Розділ 4

АНАЛІЗ ВЛАСТИВОСТЕЙ ТА ОБЛАСТЕЙ ЗАСТОСУВАННЯ ЦИФРОВИХ ПІДПИСІВ ІЗ ВІДНОВЛЕННЯМ ПОВІДОМЛЕННЯ

У сучасних автоматизованих системах управління, комп'ютерних системах і мережах, інформаційних і телекомунікаційних системах, висуваються високі вимоги до забезпечення цілісності, автентичності (справжності) та доступності інформації на всіх етапах її життєвого циклу, а також надання послуг неспростовності [29, 42, 44, 52, 180–185]. Досвід застосування та проведені дослідження підтвердили, що ці високі вимоги, особливо щодо реалізації функції причетності (неспростовності), можуть бути забезпечені тільки за рахунок застосування (електронного) цифрового підпису (ЕЦП) [17–20]. Цифровий підпис (ЕЦП), по суті, являє собою додані до інформації дані, обчислені за допомогою криптографічного перетворення захищуваної інформації, і спирається на параметри, за наявності яких можна упевнитися в цілісності й справжності інформації та її джерела, а також забезпечити захист від підробки з боку отримувача.

На цей час широкого розповсюдження набули ЕЦП з додатком [15, 16, 34] та відновленням повідомлення [27, 29, 186], що ґрунтуються на використанні асиметричних криптографічних перетворень. В ЕЦП з додатком підпису цифровий підпис приєднується до повідомлення та зберігається й передається з ним, а для перевірки ЕЦП потрібно обов'язково мати сертифікат відкритого ключа, що був використаний під час підпису. Детально ЕЦП з додатком розглянутий у розділі 3 цієї монографії. В ЕЦП з відновленням повідомлення частина або повне повідомлення можуть бути відновлені з цифрового підпису, тобто для перевірки цифрового підпису необхідно знати тільки цифровий підпис і, можливо, сертифікат відкритого ключа.

Теоретичні обґрунтування й практичні дослідження ЕЦП з відновленням повідомлення були виконані, у порівнянні з ЕЦП з додатком, пізніше. Значною мірою вони з'явилися, коли виникла необхідність в ЕЦП для коротких повідомлень. Цей напрям був успішно розвинутий у роботах [181–186]. Як наслідок у 2003 році був прийнятий міжнародний стандарт ISO/IEC 15946-4 [27]. У нього було включено 5 незалежних алгоритмів ЕЦП з відновленням повідомлення, криптографічні перетворення в якому базуються на еліптичних кривих. У подальшому цей стандарт було вдосконалено, і він був прийнятий у 2006 році як ISO/IEC 9796-3

[29] на заміну існуючому. Додатково в нього був включений алгоритм ЕЦП, що ґрунтується на перетворенні в полі Галуа.

З прийняттям стандарту в практичному аспекті виникла проблема оцінки криптографічної стійкості та практичного застосування, перш за все порівняльного аналізу схем (методів) ЕЦП, що містяться в стандарті.

Метою цього розділу є опис і аналіз алгоритмів стандарту ЕЦП як ISO/IEC 9796-3 з відновленням повідомлення, їх порівняння, оцінка захищеності та колізійної стійкості, а також аналіз алгоритмів з точки зору практичного застосування.

Особливістю схеми підпису з відновленням повідомлення є те, що в ньому висувають правила використання функції формування доповнення. Для повної перевірки абонент цифрового підпису повинен мати повну та неушкоджену збитковість повідомлення. Також схеми з відновленням повідомлення не висувають обмежень у використанні функції формування збитковості. Наприклад, частка відновлюваного повідомлення могла б мати чітко визначений розмір у 80 бітів, але в цьому випадку нівелюються всі переваги схеми. До того ж типові повідомлення належать до якоїсь групи значень, тобто мають природну збитковість тощо.

Таким чином, схеми ЕЦП із відновленням повідомлення доцільно використовувати в інформаційних системах і протоколах з чітко визначеними повідомленнями. Це є принциповою особливістю з точки зору їх застосування. Також, зважаючи на те, що ЕЦП з відновленням повідомлення є специфічним, наведемо основні символи та позначення, що використовуються в цьому розділі (таблиця 4.1).

Таблиця 4.1. **Основні символи та позначення ЕЦП з відновленням**

d, d'	Вхідні дані та відновлені вхідні дані відповідно
h, h', h''	Геш-токен (значення), відновлений (обрізаний) геш-токен, повторно обчислений (обрізаний) геш-токен відповідно
k	Ключ сеансу
l_h	Довжина (обрізаного) геш-токену (у бітах)
L_F	Розмір поля F (в октетах)
lrec, *lclr*, ln	Довжина (у бітах) M_{rec}, M_{clr} та n відповідно
L_P	Довжина p (в октетах)
l_1, l_2	Довжина (у бітах) короткої і довгої надлишковості, відповідно

Закінчення табл. 4.1

M, M_{clr}, M_{rec}	Повідомлення M, невідновлювана частина M_{clr} і відновлювана частина M_{rec} відповідно
M', M'_{rec}	Відновлене повідомлення, відновлена частина повідомлення відповідно
m	Ціле натуральне число
n, a	Порядок базової точки та первісний елемент простого поля Галуа відповідно
$F(q)$	Скінченне поле з q елементами, де q є простим ступенем
$E(F(q))$	Група з u точками на еліптичній кривій – порядок еліптичної кривої
$П$	Попередній підпис
r, r'	Перша частина підпису r, що обчислений підписувачем, і перша частина підпису r', що має перевірник, відповідно
s, s'	Друга частина підпису s, що обчислений підписувачем, і друга частина s', що має перевірник, відповідно
x_A	Особистий ключ підпису об'єкта A
Y_A	Відкритий ключ перевірки об'єкта A
P, Q	Точки, які приймають значення в залежності від обраної схеми генерації ключів. Так, $P = G$ і $Q = Y_A$ для схеми генерації ключів I, та $P = Y_A$ і $Q = G$ для схеми генерації ключів II
π	Функція перетворення точки на еліптичній кривій на ціле число
$\times$	Добуток в декартовій системі координат
XOR	Побітова сума виключного АБО (сума за модулем 2)
SYM	Використовуваний симетричний шифр, ключ якого формується за допомогою функції формування ключів KDF
KDF	Функція формування симетричних ключів, у якій на вхід подається точка еліптичної кривої, а на виході видається значення, що може бути використане як симетричний ключ симетричного шифру SYM

Окрім того, надалі будемо дотримуватись таких домовленостей щодо кодування, довжини та розміру поля Галуа:

- Усі цілі числа необхідно записувати так, щоб старший розряд (або біт, або октет) був у крайній лівій позиції.
- Якщо ціле число U знаходиться в діапазоні $2^{i-1} \le U \le 2^i$, тоді вважається, що довжина U (у бітах) дорівнює i та використовується позначення $i = lU$.
- Якщо ціле число U знаходиться в діапазоні $256^{m-1} \le U < 256^m$, то вважається, що довжина U (в октетах) дорівнює m і використовується позначення $m = L_U$. Таким чином, L_U – найменше ціле число, що задовольняє вимозі $8L_U \ge lU$.
- Якщо F є скінченним простим полем $F(p)$, то $L_F = L_p$. Якщо F є розширенням поля $F(2^m)$, то L_F є найменшим цілим числом, що задовольняє вимозі $8L_F \ge m$. Якщо F є розширенням поля $F(p^m)$, то L_F є найменшим цілим числом, що задовольняє вимозі $L_F \ge \log_{256} p^m$.

4.1. ЗАГАЛЬНА МОДЕЛЬ ЕЦП З ВІДНОВЛЕННЯМ ПОВІДОМЛЕНЬ

У стандарті ISO/IEC 9796-3:2006 визначено 6 різних схем (методів) і на їх основі 6 механізмів цифрових підписів з відновленням повідомлення. Механізми цифрового підпису, подані в стандарті, мають назву «механізм підпису з відновленням». Вони забезпечують повне або часткове відновлення повідомлення. У подальшому ЕЦП з відновленням повідомлення будемо розглядати через специфікацію процесів обчислення (генерації) параметрів, обчислення (генерації) підпису та перевіряння підпису.

4.1.1. Процес обчислення параметрів

Параметри можна розділити на доменні параметри та параметри користувача. Доменні параметри складаються з параметрів для визначення скінченного поля, параметрів для визначення еліптичної кривої над скінченним полем та іншої відкритої інформації, що є спільною, відомою та доступною для всіх об'єктів у межах домену.

Обов'язково мають бути визначені такі параметри:

– ідентифікатор схеми цифрового підпису, що використовується;

– функція гешування (Hash);

– процедури генерації параметрів користувача.

Кожен об'єкт має свої власні відкриті й особисті параметри. До параметрів користувача об'єкта A належать:

– особистий ключ підпису x_A;

– відкритий ключ перевірки Y_A;

– інша інформація (необов'язково), що є специфічною для об'єкта A, яка використовується в процесі генерації підпису і/або в процесі перевіряння.

Застосовувані параметри та ключі повинні бути дійсними – у сенсі їх цілісності та справжності (автентичності).

Впевненість у дійсності доменних параметрів може бути забезпечена в один із таких способів:

1) за умови вибору дійсних доменних параметрів з довірчого опублікованого джерела, наприклад такого, як стандарт відповідного ЕЦП;

2) коли обчислення дійсних доменних параметрів виконується третьою довірчою стороною, такою як центр сертифікації;

3) коли перевірка дійсності кандидатів до доменних параметрів здійснюється третьою довірчою стороною, такою як центри сертифікації;

4) за умови обчислення доменних параметрів підписувачем з використанням довірчої системи;

5) коли перевірка дійсності кандидатів до доменних параметрів виконується підписувачем або перевірником.

