеми електронного підпису

У певних ситуаціях можуть знадобитися схеми електронного підпису, відмінні від розглянутих класичних схем. Відомі такі спеціальні схеми електронного підпису:

– *схема „сліпої" підпису*, коли абонент A підписує документ, не знаючи його змісту (запропонована Девідом Чаумом);

– *схема групового підпису*, яка дозволяє верифікаторові переконатися у приналежності отриманого повідомлення певній групі претендентів, але хто саме з членів групи підписав документ, верифікатор визначити не в змозі;

– *схема розподіленого підпису*, яка формується лише за участі певної кількості учасників протоколу, інакше кажучи, ця схема є об'єднанням класичної схеми підпису та схеми поділу секрету;

– *схема конфіденційного підпису*, яка не може бути перевірена без участі учасника протоколу, який її сформував;

– *схема беззаперечного підпису*, в якій підробку підпису може бути доведено.

4.9 Електронні гроші

Розглянуті вище криптографічні методи часто використовуються як інструменти для розв'язування інших практично важливих завдань. Сучасна криптографія дозволяє розв'язувати проблеми, які раніше вважалися принципово нерозв'язними. Причому сьогодні багато таких можливостей криптографії використовуються в реальних комп'ютерних системах. Це і укладання комерційних угод в режимі віддаленої взаємодії учасників, і здійснення грошових розрахунків мережею, і проведення виборів з комп'ютерних мереж та багато чого іншого. Звернімо увагу на те, що криптографічні алгоритми не просто надають нові можливості користувачеві (наприклад, не треба відвідувати банк, можна виконати всі необхідні операції зі свого домашнього комп'ютера). Важливим є те, що ці алгоритми здатні забезпечувати надійність значно вищу, ніж традиційні механізми. Наприклад, якщо паперову банкноту можна підробити, і випадки підробок є надто численні, то електронну банкноту, створену за допомогою криптографічних методів, підробити практично неможливо.

У багатьох країнах сплачують за придбані речі при допомозі електронних карток, які дозволяють також замовляти авіаквитки через Інтернет, купувати найрізноманітніші товари в Інтернет-магазинах. Відомості про покупки накопичуються в магазинах і банках. Тому з'явилася нова проблема, іноді названа як „проблема Великого Брата".

Суть проблеми полягає в тому, що зникає анонімність процесу купівлі, тобто інформація про покупки будь-якої особи може стати відомою третім особам і використовуватися проти неї. Наприклад, відомості щодо придбання квитка на потяг чи літак можуть становити інтерес для злочинців тощо. Тому виникла ідея розробити такі схеми електронних платежів, які б зберігали анонімність покупця тією самою мірою, що й при розрахунку готівкою. Такі протоколи називаються *електронними*, або *цифровими грошима* (digital cache), що підкреслює їхню головну властивість – забезпечувати такий самий ступінь анонімності, як і звичайні гроші. Деякі схеми вже використовуються в реальному житті.

Описана нижче схема була запропонована Д. Чаумом (David Chaum)[11].

Розглянемо дві не надто вдалі схеми, а потім більш оптимальну, щоб легше було збагнути суть методу.

---

[11] 1982 року **Девід Чаум** заснував Міжнародну Асоціацію криптографічних Досліджень (IACR), яка сьогодні зорганізовує академічні конференції з досліджень криптографії. Зробив значний внесок у просування електронних грошей частково в ролі засновника DigiCash та електронної платіжної системи eCash. Внески Чаума у криптографію включають винахід двох мереж анонімності. Винайшов кілька важливих цифрових підписів, різні методи для анонімного мандата, перший запропонував методи для анонімних цифрових угод та цифрові гроші. Чаум є творцем системи шифрування електронного голосування.

Спочатку дамо більш точну постановку задачі. Є три учасники: банк, покупець і магазин. Покупець і магазин мають відповідні рахунки в банку, і покупець хоче придбати товар в магазині. Купівля здійснюється у вигляді триступінчастого процесу:

1) покупець знімає потрібну суму зі свого рахунку в банку;

2) покупець „пересилає" гроші до магазину;

3) магазин повідомляє про це банк, відповідна сума грошей зараховується на рахунок магазину, а покупець забирає товар (чи останній йому доставляється).

Наша мета – обрати таку схему, щоб вона була надійна; щоб банк не знав, хто саме купив товар, тобто було збережено анонімність звичайних грошей.

Опишемо *першу не надто вдалу схему* (вона базується на RSA). Банк має таку інформацію: секретні числа p, q, d й відкриті e, n.

Припустімо, покупець вирішив витратити певну, заздалегідь обумовлену з банком, суму (наприклад 100 грн.). (Спочатку розглянемо випадок, коли може використовуватися „банкнота" тільки одного номіналу, скажімо, 100 грн.) Покупець доправляє в банк число x, яке буде номером банкноти (зазвичай це потрібно, аби генерувати випадкове число у проміжку $(2,\ n-1)$). Банк обчислює число

$$s \equiv x^d \pmod{n}$$

і формує банкноту $(x,\ s)$, яку повертає покупцеві, попередньо зменшивши його рахунок на 100 грн. Параметр s в банкноті – це підпис банку. Ніхто не може підробити підпис, тому що число d є секретне.

Покупець пред'являє банкноту (x, s) в магазині, щоб купити товар. Магазин доправляє цю банкноту в банк для перевірки. Перш за все банк перевіряє справжність підпису (цю перевірку міг би зробити і магазин, використовуючи відкриті ключі банку). Але, окрім цього, банк зберігає всі номери повернутих до нього банкнот і перевіряє, чи немає числа x в цьому списку. Якщо x є в списку, то платіж не приймається (дехто намагається використати банкноту повторно), і банк повідомляє про це магазину. Якщо ж всі перевірки пройшли успішно, то банк додає 100 грн. на рахунок магазину, а магазин відпускає товар покупцеві.

Недолік цієї схеми – відсутня анонімність. Банк, а також всі, хто має доступ до відкритих ліній зв'язку, можуть запам'ятати, якому покупцеві відповідає число x, і тим самим з'ясувати, хто саме придбав товар.

Розглянемо *другу не надто вдалу схему*, яка вже забезпечує анонімність. Ця схема базується на так званому „сліпому" підпису.

Знову покупець хоче придбати товар. Він генерує число x, яке тепер не буде надсилатися у банк. Потім він генерує випадкове число r, взаємно просте з n, і обчислює число

$$g \equiv (x\, r^e) \bmod n.$$

Число g покупець доправляє до банку. Банк обчислює число

$$s' \equiv g^d \pmod n$$

і доправляє s' назад покупцеві (не забувши при цьому зняти 100 грн. з його рахунку).

Покупець обчислює

$$s \equiv (s'\, r^{-1}) \bmod n,$$

тобто отриманий підпис банку до x, але самого числа x ані банк, ані будь-хто інший не бачив. Обчислення s' називається „сліпим підписом", оскільки реальне повідомлення x підписуючий не бачить і дізнатися про нього не може. Отже, покупець має число x, яке є нікому невідоме і ніколи не передавалося каналами зв'язку. Покупець формує банкноту (x, s) і чинить так само, як в першій не надто вдалій схемі. Але тепер ніхто не знає, кому належить ця банкнота, тобто вона стала анонімною, як звичайна паперова банкнота.

Дії магазину і банку після пред'явлення покупцем банкноти (x, s) нічим не відрізняються від дій, описаних у першій схемі.

Чому ж ця схема є не надто вдала? Вона має недолік: можна сфабрикувати фальшиву банкноту, якщо відомі хоча б дві справжні. Робиться це так. Нехай зловмисник (чи покупець або магазин) має дві справжні банкноти – (x_1, s_1) та (x_2, s_2). Тоді він легко зможе виготовити фальшиву банкноту (x_3, s_3), обчисливши

$$x_3 \equiv (x_1\, x_2) \bmod n;$$
$$s_3 \equiv (s_1\, s_2) \bmod n.$$

Дійсно,

$$x_3^d \equiv (x_1\, x_2)^d \bmod n \equiv (\, x_1^d\, x_2^d\,) \bmod n \equiv (s_1\, s_2) \bmod n \equiv s_3,$$

тобто s_3 є справжнім підписом для x_3, і у банку немає жодних причин, щоб не прийняти цю фальшиву банкноту (він просто не зможе відрізнити її від справжньої). Це так звана *мультиплікативна* властивість системи RSA.

Опишемо, врешті, *більш оптимальну схему*, в якій усунено всі недоліки перших двох. В одному варіанті такої схеми використовується певна односпрямована функція $F(x)$. Функція F не є секретна і відома всім (покупцеві, банку і магазину).

Банкнота тепер визначається як пара чисел (x, s_F), де

$$s_F \equiv [F(x)]^d \bmod n,$$

тобто підписується не x, а значення $F(x)$.

Покупець генерує x (нікому його не показуючи), обчислює $F(x)$, підписує в банку за допомогою „сліпого” підпису число $F(x)$ і формує банкноту (x, s_F). Ця банкнота має всі позитивні властивості, як і в другій схемі, але підробити таку банкноту неможливо, так само як неможливо обчислити обернену функцію. Для перевірки підпису (тобто справжності банкноти) потрібно обчислити $F(x)$ і переконатися, що

$$s_F^e \pmod n \equiv F(x).$$

Зауважимо, що при виборі односпрямованих функції слід виявляти обережність. Наприклад, функція $F(x) \equiv a^2 \bmod n$ не підходить для певного протоколу. На практиці в якості $F(x)$ завжди використовуються криптографічні геш-функції, описані в розділі 3. Вся решта дій магазину та банку залишається такими самими, як і в раніш описаних схемах.

Існує ще один, більш простий, спосіб боротьби з мультиплікативною властивістю системи RSA – внесення надлишковості в повідомлення. Припустімо, що довжина модуля n – 1024 біти. Такою самою може бути й довжина числа x. Будемо записувати (випадково обираючи) номер банкноти лише в молодші 512 біт x, а в старші 512 біт x запишемо певне фіксоване число. Це фіксоване число може містити корисну інформацію, таку, приміром, як номінал банкноти та назва банку. Тепер банк при пред’явленні йому банкноти неодмінно перевірятиме наявність фіксованого заголовка в параметрі x і відкидати банкноту в разі його відсутності. Ймовірність того, що при множенні двох чисел за модулем n результат буде збіжним з ними в 512-ти бітах, є неймовірно мала. Тому отримати фальшиву банкноту за формулою не вдасться.

Приклад 4.8 Нехай за секретні параметри банку обрано числа $p = 17$, $q = 7$, $d = 77$. Відповідні їм відкриті параметри $n = 119$, $e = 5$. Для запобігання можливості підробки банкнот за їхні припустимі номери вважаються лише числа, що складаються з двох однакових десяткових цифр, наприклад 11, 77, 99. Коли покупець хоче отримати банкноту, він спочатку випадково обирає її номер з числа припустимих. Припустімо, покупець обрав $x = 33$. Потім відшукує випадкове число $r = 67$, взаємно просте з n (НСД (67, 119) = 1). Далі покупець обчислює

$$g \equiv (33 \cdot 67^5) \bmod 119 \equiv (33 \cdot 16) \bmod 119 \equiv 52.$$

Саме число $g = 52$ покупець надсилає до банку.

Банк списує з рахунку покупця 100 грн. і доправляє йому число

$$s' \equiv (52^{77}) \bmod 119 \equiv 103.$$

Покупець обчислює

$$s \equiv (103 \cdot 67^{-1}) \bmod 119 \equiv (103 \cdot 16) \bmod 119 \equiv 101$$

і отримує платоспроможну банкноту

$$(x, s) = (33, 101).$$

Цю банкноту він приносить (чи надсилає) до магазину, щоб придбати товар.

Магазин пред'являє банкноту до банку. Банк робить перевірки:

1) номер банкноти $x = 33$ складається з двох однакових десяткових цифр (тобто містить потрібну надлишковість);

2) раніше банкнота з таким номером не пред'являлася;

3) підпис банку є справжній, тобто $33^5 \pmod{119} \equiv 101$.

Якщо всі перевірки пройшли успішно, банк зараховує 100 грн. (це фіксований номінал банкноти) на рахунок магазину, про що йому й повідомляє. Магазин відпускає товар покупцеві.

На завершення розберемо ще дві проблеми, які виникають у зв'язку з розглянутою схемою електронних грошей.

Перша проблема. У поданій схемі незалежно діючі покупці чи навіть один покупець, який не пам'ятає номерів раніше використаних ним банкнот, можуть випадково згенерувати дві чи то більше банкноти з однаковими номерами. За умовами протоколу банк візьме до сплати лише одну з таких банкнот (ту, яку буде подано першою). Проте, візьмімо до уваги розміри чисел, використаних у протоколі. Якщо номер банкноти – число довжиною 512 біт і покупці генерують його насправді випадково, то ймовірність отримання одного чи двох однакових номерів є неймовірно мала.

Друга проблема полягає в тому, що у розглянутій схемі використовуються лише банкноти одного фіксованого номіналу, що, зрозуміло ж, є незручно для покупця. Вирішення проблеми використовування банкнот різного номіналу є можливий у такий спосіб. Банк заводить кілька пар (e_i, d_i) і оголошує, що e_1 відповідає, наприклад, 500 грн., e_2 – 200 грн. і т. д. Коли покупець запитує „сліпий" підпис у банку, він додатково повідомляє, якого номіналу банкноту хоче отримати. Банк знімає з його рахунку суму, що дорівнює зазначеному номіналу, і формує підпис, використовуючи відповідне секретне число d_i. Коли згодом банк отримує підписану банкноту, він використовує для перевірки підпису по черзі числа e_1, e_2 і т. д. Якщо підпис буде справжнім для певного e_i, то приймається банкнота i-того номіналу. У разі, коли параметр x банкноти містить фіксований заголовок із зазначенням її номіналу, завдання перевірки підпису полегшується – банк одразу використовує потрібний ключ e_i.

Вправа

У системі електронних грошей обрано секретні параметри банку $q = 7$, $p = 17$, $d = 25$ і відповідні їм відкриті параметри $n = 119$; $e = 5$. Сформувати електронні банкноти з такими номерами:

а) $x = 11$, $r = 5$; б) $x = 99$, $r = 6$; в) $x = 55$, $r = 10$; г) $x = 44$, $r = 15$; д) $x = 77$, $r = 30$.

ТЕСТИ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1 З яких частин складається автоматизована система?

а) мережа зв'язку, апаратно-програмні засоби, персонал, інформаційні ресурси;

б) інформаційні технології, зовнішнє середовище, обчислювальна система, персонал та інформація;

в) інформаційні ресурси, персонал, операційна система, зовнішнє середовище, розподілена обчислювальна система.

2 Які властивості інформації підлягають захисту в автоматизованих системах та системах телекомунікації?

а) конфіденційність, цілісність, доступність, захищеність;

б) конфіденційність, цілісність, доступність;

в) цілісність, доступність, секретність.

3 Що означає поняття „документ" відповідно до нормативної бази України? Це:

а) інформація, зафіксована на будь-якому матеріальному носії;

б) інформація, зафіксована на будь-якому матеріальному носії у визначеному законом порядку;

в) інформація, зафіксована на будь-якому матеріальному носії та зареєстрована у державному органі.

4 У яких трьох аспектах має розвиватися проблема захисту інформації в телекомунікаційних системах?

а) вдосконалення відповідної нормативної бази, організаційних заходів та програмно-апаратних засобів;

б) вдосконалення відповідної нормативної бази та розподільних обчислювальних мереж, розробляння сучасних захищуваних інформаційних технологій.

5 Що називають загрозами інформаційних об'єктам?

а) потенційно можливі події, які призводять до порушень політики безпеки;

б) потенційно можливі події, які призводять до втрат конфіденційності, цілісності та доступність інформації;

в) потенційно можливі події, які призводять до втрат електронних документів.

6 Що називають атаками на інформаційні об'єкти?

а) спроба реалізації загрози;

б) підготовка, реалізація несанкціонованого доступу до захищуваного об'єкта та усунення слідів нападу;

в) факт несанкціонованого доступу до захищуваного об'єкта.

7 Атака – це спроба реалізації загрози?

а) так;

б) ні.

8 Загрози, які мають суб'єктивну природу, можуть бути лише випадковими?

а) так;

б) ні.

9 Конфіденційність, цілісність та доступність – властивості інформації?

а) так;

б) ні.

10 Цілісність – це відсутність зумисного спотворення інформації?

а) так;

б) ні.

11 Автоматизована система – це:

а) система, що здійснює автоматизоване опрацьовування даних, до складу якої входять технічні засоби їхнього опрацьовування (засоби обчислювальної техніки та зв'язку), а також методи і процедури, програмне забезпечення;

б) система, за допомогою якої забезпечується зв'язок поміж користувачами.

12 Які алгоритми називають симетричними?

а) блочні, потокові, з відкритим ключем;

б) алгоритми, в яких операції шифрування виконуються одним ключем.

13 Які криптографічні перетворювання використовуються для створювання симетричних криптографічних алгоритмів?

а) гамування, переставляння і заміни;

б) функційні перетворювання, переставляння та заміни;

в) параметричні перетворювання, переставляння та заміни, гамування.

14 Які криптографічні перетворювання використовуються у стандарті DES?

а) зворотні табличні переставляння, табличні заміни, переставляння з розширенням блоків, переставляння зі стисненням блоків, циклічні переставляння;

б) табличні переставляння, табличні заміни, функційні перетворювання.

15 Які криптографічні перетворювання використовуються у стандарті ГОСТ 28147–89?

а) зворотні табличні переставляння, табличні заміни, переставляння з розширенням блоків, переставляння зі стискуванням блоків, циклічні переставляння;

б) табличні переставляння, табличні заміни та функційні перетворювання.

16 До потокових алгоритмів шифрування належить:

а) алгоритм, який використовує різні ключі зашифровування і розшифровування;

б) алгоритм, в якому кожен символ відкритого тексту зашифровується незалежно від інших і розшифровується так само.

17 ГОСТ 28147–89 працює в робочих режимах:

а) гамування зі зворотним зв’язком по виходу;

б) зчеплення блоків шифру;

в) проста заміна.

18 Циклічний регістр зсуву в алгоритмі за ГОСТом 28147–89 виконується:

а) на 11 розрядів праворуч;

б) на 11 розрядів ліворуч;

в) на 11 розрядів праворуч, потім ліворуч, залежно від номера циклу.

19 Кількість циклів шифрування в алгоритмі DES:

а) 16;

б) 32;

в) 64;

г) 128;

д) 256.

20 Розмір блока даних в алгоритмі за ГОСТом 28147–89:

а) 16;

б) 32;

в) 64;

г) 128;

д) 256.

21 Розмір ключа в алгоритмі за ГОСТом 28147–89:

а) 16;

б) 32;

в) 64;

г) 128;

д) 256.

22 Розмір ключа в алгоритмі DES:

а) 16;

б) 32;

в) 48;

г) 59;

д) 64.

23 Кількість циклів шифрування в алгоритмі за ГОСТом 28147–89:

а) 16;

б) 32;

в) 64;

г) 128;

д) 256.

24 Які параметри обираються в алгоритмі RSA:

а) два великих простих числа, ключ зашифровування;

б) функція Ейлера, ключ розшифровування;

в) ключ зашифровування і розшифровування.

25 Для чого призначено алгоритм Діффі–Хеллмана:

а) для шифрування повідомлень;

б) для генерування секретного ключа.

26 Алгоритм Ель–Гамаля можна використовувати:

а) для шифрування повідомлень і генерування секретного ключа;

б) для генерування секретного ключа;

в) для шифрування повідомлень та створювання цифрового підпису.

27 Геш-функція застосовується у криптографії:

а) для шифрування повідомлень;

б) для генерування секретного ключа;

в) у протоколах автентифікації цифрового підпису.

28 Зашифрувати повідомлення M за допомогою алгоритму RSA, якщо відомо: $n = 15$, ключ зашифровування $e = 3$, $M = 3$.

а) 16;

б) 9;

в) 12;

г) 25.

29 Розшифрувати криптограму C за допомогою алгоритму RSA, якщо відомо: $n = 21$, ключ розшифровування $d = 3$, $C = 5$.

а) 20;

б) 17;

в) 15;

г) 7.

30 Знайти найбільший спільний дільник для чисел 85, 34:

а) 1;

б) 5;

в) 7;

г) 17.

31 Обчислити $4^8 \pmod{14}$.

а) 1;

б) 2;

в) 4^4;

г) 14.

32 Знайти x, якщо $3^x \equiv 5 \pmod{7}$.

а) 3;

б) 5;

в) 7;

г) 6.

33 Розв'язати порівняння $6x \equiv 2 \pmod{11}$.

а) 3;

б) 4;

в) 6;

г) 11.

34 Обчислити функцію Ейлера $\varphi(49)$.

а) 26;

б) 36;

в) 48;

г) 42.

35 Зашифрувати повідомлення M за допомогою алгоритму Рабіна, якщо відомо: $p = 3$, $q = 7$, $M = 9$.

а) 27;

б) 60;

в) 18;

г) 21.

36 Зашифрувати повідомлення M за допомогою алгоритму Ель–Гамаля, якщо відомо: $p = 5$, відкритий ключ сторони $A - Y_a = 3$, секретний ключ сторони $B - k_b = 2$, $M = 6$.

а) 1;

б) 2;

в) 3;

г) 4.

37 Визначити ключ методом Діффі–Хеллмана, якщо відомо: $p = 5$, відкритий ключ сторони $A - Y_a = 6$, секретний ключ сторони $B - k_b = 3$.

а) 1;

б) 2;

в) 3.

ЛІТЕРАТУРА

1 **Введение** в криптографию; под общ. ред. В. В. Ященко. – СПб.: Питер, 2001. – 288 с.: ил.

2 **Барабаш А. В.** Криптография: науч.-поп. изд. Серия „Аспекты защиты” / А. В. Барабаш, Г. П. Шанкин. – М.: Солон-Р, 2002. – 511 с.

3 **Романец Ю. В.** Защита информации в компьютерных системах и сетях / Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф. – М.: Радио и связь, 2001. – 376 с.

4 **Анин Б. Ю.** Защита компьютерной информации / Анин Б. Ю. – СПб.: БХВ, 2000. – 384 с.

5 **Саломаа А.** Классическая криптография / Саломаа А.; пер. с англ.– М.: Мир, 1996. – 304 с.

6 **Петров А. А.** Компьютерная безопасность. Криптографические методы защиты / Петров А. А. – М.: ДМК, 2000. – 448 с.

7 **Жельников В.** Криптография от папируса до комп’ютера / Жельников В. – М.: ABF, 1997. – 336 с.

8 **Шнайдер Б.** Прикладная криптография / Шнайдер Б. – М.: Триумф, 2003. – 816 с.

9 **Андерсон Джеймс А.** Дискретная математика и комбинаторика / Андерсон Джеймс А.; пер. с англ. – М.: изд. дом “Вильямс”, 2003. – 960 с.

10 **Hellman M. E.** The mathematics of public-key cryptography / Hellman M. E. – Scientific American, 1979.

11 **Shamir A.** A Polynomial Time Algorithm for Breaking the Basic Merkle Hellman Cryptosystem / Shamir A. // IEEE Transactions on Information Theory. – 1984, Sep. – V. IT-30, n. 5. – Pp. 1699-1704.

12 **Rives R. L.** A Method for Obtaining Digital Signatures and Public-Key Cryptosystems / Rives R. L., Shamir A. and Adleman L. M. // Communications of the ACM. – 1978, Feb. – V. 21, n. 2. – Pp.120-126.

13 **Rives R. L.** On Digital Signatures and Public Key Cryptosystems / Rives R. L., Shamir A. and Adleman L. M. // MIT Laboratory for Computer Science: Technical Report. MIT/LCSATR-212, 1979, Jan.

14 **Simmons G. J.** A Wea Privacy Protocol Using the RSA Cryptosystem / Simmons G. J. // Cryptologia. – 1983, Apr. – V. 7, n. 2. – Pp. 180-182.

15 **DeLaurentis I. M.** A Further Weakness in the Common Modulus Protocol for the RSA Cryptosystem / DeLaurentis I. M. // Cryptologia. – 1984. – V. 8, n. 3. – Pp. 253-259.

16 **ISO/IEC 9796.** Information Technology Security Techniques. Digital Signature Scheme Giving Message Recovery. International Organization for Standardization. 1991.

17 **Rabin M. O.** Digital Signatures and Public-Key Functions as Intractable as Factorization / Rabin M. O. // MIT Laboratory for Computer Science: Technical Report. MIT/LCS/TR – 1979, Jan., p. 212.

18 **Williams H. C.** A Modification of the RSA Public-Key Encryption Procedure / Williams H. C. // IEEE Transactions on Information Theory. – 1980, Nov. – V. IT-26, n. 6. – Pp. 726-729.

19 **ElGamal T.** A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in Cryptology: Proceedings of CRYPTO 84 / ElGamal T. – Springer Verlag, 1985. – Pp. 1-18.

20 **ElGamal T.** On Computing Logarithms Over Finite Fields. Advances in Cryptology: Proceedings of CRYPTO 85 / ElGamal T. – Springer Verlag, 1986. – Pp. 396-402.

21 **Diffie W.** New Directions in Cryptography / W. Diffie and M. Hellman // IEEE Transactions on Information Theory. – 1976, Nov. – V. IT-22, n. 6. – Pp. 44-54.

22 **Болотов А. А.** Алгоритмические основы эллиптической криптографии / Болотов А. А., Гашков С. Б., Фролов А. Б. – М.: Мэй, 2000. – 100 с.

23 **Болотов А. А.** Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы / Болотов А. А., Гашков С. Б., Фролов А. Б. – М.: КомКнига, 2006. – 328 с.

24 **Болотов А. А.** Элементарное введение в эллиптическую криптографию: Протоколы криптографии на эллиптических кривих / Болотов А. А., Гашков С. Б., Фролов А. Б. – М.: КомКнига, 2006. – 280 с.

25 **Menezes A.** Handbook of Applied Cryptography / Menezes A., van Oorschot P., Vanstone S. – CRC Press, 1996. – 661 p.

26 **Blake I.** Elliptic Curves in Cryptography / Blake I., Seroussi G., Smart N. – Cambridge University Press, 1999.

27 **Menezes A.** Elliptic Curve Public Key Cryptosystems / Menezes A. – Kluwer Academic Publishers, 1993.

28 **Анохин М. И.** Криптография в банковском деле / Анохин М. И., Варновский Н. П., Сидельников В. М. – М.: МИФИ, 1997.

29 **Rivest R.** The MD5 Message-Digest Algorithm / Rivest R. – RFC 1321, MIT and RSA Data Security, Inc. 1992, Apr.

30 **Research** and Development in Advanced Communication Technologies in Europe. RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation (R1040). RACE. 1992, June. – Режим доступу: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.8213

31 **Federal** Information Processing Standards Publication 180-1 SECURE HASH STANDARD. 1995, Apr. – Режим доступу: http://www.netnsk.ru/publica/security/sec_05.htm

32 **Информационная технология.** Криптографическая защита информации. Функция хэширования: ГОСТ Р 34.11–94. (утв. Постановлением Госстандарта РФ от 23.05.1994 № 154).

33 **Системы** обработки информации. Защита криптографическая. Алгоритм криптографического преобразования: ГОСТ 28147–89. – Режим доступу: http://protect.gost.ru/document.aspx?control=7&id=139177

34 **Anderson R.** Security Engineering: A Guide to Building Dependable Distributed Systems, 2001, John Wiley & Sons, 640 p.

35 **PKCS #1** v 2.1: RSA Cryptography Standard RSA Laboratories. 2002, June. – Режим доступу: http://www.rsa.com

36 **Zheng Y., Pieprzyk J. and Seberry J.** HAVAL A One-Way Hashing Algorithm with Variable Length of Output: Springer Verlag, 1993. – Pp. 83-104.

37 **National** Institute of Standards and Technology. NIST FIPS PUB 186, Digital Signature Standard, U.S. Department of Commerce, 1994, May.

38 **ANSI X9.62–1999.** Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.

39 **ANSI X9.63–1999.** Public Key Cryptography For The Financial Services Industry: Elliptic Curve Key Agreement and Transport Protocols, 1999.

40 **IEEE Std 1363–2000.** IEEE Standard Specifications for Public-Key Cryptography, 2000.

41 **Goldwasser S., Bellare M.** Lecture notes on cryptography. 2008, July. – Режим доступу: http://www-cse.ucsd.edu/users/mihir/crypto-lecnotes.html

42 **Информационная технология**. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма: ГОСТ Р 34.10–94. – Режим доступу: http://www.securitylab.ru/informer/240665.php

43 **Информационная технология.** Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи: ГОСТ Р 34.10–2001. – Режим доступу: http://protect.gost.ru/v.aspx?control=7&id=131131

44 **Информационная технология**. Криптографическая защита информации. Функция хэширования: ГОСТ Р 34.11–94. – Режим доступу: http://protect.gost.ru/document.aspx?control=7&id=134550

45 **Информационная технология**. Криптографическая защита информации. Функция хэширования: ГОСТ 34.311–95. – Режим доступу: http://protect.gost.ru/document.aspx?control=7&id=132760

Б. Закон України „Про електронні документи та електронний документообіг”

ЗАКОН УКРАЇНИ

про електронні документи та електронний документообіг

Цей Закон встановлює основні організаційно-правові засади електронного документообігу та використання електронних документів.

Розділ І ЗАГАЛЬНІ ПОЛОЖЕННЯ

Стаття 1. Визначення термінів

У цьому Законі терміни вживаються в такому значенні:

адресат – фізична або юридична особа, якій адресується електронний документ;

дані – інформація, яка подана у формі, придатній для її оброблення електронними засобами;

посередник – фізична або юридична особа, яка в установленому законодавством порядку здійснює приймання, передавання (доставку), зберігання, перевірку цілісності електронних документів для задоволення власних потреб або надає відповідні послуги за дорученням інших суб’єктів електронного документообігу;

обов’язковий реквізит електронного документа – обов’язкові дані в електронному документі, без яких він не може бути підставою для його обліку і не матиме юридичної сили;

автор електронного документа – фізична або юридична особа, яка створила електронний документ;

суб’єкти електронного документообігу – автор, підписувач, адресат та посередник, які набувають передбачених законом або договором прав і обов’язків у процесі електронного документообігу.

Стаття 2. Сфера дії Закону

Дія цього Закону поширюється на відносини, що виникають у процесі створення, відправлення, передавання, одержання, зберігання, оброблення, використання та знищення електронних документів.

Стаття 3. Законодавство про електронні документи та електронний документообіг

Відносини, пов’язані з електронним документообігом та використанням електронних документів, регулюються Конституцією України, Цивільним кодексом України, законами України “Про інформацію”, “Про захист інформації в автоматизованих системах”, “Про державну таємницю”, “Про

зв’язок”, “Про обов’язковий примірник документів”, “Про Національний архівний фонд та архівні установи”, цим Законом, а також іншими нормативно-правовими актами.

Якщо міжнародним договором України, згода на обов’язковість якого надана Верховною Радою України, встановлено інші правила, ніж ті, що передбачені цим Законом, застосовуються правила міжнародного договору.

Стаття 4. Державне регулювання електронного документообігу

Кабінет Міністрів України та інші органи виконавчої влади в межах повноважень, визначених законом, реалізують державну політику електронного документообігу. Державне регулювання у сфері електронного документообігу спрямовано на:

реалізацію єдиної державної політики електронного документообігу;

забезпечення прав і законних інтересів суб’єктів електронного документообігу;

нормативно-правове забезпечення технології оброблення, створення, передавання, одержання, зберігання, використання та знищення електронних документів.

Розділ ІІ ЕЛЕКТРОННИЙ ДОКУМЕНТ

Стаття 5. Електронний документ

Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов’язкові реквізити документа.

Склад та порядок розміщення обов’язкових реквізитів електронних документів визначається законодавством.

Електронний документ може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму.

Візуальною формою подання електронного документа є відображення даних, які він містить, електронними засобами або на папері у формі, придатній для приймання його змісту людиною.

Стаття 6. Електронний підпис

Електронний підпис є обов’язковим реквізитом електронного документа, який використовується для ідентифікації автора та/або підписувача електронного документа іншими суб’єктами електронного документообігу.

Накладанням електронного підпису завершується створення електронного документа.

Відносини, пов’язані з використанням електронних цифрових підписів, регулюються законом.