Впевненість у дійсності відкритого ключа перевірки може бути забезпечена в один із таких способів:

1) при генерації (обчисленні) пари відкритого ключа перевірки особистого ключа підпису з використанням довірчої третьої сторони;

2) коли перевірка дійсності відкритого ключа перевірки здійснюється третьою довірчою стороною, такою, як центр сертифікації;

3) коли перевірка дійсності відкритого ключа перевірки здійснюється підписувачем або перевірником.

4.1.2. Процес обчислення підпису

До обчислення ЕЦП мають бути введені:

– дійсні доменні параметри;

– параметри користувача підписувача A, зокрема особистий ключ підпису x_A;

– повідомлення M, що має бути підписане.

Незалежно від схеми, процес обчислення підпису складається з таких окремих етапів (процедур):

1) розщеплення повідомлення на складові;

2) обчислення надлишковості, або обчислення геш-значення повідомлення (за вибором користувача);

3) обчислення в групі точок еліптичної кривої;

4) обчислення за модулем порядку групи базової точки G;

5) форматування підписаного повідомлення.

Вихідними даними процесу генерації підпису є пара цілих чисел (r, s), що є цифровим підписом повідомлення M відповідного об'єкта, наприклад A.

4.1.3. Процес перевіряння підпису

Для перевіряння підпису потрібні такі параметри та ключі:

– дійсні доменні параметри;

– відкритий ключ перевірки ЕЦП Y_A користувача A;

– невідновлюване повідомлення M'_{clr} (якщо воно є);

– ЕЦП для повідомлення M, що представлений двома цілими числами – r' та s'.

Для всіх схем, поданих у стандарті, процес перевіряння підпису складається з окремих або всіх таких етапів (процедур):

1) перевіряння розміру підпису;

2) відновлення попереднього підпису та вхідних даних;

3) відновлення повідомлення;

4) перевірка надлишковості, або обчислення геш-значення повідомлення (з правом вибору);

5) обчислення за модулем порядку групи базової точки G;

6) обчислення в групі точок еліптичної кривої;

7) перевірка підпису.

Якщо всі процедури пройшли успішно, перевірник приймає підпис, в іншому випадку підпис відхиляється.

4.1.4. Особливості ЕЦП з відновленням повідомлення

Для реалізації кожного з механізмів цифрового підпису, визначених у стандарті, повинні бути обрані такі доменні параметри конкретної схеми цифрового підпису:

– скінченне поле Галуа $F(q)$;

– еліптична крива E над полем $F(q)$, яка має унікальну циклічну підгрупу простого порядку n;

– точка G на еліптичній кривій E простого порядку n.

Користувачі повинні обрати одну з таких надлишковостей:

– природна надлишковість;

– додана надлишковість;

– обидва типи надлишковості.

Повідомлення з природною надлишковістю означає, що повідомлення за своєю природою включає надлишковість, або те, що надлишковість повідомлення може бути побічно перевірена деякими застосуваннями.

Повідомлення з доданою надлишковістю може бути сформовано із застосуванням повідомлення або відновлюваного повідомлення. Природна або додана надлишковість можуть бути чимось таким, що узгоджено взаємодіючими сторонами і може бути перевірено ними.

Повна надлишковість, яка складається з природної надлишковості і доданої надлишковості, має бути більше деякого мінімального значення, що визначено застосуванням у відповідному додатку.

Якщо використовується додана надлишковість, то типи надлишковості мають бути віднесені до таких фіксованих значень:

– коротка надлишковість;

– довга надлишковість.

Коротка надлишковість має використовуватись у випадках, коли з підпису можна відновити все повідомлення.

Довга надлишковість має використовуватись у випадках, коли з підпису можна відновити лише частину повідомлення.

Довжини короткої та довгої надлишковостей, відповідно l_1 і l_2 повинні бути зафіксованими. Причому якщо довжина повідомлення (у бітах) не більша ніж $l_n - l_1 - 1$, то з підпису можна відновити все повідомлення і використовується коротка надлишковість. Якщо довжина (у бітах) повідомлення більша ніж $l_n - l_1 - 1$, то частина відновлюваного повідомлення не може бути більшою ніж $l_n - l_2 - 1$ бітів, а в цілому використовується довга надлишковість.

Типовими значеннями l_1 є 64 або 80. Типові значення len_2 змінюються у діапазоні від 136 до 168. Також допускається встановлювати $l_1 = l_2$.

4.1.5. Перелік функцій і процедур

Схеми підпису, що розглядаються, дозволяють відновлювати повідомлення, тобто здійснити відновлення деяких з даних, що використовуються при обчисленні підпису. Вони є частиною процедури перевіряння підпису.

Підпис виконується через виконання таких процедур (етапів):

1) генерування (обчислення) доменних параметрів;
2) генерування (обчислення) асиметричної ключової пари ЦП;
3) формування ключа сеансу та попереднього підпису;
4) обчислення першої частини підпису;
5) обчислення другої частини підпису;
6) відновлення попереднього підпису;
7) відновлення вхідних даних.

4.1.6. Генерування (обчислення) асиметричної ключової пари ЦП

Для обчислення асиметричної ключової пари ЦП може застосовуватись один із двох наступних методів.

Метод генерації ключа I

Для заданого дійсного набору доменних параметрів еліптичної кривої асиметрична пара, тобто особистий ключ підпису x_A та відповідний йому відкритий ключ Y_A, повинні генеруватись таким чином.

1. Обрати (згенерувати) з множини $[2, n-2]$ випадкове або псевдовипадкове ціле число x_A, тобто особистий ключ. Ключ x_A має бути захищеним від несанкціонованого розкриття і бути непередбачуваним.

2. Обчислити відкритий ключ як точку еліптичної кривої $Y_A = x_A G$.

3. Асиметричною ключовою парою є пара (Y_A, x_A).

Надалі для уніфікації позначення та однакового представлення в цьому розділі приймемо позначення $P := G$ і $Q := Y_A$.

Метод генерації ключа II

Для другого методу та заданого дійсного набору доменних параметрів еліптичної кривої асиметрична ключова пара, тобто особистий ключ підпису x_A та відповідний йому відкритий ключ Y_A, повинні генеруватися таким чином.

1. Обрати (згенерувати) з множини $[2, n-2]$ випадкове або псевдовипадкове ціле число e та обчислити ціле число x_A в інтервалі $[2, n-2]$ – таке, що $x_A e = 1 \bmod n$.

Обидва цілі числа x_A і e мають бути захищеними від несанкціонованого розкриття та бути непередбачуваними.

2. Обчислити точку еліптичної кривої $Y_A = x_A G$.

3. Ключовою парою є (Y_A, x_A), де, як і раніше, Y_A – відкритий ключ перевірки, а x_A – особистий ключ ЦП.

Перед використанням відкритого ключа перевірки перевірник повинен мати гарантію його дійсності та володіння.

4.1.7. Генерування (обчислення) ключа сеансу та попереднього ЦП

Перед кожним обчисленням підпису об'єкт, що підписує, повинен мати доступ до нового, захищеного значення ключа сеансу k та його використати.

Ключем сеансу є ціле число k – таке, що $1 < k < n-1$. Реалізація схеми підпису має забезпечувати виконання таких двох вимог:

1) використані ключі сеансу k не повинні ніколи розкриватися, й одразу після використання кожен ключ повинен бути знищеним;

2) ключі сеансу k повинні генеруватися таким чином, щоб імовірність використання одного й того самого ключа сеансу при формуванні підписів для двох різних повідомлень була зневажливо малою.

Попередній підпис у всіх ЦП з відновленням повідомлення обчислюється як функція ключа сеансу. При цьому необхідно мати на увазі, що розкриття ключа сеансу k (після його використання) може призвести до компрометації особистого ключа підпису x_A. Але оскільки кожен із ключів сеансу, що вже використаний, більше не використовується ні підписувачем, ні перевірником, то він може (повинен) бути знищеним відразу після обчислення підпису.

4.1.8. Обчислення першої і другої частин підпису

Перша частина підпису r у всіх алгоритмах ЦП обчислюється як функція попереднього підпису Π і вхідних даних D, які є цілим числом, що залежить від повідомлення, причому $0 \le D < n$. Вона є цілим числом r – таким, що $0 < r < n$, де n є порядок базової точки.

Друга частина підпису s у всіх алгоритмах обчислюється із застосуванням особистого ключа підпису x_A, першої частини підпису r і ключа сеансу k. Вона є цілим числом s – таким, що $0 \le s < n$, де n є порядок базової точки.

4.1.9. Відновлення попереднього підпису та вхідних даних

Відновлення попереднього підпису Π здійснюється з використанням відкритого ключа підписувача, наприклад Y_A, та самого підпису (r, s).

Відновлення вхідних даних d виконується з використанням заданої першої частини r підпису та відновленого попереднього підпису Π'.

4.2. Обчислення цифрового підпису

Алгоритм обчислення цифрового підпису складається з таких кроків.

1. Формування ключа сеансу та обчислення попереднього підпису (передпідпису);

2. Розщеплювання повідомлення M на відновлювану частину M_{rec} та невідновлювану частину повідомлення M_{clr};

3. Формування вхідних даних d;

4. Обчислення цифрового підпису (r, s);

5. Форматування підписаного повідомлення.

На рис. 4.1 наведено алгоритм обчислення підпису.