Використання інших видів електронних підписів в електронному документообігу здійснюється суб'єктами електронного документообігу на договірних засадах.

Стаття 7. Оригінал електронного документа

Оригіналом електронного документа вважається електронний примірник документа з обов'язковими реквізитами, у тому числі з електронним цифровим підписом автора.

У разі надсилання електронного документа кільком адресатам або його зберігання на кількох електронних носіях інформації, кожний з електронних примірників вважається оригіналом електронного документа.

Якщо автором створюються ідентичні за документарною інформацією та реквізитами електронний документ та документ на папері, кожен з документів є оригіналом і має однакову юридичну силу.

Оригінал електронного документа повинен давати змогу довести його цілісність та справжність у порядку, визначеному законодавством; у визначених законодавством випадках може бути пред'явлений у візуальній формі відображення, в тому числі у паперовій копії.

Електронна копія електронного документа засвідчується у порядку, встановленому законом.

Копією документа на папері для електронного документа є візуальне подання електронного документа на папері, яке засвідчене в порядку, встановленому законодавством.

Стаття 8. Правовий статус електронного документа та його копії

Юридична сила електронного документа не може бути заперечена виключно через те, що він має електронну форму.

Допустимість електронного документа як доказу не може заперечуватися виключно на підставі того, що він має електронну форму.

Електронний документ не може бути застосовано як оригінал:

1) свідоцтва про право на спадщину;

2) документа, який відповідно до законодавства може бути створений лише в одному оригінальному примірнику, крім випадків існування централізованого сховища оригіналів електронних документів;

3) в інших випадках, передбачених законом.

Нотаріальне посвідчення цивільно-правової угоди, укладеної шляхом створення електронного документа (електронних документів), здійснюється у порядку, встановленому законом.

Розділ ІІІ ЗАСАДИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Стаття 9. Електронний документообіг

Електронний документообіг (обіг електронних документів) – сукупність процесів створення, оброблення, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів.

Порядок електронного документообігу визначається державними органами, органами місцевого самоврядування, підприємствами, установами та організаціями всіх форм власності згідно з законодавством.

Стаття 10. Відправлення та передавання електронних документів

Відправлення та передавання електронних документів здійснюється автором або посередником в електронній формі за допомогою засобів інформаційних, телекомунікаційних, інформаційно-телекомунікаційних систем або шляхом відправлення електронних носіїв, на яких записано цей документ.

Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, датою і часом відправлення електронного документа вважається дата і час, коли відправлення електронного документа не може бути скасовано особою, яка його відправила. У разі відправлення електронного документа шляхом пересилання його на електронному носії, на якому записано цей документ, датою і часом відправлення вважається дата і час здавання його для пересилання.

Вимоги підтвердження факту одержання документа встановлені законодавством у випадках відправлення документів рекомендованим листом або передавання їх під розписку, не поширюються на електронні документи. У таких випадках підтвердження факту одержання електронних документів здійснюється згідно з вимогами цього Закону.

Стаття 11. Одержання електронних документів

Електронний документ вважається одержаним адресатом з часу надходження авторові повідомлення в електронній формі від адресата про одержання цього електронного документа автора, якщо інше не передбачено законодавством або попередньою домовленістю між суб’єктами електронного документообігу.

Якщо попередньою домовленістю між суб’єктами електронного документообігу не визначено порядок підтвердження факту одержання електронного документа, таке підтвердження може бути здійснено в будь-якому порядку автоматизованим чи іншим способом в електронній формі або у формі документа на папері. Зазначене підтвердження повинно містити дані про

факт і час одержання електронного документа та про відправника цього підтвердження.

У разі ненадходження до автора підтвердження про факт одержання цього електронного документа вважається, що електронний документ не одержано адресатом.

Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, електронний документ вважається відправленим автором та одержаним адресатом за їх місцезнаходженням (для фізичних осіб – місцем проживання), у тому числі якщо інформаційна, телекомунікаційна, інформаційно-телекомунікаційна системи, за допомогою яких одержано документ, знаходиться в іншому місці. Місцезнаходження (місце проживання) сторін визначається відповідно до законодавства.

Стаття 12. Перевірка цілісності електронного документа

Перевірка цілісності електронного документа проводиться шляхом перевірки електронного цифрового підпису.

Стаття 13. Зберігання електронних документів та архіви електронних документів

Суб'єкти електронного документообігу повинні зберігати електронні документи на електронних носіях інформації у формі, що дає змогу перевірити їх цілісність на цих носіях.

Строк зберігання електронних документів на електронних носіях інформації повинен бути не меншим від строку, встановленого законодавством для відповідних документів на папері.

У разі неможливості зберігання електронних документів на електронних носіях інформації протягом строку, встановленого законодавством для відповідних документів на папері, суб'єкти електронного документообігу повинні вживати заходів щодо дублювання документів на кількох електронних носіях інформації та здійснювати їх періодичне копіювання відповідно до порядку обліку та копіювання документів, встановленого законодавством. Якщо неможливо виконати зазначені вимоги, електронні документи повинні зберігатися у вигляді копій документів на папері (у разі відсутності оригіналу цього документа на папері). При копіюванні електронного документа з електронного носія інформації обов'язково здійснюється перевірка цілісності даних на цьому носії.

При зберіганні електронних документів обов'язкове додержання таких вимог:

1) інформація, що міститься в електронних документах, повинна бути доступною для її подальшого використання;

2) має бути забезпечена можливість відновлення електронного документа у тому форматі, в якому він був створений, відправлений або одержаний;

3) у разі наявності повинна зберігатися інформація, яка дає змогу встановити походження та призначення електронного документа, а також дату і час його відправлення чи одержання.

Суб'єкти електронного документообігу можуть забезпечувати додержання вимог щодо збереження електронних документів шляхом використання послуг посередника, у тому числі архівної установи, якщо така установа додержується вимог цієї статті. Створення архівів електронних документів, подання електронних документів до архівних установ України та їх зберігання в цих установах здійснюється у порядку, визначеному законодавством.

Розділ IV ОРГАНІЗАЦІЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Стаття 14. Організація електронного документообігу

Електронний документообіг здійснюється відповідно до законодавства України або на підставі договорів, що визначають взаємовідносини суб'єктів електронного документообігу.

Використання електронного документа у цивільних відносинах здійснюється згідно із загальними вимогами вчинення правочинів, встановлених цивільним законодавством.

Стаття 15. Обіг електронних документів, що містять інформацію з обмеженим доступом

Суб'єкти електронного документообігу, які здійснюють його на договірних засадах, самостійно визначають режим доступу до електронних документів, які містять конфіденційну інформацію, та встановлюють для них систему (способи) захисту.

В інформаційних, телекомунікаційних, інформаційно-телекомунікаційних системах, які забезпечують обмін електронними документами, що містять інформацію, яка є власністю держави, або інформацію з обмеженим доступом, повинен забезпечуватися захист цієї інформації відповідно до законодавства.

Стаття 16. Права та обов'язки суб'єктів електронного документообігу

Суб'єкти електронного документообігу користуються правами та мають обов'язки, які встановлено для них законодавством.

Якщо в процесі організації електронного документообігу виникає необхідність у визначенні додаткових прав та обов'язків суб'єктів електронного документообігу, що не визначені законодавством, такі права та обов'язки можуть встановлюватися цими суб'єктами на договірних засадах.

Стаття 17. Вирішення спорів між суб'єктами електронного документообігу

Вирішення спорів між суб'єктами електронного документообігу

здійснюється в порядку, встановленому законом.

Стаття 18. Відповідальність за порушення законодавства про електронні документи та електронний документообіг

Особи, винні в порушенні законодавства про електронні документи та електронний документообіг, несуть відповідальність згідно з законами України.

Розділ V ПРИКІНЦЕВІ ПОЛОЖЕННЯ

1) Цей Закон набирає чинності через шість місяців з дня його опублікування.

2) Кабінету Міністрів України протягом шести місяців з дня набрання чинності цим Законом:

підготувати та подати на розгляд Верховної Ради України відповідні пропозиції про внесення змін до законодавчих актів України;

забезпечити прийняття нормативно-правових актів, передбачених цим Законом;

забезпечити перегляд і скасування міністерствами, іншими центральними органами виконавчої влади України їх нормативно-правових актів, що суперечать цьому Закону;

разом з Національним банком України розробити та внести на розгляд Верховної Ради України програму заходів щодо впровадження електронних документів, електронного документообігу та електронного цифрового підпису, стимулювання підприємств, установ і організацій, які впроваджують електронний документообіг.

м. Київ
22 травня 2003 року
№ 851–IV

В. Закон України „Про електронний цифровий підпис”

ЗАКОН УКРАЇНИ
про електронний цифровий підпис

Цей Закон визначає правовий статус електронного цифрового підпису та регулює відносини, що виникають при використанні електронного цифрового підпису.

Дія цього Закону не поширюється на відносини, що виникають під час використання інших видів електронного підпису, в тому числі переведеного у цифрову форму зображення власноручного підпису.

Якщо міжнародним договором, згода на обов’язковість якого надана Верховною Радою України, встановлено інші правила, ніж ті, що передбачені цим Законом, застосовуються правила міжнародного договору.

Стаття 1. Визначення термінів

У цьому Законі терміни вживаються у такому значенні:

електронний підпис – дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов’язані та призначені для ідентифікації підписувача цих даних;

електронний цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа;

засіб електронного цифрового підпису – програмний засіб, програмно-апаратний або апаратний пристрій, призначені для генерації ключів, накладення та/або перевірки електронного цифрового підпису;

особистий ключ – параметр криптографічного алгоритму формування електронного цифрового підпису, доступний тільки підписувачу;

відкритий ключ – параметр криптографічного алгоритму перевірки електронного цифрового підпису, доступний суб’єктам відносин у сфері використання електронного цифрового підпису;

засвідчення чинності відкритого ключа – процедура формування сертифіката відкритого ключа;

сертифікат відкритого ключа (далі – сертифікат ключа) – документ, виданий центром сертифікації ключів, який засвідчує чинність і належність відкритого ключа підписувачу. Сертифікати ключів можуть розповсюджуватися в електронній формі або у формі документа на папері та використовуватися для ідентифікації особи підписувача;

посилений сертифікат відкритого ключа (далі – посилений сертифікат ключа) – сертифікат ключа, який відповідає вимогам цього Закону, виданий акредитованим центром сертифікації ключів, засвідчувальним центром, центральним засвідчувальним органом;

акредитація – процедура документального засвідчення компетентності центра сертифікації ключів здійснювати діяльність, пов'язану з обслуговуванням посилених сертифікатів ключів;

компрометація особистого ключа – будь-яка подія та/або дія, що призвела або може призвести до несанкціонованого використання особистого ключа;

блокування сертифіката ключа – тимчасове зупинення чинності сертифіката ключа;

підписувач – особа, яка на законних підставах володіє особистим ключем та від свого імені або за дорученням особи, яку вона представляє, накладає електронний цифровий підпис під час створення електронного документа;

послуги електронного цифрового підпису – надання у користування засобів електронного цифрового підпису, допомога при генерації відкритих та особистих ключів, обслуговування сертифікатів ключів (формування, розповсюдження, скасування, зберігання, блокування та поновлення), надання інформації щодо чинних, скасованих і блокованих сертифікатів ключів, послуги фіксування часу, консультації та інші послуги, визначені цим Законом;

надійний засіб електронного цифрового підпису – засіб електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації. Підтвердження відповідності та проведення державної експертизи цих засобів здійснюється у порядку, визначеному законодавством.

Стаття 2. Суб'єкти правових відносин у сфері послуг електронного цифрового підпису

Суб'єктами правових відносин у сфері послуг електронного цифрового підпису є:

підписувач;

користувач;

центр сертифікації ключів;

акредитований центр сертифікації ключів;

центральний засвідчувальний орган;

засвідчувальний центр органу виконавчої влади або іншого державного органу (далі – засвідчувальний центр);

контролюючий орган.

Стаття 3. Правовий статус електронного цифрового підпису

Електронний цифровий підпис за правовим статусом прирівнюється до власноручного підпису (печатки) у разі, якщо:

електронний цифровий підпис підтверджено з використанням посиленого сертифіката ключа за допомогою надійних засобів цифрового підпису;

під час перевірки використовувався посилений сертифікат ключа, чинний на момент накладення електронного цифрового підпису;

особистий ключ підписувача відповідає відкритому ключу, зазначеному у сертифікаті.

Електронний підпис не може бути визнаний недійсним лише через те, що він має електронну форму або не ґрунтується на посиленому сертифікаті ключа.

Стаття 4. Призначення електронного цифрового підпису

Електронний цифровий підпис призначений для забезпечення діяльності фізичних та юридичних осіб, яка здійснюється з використанням електронних документів.

Електронний цифровий підпис використовується фізичними та юридичними особами – суб'єктами електронного документообігу для ідентифікації підписувача та підтвердження цілісності даних в електронній формі.

Використання електронного цифрового підпису не змінює порядку підписання договорів та інших документів, встановленого законом для вчинення правочинів у письмовій формі.

Нотаріальні дії із засвідчення справжності електронного цифрового підпису на електронних документах вчиняються відповідно до порядку, встановленого законом.

Стаття 5. Особливості застосування електронного цифрового підпису

Органи державної влади, органи місцевого самоврядування, підприємства, установи та організації державної форми власності для засвідчення чинності відкритого ключа використовують лише посилений сертифікат ключа.

Інші юридичні та фізичні особи можуть на договірних засадах засвідчувати чинність відкритого ключа сертифікатом ключа, сформованим центром сертифікації ключів, а також використовувати електронний цифровий підпис без сертифіката ключа.

Розподіл ризиків збитків, що можуть бути заподіяні підписувачам, користувачам та третім особам, які користуються електронними цифровими підписами без сертифіката ключа, визначається суб'єктами правових відносин у сфері послуг електронного цифрового підпису на договірних засадах.

Захист прав споживачів послуг електронного цифрового підпису, а також механізм реалізації захисту цих прав регулюються цим Законом та Законом

України “Про захист прав споживачів”.

У випадках, коли відповідно до законодавства необхідне засвідчення дійсності підпису на документах та відповідності копій документів оригіналам печаткою, на електронний документ накладається ще один електронний цифровий підпис юридичної особи, спеціально призначений для таких цілей.

Порядок застосування електронного цифрового підпису органами державної влади, органами місцевого самоврядування, підприємствами, установами та організаціями державної форми власності визначається Кабінетом Міністрів України.

Порядок застосування цифрового підпису в банківській діяльності визначається Національним банком України.

Стаття 6. Вимоги до сертифіката ключа

Сертифікат ключа містить такі обов’язкові дані:

найменування та реквізити центру сертифікації ключів (центрального засвідчувального органу, засвідчувального центру);

зазначення, що сертифікат виданий в Україні;

унікальний реєстраційний номер сертифіката ключа;

основні дані (реквізити) підписувача – власника особистого ключа;

дату і час початку та закінчення строку чинності сертифіката;

відкритий ключ;

найменування криптографічного алгоритму, що використовується власником особистого ключа;

інформацію про обмеження використання підпису.

Посилений сертифікат ключа, крім обов’язкових даних, які містяться в сертифікаті ключа, повинен мати ознаку посиленого сертифіката ключа.

Інші дані можуть вноситися у посилений сертифікат ключа на вимогу його власника.

Стаття 7. Права та обов’язки підписувача

Підписувач має право:

вимагати скасування, блокування або поновлення свого сертифіката ключа;

оскаржити дії чи бездіяльність центру сертифікації ключів у судовому порядку.

Підписувач зобов’язаний:

зберігати особистий ключ у таємниці;

надавати центру сертифікації ключів дані згідно з вимогами статті 6 цього Закону для засвідчення чинності відкритого ключа;

своєчасно надавати центру сертифікації ключів інформацію про зміну даних, відображених у сертифікаті ключа.

Стаття 8. Центр сертифікації ключів

Центром сертифікації ключів може бути юридична особа незалежно від форми власності або фізична особа, яка є суб'єктом підприємницької діяльності, що надає послуги електронного цифрового підпису та засвідчила свій відкритий ключ у центральному засвідчувальному органі або засвідчувальному центрі з дотриманням вимог статті 6 цього Закону.

Обслуговування фізичних та юридичних осіб здійснюється центром сертифікації ключів на договірних засадах.

Центр сертифікації ключів має право:

надавати послуги електронного цифрового підпису та обслуговувати сертифікати ключів;

отримувати та перевіряти інформацію, необхідну для реєстрації підписувача і формування сертифіката ключа безпосередньо у юридичної або фізичної особи чи у її уповноваженого представника.

Центр сертифікації ключів зобов'язаний:

забезпечувати захист інформації в автоматизованих системах відповідно до законодавства;

забезпечувати захист персональних даних, отриманих від підписувача, згідно з законодавством;

встановлювати під час формування сертифіката ключа належність відкритого ключа та відповідного особистого ключа підписувачу;

своєчасно скасовувати, блокувати та поновлювати сертифікати ключів у випадках, передбачених цим Законом;

своєчасно попереджувати підписувача та додавати в сертифікат відкритого ключа підписувача інформацію про обмеження використання електронного цифрового підпису, які встановлюються для забезпечення можливості відшкодування збитків сторін у разі заподіяння шкоди з боку центру сертифікації ключів;

перевіряти законність звернень про скасування, блокування та поновлення сертифікатів ключів та зберігати документи, на підставі яких були скасовані, блоковані та поновлені сертифікати ключів;

цілодобово приймати заяви про скасування, блокування та поновлення сертифікатів ключів;

вести електронний перелік чинних, скасованих і блокованих сертифікатів ключів;

забезпечувати цілодобово доступ користувачів до сертифікатів ключів та відповідних електронних переліків сертифікатів через загальнодоступні телекомунікаційні канали;

забезпечувати зберігання сформованих сертифікатів ключів протягом строку, передбаченого законодавством для зберігання відповідних документів

на папері;

надавати консультації з питань, пов'язаних з електронним цифровим підписом.

Зберігання особистих ключів підписувачів та ознайомлення з ними в центрі сертифікації ключів забороняються.

Стаття 9. Акредитований центр сертифікації ключів

Центр сертифікації ключів, акредитований в установленому порядку, є акредитованим центром сертифікації ключів.

Акредитований центр сертифікації ключів має право:

надавати послуги електронного цифрового підпису та обслуговувати виключно посилені сертифікати ключів;

отримувати та перевіряти інформацію, необхідну для реєстрації підписувача і формування посиленого сертифіката ключа, безпосередньо у юридичної або фізичної особи чи її представника.

Акредитований центр сертифікації ключів має виконувати усі зобов'язання та вимоги, встановлені законодавством для центру сертифікації ключів, та додатково зобов'язаний використовувати для надання послуг електронного цифрового підпису надійні засоби електронного цифрового підпису.

Порядок акредитації та вимоги, яким повинен відповідати акредитований центр сертифікації ключів, встановлюються Кабінетом Міністрів України.

Стаття 10. Засвідчувальний центр

Кабінет Міністрів України за необхідності визначає засвідчувальний центр центрального органу виконавчої влади для забезпечення реєстрації, засвідчення чинності відкритих ключів та акредитації групи центрів сертифікації ключів, які надають послуги електронного цифрового підпису цьому органу і підпорядкованим йому підприємствам, установам та організаціям.

Інші державні органи за необхідності, за погодженням з Кабінетом Міністрів України, визначають свої засвідчувальні центри, призначені для виконання функцій, зазначених у частині першій цієї статті.

Засвідчувальний центр по відношенню до групи центрів сертифікації ключів, зазначених у частині першій цієї статті, має ті ж функції і повноваження, що й центральний засвідчувальний орган стосовно центрів сертифікації ключів.

Засвідчувальний центр відповідає вимогам, встановленим законодавством для акредитованого центру сертифікації ключів.

Засвідчувальний центр реєструється, засвідчує свій відкритий ключ і акредитується у центральному засвідчувальному органі.

Положення про засвідчувальний центр центрального органу виконавчої влади затверджується Кабінетом Міністрів України.

Стаття 11. Центральний засвідчувальний орган

Центральний засвідчувальний орган визначається Кабінетом Міністрів України.

Центральний засвідчувальний орган:

формує і видає посилені сертифікати ключів засвідчувальним центрам та центрам сертифікації ключів з дотриманням вимог статті 6 цього Закону;

блокує, скасовує та поновлює посилені сертифікати ключів засвідчувальних центрів та центрів сертифікації ключів у випадках, передбачених цим Законом;

веде електронні реєстри чинних, блокованих та скасованих посилених сертифікатів ключів засвідчувальних центрів та центрів сертифікації ключів;

веде акредитацію центрів сертифікації ключів, отримує та перевіряє інформацію, необхідну для їх акредитації;

забезпечує цілодобово доступ засвідчувальних центрів та центрів сертифікації ключів до посилених сертифікатів ключів та відповідних електронних реєстрів через загальнодоступні телекомунікаційні канали;

зберігає посилені сертифікати ключів засвідчувальних центрів та центрів сертифікації ключів;

надає засвідчувальним центрам та центрам сертифікації ключів консультації з питань, пов’язаних з використанням електронного цифрового підпису.

Центральний засвідчувальний орган відповідає вимогам, встановленим законодавством для акредитованого центру сертифікації ключів.

Положення про центральний засвідчувальний орган затверджується Кабінетом Міністрів України.

Стаття 12. Контролюючий орган

Функції контролюючого органу здійснює спеціально уповноважений центральний орган виконавчої влади у сфері криптографічного захисту інформації.

Контролюючий орган перевіряє дотримання вимог цього Закону центральним засвідчувальним органом, засвідчувальними центрами та центрами сертифікації ключів.

У разі невиконання або неналежного виконання обов’язків та виявлення порушень вимог, встановлених законодавством для центру сертифікації ключів, засвідчувального центру, контролюючий орган дає розпорядження центральному засвідчувальному органу про негайне вжиття заходів, передбачених законом.

Стаття 13. Скасування, блокування та поновлення посиленого сертифіката ключа

Акредитований центр сертифікації ключів негайно скасовує сформований ним посилений сертифікат ключа у разі:

закінчення строку чинності сертифіката ключа;

подання заяви власника ключа або його уповноваженого представника;

припинення діяльності юридичної особи – власника ключа;

смерті фізичної особи – власника ключа або оголошення його померлим за рішенням суду;

визнання власника ключа недієздатним за рішенням суду;

надання власником ключа недостовірних даних;

компрометації особистого ключа.

Центральний засвідчувальний орган негайно скасовує посилений сертифікат ключа центру сертифікації ключів, засвідчувального центру у разі:

припинення діяльності з надання послуг електронного цифрового підпису;

компрометації особистого ключа.

Центральний засвідчувальний орган, засвідчувальний центр, акредитований центр сертифікації ключів негайно блокують посилений сертифікат ключа:

у разі подання заяви власника ключа або його уповноваженого представника;

за рішенням суду, що набрало законної сили;

у разі компрометації особистого ключа.

Скасування і блокування посиленого сертифіката ключа набирає чинності з моменту внесення до реєстру чинних, скасованих і блокованих посилених сертифікатів із зазначенням дати та часу здійснення цієї операції.

Центральний засвідчувальний орган, засвідчувальний центр, акредитований центр сертифікації ключів негайно повідомляють про скасування або блокування посиленого сертифіката ключа його власника.

Блокований посилений сертифікат ключа поновлюється:

у разі подання заяви власника ключа або його уповноваженого представника;

за рішенням суду, що набрало законної сили;

у разі встановлення недостовірності даних про компрометацію особистого ключа.

Стаття 14. Припинення діяльності центру сертифікації ключів

Центр сертифікації ключів припиняє свою діяльність відповідно до законодавства.

Про рішення щодо припинення діяльності центр сертифікації ключів повідомляє підписувачів за три місяці, якщо інші строки не визначено законодавством. Підписувачі мають право обирати за власним бажанням будь-який центр сертифікації ключів для подальшого обслуговування, якщо інше не передбачено законодавством. Після повідомлення про припинення діяльності, центр сертифікації ключів не має права видавати нові сертифікати ключів. Усі сертифікати ключів, що були видані центром сертифікації ключів, після припинення його діяльності скасовуються.

Центр сертифікації ключів, що повідомив про припинення своєї діяльності, зобов'язаний забезпечити захист прав споживачів шляхом повернення грошей за послуги, що не можуть надаватися в подальшому, якщо вони були попередньо оплачені.

Акредитований центр сертифікації ключів додатково повідомляє про рішення щодо припинення діяльності центральний засвідчувальний орган або відповідний засвідчувальний центр.

Акредитований центр сертифікації ключів протягом доби, визначеної як дата припинення його діяльності, передає посилені сертифікати ключів, відповідні реєстри посилених сертифікатів ключів та документовану інформацію, яка підлягає обов'язковій передачі, відповідному засвідчувальному центру або центральному засвідчувальному органу.

Порядок передачі акредитованим центром сертифікації ключів посилених сертифікатів ключів, відповідних реєстрів посилених сертифікатів ключів та документованої інформації, яка підлягає обов'язковій передачі, встановлюється Кабінетом Міністрів України.

Стаття 15. Відповідальність за порушення законодавства про електронний цифровий підпис

Особи, винні у порушенні законодавства про електронний цифровий підпис, несуть відповідальність згідно з законом.

Стаття 16. Розв'язання суперечок

Суперечки, які виникають у сфері надання послуг електронного цифрового підпису, розв'язуються в порядку, встановленому законом.

Стаття 17. Визнання іноземних сертифікатів ключів

Іноземні сертифікати ключів, засвідчені відповідно до законодавства тих держав, де вони видані, визнаються в Україні чинними у порядку, встановленому законом.

Стаття 18. Прикінцеві положення

1) Цей Закон набирає чинності з 1 січня 2004 року.

2) До приведення законів України та інших нормативно-правових актів у відповідність із цим Законом вони застосовуються у частині, що не суперечить цьому Закону.

3) Пункт 14 статті 9 Закону України “Про ліцензування певних видів господарської діяльності” (Відомості Верховної Ради України, 2000 р., № 36, ст. 299) після слів “надання послуг в галузі криптографічного захисту інформації” доповнити словами “(крім послуг електронного цифрового підпису)”.

4) Кабінету Міністрів України протягом шести місяців з дня набрання чинності цим Законом:

підготувати та внести до Верховної Ради України пропозиції про внесення змін до законів України, що випливають із цього Закону;

забезпечити приведення своїх нормативно-правових актів, а також нормативно-правових актів міністерств та інших центральних органів виконавчої влади у відповідність з цим Законом;

визначити центральний засвідчувальний орган;

забезпечити прийняття нормативно-правових актів, передбачених цим Законом.

5) Національному банку України протягом шести місяців з дня набрання чинності цим Законом привести свої нормативно-правові акти у відповідність з цим Законом.

6) Кабінету Міністрів України разом з Національним банком України, іншими органами державної влади протягом шести місяців з дня набрання чинності цим Законом розробити та внести на розгляд Верховної Ради України програму заходів щодо впровадження електронного документа, електронного документообігу та електронного цифрового підпису.

м. Київ
22 травня 2003 року
№ 852–IV

Г. Математичне обґрунтування атак, у підґрунті яких лежить парадокс щодо днів народження

Парадокс завдання про дні народження часто використовують в елементарних курсах з теорії ймовірностей, щоб продемонструвати, що результати теорії ймовірностей іноді суперечать інтуїтивним уявленням. Проблема може бути сформульована так: чому одне мінімальне значення k, за якого ймовірність того, що принаймні у двох із групи k людей дні народження збігаються, виявляється рівною 0,5? Зігноруємо 29 лютого і припустімо, що кожен день народження є однаково ймовірний. Щоб відповісти на поставлене запитання, визначимо $P(n, k)$ (має місце принаймні один збіг з-посеред k елементів, де кожний елемент набуває одне з n однаково ймовірних значень від 1 до n).

Отже, потрібно знайти найменше значення k, за якого $P(365, k) \geq 0{,}5$. Спочатку визначимо ймовірність того, що збігів не станеться. Позначимо її як $Q(365, k)$. Для $k > 365$ є неможливо, щоб всі значення були різними. Тому можна припустити, що $k \leq 365$. Розглянемо кількість різних способів N, що дозволяють здобути k значень без повторювань. Для першого елемента ми маємо на вибір будь-яке з 365-ти значень, для другого елемента – будь-яке з 364-х, що залишилися, і т. д. Тому для кількості відповідних способів одержуємо:

$$N = 365 \cdot 364 \ldots (365 - k + 1) = \frac{365!}{(365-k)!}.$$

Якщо відкинути умову відсутності збігів, то кожен елемент може набувати будь-яке з 365-ти можливих значень, що в сумі дає 365^k варіантів. Тому ймовірність відсутності збігів дорівнює відношенню кількості варіантів без збігів до сумарної кількості варіантів:

$$Q(365, k) = \frac{365!/(365-k)!}{365^k} = \frac{365!}{365^k (365-k)!};$$

$$P(365, k) = 1 - Q(365, k) = 1 - \frac{365!}{365^k (365-k)!}.$$

Цю функцію подано на рис. Г.1.

Ймовірності можуть видатися надто великими, якщо ви не стикалися з подібною проблемою раніш. Багато хто думає, що для здобуття ймовірності хоча б одного збігу, що перевищує 0,5, у групі має бути близько 100 осіб, а насправді вистачить усього 23-х, оскільки $P(365, 23) = 0{,}5073$. Для $k = 100$ ймовірність принаймні б одного збігу дорівнює 0,9999997.

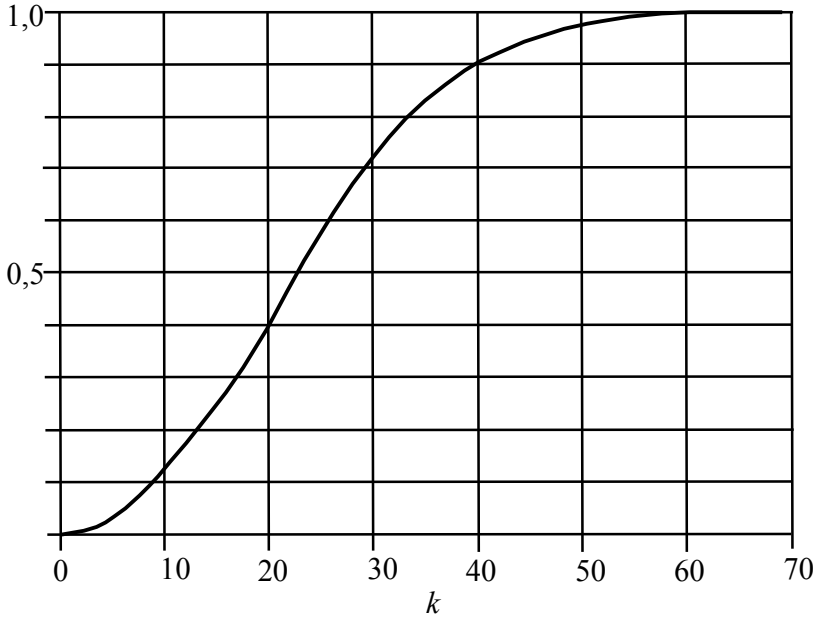


Рисунок Г.1 – Парадокс завдання щодо днів народження

Поданий результат видається вражаючим, можливо, тому, що для кожної окремої людини в групі ймовірність того, що з її днем народження збігатиметься день народження когось іншого в групі, є дуже незначна. Але ми розглядаємо ймовірність того, що для якоїсь пари людей у групі дні народження збігатимуться. У групі з 23-х осіб є $\frac{23(23-1)}{2} = 253$ різні пари, тому й згадані імовірності є такі високі.

Одна корисна нерівність. Перед тим як приступити до розгляду узагальнення парадоксу завдання щодо днів народження, доведемо одну корисну нерівність:

$$(1 - x) \le \mathrm{e}^{-x} \qquad \text{для усіх } x \ge 0.$$

Ця нерівність ілюструється графіком на рис. Г.2.

Щоб переконатися в істинності поданої нерівності, слід звернути увагу на те, що нижня лінія є пряма, дотична до графіка функції e^{-x} в точці $x = 0$. Нахил цієї прямої в точності дорівнює похідній e^{-x} в точці $x = 0$:

$$f(x) = \mathrm{e}^{-x};$$
$$f'(x) = \frac{d}{dx}\mathrm{e}^{-x} = -\mathrm{e}^{-x};$$
$$f'(0) = -1.$$

Дотичною до e^{-x} в точці $x = 0$ є пряма вигляду $ax + b$, для якої $a = -1$ і яка набуває значення $\mathrm{e}^0 = 1$ в точці $x = 0$. Такою функцією є $(1 - x)$. Зауважимо також, що для малих x є $(1 - x) \approx \mathrm{e}^{-x}$.