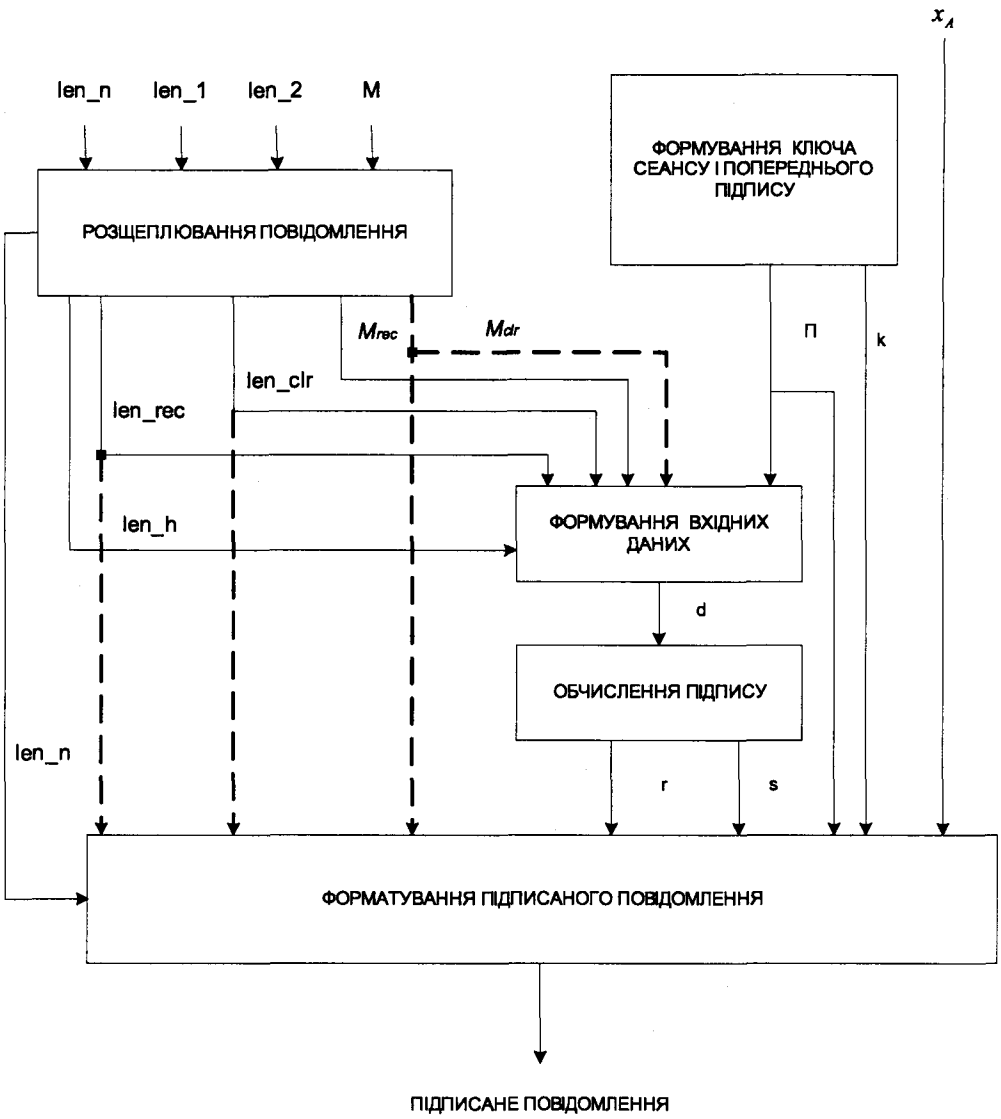

Рис. 4.1. Алгоритм обчислення цифрового підпису

4.2.1. Формування ключа сеансу та попереднього підпису

Попередній підпис $П$ є проміжним елементом даних, що виробляється до початку процесу обчислення підпису в будь-якому захищеному від його компрометації механізмі цифрового підпису. При формуванні передпідпису спершу у відповідності з вимогами, що викладені вище, формується особистий ключ сеансу k. Термін «особистий» застосовується у тому сенсі, що відносно нього повинна бути забезпечена конфіденційність. Тому значення ключа сеансу k має

бути доступним тільки процесу генерації цифрового підпису. Попередній підпис $П$ є відкритим елементом даних, а по суті відкритим ключем сеансу, тоді як значення ключа сеансу k має бути доступне тільки процесу обчислення підпису й бути конфіденційним. Необхідно також відзначити, що особисті ключі сеансу k можуть формуватися, а відповідні попередні підписи $П$ можуть обчислюватися автономно та зберігатися безпечним способом для використання при виконанні наступних цифрових підписів.

4.2.2. Розщеплення повідомлення

Повідомлення M розщеплюється на відновлювану частину M_{rec} та невідновлювану частину повідомлення M_{clr}, при цьому відповідно l_{rec} і l_{clr} визначені як довжини відновлюваної частини M_{rec} і невідновлюваної частини M_{clr}. При цьому додана надлишковість повідомлення M може бути розщеплена таким чином.

- Якщо довжина l_M (у бітах) повідомлення M задовольняє умові

$$l_M \le l_n - l_1 - 1,$$

то з підпису можна відновити все повідомлення M. У цьому випадку правильними є рівняння: $M_{rec} = M$ та $l_{rec} = l_M$. Окрім того, $l_{clr} = 0$ та $l_h = l_1$.

- Якщо довжина l_M (у бітах) повідомлення M задовольняє умові

$$l_M > l_n - l_1 - 1,$$

то $lrec$ визначається як ціле число, таке що $l_{rec} \le l_n - l_2 - 1$. Крайні ліві l_{rec} біти M складають відновлювану частину повідомлення M_{rec}. А крайні праві $l_M - l_{rec}$ біти складають невідновлювану частину M_{clr} повідомлення M, і правильними є рівняння $l_{rec} = l_M - l_{clr}$ та $l_h = l_2$.

4.2.3. Формування вхідних даних та обчислення ЦП

Вхідними даними функції формування даних є:

– l_{rec}, l_{clr} та попередній підпис $П$;

– невідновлювана частина повідомлення M_{clr} (необов'язково) і геш-токен повідомлення, що відновлюється;

– M_{rec} з доданою надлишковістю, або повідомлення, що відновлюється, M_{rec} з природною надлишковістю.

Геш-токен є або самим геш-значенням, або геш-значенням, до якого праворуч приєднано ідентифікатор геш-функції, якщо геш-значення обчислюється способом гешування повідомлення. Вибір щодо включення до геш-токену ідентифікатора геш-функції має обумовлюватися доменними параметрами. Виходом функції введення даних є значення D, яке після перетворення на ціле число знаходиться в діапазоні $0 \le D < n$.

4.2.4. Порядок обчислення цифрового підпису

Цифрові підписи, що формуються, мають дві частини – r та s. Перша частина r обчислюється як функція вхідних даних D і попереднього підпису $П$. Друга частина s обчислюється як функція від першої частини підпису r, ключа сеансу k та особистого ключа підпису x_A.

4.3. ПЕРЕВІРЯННЯ ЦИФРОВОГО ПІДПИСУ

На рис. 4.2 наведено алгоритм (процес) перевіряння підпису. Він складається з таких кроків:

1) відкриття повідомлення з цифровим підписом;
2) перевіряння розміру компонент r та s цифрового підпису;
3) відновлення попереднього підпису та/або вхідних даних;
4) відновлення вхідних даних або повідомлення;
5) повторне обчислення геш-токену (необов'язкове);
6) зіставлення підпису та прийняття рішення.

Зіставлення підпису складається із:

1) порівняння відновлених і повторно обчислених (обрізаних) геш-токенів;
2) перевіряння надлишковості.

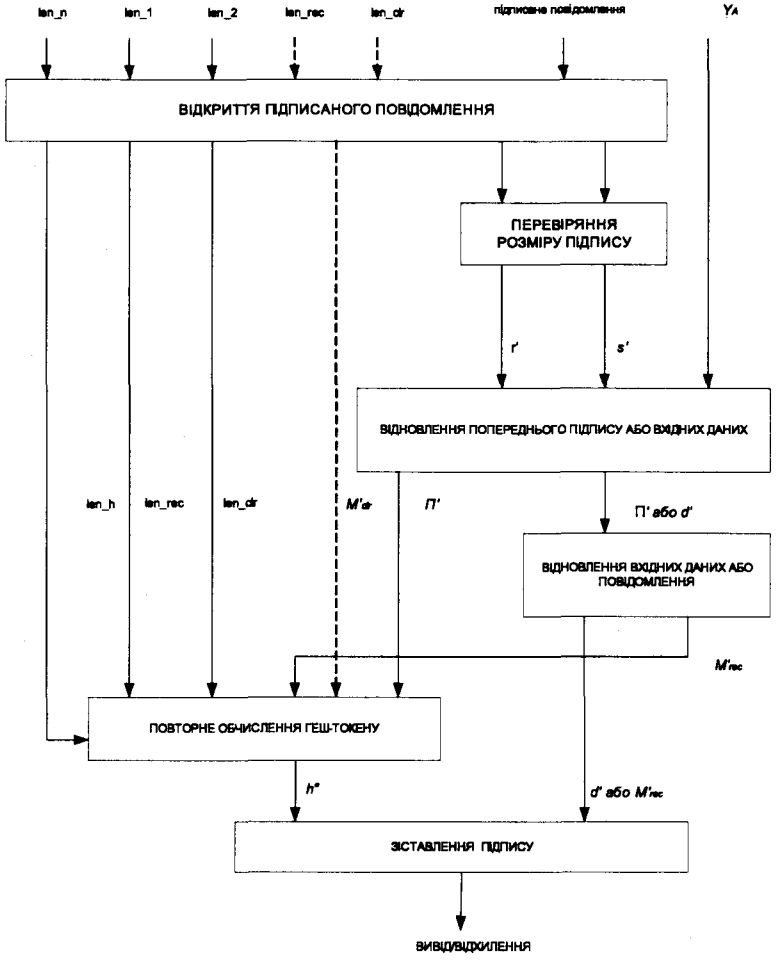

Рис. 4.2. Процес (алгоритм) перевіряння підпису

4.3.1. Відкриття підписаного повідомлення

На початку цього кроку перевірник повинен мати доступ до такої інформації:

– довжини різних частин повідомлення, що містяться в підписаному повідомленні;

– значень параметрів l_n, l_1 та l_2.