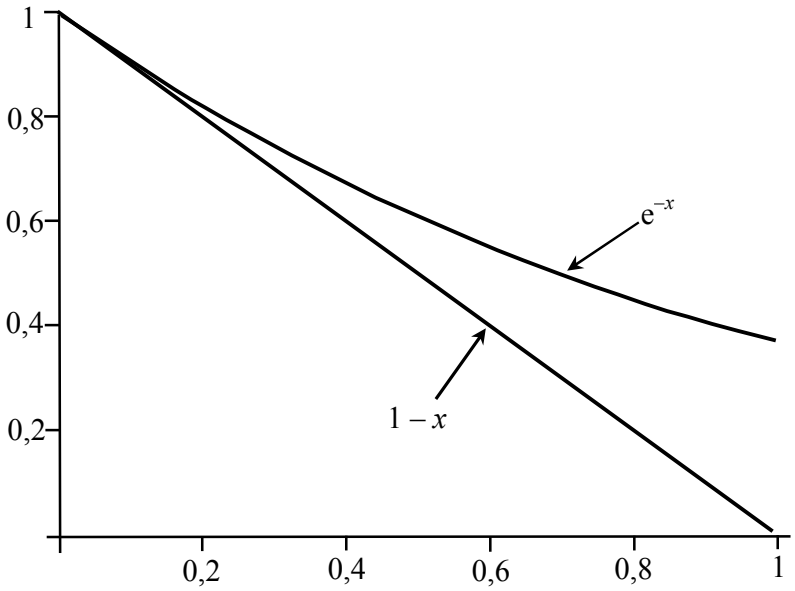


Рисунок Г.2 – Нерівність $(1 - x) \le e^{-x}$

Загальний випадок. Завдання щодо днів народження можна узагальнити: якщо певна цілочисельна випадкова величина з рівномірним розподілом значень від 1 до n і є вибірка, що складається з k значень цієї випадкової величини ($k \le n$), то яка буде ймовірність $P(n, k)$ того, що з-посеред значень у вибірці принаймні два збігаються? Завдання, до якого зводиться парадокс щодо днів народження, є окремим випадком сформульованої перед тим проблеми (за $n = 365$). Використовуючи аргументацію, подану вище, здобуваємо узагальнення рівності

$$P(n, k) = 1 - \frac{n!}{(n-k)!\,n^k}.$$

Можна подати його у вигляді

$$P(n, k) = 1 - \frac{n(n-1)\ldots(n-k+1)}{n^k} = 1 - \left[\frac{n-1}{n}\frac{n-2}{n}\cdots\frac{n-k+1}{n}\right] =$$

$$= 1 - \left[\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{k-1}{n}\right)\right];$$

$$P(n, k) > 1 - \left[\left(e^{-1/n}\right)\left(e^{-2/n}\right)\ldots\left(e^{-(k-1)/n}\right)\right] > 1 - e^{-[(1/n)+(2/n)+\ldots+((k-1)/n)]} > 1 - e^{-(k(k-1))/2n}.$$

Тепер запитаємо: за якого значення k можна здобути нерівність $P(n, k) > 0{,}5$? Для цього потрібно, щоб

$$1/2 = 1 - e^{-(k(k-1))/2n};$$

$$2 = e^{(k(k-1))/2n};$$

$$\ln(2) = \frac{k(k-1)}{2n}.$$

Для великих k значень k (k – 1) можна замінити на k^2 і тоді одержимо

$$k = \sqrt{2\ln(2)n} = 1{,}18\sqrt{n} \approx \sqrt{n}\,.$$

Для перевірки припустімо, що n = 365, тоді $k = 1{,}18\ \sqrt{365} = 22{,}54$, що є дуже наближено до правильного значення 23.

Тепер можна сформулювати суть атаки, в основі якої лежить парадокс завдання щодо днів народження. Припустімо, що є функція H, яка припускає 2^m варіантів виведення (тобто m-бітове виведення). Якщо на вхід H подати k варіантів випадкового введення, то яким має бути значення k, щоб можна було сподіватися на виході хоча б одного збігу [тобто H (y) = H (x) для певних введених (x, y)]? Дістаємо:

$$k = \sqrt{2\ln(2)2^m} = 1{,}18\ \sqrt{2^m} \approx \sqrt{2^m}\,.$$

Д. Обґрунтування алгоритму цифрового підпису

Метою цього додатку є доведення того, що в процесі верифікування підпису буде здобуто рівність $v = r$, якщо підпис є справжній.

Лема 1 Для будь-якого цілого числа t:

$$\text{якщо } g \equiv h^{(p-1)/q} \pmod p, \text{ то } g^t \pmod p \equiv g^{t \bmod q} \pmod p.$$

Доведення. За теоремою Ферма, оскільки h є взаємно простим по відношенню до p, маємо $h^{p-1} \pmod p \equiv 1$. Отже, для кожного невід'ємного цілого числа n

$$g^{nq} \pmod p \equiv h^{(p-1)/q} \pmod p \equiv h^{((p-1)/q)nq} \pmod p \equiv h^{(p-1)n} \pmod p \equiv$$
$$\equiv (h^{(p-1)} \bmod p)^n \bmod p.$$

Тому для невід'ємних цілих чисел n та z одержуємо

$$(g^{nq+z}) \bmod p \equiv (g^{nq} g^z) \bmod p \equiv [(g^{nq} \bmod p)\,(g^z \bmod p)] \bmod p \equiv g^z \pmod p.$$

Кожне ціле невід'ємне число t може бути подано у єдиний спосіб у формі $t = nq + z$, де n та z є цілими невід'ємними числами і $0 < z < q$. Тому $z \equiv t \pmod q$, звідки й випливає потрібний результат.

Лема 2 $(y^{(rw) \bmod q}) \bmod p \equiv (g^{(krw) \bmod q}) \bmod p$.

Доведення. За означенням, $y \equiv g^k \pmod p$. Тоді

$$(y^{(rw) \bmod q}) \bmod p \equiv (g^k \bmod p)^{(rw) \bmod q} \bmod p \equiv (g^{k((rw) \bmod q)}) \bmod p \equiv$$
$$\equiv (g^{(k((rw) \bmod q)) \bmod q}) \bmod p \equiv (g^{(krw) \bmod q}) \bmod p.$$

Лема 3 $((H(M) + kr)w) \bmod q \equiv x$.

Доведення. За означенням, $s \equiv [x^{-1}(H(M) + kr)] \bmod q$. Також, з огляду на те що q є простим, кожне невід'ємне ціле число менше за q має мультиплікативно обернене. Тому $(xx^{-1}) \bmod q \equiv 1$. Далі маємо

$$(xs) \bmod q \equiv (x(x^{-1}(H(M) + kr)) \bmod q)) \bmod q \equiv$$
$$\equiv (x(x^{-1}(H(M) + kr))) \bmod q \equiv (((xx^{-1}) \bmod q)((H(M) + kr) \bmod q)) \bmod q \equiv$$
$$\equiv (H(M) + kr) \bmod q.$$

За визначенням, $w \equiv s^{-1} \bmod q$, тому $(ws) \bmod q \equiv 1$. Отже,

$$((H(M) + kr)w) \bmod q \equiv (((H(M) + kr)w) \bmod q)(w \bmod q)) \bmod q \equiv$$
$$\equiv [((xs) \bmod q)(w \bmod q)] \bmod q \equiv (wxs) \bmod q \equiv$$
$$\equiv [(x \bmod q)((ws) \bmod q)] \bmod q \equiv x \pmod q.$$

Оскільки $0 < x < q$, одержуємо $x \bmod q \equiv x$. Що й треба було довести.

Теорема У позначеннях табл. 4.1 має місце рівність $v = r$.

Доведення.

$$v \equiv ((g^{u_1} Y^{u_2}) \bmod p) \bmod q \equiv ((g^{(H(M)w) \bmod q} y^{(rw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w) \bmod q} g^{krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w) \bmod q + (krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w + krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M) + kr)w) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv (g^x \bmod p) \bmod q \equiv r.$$

Що й треба було довести.

Е. Приклади обчислювання цифрового підпису згідно з ДСТУ 4145–2002

Нижче подаються витяги з національного стандарту України „Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих”. Стандарт встановлює механізм створення цифрового підпису, який ґрунтується на властивостях груп точок еліптичних кривих над полями $GF(2^m)$.

1 Обчислення загальних параметрів цифрового підпису

Загальні параметри цифрового підпису можуть бути однаковими для довільної кількості користувачів цифрового підпису. Цей розділ встановлює правила обчислення загальних параметрів цифрового підпису.

1.1 Вибір основного поля

Якщо використовують оптимальний нормальний базис, то основне поле треба обирати з-посеред полів $GF(2^m)$, степені яких подано в табл. Е.1.

Таблиця Е.1 – Припустимі основні поля з оптимальним нормальним базисом

Степінь поля m	173	179	191	233	239	251	281
Степінь поля m	293	359	419	431	443	491	509

Для реалізації цього стандарту можна використовувати еліптичні криві, подані в табл. Е.2.

Таблиця Е.2 – Рекомендовані еліптичні криві в поліноміальному базисі

№ пп.	m	Еліптична крива
1	163	A = 1 B = 5FF6108462A2DC8210AB403925E638A19C1455D21 n = 400000000000000000002BEC12BE2262D39BCF14D
2	167	A = 1 B = 6EE3CEEB230811759F20518A0930F1A4315A827DAC n = 3FFFFFFFFFFFFFFFFFFFFB12EBCC7D7F29FF7701F
3	173	A = 0 B = 108576C80499DB2FC16EDDF6853BBB278F6B6FB437D9 n = 800000000000000000000189B4E67606E3825BB2831
4	179	A = 1 B = 4A6E0856526436F2F88DD07A341E32D04184572BFB710 n = 3FFFFFFFFFFFFFFFFFFFFB981960435FE5AB64236EF
5	191	A = 1 B = 7BC86E2102902EC4D5890E8B6B4981FF27E0482750FEFC03 n = 4000000000000000000000069A779CAC1DABC6788F7474F
6	233	A = 1 B = 06973B15095675534C7CF7E64A21BD54EF5DD3B8A0326AA936ECE454D2C n = 1000000000000000000000000000013E974E72F8A6922031D2603CFE0D7

Дозволено задавати основне поле поліноміальним або оптимальним нормальним базисом. Якщо використовують поліноміальний базис, то основне поле треба обирати з-посеред полів $GF(2^m)$, степені яких подано в табл. Е.3. Поліноміальний базис задають примітивними тричленами чи примітивними п'ятичленами. Використовування примітивних многочленів, поданих у табл. Е.3, не є обов'язковим.

Таблиця Е.3 – Припустимі основні поля з поліноміальним базисом і рекомендовані примітивні многочлени

№ пп.	Степінь поля m	Примітивний многочлен	№ пп.	Степінь поля m	Примітивний многочлен
1	163	$x^{163}+x^7+x^6+x^3+1$	31	337	$x^{337}+x^{10}+x^6+x+1$
2	167	$x^{167}+x^6+1$	32	347	$x^{347}+x^{17}+x^6+x+1$
3	173	$x^{173}+x^{10}+x^2+x+1$	33	349	$x^{349}+x^6+x^5+x^2+1$
4	179	$x^{179}+x^4+x^2+x+1$	34	353	$x^{353}+x^{26}+x^7+x^3+1$
5	181	$x^{181}+x^7+x^6+x+1$	35	359	$x^{359}+x^{18}+x^4+x^2+1$
6	191	$x^{191}+x^9+1$	36	367	$x^{367}+x^{21}+1$
7	193	$x^{193}+x^{15}+1$	37	373	$x^{373}+x^9+x^6+x+1$
8	197	$x^{197}+x^{21}+x^2+x+1$	38	379	$x^{379}+x^{17}+x^6+x+1$
9	199	$x^{199}+x^{11}+x^2+x+1$	39	383	$x^{383}+x^9+x^5+x+1$
10	211	$x^{211}+x^{12}+x^6+x+1$	40	389	$x^{389}+x^{17}+x^{10}+x+1$
11	223	$x^{223}+x^{12}+x^2+x+1$	41	397	$x^{397}+x^{22}+x^3+x+1$
12	227	$x^{227}+x^{21}+x^2+x+1$	42	401	$x^{401}+x^{29}+x^4+x+1$
13	229	$x^{229}+x^{21}+x^2+x+1$	43	409	$x^{409}+x^{15}+x^6+x+1$
14	233	$x^{233}+x^9+x^4+x+1$	44	419	$x^{419}+x^{21}+x^{14}+x+1$
15	239	$x^{239}+x^{15}+x^2+x+1$	45	421	$x^{421}+x^7+x^4+x+1$
16	241	$x^{241}+x^{15}+x^4+x+1$	46	431	$x^{431}+x^6+x^3+x+1$
17	251	$x^{251}+x^{14}+x^4+x+1$	47	433	$x^{433}+x^{15}+x^5+x+1$
18	257	$x^{257}+x^{12}+1$	48	439	$x^{439}+x^8++x^3+x^2+1$
19	263	$x^{263}+x^{27}+x^2+x+1$	49	443	$x^{443}+x^{28}+x^3+x+1$
20	269	$x^{269}+x^7+x^6+x+1$	50	449	$x^{449}+x^{25}+x^5+x^3+1$
21	271	$x^{271}+x^{16}+x^3+x+1$	51	457	$x^{457}+x^{16}+1$
22	277	$x^{277}+x^{23}+x^3+x^2+1$	52	461	$x^{461}+x^{23}+x^4+x+1$
23	281	$x^{281}+x^9+x^4+x+1$	53	463	$x^{463}+x^{24}+x^3+x+1$
24	283	$x^{283}+x^{26}+x^9+x+1$	54	467	$x^{467}+x^{28}+x^3+x+1$
25	293	$x^{293}+x^{11}+x^6+x+1$	55	479	$x^{479}+x^{25}+x^6+x+1$
26	307	$x^{307}+x^8+x^4+x^2+1$	56	487	$x^{487}+x^{15}+x^2+x+1$
27	311	$x^{311}+x^{29}+x^4+x+1$	57	491	$x^{491}+x^{17}+x^6+x^2+1$
28	313	$x^{313}+x^7+x^3+x+1$	58	499	$x^{499}+x^{29}+x^6+x^2+1$
29	317	$x^{317}+x^9+x^5+x^2+1$	59	503	$x^{503}+x^3+1$
30	331	$x^{331}+x^{12}+x^5+x^2+1$	60	509	$x^{509}+x^{23}+x^3+x^2+1$

1.2 Перетворювання даних

1.2.1 Перетворювання елемента основного поля на ціле число

Установимо алгоритм перетворювання елемента основного поля $x \in GF(2^m)$ на ціле число a.

Вхідні дані алгоритму: елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1}, ..., x_0)$, порядок базової точки еліптичної кривої n.

Результат виконання алгоритму – ціле число $a = (a_{L-1}, ..., a_0)$, яке задовольняє умові $L = L(a) < L(n)$.

Алгоритм перетворювання елемента основного поля на ціле число:

– якщо елемент x основного поля дорівнює 0, то $a \leftarrow 0$ $L = L(a) \leftarrow 1$, кінець алгоритму;

– обчислюють ціле число $k = L(n) - 1$;

– приймають $a_i = x_i$ для $i = 0, ..., k - 1$ і знаходять j, що дорівнює найбільшому індексові i, при якому $a_i = 1$. Якщо такого індексу нема, то приймають $a \leftarrow 0$ і завершують виконування алгоритму;

– двійковий рядок $(a_j, ..., a_0)$ довжини $L = L(a) = j + 1$ зображує ціле число a, яке є результат виконання алгоритму.

1.2.2 Перетворювання геш-коду на елемент основного поля

Установимо алгоритм перетворювання результату обчислення функції гешування $(h_{L_H-1}, ..., h_0)$ на елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1}, ..., x_0)$.

Вхідні дані алгоритму: геш-код $(h_{L_H-1}, ..., h_0)$.

Результат виконання алгоритму – елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1}, ..., x_0)$.

Алгоритм перетворювання результату обчислення функції гешування на елемент основного поля:

– обчислюють ціле число $k = \min(m, L_H)$;

– приймають $x_i = h_i$ для $i = 0, ..., k - 1$;

– якщо $k < m$, то приймають $x_i = 0$ для $i = 0, ..., m - 1$;

– двійковий рядок $(x_{m-1}, ..., x_0)$ зображує елемент x основного поля, який є результатом виконання алгоритму.

1.2.3 Перетворювання пари цілих чисел на цифровий підпис

Установимо алгоритм перетворювання пари цілих чисел (r, s), які задовольняють умовам $0 < r < n$, $0 < s < n$, на цифровий підпис $D = (D_{L_D-1}, ..., D_0)$.

Вхідні дані алгоритму: пара цілих чисел (r, s) у двійковому зображенні: $r = (r_{L(r)-1}, ..., r_0)$, $s = (s_{L(s)-1}, ..., s_0)$, $0 < r < n$, $0 < s < n$, довжина цифрового підпису L_D: $L_D \geq 2L(n)$, L_D є кратне до 16.

Результат виконання алгоритму – цифровий підпис $D = (D_{L_D-1}, ..., D_0)$ довжини L_D.

Алгоритм перетворювання пари цілих чисел на цифровий підпис:

– приймають $l = L_D/2$;

– утворюють двійковий рядок R за правилом:

приймають $R_i = r_i$ для $i = 0, ..., L(r) - 1$,

приймають $R_i = 0$ для $i = L(r), ..., l - 1$;

– утворюють двійковий рядок S за правилом:

приймають $S_i = s_i$ для $i = 0, ..., L(s) - 1$,

приймають $S_i = 0$ для $i = L(s), ..., l - 1$;

– рядок D є конкатенація двох рядків $S \| R$;

– двійковий рядок $D = (D_{L_D-1}, ..., D_0)$ довжини L_D є результат виконання алгоритму.

1.2.4 Перетворювання двійкового рядка на пару цілих чисел

Установимо алгоритм перетворювання двійкового рядка D парної довжини L_D на пару цілих чисел $r = (r_{L(r)-1}, ..., r_0)$, $s = (s_{L(s)-1}, ..., s_0)$.

Вхідні дані алгоритму: двійковий рядок $D = (D_{L_D-1}, ..., D_0)$ парної довжини L_D.

Результат виконання алгоритму – пара цілих чисел $r = (r_{L(r)-1}, ..., r_0)$ та $s = (s_{L(s)-1}, ..., s_0)$.

Алгоритм перетворювання двійкового рядка на пару цілих чисел:

1) обчислюють ціле число $l = L_D/2$;
2) приймають $r_i = D_i$ для $i = 0, ..., l - 1$;
3) визначають j як найбільше i, $i = 0, ..., l - 1$, для якого $r_i = 1$;
4) якщо такого індексу нема, то $r \leftarrow 0$, $j = 0$ і переходять до кроку 6;
5) двійковий рядок $(r_{L(r)-1}, ..., r_0)$, $L(r) = j + 1$, зображає ціле число r;
6) приймають $s_i = D_{i+l}$ для $i = 0, ..., l - 1$;
7) визначають індекс j як найбільше i, $i = 0, ..., l - 1$, для якого $s_i = 1$;
8) якщо такого індексу нема, то $s \leftarrow 0$, $j = 0$ і переходять до кроку 10;
9) двійковий рядок $(s_{L(s)-1}, ..., s_0)$, $L(s) = j + 1$, зображає ціле число s;
10) пара цілих чисел r та s є результат виконання алгоритму.

2 Приклади обчислень цифрового підпису

2.1 Обчислення й перевіряння цифрового підпису в поліноміальному базисі

Нижче подамо приклади обчислення й перевіряння цифрового підпису з використанням поліноміального та оптимального нормального базисів. У прикладах обчислень двійкові рядки поданом у вигляді рядків шістнадцяткових цифр: двійковий рядок у разі потреби доповнюють ліворуч у такий спосіб, щоб довжина рядка стала кратною до чотирьох, потім рядок поділяють на групи по чотири двійкових розряди, а кожну таку групу замінюють на шістнадцяткову цифру, що відповідає цій групі двійкових символів.

Вибір загальних параметрів. За основне поле використовують скінченне поле $GF(2^{163})$. Елементи цього поля зображують у поліноміальному базисі, що відповідає примітивному многочленові $x^{163} + x^7 + x^6 + x^3 + 1$ (див. табл. Е.3).

Використовується еліптична крива над полем $GF(2^{163})$:

$$y^2 + xy = x^3 + x^2 + 5FF6108462A2DC8210AB403925E638A19C1455D21.$$

Порядок цієї еліптичної кривої ділиться на просте число

$$n = 40000000000000000002BEC12BE2262D39BCF14D,$$

яке є порядком базової точки.

Обчислення базової точки еліптичної кривої здійснюють у такий спосіб. Обчислюємо випадкову точку еліптичної кривої

$$P = (x_P, y_P) = (72D867F93A93AC27DF9FF01AFFE74885C8C540420,$$
$$0224A9C3947852B97C5599D5F4AB81122ADC3FD9B).$$

Оскільки $nP = O$, то точка P – шукана базова точка еліптичної кривої.

За особистий ключ цифрового підпису візьмемо ціле число

$$d = 183F60FDF7951FF47D67193F8D073790C1C9B5A3E.$$

Обчислимо відкритий ключ цифрового підпису, що відповідає обраному особистому ключеві:

$$Q = -dP = (x_Q, y_Q) = (057DE7FDE023FF929CB6AC785CE4B79CF64ABDC2DA,$$
$$3E85444324BCF06AD85ABF6AD7B5F34770532B9AA).$$

Нехай використовується довжина цифрового підпису $L_D = 512$.

Припустімо, що функцію гешування обрано згідно з ГОСТ 34.311–95 [45] ($iH = 1$) і її використовують без додаткових вказівок. У цьому разі $L_H = 256$. Без додаткових вказівок приймемо також, що iH не передається.

Обчислення цифрового підпису. Обчислимо геш-функцію за повідомленням T. Нехай результат гешування буде

$$H(T) = 09C9C44277910C9AAEE486883A2EB95B7180166DDF73532EEB76EDAEF52247FF.$$

Перетворимо результат обчислення функції гешування $H(T)$ на елемент основного поля згідно з 1.2.2. Перетворення цього рядка на елемент основного поля полягає у відокремленні з цього рядка $\min(m, L_H) = 163$ молодших розрядів. Унаслідок перетворення знайдемо елемент основного поля

$$h = 03A2EB95B7180166DDF73532EEB76EDAEF52247FF.$$

Нехай ціле число

$$e = 1025E40BD97DB012B7A1D79DE8E12932D247F61C6.$$

Обчислимо точку eP:

$$eP = (x_{eP}, y_{eP}) = (42A7D756D70E1C9BA62D2CB43707C35204EF3C67C,$$
$$5310AE5E560464A95DC80286F17EB762EC544B15B).$$

Тоді F_e дорівнює координаті x_{eP} цієї точки:

F_e = 42A7D756D70E1C9BA62D2CB43707C35204EF3C67C.

Обчислимо добуток елементів основного поля

$y = hF_e$ = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Перетворимо елемент основного поля y на ціле число r згідно з 1.2.1:

r = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Обчислимо ціле число

$s \equiv (e + dr) \bmod n \equiv$ 2100D86957331832B8E8C230F5BD6A332B3615ACA.

Перетворимо пару цілих чисел (r, s) на цифровий підпис D згідно з 1.2.3:

D = 000000000000000000000000002100D86957331832B8E8C230F5BD6A332B3615ACA
000000000000000000000000274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Підписане повідомлення довжини $L = L_T + L_D$ має форму $T \| D$.

Перевірення цифрового підпису. Перевіримо цифровий підпис, обчислений вище. При перевіренні цифрового підпису використовують ті самі загальні параметри: обчислений вище відкритий ключ та геш-функцію без додаткових вказівок (iH = 1, L_H = 256, iH не передається).

Перевіряємо цифровий підпис:

D = 000000000000000000000000002100D86957331832B8E8C230F5BD6A332B3615ACA
000000000000000000000000274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Перевіримо довжину цифрового підпису: L_D = 512, тобто чи є це число кратне до 16 і більше за подвоєну довжину двійкового зображення порядку базової точки n.

Обчислюємо $L_T = L - L_D$. Вважаємо, що підписаний текст прийнято без спотворень, тому $L_T > 0$.

Обчислюємо $H(T)$. Підписаний текст прийнято без спотворень, тому результат обчислення функції гешування є, як і при обчисленні цифрового підпису,

$H(T)$ = 09C9C44277910C9AAEE486883A2EB95B7180166DDF73532EEB76EDAEF52247FF.

Перетворюємо результат обчислення функції гешування на елемент основного поля згідно з 1.2.2:

h = 03A2EB95B7180166DDF73532EEB76EDAEF52247FF.

Перетворюємо цифровий підпис на пару цілих чисел (r, s) згідно з 1.2.4:

r = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7;
s = 2100D86957331832B8E8C230F5BD6A332B3615ACA.

Переконуємося, що $0 < r < n$ та $0 < s < n$.

Обчислюємо точку еліптичної кривої

$R = sP + rQ = (x_R, y_R)$ = (42A7D756D70E1C9BA62D2CB43707C35204EF3C67C,
5310AE5E560464A95DC80286F17EB762EC544B15B).

Обчислюємо елемент основного поля

$y = hx_R$ = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Перетворюємо елемент основного поля y на ціле число $\tilde{r}$ згідно з 1.2.1:

$\tilde{r}$ = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Оскільки $r = \tilde{r}$, то підпис є справжній.

2.2 Обчислення й перевірення цифрового підпису в оптимальному нормальному базисі

Загальні параметри цифрового підпису. За основне поле візьмемо поле $GF(2^{173})$ в оптимальному нормальному базисі табл. Е.1. Нехай еліптичну криву визначено рівнянням

$y^2 + xy = x^3$ + 043D7E139319F43BA00944915740E1E6651B06E278C7.

Порядок базової точки

n = 800000000000000000000189B4E67606E3825BB2831.

За базову точку візьмемо

$P = (x_P, y_P)$ = (1D26E5282C2647E1B0327CB4F6087932A0A1F006A89F,
0322218E4551CF8F6E3345644A10B2408F5DF19EC2C7).

Особистий та відкритий ключі цифрового підпису оберемо в такий спосіб:
особистий ключ

d = 49FF09C848613AEA23699F78C960D5174617311ADCC;

відкритий ключ

$Q = –dP = (x_Q, y_Q)$ = (16C1FF796B32D6D2FC83CA9530B368B322F7874B2D01,
1F253334197826B6ADAE44EDEAA7EA281D1C5E597A37).

Нехай функцію гешування обчислено без додаткових вказівок згідно з ГОСТ 34.311–95 (iH = 1, L_H = 256) і при цьому прийнято, що iH передається. Результат обчислення є

$H(T)$ = 2A681ECE118389B27A108137187EA862117EF1484289470ECAC802C5A651FDA8.

Припустімо, що довжина цифрового підпису дорівнює найменшому припустимому значенню L_D = 352.

Обчислення цифрового підпису. Обчислимо $H(T)$ і перетворимо результат обчислення на елемент основного поля h згідно з 1.2.2. Внаслідок перетворення одержимо елемент основного поля

h = 0137187EA862117EF1484289470ECAC802C5A651FDA8.

Нехай ціле число

e = 70516411E5D9886B8486ECE54A30E9403D103B95F90.

Обчислимо точку eP:

$eP = (x_{eP}, y_{eP})$ = (028886EA28A7C2951FA6473EB3EBC861D3EDB1FBB031,
19059E90F7C7725079CFFE312A389B265140F5BDA493).

Тоді F_e дорівнює координаті x_{eP} цієї точки:

F_e = 028886EA28A7C2951FA6473EB3EBC861D3EDB1FBB031.

Обчислимо добуток елементів основного поля

$y = hF_e$ = 0477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Перетворимо елемент основного поля y на ціле число r згідно з 1.2.1:

r = 477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Обчислимо ціле число

$s \equiv (e + dr) \bmod n \equiv$ 472EA56AE478F95F1EC9F628FF43857E168B50FB819.

Перетворимо пару цілих чисел (r, s) на цифровий підпис D згідно з 1.2.3:

D = 0472EA56AE478F95F1EC9F628FF43857E168B50FB819
0477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Підписане повідомлення довжини $L = 8 + L_T + L_D$ має вигляд $iH \,||\, T \,||\, D$, де iH – двійковий рядок вигляду 00000001.

Перевірення цифрового підпису. Перевіримо цифровий підпис, обчислений вище. При перевіренні цифрового підпису використовують ті самі загальні параметри, обчислений вище відкритий ключ та геш-функцію без додаткових вказівок (iH = 1, L_H = 256, iH передається).

Перевіряється цифровий підпис

D = 0472EA56AE478F95F1EC9F628FF43857E168B50FB819
0477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Без додаткових вказівок, перші 8 бітів повідомлення задають iH. Перевіряємо, чи геш-функція з номером 00000001 є діюча на даний час, і визначаємо відповідну довжину геш-коду L_H = 256.

Перевіряємо довжину цифрового підпису: L_D = 352, тобто чи є це число кратне до 16 і більше за подвоєну довжину двійкового зображення порядку базової точки n.

Обчислюємо $L_T = L - L_D - L(iH)$. Вважаємо, що підписаний текст прийнято без спотворень, тому $L_T > 0$.

Обчислюємо $H(T)$. Підписаний текст прийнято без спотворень, тому результат обчислення функції гешування є, як і при обчисленні цифрового підпису,

$H(T)$ = 2A681ECE118389B27A108137187EA862117EF1484289470ECAC802C5A651FDA8.

Перетворюємо результат обчислення на елемент h основного поля згідно з 1.2.2:

h = 0137187EA862117EF1484289470ECAC802C5A651FDA8.

Перетворюємо цифровий підпис на пару цілих чисел (r, s) згідно з 1.2.4:

r = 477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A;
s = 472EA56AE478F95F1EC9F628FF43857E168B50FB819.

Переконуємося, що $0 < r < n$ та $0 < s < n$.

Обчислюємо точку еліптичної кривої

$R = sP + rQ = (x_R, y_R)$ = (028886EA28A7C2951FA6473EB3EBC861D3EDB1FBB031,
19059E90F7C7725079CFFE312A389B265140F5BDA493).

Обчислюємо елемент основного поля

$y = hx_R$ = 0477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Перетворюємо елемент основного поля y на ціле число $\tilde{r}$ згідно з 1.2.1:

$\tilde{r}$ = 477ECC260F390FB6D0AE4AE3B7A78120F8EC458EF9A.

Оскільки $r = \tilde{r}$, то підпис є справжній.

Ж. Приклад обчислювання геш-функції за ГОСТом Р 34.10–94

За стартовий вектор гешування взято нульове значення, тобто

$GOST_0$ = 0×00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000.

Вхідна інформаційна послідовність M:

M = 0×73657479 62203233 3D687467 6E656C20 2C656761 7373656D 20736920 73696854.

Оскільки довжина повідомлення становить 256 біт, то немає потреби дописувати нулі.

1) Обчислюємо $GOST_1 = h\ (M, GOST_0)$.

Генерування ключів:

k_A = 0×733D2C20 65686573 74746769 79676120 626E7373 20657369 326C6568 33206D54;

k_B = 0×110C733D 0D166568 130E7474 06417967 1D00626E 161A2065 090D326C 4D393320;

k_C = 0×80B111F3 730DF216 850013F1 C7E1F941 620C1DFF 3ABAE91A 3FA109F2 F513B239;

k_D = 0×A0E2804E FF1B73F2 ECE27A00 E7B8C7E1 EE1D620C AC0CC5BA A804C05E A18B0AEC.

Шифрувальне перетворювання

Для шифрувального перетворювання використовується алгоритм ГОСТ 28147–89 в режимі простої заміни. При цьому заповнення вузлів заміни (S-box) є таке:

Номер S-box	Значення															
1	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
2	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
3	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
4	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
5	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
6	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
7	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
8	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

E_{k_A} = 0×42ABBCCE 32BC0B1B;

E_{k_B} = 0×5203EBC8 5D9BCFFD;

E_{k_C} = 0×8D345899 00FF0E28;

E_{k_D} = 0×E7860419 0D2A562D;

$E(M)$ = 0xE7860419 0D2A562D 8D345899 00FF0E28 5203EBC8 5D9BCFFD 42ABBCCE 32BC0B1B.