Перевірник вибирає різні частини – підписаного повідомлення в такому порядку:

– невідновлювану частину повідомлення;

– першу частину r' підпису;

– другу частину s' підпису.

4.3.2. Перевіряння розміру підпису, відновлення попереднього підпису та вхідних даних

Перевіряння розміру підпису

Перевірник має перевірити розмір частин підпису, тобто що $0 < r' < n$ та $0 \le s' < n$, де n – порядок базової точки.

На початку цього кроку перевірник повинен мати доступ до такої інформації:

– відкриті параметри, які визначають схему підпису, що використовується;

– відкритий ключ перевірки Y_A об'єкта, що підписав.

Обчислення на цьому кроці залежать від схеми підпису, що використовується. Попередній підпис і вхідні дані відновлюються з підпису. Відновленим попереднім підписом є Π', а відновленими вхідними даними є d'.

Відновлення вхідних даних або повідомлення

Відновлені вхідні дані d' перетворюються на рядок бітів. Вхідними даними d' є:

– значення геш-токену;

– відновлюване повідомлення з доданою надлишковістю;

– відновлюване повідомлення з природною надлишковістю.

4.3.3. Повторне обчислення геш-значення та зіставлення підпису

Повторне обчислення геш-значення

Перед обчисленням геш-токену спершу ідентифікується геш-функція, що використовувалась об'єктом при обчисленні підпису, наприклад, на основі ідентифікатора геш-функції з відновленого геш-значення. Потім геш-значення повторно обчислюється шляхом гешування повідомлення, що мається у перевірника.

Повторно обчислене геш-значення використовується для отримання повторно обчисленого геш-токену шляхом необов'язкового поєднання з ідентифікатором геш-функції.

Зіставлення підпису

Зіставлення підпису складається із:

1) порівняння відновленого та повторно обчисленого (обрізаного) геш-токенів;

2) перевіряння надлишковості.

Процедура порівняння складається з порівняння повторно обчисленого (обрізаного) геш-токену h'' та відновленого (обрізаного) геш-токену h'. Підпис має відхилятися, якщо ці два значення не рівні.

Процедура перевіряння надлишковості призначена для перевірки додаткової та/або природної надлишковості відновленого повідомлення. Підпис повинен відхилятися, якщо надлишковість не підтверджено.

4.3.4. Форматування підписаного повідомлення

Для успішного відкриття й перевірки підписаного повідомлення необхідно знати довжину відновлюваної частини повідомлення та довжину невідновлюваної частини повідомлення. Якщо ця інформація не міститься в доменних параметрах, то вона повинна бути включена до підписаного повідомлення.

Підписане повідомлення складається з таких елементів даних:

– невідновлювана частина повідомлення M_{clr} та довжина повідомлення l_m;

– перша r частина підпису;

– друга s частина підпису.

4.3.5. Функції перетворення та генерації маски

Надалі будемо застосовувати такі функції перетворення та генерації маски.

BS2IP	– примітив перетворення бітових рядків на цілі числа.
BS2OSP	– примітив перетворення бітових рядків на октетові (байтові) рядки.
EC2OSP	– примітив перетворення точки еліптичної кривої на октетові рядки.
FE2IP	– примітив перетворення елементів кінцевого поля на цілі числа.
FE2OSP	– примітив перетворення елементів кінцевого поля на октетові рядки.
I2BSP	– примітив перетворення цілих чисел на бітові рядки.
I2OSP	– примітив перетворення цілих чисел на октетові рядки.
MGF1, MGF2	– функції генерації маски.
OS2BSP	– примітив перетворення октетових рядків на бітові рядки.
OS2ECP	– примітив перетворення октетових рядків на точку еліптичної кривої.
OS2FEP	– примітив перетворення октетових рядків на елементи кінцевого поля.
OS2IP	– примітив перетворення октетових рядків на цілі числа.

В іншій формі всі ЕЦП з відновленням повідомлення, що розглядаються в цьому розділі, записані на електронному диску, який додається до монографії.

4.4. ЦИФРОВИЙ ПІДПИС НІБЕРГА-РЮПЕЛЯ У СКІНЧЕННОМУ ПОЛІ (NYRBERG–RUEPPEL MESSAGE RECOVERY SIGNATURE)

У 1993 році Ніберг та Рюпель запропонували схему ЕЦП з відновленням повідомлення, що була заснована на проблемі дискретного логарифмування в полі Галуа [29, 184]. Вона дає перевагу при застосуванні з повідомленнями невеликого розміру. ЕЦП, розроблений за такою схемою, може ефективно використовуватися в інфраструктурах з відкритими ключами, у протоколах з малим розміром повідомлення, наприклад, електронних магазинах, а по суті для захисту товарів і послуг тощо. Нижче у скороченому вигляді наведено процедури обчислення передпідпису та підпису, а також перевіряння підпису для 6 алгоритмів ЦП з відновленням повідомлення, що подані спочатку в міжнародному стандарті ISO/IEC 15946-4, а потім у новій версії ISO/IEC 9796-3 [29, 184].

Основними складовими для обчислення підпису Ніберга-Рюпеля, як уже вказано в п. 4.2, 4.3, є процедури обчислення передпідпису та підпису, а також перевіряння підпису. Будемо вважати, що попередньо генеровані загальні параметри та асиметрична ключова пара – особистий ключ x та відкритий ключ Q, щодо яких забезпечена їх цілісність, справжність і доступність, а щодо особистого ключа ще й конфіденційність.

Передпідпис формується шляхом генерування випадково особистого ключа сеансу k, піднесення первісного елемента a до ступеня k за модулем P, а також перетворення елемента кінцевого поля R на октетові рядки $П$, тобто

$$k = rand([1, n-1]),$$
$$R = a^k \pmod P,$$
$$П = FE2OSP_f(R).$$

Підпис виконується відповідно до загальної схеми п. 4.2, тобто способом перетворення октетових рядків данних d та передпідпису $П$ на цілі числа δ та π, а також безпосереднього обчислення самого підпису – компонент (r, s), причому на завершувальному етапі r компонента повинна бути перетворена з октетових (байтових) рядків на ціле число. Указані операції виконуються в такій послідовності:

$$\delta = OS2IP(d), \qquad \delta \in [0, n-1],$$
$$\pi = OS2IP(П) \bmod n,$$
$$\tilde{r} = (\delta + \pi) \bmod n,$$
$$s = (k - x_A\tilde{r}) \bmod n,$$
$$r = OS2IP(\tilde{r}, L(n)).$$

Таким чином, вихідними даними за результатами обчислення цифрового підпису є пара цілих чисел (r, s). Якщо $s = 0$ або $r = 0$, то процес генерації підпису має бути повторений із новим значенням k.

Для перевірки підпису проводиться відновлення передпідпису та відновлення частини з r компоненти підпису. Детальний опис схеми перевірки підпису наведений у п. 4.3. Рішення щодо справжності підпису приймається після виконання

нижче наведених операцій перевіряння підпису. Спочатку отримана або така, що міститься при повідомленні, інформація відносно компоненти r' переводиться з октетових рядків у ціле число $\tilde{r}$. Потім R' обчислюється як елемент поля, причому R' переводиться з елемента поля спочатку в октетові рядки $П'$, а потім $П'$ переводиться із октетових рядків у ціле число π'. На завершення обчислені при перевірці з δ' дані d' також переводяться з цілого числа в октетові рядки.

$$\tilde{r}' = OS2IP(r'),$$
$$R' = P^{s'}Q^{\tilde{r}'} \pmod{P},$$
$$П' = FE2OSP_F(R'),$$
$$\pi' = OS2IP(П') \bmod n,$$
$$d' = I2OSP(\delta', L_{dat}).$$

Зіставлення підпису складається з перевіряння надлишковості d'.

4.5. ЦИФРОВИЙ ПІДПИС НІБЕРГА-РЮПЕЛЯ В ГРУПІ ТОЧОК ЕЛІПТИЧНИХ КРИВИХ (ELLIPTIC CURVE NYRBERG–RUEPPEL (ECNR) MESSAGE RECOVERY SIGNATURE)

Цифровий підпис ECNR засновано на схемі підпису, яку визначено в [184]. Розглянемо її детально за етапами, зважаючи на ґрунтовно визначену в 4.2–4.3 загальну модель схем цифрового підпису з відновленням повідомлення.

4.5.1. Ключі цифрового підпису ECNR

Асиметрична ключова пара цифрового підпису ECNR повинна формуватись таким чином:

1)спочатку генерується або вибирається особистий (довгостроковий) ключ цифрового підпису x_A об'єкта A, причому x_A – випадкове ціле число з інтервалу $[2, n-2]$;

2)обчислюється відкритий ключ перевірки цифрового підпису Y_A об'єкта згідно 4.1.6 (спосіб I або спосіб II).

4.5.2. Обчислення цифрового підпису

Вхідними даними до процесу обчислення підпису є такі:

– дійсні доменні параметри – $a, b, n, G, u, P(f(x)), h$;

– особистий ключ цифрового підпису x_A;

– вхідні дані d як елемент простого поля Галуа $F(n)$.