Змішувальне перетворювання:

$GOST_1$ = 0×CF9A8C65 505967A4 68A03B8C 42DE7624 D99C4124 883DA687 561C7DE3 3315C034.

2) Обчислюємо $GOST_L = h\ (L, GOST_1)$.

Довжина повідомлення, яке підлягає гешуванню, становить 256 біт, тоді

L = 0×00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000100.

Генерування ключів:

k_A = 0×CF68D956 9AA09C1C 8C3B417D 658C24E3 50428833 59DE3D15 6776A6C1 A4248734;

k_B = 0×8FCF68D9 809AAC9C 3C8C3B41 C7658C24 BB504288 2859DE3D 666676A6 B3A42487;

k_C = 0×4E70CF97 3C8065A0 853C8CC4 57389A8C CABB50BD E3D7A6DE D1996788 5CB35B24;

k_D = 0×584E70CF C53C8065 48853C8C 1657389A EDCABB50 78E3D7A6 EED19867 7F5CB35B.

Шифрувальне перетворювання:

E = 0×66B70F5E F163F461 468A9528 61D60593 E5EC8A37 3FD42279 3CD1602D DD783E86.

Змішувальне перетворювання:

$GOST_L$ = 0×2B6EC233 C7BC89E4 2ABC2692 5FEA7285 DD3848D1 C6AC997A 24F74E2B 09A3AEF7.

3) Обчислюємо GOST = h (I, $GOST_L$).

I = 0×73657479 62203233 3D687467 6E656C20 2C656761 7373656D 20736920 73696854.

Генерування ключів:

k_A = 0×5817F104 0BD45D84 B6522F27 4AF5B00B A531B57A 9C8FDFCA BB1EFCC6 D7A517A3;

k_B = 0×E82759E0 C278D95E 15CC523C FC72EBB6 D2C73DA8 19A6CAC9 3E8440F5 C0DDB66A;

k_C = 0×77483AD9 F7C29CAA EB06D1D7 641BCAD3 FBC3DAA0 7CB555F0 D4968080 0A9E56BC;

k_D = 0×A1157965 2D9FBC9C 088C7CC2 46FB3DD2 7681ADCB FA4ACA06 53EFF7D7 C0748708.

Шифрувальне перетворювання:

E = 0×2AEBFA76 A85FB57D 6F164DE9 2951A581 C31E7435 4930FD05 1F8A4942 550A582D.

Змішувальне перетворювання:

$GOST_I$ = 0×FAFF37A6 15A81669 1CFF3EF8 B68CA247 E09525F3 9F811983 2EB81975 D366C4B1.

Отже, результат гешування:

GOST = 0×FAFF37A6 15A81669 1CFF3EF8 B68CA247 E09525F3 9F811983 2EB81975 D366C4B1.

3. Визначення термінів та понять

Інформація (information) – будь-які відомості та/або дані, які можуть бути збережені на матеріальних носіях або відображені в електронному вигляді.

Документ (document, paper) – матеріальний носій, що містить інформацію, основними функціями якого є її збереження та передавання у часі та просторі.

Державні інформаційні ресурси (state informative resources) – інформація, яка є власністю держави та необхідність захисту якої визначено законодавством.

Власник інформації (proprietor of information) – фізична або юридична особа, якій належить право власності на інформацію.

Політика безпеки інформації (information security policy) – сукупність законів, правил, обмежень, рекомендацій, інструкцій тощо, які регламентують порядок обробки інформації.

Модель політики безпеки (security policy model) – абстрактний формалізований або неформалізований опис політики безпеки інформації.

Користувач інформації в системі (user of information in the system) – фізична або юридична особа, яка в установленому законодавством порядку отримала право доступу до інформації в системі.

Правила розмежування доступу (ПРД) (access mediation rules) – частина політики безпеки, що регламентує правила доступу користувачів і процесів до пасивних об'єктів.

Безпека інформації (information security) – стан інформації, в якому забезпечується збереження визначених політикою безпеки властивостей інформації.

Комп'ютерна система (КС) (computer system) – сукупність програмно-апаратних засобів, яка подана для оцінки.

Домен комп'ютерної системи (domain) – ізольована логічна область КС, що характеризується унікальним контекстом, всередині якої об'єкти володіють певними властивостями, повноваженнями і зберігають певні відносини між собою.

Оцінка безпеки інформації (information security evaluation) – процес, метою якого є визначення відповідності стану безпеки інформації в КС встановленим вимогам.

Захист інформації (information protection, information security) – сукупність правових, адміністративних, організаційних, технічних та інших заходів, що забезпечують збереження, цілісність інформації та належний порядок доступу до неї.

Інформація з обмеженим доступом (ІзОД) (information with the limited access) – інформація, право доступу до якої обмежено встановленими правовими нормами і (чи) правилами. Інформацією з обмеженим доступом є конфіденційна, таємна та службова інформація.

Конфіденційна інформація (confidential information, inside information, privileged information) – інформація про фізичну особу, а також інформація, доступ до якої обмежено фізичною або юридичною особою, крім суб'єктів владних повноважень. Конфіденційна інформація може поширюватися за бажанням (згодою) відповідної особи у визначеному нею порядку відповідно до передбачених нею умов, а також в інших випадках, визначених законом.

Довірча конфіденційність (discretionary confidentiality) – послуга, що забезпечує конфіденційність інформації відповідно до принципів довірчого керування доступом.

Адміністративна конфіденційність (mandatory confidentiality) – послуга, що забезпечує конфіденційність інформації відповідно до принципів адміністративного керування доступом.

Таємна інформація (secret information) – інформація з обмеженим доступом, яка містить відомості, що становлять державну або іншу передбачену законом таємницю.

Цілісність інформації (information integrity) – властивість інформації, яка полягає в тому, що інформація не може бути модифікована неавторизованим користувачем і/або процесом.

Довірча цілісність (discretionary integrity) – послуга, що забезпечує цілісність інформації відповідно до принципів довірчого керування доступом.

Адміністративна цілісність (mandatory integrity) – послуга, що забезпечує цілісність інформації відповідно до принципів адміністративного керування доступом.

Цілісність системи (system integrity) – властивість системи, яка полягає в тому, що жоден її компонент не може бути усунений, модифікований або доданий з порушенням політики безпеки.

Потік інформації (information flow) – передавання інформації від одного до іншого об'єкта КС.

Доступ до інформації (access to information) – вид взаємодії двох об'єктів КС, внаслідок якого створюється потік інформації від одного об'єкта до іншого і/або відбувається зміна стану системи.

Доступ до інформації в системі (access to information in the system) – отримання користувачем можливості обробляти інформацію в системі.

Тип доступу (access type) – суттєвість доступу до об'єкта, що характеризує зміст здійснюваної взаємодії, а саме: проведені дії, напрям потоків інформації, зміни в стані системи (наприклад, читання, запис, запуск на виконання, видалення, дозапис).

Довірче керування доступом (discretionary access control) – принцип керування доступом, який полягає в тому, що звичайним користувачам дозволено керувати (довіряють керування) потоками інформації між іншими користувачами і об'єктами свого домена (наприклад, на підставі права володіння об'єктами) без втручання адміністратора.

Адміністративне керування доступом (mandatory access control) – принцип керування доступом, який полягає в тому, що керувати потоками інформації між користувачами і об'єктами дозволено тільки спеціально авторизованим користувачам, а звичайні користувачі не мають можливості створити потоки інформації, які могли б призвести до порушення встановлених ПРД.

Диспетчер доступу (reference monitor) – реалізація концепції абстрактного автомата, яка забезпечує дотримання ПРД і характеризується такими трьома особливостями: забезпечує безперервний і повний контроль за доступом, захищений від модифікації і має невеликі розміри.

Обробка інформації в системі (information processing in the system) – виконання однієї або кількох операцій, зокрема: збирання, введення, записування, перетворення, зчитування, зберігання, знищення, реєстрація, приймання, отримання, передавання, які здійснюються в системі за допомогою технічних і програмних засобів.

Порядок доступу до інформації в системі (order of access to information in the system) – умови отримання користувачем можливості обробляти інформацію в системі та правила обробки цієї інформації.

Ознайомлення (disclosure) – одержання користувачем або процесом інформації, що міститься в об'єкті.

Модифікація (modification) – зміна користувачем або процесом інформації, що міститься в об'єкті.

Критична інформація (sensitive information) – інформація, що вимагає захисту; будь-яка інформація, втрата або неправильне використання якої (модифікація, ознайомлення) може нанести шкоду власникові інформації або АС, або будь-якій іншій фізичній (юридичній) особі чи групі осіб.

Обчислювальна система (ОС) (computer system) – сукупність програмно-апаратних засобів, призначених для обробки інформації.

Автоматизована система (АС) (automated system) – організаційно технічна система, що реалізує інформаційну технологію і об'єднує ОС, фізичне середовище, персонал і інформацію, яка обробляється.

Інформаційно-телекомунікаційна система (information telecommunication system) – сукупність інформаційних та телекомунікаційних систем, які у процесі обробки інформації діють як єдине ціле.

Телекомунікаційна система (telecommunication system) – сукупність технічних і програмних засобів, призначених для обміну інформацією шляхом передавання, випромінювання або приймання її у вигляді сигналів, знаків, звуків, рухомих або нерухомих зображень чи в інший спосіб.

Захищена комп'ютерна система (trusted computer system, computer system security) – комп'ютерна система, яка здатна забезпечувати захист оброблюваної інформації від певних загроз.

Захист інформації в АС (information protection automated system, information security automated system) – діяльність, яка спрямована на забезпечення безпеки оброблюваної в АС інформації та АС в цілому, і дозволяє запобігти або ускладнити можливість реалізації загроз, а також знизити величину потенційних збитків внаслідок реалізації загроз.

Технічний захист інформації (ТЗІ) (technical protection of the information) – вид захисту інформації, спрямований на забезпечення за допомогою інженерно-технічних заходів та/або програмних і технічних засобів унеможливлення витоку, знищення та блокування інформації, порушення цілісності та режиму доступу до інформації.

Засіб забезпечення ТЗІ – технічний засіб із захистом, засіб технічного захисту інформації або засіб контролю за ефективністю технічного захисту інформації.

Засіб забезпечення ТЗІ загального призначення – такий засіб забезпечення ТЗІ, при створенні якого не передбачається його використання у складі конкретного об’єкта.

Захист інформації в системі (information protection system) – діяльність, спрямована на запобігання несанкціонованим діям щодо інформації в системі.

Комплексна система захисту інформації (КСЗІ) (complex information protection system) – взаємопов’язана сукупність організаційних, інженерно-технічних та програмно-апаратних засобів і методів які забезпечують захист інформації.

Комплекс засобів захисту (КЗЗ) (trusted computing base) – сукупність програмно-апаратних засобів, які забезпечують реалізацію політики безпеки інформації.

Керування потоками (flow control) – сукупність функцій і процедур, які забезпечують неможливість передачі інформації прихованими каналами, тобто в обхід КЗЗ. В більш вузькому значенні часто розуміється сукупність процедур, які забезпечують неможливість передачі інформації від об’єкта КС з більш високим рівнем доступу до об’єкта КС з більш низьким рівнем доступу.

Санкціонований доступ до інформації (authorized access to information) – доступ до інформації, що не порушує ПРД.

Несанкціонований доступ до інформації (НСД до інформації) (unauthorized access to information) – доступ до інформації, здійснюваний з порушенням ПРД.

Захист від НСД (protection from unauthorized access) – запобігання або істотне утруднення несанкціонованого доступу до інформації.

Право доступу (access right) – дозвіл або заборона здійснення певного типу доступу.

Повноваження (privilege) – права користувача або процесу на виконання певних дій, зокрема на одержання певного типу доступу до об'єктів.

Керування доступом (access control) – сукупність заходів з визначення повноважень і прав доступу, контролю за додержанням ПРД.

Розмежування доступу (access mediation) – сукупність процедур, що реалізують перевірку запитів на доступ і оцінку на підставі ПРД можливості надання доступу.

Доступність (availability) – властивість ресурсу системи (КС, послуги, об'єкта КС, інформації), яка полягає в тому, що користувач і/або процес, який володіє відповідними повноваженнями, може використовувати ресурс відповідно до правил, встановлених політикою безпеки, не очікуючи довше заданого (малого) проміжку часу, тобто коли він знаходиться у вигляді, необхідному користувачеві, в місці, необхідному користувачеві, і в той час, коли він йому необхідний.

Атрибут доступу (tag, access mediation information) – будь-яка зв'язана з об'єктом КС інформація, яка використовується для керування доступом.

Матриця доступу (access matrix) – *n*-мірна таблиця, вздовж кожного виміру якої відкладені ідентифікатори об'єктів КС одного типу (об'єктів-користувачів, об'єктів-процесів чи пасивних об'єктів), і містить як елементи права доступу за кожним із типів доступу.

Список доступу (access control list) – перелік користувачів і/або процесів з зазначенням їх прав доступу до об'єкта КС, з яким пов'язаний цей перелік.

Список повноважень (privilege list, profile) – перелік об'єктів з зазначенням прав доступу до них з боку користувача або процесу, з яким пов'язаний цей перелік.

Мітка (label) – атрибут доступу, що відображає категорію доступу об’єкта КС.

Категорія доступу (security level) – комбінація ієрархічних і неієрархічних атрибутів доступу, що відображає рівень критичності (наприклад, конфіденційності) інформації або повноважень користувача щодо доступу до такої інформації.

Рівень доступу (access level) – ієрархічна частина категорії доступу пасивного об’єкта.

Рівень допуску (clearance) – ієрархічна частина категорії доступу користувача або процесу, що визначає максимальний рівень доступу пасивного об’єкта, до якого може одержати доступ користувач чи процес.

Авторизація (authorization) – надання повноважень; встановлення відповідності між повідомленням (пасивним об’єктом) і його джерелом (створившим його користувачем або процесом).

Авторизований користувач (authorized user) – користувач, що володіє певними повноваженнями.

Адміністратор безпеки (security administrator) – адміністратор, відповідальний за дотримання політики безпеки.

Спостереженість (accountability) – властивість КС, що дозволяє фіксувати діяльність користувачів і процесів, використання пасивних об’єктів, а також однозначно установлювати ідентифікатори причетних до певних подій користувачів і процесів з метою запобігання порушення політики безпеки і/або забезпечення відповідальності за певні дії.

Загроза (threat) – будь-які обставини або події, що можуть бути причиною порушення політики безпеки інформації і/або нанесення збитків АС.

Атака (attack) – спроба реалізації загрози.

Блокування інформації в системі (access lock to information in the system) – дії, внаслідок яких унеможливлюється доступ до інформації в системі.

Знищення інформації в системі (erase [deletion, destruction] to information in the system) – дії, внаслідок яких інформація в системі зникає.

Виток інформації (information disclosure) – результат дій, внаслідок яких інформація в системі стає відомою чи доступною фізичним та/або юридичним особам, що не мають права доступу до неї;

Несанкціоновані дії щодо інформації в системі – дії, що провадяться з порушенням порядку доступу до цієї інформації, установленого відповідно до законодавства.

Порушник (user violator) – користувач, який здійснює несанкціонований доступ до інформації.

Порушення цілісності інформації в системі (violator information integrity in the system) – несанкціоновані дії щодо інформації в системі, внаслідок яких змінюється її вміст.

Проникнення (penetration) – успішне подолання механізмів захисту системи.

Тестування на проникання (penetration testing) – випробування, метою яких є здійснення спроби обминути або відключити механізми захисту.

Вразливість системи (system vulnerability) – нездатність системи протистояти реалізації певної загрози або сукупності загроз.

Оцінка вразливості (vulnerability assessment) – дослідження об'єкта оцінки з метою визначення можливості реалізації загроз.

Компрометація (compromise) – порушення політики безпеки; несанкціоноване ознайомлення.

Втрата інформації (information leakage) – неконтрольоване розповсюдження інформації, що веде до її несанкціонованого одержання.

Прихований канал (covert channel) – спосіб одержання інформації за рахунок використання шляхів передачі інформації, існуючих у КС, але не керованих КЗЗ, або спостереження за існуючими потоками інформації.