Дані d формуються з повідомлення. Виходом процесу обчислення підпису є пара чисел $(r, s) \in F(n) \times F(n)$, що є цифровим підписом A для даних.

Підпис даних d об'єкт A виконує в такій послідовності.

1. Генерація ключа сеансу й попереднього підпису:

а) обрати (генерувати) випадкове ціле k з інтервалу $[2, n-2]$;

б) обчислити точку еліптичної кривої $(x_1, y_1) = kP$, де P – базова точка порядку n.

2. Обчислення за модулем n порядку групи G:

а) визначити попередній підпис $\Pi \equiv \pi(kP) (\mathrm{mod}\, n) = x_1 (\mathrm{mod}\, n)$;

б) обчислити першу складову ЦП $r \equiv d + \Pi (\mathrm{mod}\, n)$;

в) обчислити другу складову ЦП $s \equiv k - x_A r (\mathrm{mod}\, n)$;

г) знищити особистий ключ сеансу k.

Якщо $s = 0$ або $r = 0$, то процес обчислення підпису має бути повторено з новим випадковим значенням k.

3. Форматування підписаного повідомлення:

Пара $(r, s) \in F(n) \times F(n)$ є підписом A для даних d.

4.5.3. Перевіряння цифрового підпису

Процес перевіряння цифрового підпису складається з таких трьох кроків:

Вхідними даними до процесу перевіряння підпису є такі:

– доменні параметри еліптичної кривої – $a, b, n, G, u, P(f(x)), h$;

– відкритий ключ перевіряння цифрового підпису Y_A об'єкта A;

– підпис для даних d, представлений двома цілими числами r' і s';

– невідновлюване повідомлення M'_{clr} (за наявності).

Для перевірки підпису об'єкта A для d об'єкт B повинен виконати такі кроки.

1. Зробити перевірку розміру підпису, тобто перевірити, що $0 < r' < n$ та $0 < s' < n$; якщо це не так, то відхилити підпис.

2. Відновлення попереднього (старого) підпису та вхідних даних (у групі точок еліптичної кривої):

1) Обчислити точку на ЕС $R_1' = s'P + r'Q$, де P – базова точка ЕС, а Q – відкритий ключ.

2) Обчислити попередній підпис $\Pi' \equiv \pi(R'_1) (\mathrm{mod}\, n) = x_1 (\mathrm{mod}\, n)$.

3. Відновлення вхідних даних або повідомлення.

4. Обчислення $d' \equiv r' - \Pi' (\mathrm{mod}\, n)$.

5. Зіставлення підпису.

Зіставлення надлишковості має здійснюватись за обчисленої надлишковості та відомої перевірнику. Якщо $d' = d$, то вважається, що повідомлення цілісне й справжнє, інакше робиться відхилення прийнятого повідомлення (підпису).

Необхідно відзначити, що в процесі обчислення та перевіряння підпису необхідно застосовувати відповідні функції перетворення та маски, наприклад, як у п. 4.4. Тоді обчислення передпідпису необхідно виконати із забезпеченням таких перетворень:

$$k = rand([1, n-1]),$$
$$R = kP,$$
$$\Pi = FC2OSP_E(R, compressed).$$

Обчислення цифрового підпису виконується із застосуванням таких функцій і перетворень:

$$\delta = OS2IP(d), \qquad \delta \in [0, n-1],$$
$$\pi = OS2IP(\Pi) \bmod n,$$
$$\tilde{r} = (\delta + \pi) \bmod n,$$
$$s = (k - x_A \tilde{r}) \bmod n,$$
$$r = OSP(\tilde{r}, L(n)).$$

Перевіряння цифрового підпису має бути виконане з дотриманням таких вимог:

$$\tilde{r}' = OS2IP(r'),$$
$$R' = s'P + \tilde{r}'Q,$$
$$\Pi = EC2OSP(R', compressed),$$
$$\delta' = (r' - \pi') \bmod n,$$
$$d' = I2OSP(\delta', L_{dat}).$$

4.6. ЦИФРОВИЙ ПІДПИС МІЯДЖІ З ВІДНОВЛЕННЯМ ПОВІДОМЛЕННЯ В ГРУПІ ТОЧОК ЕЛІПТИЧНОЇ КРИВОЇ (ELLIPTIC CURVE MIYAJI MESSAGE RECOVERY SIGNATURE (ECMR))

Цифровий підпис ECMR засновано на схемі підпису, яку визначено у [27, 29, 187]. Розглянемо її детально за етапами, визначеними в 4.2–4.3.

4.6.1. Доменні параметри та параметри користувача

Ключі для ECMR формуються таким чином:

1) генерується або вибирається особистий (довгостроковий) ключ підпису x_A об'єкта A, причому x_A – випадкове ціле число з інтервалу $[2, n-2]$;

2) відкритий ключ перевірки цифрового підпису Y_A об'єкта A обчислюється згідно з 4.1.6 (спосіб I або спосіб II).

Об'єкт A також вибирає геш-функцію:

$$Hash: E(F(q) \to \{0,1\}^{l_n}$$

4.6.2. Процес обчислення (генерації) підпису

Вхідні дані для процесу генерації підпису повинні містити:

– доменні параметри EC;

– особистий ключ цифрового підпису x_A;

– дані d з доданою або природною надлишковістю у вигляді $\{0,1\}^{l_n}$.

Дані d формуються з повідомлення [див. розділ 4.2]. Виходом процесу генерації підпису є пара $(r,s) \in \{0,1\}^{len-n} \times F(n)$, що є цифровим підписом A для даних.

Цифровий підпис даних d об'єкт A має виконувати в такій послідовності.

1. Генерація ключа сеансу й попереднього підпису:

а) обрати (генерувати) випадкове ціле k з інтервалу $[2, n-2]$;

б) Обчислити точку еліптичної кривої $(x_1, y_1) = kP$, де P – базова точка порядку n.

2. Обчислення за модулем порядку n групи G:

а) визначити передпідпис як $П = Hash(kP)$;

б) обчислити першу складову підпису $r = d \; XOR \; П$;

в) обчислити другу складову підпису $s = (rk - r - 1)/(x_A + 1) (\mathrm{mod}\, n)$;

г) знищити k.

Якщо $s = 0$ або $r = 0 \bmod n$, то процес генерації підпису повинен бути повторений з новим випадковим значенням k.

3. Форматування підписаного повідомлення:

Пара $(r, s) \in \{0,1\}^{len_n} \times F(n)$ є підписом об'єкта A для даних d.

4.6.3. Перевіряння підпису

Вхідні дані для процесу перевіряння підпису повинні містити:

– доменні параметри ЕК;

– відкритий ключ перевірки Y_A об'єкта A;

– одержаний підпис для d, представлений двома цілими числами r' та s';

– невідновлюване повідомлення M'_{clr} (за наявності).

1. Для перевірки підпису об'єкта A для d об'єкт B повинен виконати такі кроки:

а) перевіряння розміру підпису;

б) перевірити, що $r' \neq 0 \bmod n$ та $r' \in \{0,1\}^{len_n}$, та $0 < s' < n$. Якщо хоча б одна з умов не виконується, то відхилити підпис.

2. Відновлення попереднього підпису (у групі точок еліптичної кривої):

а) Обчислити $R_1' = \dfrac{1 + r' + s'}{r'} P + \dfrac{s'}{r'} Q$;

б) Обчислити $П' = Hash(R_1')$.

3. Відновлення вхідних даних або повідомлення:

а) обчислити $d' = r' \; XOR \; П'$;

б) зіставлення підпису.

Зіставлення надлишковості має здійснюватися за обчисленої надлишковості та відомої перевірнику. Якщо $d' = d$, то вважається, що повідомлення цілісне й справжнє, інакше робиться відхилення прийнятого повідомлення (підпису).

Таким чином, схема використовує функції маскування та гешування з однаковою довжиною та n-бітовою довжиною результату.

$$Mask: \{0,1\}^{8^*} \to \{0,1\}^{8L(n)},$$

$$Hash: \{0,1\}^{8^*} \to \{0,1\}^{8L(Hash)}$$

6. При обчисленні передпідпису $П$ необхідно виконати такі умови:

$$k = rand([1, n-1]),$$

$$R = kP,$$

$$П = Mask(EC2OSP_E(R, compressed)).$$

7. При обчисленні відновлюваного цифрового підпису дані маскуються перед-підписом за допомогою складання за модулем 2, а обчислення s необхідно виконати згідно з нижче наведеним:

$$r = d \oplus \Pi$$

$$s = (OS2IP(r)k - OS2IP(r) - 1/(x_A + 1) \bmod r$$

8. При перевірянні підпису необхідно виконати умови щодо представлення даних в ході обчислень:

$$R' = ((1 + OS2IP(r') + s')/OS2IP(r'))P + (s'/OS2IP(r'))Q$$

$$\Pi' = Mask(EC2OSP_E(R', uncompessed))$$

$$d' = r' \oplus \Pi$$

4.7. ЦИФРОВИЙ ПІДПИС АБІ-ОКАМОТО З ВІДНОВЛЕННЯМ ПОВІДОМЛЕННЯ В ГРУПІ ТОЧОК ЕЛІПТИЧНОЇ КРИВОЇ (ELLIPTIC CURVE ABE-OKAMOTO MESSAGE RECOVERY SIGNATURE (ECAO))

Цифровий підпис ECAO засновано на схемі підпису, яку визначено у [27, 29, 180]. Розглянемо її детально за етапами, визначеними в 4.2–4.3.

4.7.1. Доменні параметри та параметри користувача

Асиметрична пара ключів для ECAO формуються таким чином:

1) генерується або вибирається особистий (довгостроковий) ключ підпису x_A об'єкта A, причому x_A – випадкове ціле число з інтервалу $[2, n-2]$;

2) відкритий ключ перевірки цифрового підпису Y_A об'єкта A обчислюється згідно з 4.1.6 (спосіб I або спосіб II).