Комп'ютерний вірус (computer virus) – програма, що володіє здатністю до самовідтворення і, як правило, здатна здійснювати дії, які можуть порушити функціонування КС і/або зумовити порушення політики безпеки.

Програмна закладка (program bug) – потайно впроваджена програма або недокументовані властивості програмного забезпечення, використання яких може призвести до обходу КЗЗ і/або порушення політики безпеки.

Люк (trap door) – залишені розробником недокументовані функції, використання яких дозволяє обминути механізми захисту.

Модель загроз (model of threats) – абстрактний формалізований або неформалізований опис методів і засобів здійснення загроз.

Модель порушника (user violator model) – абстрактний формалізований або неформалізований опис порушника.

Ризик (risk) – функція ймовірності реалізації певної загрози, виду і величини завданих збитків.

Аналіз ризику (risk analysis) – процес визначення загроз безпеці інформації та їх характеристик, слабких сторін КСЗІ (відомих і припустимих), оцінки потенційних збитків від реалізації загроз та ступеню їх прийнятності для експлуатації АС.

Керування ризиком (risk management) – сукупність заходів, що проводяться протягом всього життєвого циклу АС щодо оцінки ризику, вибору, реалізації і впровадження заходів забезпечення безпеки, спрямована на досягнення прийнятного рівня залишкового ризику.

Залишковий ризик (residual risk) – ризик, що залишається після впровадження заходів забезпечення безпеки.

Заходи забезпечення безпеки (safeguards) – послуги, функції, механізми, правила і процедури, призначені для забезпечення захисту інформації.

Послуга безпеки (security service) – сукупність функцій, що забезпечують захист від певної загрози або від множини загроз.

Механізми захисту (security mechanism) – конкретні процедури і алгоритми, що використовуються для реалізації певних функцій і послуг безпеки.

Засоби захисту (protection facility) – програмні, програмно-апаратні та апаратні засоби, що реалізують механізми захисту.

Політика безпеки послуги (service security policy) – правила, згідно з якими функціонують механізми, що реалізують послугу.

Рівень послуги (level of service) – міра ефективності і/або стійкості механізмів, що реалізують послугу, відносно до введеної для даної послуги шкали оцінки.

Гарантії (assurance) – сукупність вимог (шкала оцінки) для визначення міри упевненості, що КС коректно реалізує політику безпеки.

Рівень гарантій (assurance level) – міра упевненості в тому, що КС коректно реалізує політику безпеки.

Критерії оцінки захищеності (security evaluation criteria) – сукупність вимог (шкала оцінки), що використовується для оцінки ефективності функціональних послуг безпеки і коректності їх реалізації.

Рейтинг (rating) – упорядкований перелік рівнів послуг і рівня гарантій, виявлених в процесі оцінки КС.

Функціональний профіль (functionality profile) – упорядкований перелік рівнів функціональних послуг, який може використовуватись як формальна специфікація функціональності КС.

Експорт інформації (information export) – виведення інформації з-під керування КЗЗ назовні.

Імпорт інформації (information import) – уведення інформації ззовні під керування КЗЗ.

Ідентифікація (identification) – процедура присвоєння ідентифікатора об’єкту КС або встановлення відповідності між об’єктом і його ідентифікатором; впізнання.

Автентифікація (authentication) – процедура перевірки відповідності пред’явленого ідентифікатора об’єкта КС на предмет належності його цьому об’єкту; встановлення або підтвердження автентичності.

Інформація автентифікації (authentication information) – інформація, що використовується для автентифікації.

Персональний ідентифікаційний номер (ПІН) (personal identification number, PIN) – вид паролю, що звичайно складається тільки із цифр, і який, як правило, має бути пред'явлений нарівні з носимим ідентифікатором.

Пароль (password) – секретна інформація автентифікації, що являє собою послідовність символів, яку користувач повинен ввести через обладнання вводу інформації, перш ніж йому буде надано доступ до КС або до інформації.

Достовірний канал (trusted path) – захищений шлях передачі інформації між користувачем і КЗЗ, що не може бути імітований, а інформація, що передається ним, не може бути отримана або модифікована стороннім користувачем або процесом.

Реєстрація (audit, auditing) – послуга, що забезпечує збирання і аналіз інформації щодо використання користувачами і процесами функцій і об'єктів, контрольованих КЗЗ.

Журнал реєстрації (audit trail) – упорядкована сукупність реєстраційних записів, кожен з яких заноситься КЗЗ за фактом здійснення контрольованої події.

Розділюваний об'єкт (shared object) – об'єкт КС, який одночасно або почергово використовується різними користувачами і/або процесами.

Повторне використання об'єкта (object reuse) – послуга, що забезпечує очищення пам'яті і призупинення дії повноважень щодо розділюваного об'єкта, який раніше використовувався одним користувачем або процесом, перед наданням його іншому користувачеві або процесу.

Аналіз прихованих каналів (covert channels analysis) – послуга, яка забезпечує гарантію того, що приховані канали в КС відсутні, знаходяться під наглядом або, принаймні, відомі.

Відкат (rollback) – послуга, що забезпечує повернення об'єкта КС до відомого попереднього стану після виконання над об'єктом певної операції або серії операцій.

Квота (quota) – обмеження можливості використання певного ресурсу КС користувачем або процесом.

Стійкість до відмов (fault tolerance) – послуга, що забезпечує здатність КС продовжувати функціонування в умовах виникнення збоїв і відмов окремих компонентів.

Ініціалізація (initialization) – встановлення системи або об'єкта у відомий чи визначений стан.

Ядро захисту (security kernel) – частина КЗЗ, в якій зосереджено мінімально необхідний набір механізмів, що реалізують ПРД.

Національна система конфіденційного зв'язку – сукупність спеціальних телекомунікаційних систем подвійного призначення, які за допомогою криптографічних та/або технічних засобів забезпечують обмін конфіденційною інформацією в інтересах органів державної влади та органів місцевого самоврядування, створюють належні умови для їх взаємодії в мирний час та у разі введення надзвичайного і воєнного стану.

Спеціальний зв'язок – передавання, випромінювання та/або приймання знаків, сигналів, письмового тексту, зображень та звуків або повідомлень, які містять інформацію з обмеженим доступом, по радіо, проводових, оптичних або інших електромагнітних системах з використанням засобів криптографічного та/або технічного захисту інформації з додержанням вимог законодавства щодо її захисту.

Урядовий зв'язок – вид спеціального зв'язку, надання якого забезпечується державною системою урядового зв'язку.

Державна система урядового зв'язку – система спеціального зв'язку, яка призначена для забезпечення управління державою в мирний час, в умовах надзвичайного та воєнного стану, а також у разі виникнення надзвичайних ситуацій та забезпечує додержання вимог законодавства з питань захисту інформації, яка містить державну таємницю.

Об'єкт інформаційної діяльності (object of informative activity) – інженерно-технічна споруда (приміщення), де здійснюється діяльність, пов'язана з інформацією, що підлягає захисту.

База персональних даних (base of the personal information) – іменована сукупність упорядкованих персональних даних в електронній формі та/або у формі картотек персональних даних.

Криптографічний захист інформації (cryptographic protection [security] of information) – вид захисту інформації, що реалізується шляхом перетворення інформації з використанням спеціальних (ключових) даних з метою приховування/відновлення змісту інформації, підтвердження її справжності, цілісності, авторства тощо.

Криптологія (cryptology) – дисципліна, що вивчає проблеми криптографічного захисту інформації та криптографічного аналізу.

Криптографія (cryptography) – дисципліна, що вивчає принципи побудови та властивості криптографічних перетворень.

Криптографічне перетворення (cryptographic transformation) – перетворення інформації з метою приховування або відновлення її змісту, підтвердження справжності, цілісності, авторства, захисту від несанкціонованого доступу тощо, що здійснюється з використанням спеціальних (ключових) даних. Розрізняють такі види криптографічних перетворень: зашифровування, розшифровування, формування та/або перевіряння цифрового підпису, вироблення імітовставки.

Криптографічний алгоритм (cryptographic algorithm) – набір математичних правил та процедур, за допомогою яких здійснюється криптографічне перетворення.

Криптографічний аналіз (cryptanalysis) – напрям криптології, що вивчає методи аналізу криптографічних систем з метою оцінки їх криптографічної стійкості до дешифрування, знаходження способів несанкціонованого доступу до системи, підробки даних, повідомлень та підписів, порушення інформаційних процесів тощо.

Криптографічний протокол (cryptographic protocol) – визначена послідовність дій обміну інформацією, в якій хоча б одна дія є криптографічним перетворенням повідомлень.

Засіб криптографічного захисту інформації (facility for cryptographic protection of information) – програмний, апаратно-програмний або апаратний засіб, призначений для криптографічного захисту інформації.

Шифр (cipher) – сукупність оборотних відображень множини відкритих текстів у множину шифртекстів, які здійснюються за певними правилами із застосуванням ключів.

Асиметричне криптографічне перетворення (asymmetric cryptographic transformation) – пряме та однозначно визначене обернене криптографічні перетворення інформації, які здійснюються за допомогою пов'язаних між собою різних відкритого та таємного ключів.

Симетричне криптографічне перетворення (symmetric cryptographic transformation) – пряме та однозначно визначене обернене криптографічні перетворення інформації, які здійснюються з використанням одного і того ж таємного ключа.

Потоковий шифр (stream cipher) – шифр, в якому криптографічне перетворення задається на символах відкритого тексту з наступним зашифровуванням кожного символу окремо.

Блоковий шифр (block cipher) – шифр, в якому криптографічне перетворення задається на блоках відкритого тексту з наступним зашифровуванням кожного блоку окремо.

Шифрування (ciphering, encryption) – процеси зашифровування та розшифровування. Перетворення електронного документа із застосуванням криптографічних методів з метою захисту його змісту.

Шифртекст [зашифроване повідомлення] (ciphertext) – результат перетворення відкритого тексту криптографічною системою, яке визначається ключем.

Криптограма (cryptogram) – повідомлення, яке містить шифртекст та, у разі необхідності, іншу додаткову інформацію.

Зашифровування даних (enciphered data, data encryption) – процес перетворення відкритого тексту у шифртекст.

Розшифровування даних (deciphered data, data decryption) – процес перетворення шифртексту у відкритий текст.

Дешифрування (attack on a ciphertext, deciphering, decryption) – перетворення шифртексту у відкритий текст без знання ключа.

Імітозахист (imitation security) – захист повідомлення від його модифікації.

Імітовставка [код автентифікації повідомлення] (data authentication code) – блок інформації фіксованої довжини, що одержується з відкритого тексту і ключа та доданий до відкритого тексту (шифртексту) для забезпечення імітозахисту.

Імітостійкість (криптографічної системи) (imitation resistance) – здатність криптографічної системи протистояти нав'язуванню неправдивої інформації будь-якими відомими способами.

Ключ (key) – конкретний стан деяких параметрів алгоритму криптографічного перетворення, що забезпечує вибір одного перетворення із сукупності можливих для даного алгоритму.

Управління ключами (key management) – сукупність функцій, що пов'язані з генеруванням, реєструванням, сертифікацією, розподіленням, зберіганням, архівуванням, скасуванням, зняттям з реєстрації та знищенням ключів.

Відкритий ключ (public key) – ключ, який не приховується на кожному із етапів його життєвого циклу.

Особистий [**таємний**] **ключ** (secret [private] key) – ключ, який не повинен бути доступним стороннім на кожному із етапів його життєвого циклу.

Разовий ключ (session key) – ключ, який застосовується для зашифровування тільки одного повідомлення.

Сеансовий ключ (short-term key) – ключ, який застосовують для зашифровування лише впродовж одного сеансу зв'язку або визначеного обмеженого часу.

Довготривалий ключ (long-term key) – ключ, який можна вживати впродовж визначеного довготривалого часу.

Ключ для зашифровування (enciphered key, encryption key) – ключ, який визначає криптографічне перетворення відкритого тексту у шифртекст.

Ключ для розшифровування (deciphered key, decryption key) – ключ, який визначає криптографічне перетворення шифртексту у відкритий текст.

Майстер ключ (master key) – ключ найвищого рівня у ієрархії ключів при їх зашифровуванні, який, як правило, не шифрується і розміщується в захищеній частині засобу криптографічного захисту інформації.

Системний ключ (system key) – ключ, який є постійним для певної криптографічної системи.

Односпрямована функція (one-way function) – функція, у якої для всіх значень аргументів існує поліноміальний алгоритм обчислення значення функції, але майже для всіх значень функції не існує поліноміального алгоритму обернення функції.

Геш-функція (hash function) – функція, що перетворює послідовність елементів даних довільної довжини у послідовність елементів даних фіксованої довжини.

Гешування (hashing) – процес застосування геш-функції до відповідних даних.

Криптостійкість (cryptographic strength) – називається характеристика шифру, що визначає його стійкість до дешифруванню без знання ключа (тобто криптоаналізу).

Завірення (notarization) – реєстрація даних у довіреної третьої особи з метою забезпечення надалі впевненості в правильності таких характеристик як зміст, джерело даних, час відправлення чи одержання тощо.

Електронний підпис (electronic signature) – дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов’язані та призначені для ідентифікації підписувача цих даних.

Електронний цифровий підпис (ЕЦП) (electronic digital signature, EDS) – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. ЕЦП накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Засвідчення чинності відкритого ключа – процедура формування сертифіката відкритого ключа.

Засіб електронного цифрового підпису (facility for electronic digital signature) – програмний засіб, програмно-апаратний або апаратний пристрій, призначені для генерації ключів, накладення та/або перевірки електронного цифрового підпису.

Особистий ключ ЕЦП (secret [private] key EDS) – параметр криптографічного алгоритму формування електронного цифрового підпису, доступний тільки підписувачу.

Відкритий ключ ЕЦП (public key EDS) – параметр криптографічного алгоритму перевірки електронного цифрового підпису, доступний суб'єктам відносин у сфері використання електронного цифрового підпису.

Сертифікат відкритого ключа (certificate public key) – документ, виданий центром сертифікації ключів, який засвідчує чинність і належність відкритого ключа підписувачу. Сертифікати ключів можуть розповсюджуватися в електронній формі або у формі документа на папері та використовуватися для ідентифікації особи підписувала.

Посилений сертифікат відкритого ключа – сертифікат ключа, виданий акредитованим центром сертифікації ключів, засвідчувальним центром, центральним засвідчувальним органом.

Акредитація (accreditation) – процедура документального засвідчення компетентності центра сертифікації ключів здійснювати діяльність, пов'язану з обслуговуванням посилених сертифікатів ключів.

Компрометація особистого ключа (compromising of the secret [private] key) – будь-яка подія та/або дія, що призвела або може призвести до несанкціонованого використання особистого ключа.

Блокування сертифіката ключа (access lock to certificate key) – тимчасове зупинення чинності сертифіката ключа.

Послуги електронного цифрового підпису (services of electronic digital signature) – надання у користування засобів електронного цифрового підпису, допомога при генерації відкритих та особистих ключів, обслуговування сертифікатів ключів (формування, розповсюдження, скасування, зберігання, блокування та поновлення), надання інформації щодо чинних, скасованих і блокованих сертифікатів ключів, послуги фіксування часу, консультації та інші послуги.

Підписувач (signed) – особа, яка на законних підставах володіє особистим ключем та від свого імені або за дорученням особи, яку вона представляє, накладає електронний цифровий підпис під час створення електронного документа.

Надійний засіб електронного цифрового підпису – засіб електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації.

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Навчальне видання

Захарченко Микола Васильович
Онацький Олексій Віталійович
Йона Лариса Григорівна
Шинкарчук Тетяна Миколаївна

АСИМЕТРИЧНІ МЕТОДИ ШИФРУВАННЯ В ТЕЛЕКОМУНІКАЦІЯХ

Навчальний посібник

Редактор *В. Т. Гусак*
Комп'ютерне верстання *Ж. А. Гардиман*

Здано до набору 13.05.11. Підписано до друку 20.05.11.
Формат 60х90/16. Зам. 4725
Наклад 300 прим. 10,25 умовн. друк. арк.
Віддруковано на видавничому устаткуванні фірми RISO
у друкарні редакційно-видавничого центру ОНАЗ ім. О. С. Попова
м. Одеса, вул. Старопортофранківська, 61
Тел. 720-78-94
ОНАЗ, 2011