Об'єкт A також вибирає геш-функцію:

$Hash(\{0,1\}^* \to \{0,1\}^{len})$, де $len \geq \max(8L_2, 8L_F - 8L_2$, кількість бітів n) і застосовує три геш-функції, де $\{0,1\}^*$ – випадкові бітові дані.

$Hash_1(\{0,1\}^* \to \{0,1\}^{8L_2})$, $Hash_2(\{0,1\}^* \to \{0,1\}^{8L_F - 8L_2})$ та $Hash(\{0,1\}^* \to [1, n-2])$, використовуючи необхідну кількість бітів з виходу $Hash$.

Вище використано такі позначення:

$\{X\}^a$ a бітів X з найбільш значущого октету;

$\{X\}^b$ b бітів X з найменш значущого октету.

4.7.2. Обчислення цифрового підпису

Вхідні дані до процесу вироблення вхідних даних повинні містити:

– доменні параметри EC;

– повідомлення M_{rec}.

Вихідними даними процесу вироблення є дані d, що обчислені як

$$d = Hash_1(M_{rec}) \,|\, (Hash_2(Hash_1(M_{rec}) \;\; XOR \; M_{rec}).$$

Причому результуюче d знаходиться в діапазоні $0 \le d < 2^{8L_F}$.

Вхідні дані процесу обчислення (генерації) цифрового підпису містять:

– доменні параметри ЕС;

– особистий ключ підпису x_A;

– дані d;

– повідомлення M_{clr} (за наявності).

Виходом процесу генерації підпису є пара $(r,s) \in \{0,1\}^{8L_F} \times F(n)$, що є цифровим підписом A для даних.

Для підписання даних dA виконує такі кроки.

1. Генерація ключа сеансу та попереднього підпису:

а) обрати (генерувати) випадкове ціле k з інтервалу $[2, n-2]$;

б) обчислити точку еліптичної кривої kP та перетворити її на ціле $x_1 = \pi(kP)$.

2. Обчислення за модулем порядку групи G:

а) обчислити $r = d\ XOR\ [x_1]_{8L_F}$;

б) обчислити $\Pi = Hash_3(r \| M_{clr})$;

в) обчислити $s = k - x_A \Pi (\mathrm{mod}\, n)$;

г) знищити k.

Якщо $r = 0$ або $s = 0$, то процес генерації підпису має бути повторено з новим випадковим значенням k.

3. Форматування підписаного повідомлення:

Пара $(r,s) \in \{0,1\}^{8L_F} \times F(n)$ є підписом A для даних d.

4.7.3. Перевіряння підпису

Процес перевіряння підпису складається з таких кроків:

Вхідні дані процесу перевіряння підпису:

– доменні параметри ЕС;

– відкритий ключ перевірки Y_A об'єкта A;

– підпис для d представлений двома цілими числами r' і s';

– геш-функції $Hash_1$, $Hash_2$, $Hash_3$;

– невідновлюване повідомлення M'_{clr} (за наявності).

Для перевірки підпису об'єкта A для даних d об'єкт B повинен виконати таке:

– перевірити, що $r' \neq 0$, $r' \in \{0,1\}^{8L_F}$ та $0 < s' < n$; якщо хоча б одна з умов не виконується, то відхилити підпис.

2. Відновлення попереднього підпису та вхідних даних повинне здійснюватись способом виконання обчислень у групі точок еліптичної кривої з виконанням таких кроків:

а) обчислити $\Pi' = Hash_3(r' \| M'_{clr})$;

б) обчислити точку еліптичної кривої $x'_1 = \pi(s'P + \Pi' Q)_{(Q=Y, P=G)}$;

в) обчислити $d' = r' XOR [x'_1]_{8L_F}$.

3. Відновлення вхідних даних або повідомлення.

4. Обчислити $M'_{rec} = [d']_{8L_F - 8L_2}\ XOR\ Hash_2([d']^{8L_2})$.

5. Зіставлення підпису.

6. Перевірити рівняння: $[d']^{8L_2} = Hash_1(M'_{rec})$. Якщо рівняння виконується, то вивести $M'_{rec} \| M'_{clr}$, інакше відхилити підпис.

4.8. ЦИФРОВИЙ ПІДПИС ПІНТСОВА-ВАНСТОНА В ГРУПІ ТОЧОК ЕЛІПТИЧНИХ КРИВИХ (ELLIPTIC CURVE PINTSOV-VANSTONE (ECPV) MESSAGE RECOVERY SIGNATURE)

Схема підпису Пінтсова-Ванстона із частковим відновленням повідомлення є варіантом схеми Шнора [27, 29, 185, 186] та підпису Ніберга-Рюпеля. Це дає можливість формувати дуже малі підписи для повідомлень із збитковістю. Для 80 бітів розмір підпису містить 20–30 байтів, залежно від кількості збитковості в повідомленні. (Для порівняння: підпис ECDSA із такими самими параметрами домену має мінімальний розмір не менше 40 байтів). У схемі використовується блоковий симетричний шифр. Можуть використовуватись різні стандарти БСШ, наприклад AES, 3DES, а в цілому ISO/IEC 18033-3 тощо, а також інші, що задовольняють вимогам та допускаються до використання національним законодавством.

Необхідно також відзначити, що деякі схеми підпису з повним або частковим відновленням повідомлення мають обмеження на довжину. Наприклад, у схемі Nyberg-Rueppel це обмеження визначає такі недоліки:

1) для дуже короткого повідомлення примусово визначена довжина підпису вимагає використовувати більше доповнення;

2) для повідомлення, що є більшим, ніж означена довжина підпису, неможливо забезпечити повне покриття підписом тощо.

Схема Пінтсова-Ванстона може використовуватися без таких обмежень. При використанні схеми з еліптичними кривими може бути отриманий найкращий результат з точки зору меншого розміру підпису.

Цифровий підпис ECPV засновано на схемі підпису, яку визначено у [8]. Розглянемо її детально за етапами, визначеними в 4.2–4.3.

4.8.1. Доменні параметри та параметри користувача

Ключі для ECPV формуються таким чином:

1) генерується або вибирається особистий (довгостроковий) ключ підпису x_A об'єкта A, причому x_A – випадкове ціле число з інтервалу $[2, n-2]$;

2) відкритий ключ перевірки цифрового підпису Y_A об'єкта A обчислюється згідно 4.1.6 (спосіб I або спосіб II).

Взаємодіючі сторони також повинні узгоджено використовувати стандарти геш-функції $Hash$, симетричного шифру SYM та функції формування ключів KDF, що використовуються в схемі, причому:

$$Hash: \{0,1\}^* \to [1, n-1];$$
$$SYM: \{0,1\}^* \to [0,1]^{l_n};$$
$$KDF: \{0,1\}^* \times \{0,1\}^{l_n} \to [0,1].$$

Розмір симетричного ключа повинен відповідати параметру безпеки і має обиратися для досягнення цілей безпеки, наприклад, забезпечувати задовільний, добрий чи високий рівень стійкості [111]. Симетричний шифр може бути потоковим, тобто використовувати потокове шифрування з операцією XOR (за модулем 2). А функція формування ключів може ґрунтуватись на використанні геш-функції.

4.8.2. Обчислення цифрового підпису

Вхідні дані до процесу вироблення вхідних даних повинні містити:

– доменні параметри ЕС;

– особистий ключ підпису x_A;

– повідомлення M, що має бути підписане;

– M розщеплюється на відновлювану частину M_{rec} (крайні октети M згідно з угодою (умовою, домовленістю), та M_{clr} з використанням тих октетів, що залишилися;

– довжини обох повідомлень M_{rec} і M_{clr} мають бути зафіксовані й узгоджені сторонами, що взаємодіють.

Результатом обчислення підпису є пара $(r,s)\in(0,1)^* \times F(n)$, що є цифровим підписом A для повідомлення M.

Для підписання повідомлення M користувач A виконує такі кроки:

1. Генерація ключа сеансу й обчислення попереднього підпису:

а) обрати (згенерувати) випадкове ціле k з інтервалу $[2, n-2]$;

б) обчислити точку еліптичної кривої $R = (x_1, y_1) = kP$ та перетворити її на ціле число.

2. Обчислення вхідних даних та ключа сеансу:

а) сформувати вхідні дані d, використовуючи M_{rec} і додаткову надлишковість згідно з угодою, а також частину доменних параметрів;

б) обчислити симетричний ключ $\sigma = KDF(R)$;

в) зашифрувати дані $r = SYM(d,\sigma)$;

г) обчислити передпідпис $П = Hash(r \| M_{clr})$.

3. Обчислення цифрового підпису:

а) обчислити $s \equiv k - x_A П \pmod n$;

б) знищити k;

в) як результат вивести підпис (r,s) і частину повідомлення M_{clr} (що може бути нульовою).

Якщо $s = 0$, то процес генерації підпису має бути повторено з новим випадковим значенням k.

4. Форматування підписаного повідомлення:

– пара $(r,s)\in(0,1)^* \times F(n)$ є цифровим підписом користувача A для повідомлення M.

4.8.3. Перевіряння цифрового підпису

Вхідні дані до процесу перевіряння підпису повинні містити:

– доменні параметри, що зазначені в п. 4.8.1.

– відкритий ключ перевірки Y_A об'єкта A;

– підпис для M, представлений двома цілими числами, r' і s';

– невідновлюване повідомлення M'_{clr} (за наявності).

Для перевірки підпису об'єктом A для d об'єкт B виконує такі кроки.

1. Перевіряння розміру підпису:

– перевірити умову $0 < s' < n$; якщо вона не виконується, відхилити підпис.

2. Відновлення попереднього підпису та вхідних даних (обчислення в групі точок еліптичної кривої):

а) обчислити передпідпис $\Pi' = Hash(r' \| M'_{clr})$;

б) обчислити точку ЕС $R'_1 = s'P + \Pi'Q$ $_{(Q=Y,\ P=G)}$.

3. Відновлення вхідних даних або повідомлення:

а) якщо R'_1 є точкою нескінченності, відхилити підпис. Інакше виконати такі кроки;

б) обчислити симетричний ключ $\sigma' = KDF(R_1')$;

в) розшифрувати дані $d' = SYM^{-1}(r', \sigma')$;

г) перевірити додану надлишковість d' та відновити M'_{rec}. Якщо додана надлишковість неправильна, відхилити підпис;

д) сформувати вихідне повідомлення $M' = M'_{rec} \| M_{clr}$.

4. Зіставлення підпису.

Зіставлення надлишковості має здійснюватись за обчисленої надлишковості та відомої перевірнику. Якщо $d' = d$, то вважається що повідомлення цілісне й справжнє, в іншому випадку робиться відхилення прийнятого повідомлення (підпису).

4.9. ЦИФРОВИЙ ПІДПИС KCDSA У ГРУПІ ТОЧОК ЕЛІПТИЧНОЇ КРИВОЇ (ELLIPTIC CURVE KCDSA/NURBERG–RUEPPEL (ECKNR)MESSAGE RECOVERY SIGNATURE)

Цифровий підпис ECKNR засновано на схемі підпису, яку визначено в [27, 29, 184]. Розглянемо її детально за етапами, визначеними в 4.1–4.3.

4.9.1. Доменні параметри та параметри користувача

Ключі для ECKNR формуються таким чином:

1) генерується або вибирається особистий (довгостроковий) ключ підпису x_A об'єкта A, причому x_A – випадкове ціле число з інтервалу $[2, n-2]$;

2) відкритий ключ перевірки цифрового підпису $Y_A = (x_0, y_0)$ об'єкта A обчислюється згідно з 4.1.6 (спосіб I або спосіб II).

Сторона A також використовує функцію генерації маски MGL.

В ЕЦП ECKNR використовуються такі параметри користувача:

– геш-значення z_A даних сертифікації об'єкта A, тобто $z_A = Hash_2$ (*Cert_Data*). Ці дані сертифікації вважаються відкритою інформацією і мають бути доступними всім сторонам, що залучені до підписання й перевіряння повідомлення:

– під *Cert_Data* розуміють дані сертифікації A, які щонайменше містять відкритий ключ перевірки Y_A об'єкта A, а також можуть містити розпізнавальний ідентифікатор A або деякі доменні параметри. В Україні вимоги до сертифікату відкритого ключа містяться в [6];

– від геш-функції $Hash_2$ не вимагається стійкості до колізій. Найпростішою реалізацією є використання об'єктом A як відкритого ключа перевірки

Cert_Data. *Cert_Data* може бути обрано як $Cert\_Data = \pi(x_0,0) + \pi(y_0,0)$, де відкритий ключ перевірки A є $Y_A = (x_0, y_0)$.

Сторони, що взаємодіють, також повинні узгоджувати функцію гешування $Hash_1 E(F(q)) \to [1, n-1]$.

4.9.2. Обчислення цифрового підпису

Вхідні дані до процесу обчислення цифрового підпису повинні містити:

– доменні параметри згідно 4.9.1;

– особистий ключ підпису x_A об'єкта A;

– геш-значення даних сертифікації z_A об'єкта A;

– дані d з доданою або природною надлишковістю $\{0,1\}^{l_n}$;

– невідновлюване повідомлення M_{clr} (за наявності).

Дані d формуються з повідомлення згідно з 4.2. Виходом процесу генерації підпису є пара $(r, s) \in \{0,1\}^{l_n} \times F(n)$, що є цифровим підписом об'єкта A для підписуваних даних.

1. Генерація ключа сеансу та попереднього підпису:

а) обрати (генерувати) випадкове ціле k з інтервалу $[2, n-2]$;

б) обчислити точку еліптичної кривої $(x_1, y_1) = kP$;

в) обчислити геш-значення від (x_1, y_1) за узгодженим правилом і прийняти його як передпідпис $П = Hash_1(kP)$.

2. Обчислення за модулем порядку групи G (арифметичні дії в полі $F(n)$):

а) обчислити $r = d \; XOR \; П \; Hash_1(z_A \| M_{clr})$;

б) обчислити $s \equiv k - x_A r \pmod n$;

в) знищити k.

Якщо $s = 0$ або $r = 0 \bmod n$, то процес генерації підпису має бути повторено з новим випадковим значенням k.

3. Форматування підписаного повідомлення:

– пара $(r, s) \in (0,1)^{l_n} \times F(n)$ є підписом A для даних d.

4.9.3. Перевіряння цифрового підпису

Процес перевіряння підпису складається з трьох кроків:

1) обчислення дайджесту повідомлення;

2) обчислення в групі точок еліптичної кривої;

3) зіставлення підпису.

Вхідні дані до процесу перевіряння підпису повинні містити:

– доменні параметри згідно з 4.9.1;

– відкритий ключ перевірки Y_A об'єкта A;

– геш-значення даних сертифікації z_A об'єкта A;

– підпис для даних d, що представлений двома цілими числами, r' і s';

– невідновлюване повідомлення M'_{clr} (за наявності).

1. Перевіряння розміру підпису:

– перевірити умови $r' \neq 0 \bmod n$ та $r' \in \{0,1\}^{ln}$, та $0 < s' < n$; якщо вони не виконуються, відхилити підпис.

2. Відновлення попереднього підпису та вхідних даних (обчислення в групі точок еліптичної кривої):

а) обчислити точку еліптичної кривої $R' = s'P + r'Q_{(Q=Y,\ P=G)}$;

б) обчислити з використанням конкатенації координат (x_R, y_R) $П' = Hash_1(R')$.

3. Відновлення вхідних даних або повідомлення:

Для відновлення даних обчислити $d' = r' XOR\ П'\ XOR\ Hash_1(z_A \| M'_{clr})$.

Зіставлення підпису:

– зіставлення надлишковості має здійснюватись на основі обчисленої надлишковості та відомої перевірнику. Якщо $d' = d$, то вважається, що повідомлення цілісне й справжнє, в іншому випадку робиться відхилення прийнятого повідомлення (підпису).

4.10. ПРИКЛАДИ ВИКОРИСТАННЯ ЕЦП З ВІДНОВЛЕННЯМ ПОВІДОМЛЕННЯ

4.10.1. Поштові марки

Поштові марки мають містити поштові дані з розміром у межах від 20 до 50 байтів. Деякі частини поштових даних, включаючи дату і поштовий код відправника, відправляються в чистому вигляді. Інші частини даних, такі, як серійний номер повідомлення, поштова адреса відправника, включаються до частини, що буде відновлена. Мінімально ця інформація займає від 13 байтів даних. Натуральна збитковість може бути на рівні 7 байтів. Щоб отримати 10 байтів збитковості, 3 байти збитковості вирівнюють. Таким чином, відновлювана частина буде містити 16 байтів.

Розробник рекомендує використовувати в таких випадках 20-байтову еліптичну криву (160–163 біти), SHA-1 як 20-байтову функції гешування та 3DES. Таким чином, інша частина підпису буде займати 20 байтів і 3 байти буде додано до збитковості. Сумарне перебільшення складе 23 байти за рівня безпеки 2^{-80}.

4.10.2. Підписування дуже короткого повідомлення

Розглянемо підписування малого повідомлення довжиною в 1 байт, таке як: так/ ні, придбати/ продати/ затримати тощо. Для того щоб запобігти атакам повтору, до таких коротких повідомлень треба додати номер послідовності у розмірі від 3-х байтів. У разі такого використання, треба збільшити стійкість до підробки. Для цього додається вирівнювання розміром 4 байти. Таким чином, отримуємо частку повідомлення у 8 байтів. Із DES, SHA-1 та 20-байтовою еліптичною кривою підпис буде мати 28 байтів, 24 з яких є криптографічним надлишком. Номер послідовності є необхідним елементом і таким чином буде природною збитковістю. Тому надлишок складає 7 байтів, що дає 2^{-56} стійкість до екзистенційної підробки (повна стійкість складає 2^{-80}).

4.10.3. Підписування та відновлення повідомлень із надлишком у 20 байтів

Якщо повідомлення, що має бути відновлене, довше ніж 20 байт, можна розраховувати на те, що деякі вимоги до форматування повідомлення становлять щонайменше 10 байтів натуральної збитковості. Тоді надлишок складе 20 байтів – друга частина підпису. У гіршому випадку, якщо натуральна збитковість відсутня, можна додати 10 байтів збитковості.

4.11. СТІЙКІСТЬ ЕЦП ECPV (ПІНТСОВА-ВАНСТОНА)

Стійкість ЕЦП з додатком розглянемо на прикладі ECPV (Пінтсова-Ванстона). Її зумовлюють такі компоненти, як стійкість перетворень у групі точок еліптичної кривої, стійкість функції гешування, стійкість БСШ та величина збитковості. Методика порівняльного аналізу наведена у [186].

Окрім того, стійкість ECPV залежить від незалежності цих чотирьох компонент. Наприклад, функція гешування не повинна бути означеною у термінах групи точок еліптичної кривої, і множина збитковості не повинна включати в себе множину всіх M_{clr}, таких що $S_{V'}(M_{clr}) = r$ для деяких фіксованих r.

Заради швидкості деякі реалізації можуть використовувати відсікання або доповнення замість безпечнішої функції встановлення ключа на базі функції гешування. Це не впливає на загальну безпечність підпису.

Для більш точного визначення правил використання підпису розглянемо декілька припущень.

Припустимо, що f – зловмисник, який здійснює підробку. Із значною ймовірністю ми можемо припустити, що f може опитувати $H()$ та $S()$. Ґрунтуючись на порядку виконання, ми можемо отримати декілька варіантів.

Спочатку виконується гешування $H(r \parallel M_{1rec})$, потім $S'_V(s)$.

Оскільки $S_V^{-1}(r) = d$, ми маємо $S'_V(d) = r$. Але значення $S'_V(d)$ отримується випадково, таким чином ймовірність того, що $S'_V(d) = r$, є незначною:

1. $H(r \parallel M_{clr}) \rightarrow S_{V'}^{-1}(r)$. У цьому випадку $d = S_{V'}^{-1}(r)$ повинна обиратися випадково, таким чином, імовірність того, що $s \in N$ є 2^{a-b}.

2. $S_{V'}^{-1}(r) \rightarrow H(r \parallel M_{clr})$ та $S_{V'}^{-1}(s) \rightarrow H(r \parallel M_{clr})$. Використовується лема Понтчеваля та Штерна [9, 11]. Випадково обирається індекс t, та f застосовується два рази, але t змінює випадкове значення H, що було отримане f. Оскільки загальна кількість опитування є біноміальною, існує значний шанс, що t запитання $H()$ буде $H(r \parallel M_{clr})$ в обох випадках. Якщо h та h' – випадкові значення, отримані від $H()$ у цьому випадку, та (r, s) і (r', s'), тоді $dG - hW = V = d'G - h'W$, оскільки значення V отримано f, як і в першому опитуванні. Оскільки $sG = W$ та $(h - h')W = (d - d')G$, то $s = (h - h')^{-1}(d - d') \bmod r$.

Таким чином, якщо f може досягати успіху досить часто, буде можливість використати один із означених варіантів. Тому необхідно, щоб $S()$ та $H()$ були ідеальними.

Сильна геш-функція

Нехай $H()$ буде функцією гешування. $H()$ є сильною функцією гешування, якщо не існує поліноміального за часом алгоритму $A()$, який спочатку знайде значення h або l_0, а потім випадкове вхідне значення c та деяке l – таке, що $H(c \| l) = h$, або $H(c \| l) = H(c \| l_0)$ з великою ймовірністю.

Спеціальні функції

Схема використовує функції маскування та гешування з однаковою довжиною та n-бітовою довжиною результату.

$$Hash: \{0,1\}^{8*} \rightarrow \{0,1\}^{8L_{red-1}}$$

$$MGF: \{0,1\}^{8*} \rightarrow \{0,1\}^{8(L_{key})}$$

$$Sym: \{0,1\}^{8*} \times \{0,1\}^{8L_{key}} \rightarrow \{0,1\}^{8*}$$

Схема визначає свій спосіб формування відновлюваної частини:

$$C_{red} = Oct(L_{red})$$

$$d = \tilde{C}_{red} \| M_{rec}$$

У підписі бере участь функція $Sym()$ симетричного шифрування як функція маскування в стандартній схемі Ніберга-Рюпеля:

$$r = Sym(d, \Pi)$$

$$u = Hash(r \| M_{clr})$$

$$t = OS2IP(u)$$

$$s = (k - x_a t) \bmod n$$

Перевірка:

$$u' = Hash(r' \| M'_{clr})$$

$$t' = OS2IP(u')$$

$$R' = s'P + t'Q = (x', y')$$

$$\Pi' = KDF(FE2OSP_f(x'))$$

$$d' = Sym^{-1}(r', \Pi')$$

$$C_{red} = Oct(L_{red})$$

$$\tilde{C}_{red} = [d']^{8L_{red}}$$

$$M'_{rec} = [d']^{8(L(d') - L_{red})}$$

4.12. СТІЙКІСТЬ ЕЦП З ВІДНОВЛЕННЯМ ПОВІДОМЛЕНЬ ДО КОЛІЗІЙ

За основу досліджень колізійної стійкості, щодо захисту від підробок, було взято ідеї з [181, 186]. У табл. 4.2 наведені загальні показники ЕЦП з відновленням повідомлень.

Таблиця 4.2. **Показники ЕЦП**

	$L(d)$	$L(H^*(M_{rec}))$	$L(H^*(M_{clr}))$	(r, s)
NR	$L(n)-1$	d: $L(d)-L(M_{rec})$		$\{0,1\}^{8L(n)}\times[1, n-1]$
ECNR	$L(n)-1$	d: $L(d)-L(M_{rec})$		$\{0,1\}^{8L(n)}\times[1, n-1]$
ECMR	L(n): $L(\{0,1\}^{8(2L+1)})$	d: $L(d)-L(M_{rec})$		$\{0,1\}^{8L(n)}\times[1, n-1]$
ECAO	$L(\{0,1\}^{8(L_F+1)})$	$L(M_{rec})_1+[L(d)-M_{rec}]_2$	$L(n)\times[1, n]$	$\{0,1\}^{8(L_F+1)}\times[1, n-1]$
ECPV	*	*	$L(n)\times[1, n]$	$\{0,1\}^{8^*}\times[1, n-1]$
ECKNR	$L(\{0,1\}^{8L(n)})$	d: $L(d)-L(M_{rec})$	$L(d)-L(M_{rec})+L(n)$	$\{0,1\}^{8L(n)}\times[1, n-1]$

Модель компоненти захисту від підробки

Для захисту від підробки використовується код виробу, що складається з коду групи товарів і серійного номеру виробу.

Будемо вважати, що серійний номер виробу є унікальним і не містить природної надлишковості. Код групи товарів навпаки – має деяку природну надлишковість.

Код виробу вкладається в частину відновлюваного підпису.

Опціонально може бути включена додаткова відкрита інформація (можливо, як частина параметрів домену виробника), що буде розглядатися як відкрита частина повідомлення.

4.13. СТІЙКІСТЬ ДО КОЛІЗІЙ ВІДНОВЛЮВАНОЇ ЧАСТИНИ ПОВІДОМЛЕННЯ

Розглянемо можливість виникнення колізій компонентів означених моделей і граничну кількість одиниць товару, що можна маркувати за допомогою означеної моделі.

Імовірність колізії для підписів з відновленням повідомлення серед ω продуктів, з точки зору всього підпису, можна оцінити як [95, 96, 188]

$$P(\omega,(r,s))=1-e^{-\omega\frac{\omega-1}{2(2^{8L(r)+8L(s)}-1)}} \qquad (4.1)$$

Але при перевірці підпису використовується також семантичний критерій остаточного вирішення питання справжності (дійсності) підпису. Для зменшення ймовірності підтвердження та відновлення некоректного повідомлення використовується збільшення надлишковості повідомлення, пов'язаного із гешуванням відновлюваної частини повідомлення.

Максимальна довжина відновлюваної частини має величину $L(n) - 1$. Для скінченного поля нині дозволяється використовувати довжину $n \in \{2048, 3072\}$. Для групи точок еліптичних кривих $n \in 160, 163, ...431, ...$ Оскільки рішення щодо справжності підпису приймається після перевірки надлишковості, стійкість усієї схеми залежить від її розміру. Тому максимальна кількість корисної інформації, що може бути відновлена з підпису, зменшується на необхідний розмір надлишковості.

Для ЕЦП NR, ECMR, ECKNR імовірність колізії P кодів, вироблених з ключем у n бітів у партії з ω виробів, згідно парадоксу про день народження, зі збільшенням розміру корисного повідомлення $8L(M_{rec})$ може бути визначена як [188]:

$$P(M_{rec}, \omega, n) = 1 - e^{-\omega \frac{\omega - 1}{2(2^{8L(n) - 8L(M_{rec})} - 1)}} \tag{4.2}$$

Формула (4.2) також дозволяє обчислити ймовірність колізії у відкритій частині тексту.

Хоча в підпису ECPV умовно немає залежності ймовірності колізії відновлюваного повідомлення від параметрів ЕС, у реальних умовах вираз (4.2) набуває вигляду:

$$P(M_{rec}, \omega, Sym_{glen}) = 1 - e^{-\omega \frac{\omega - 1}{2(2^{8L(Sym_{glen}) - 8L(M_{rec})} - 1)}} \tag{4.3}$$

де Sym_{glen} – максимальний вихід блоку чи гамми симетричного шифру в октетах.

Для ECAO вираз (4.3) правильний з деякими уточненнями:

$$P(M_{rec}, \omega, L_F) = 1 - e^{-\omega \frac{\omega - 1}{2(2^{8L_F) - 8L(M_{rec}) + 8L(M_{rec})} - 1)}} \tag{4.4}$$

де L_F – розмір поля, над яким будується еліптична крива.

Вирази (4.2), (4.3), (4.4) можуть бути трансформовані для визначення максимальної кількості товарів, що можна маркувати ЕЦП із заданими параметрами:

$$\omega(P, M_{rec}, n) = \sqrt{2\left(2^{8L(n) - 8L(M_{rec})} - 1\right)\ln\left(\frac{1}{1 - P}\right)} \tag{4.5}$$

$$\omega(P, M_{rec}, Sym_{glen}) = \sqrt{2\left(2^{8L(Sym_{glen}) - 8L(M_{rec})} - 1\right)\ln\left(\frac{1}{1 - P}\right)} \tag{4.6}$$

$$\omega(P, M_{rec}, L_F) = \sqrt{2\left(2^{8L_F - 8L(M_{rec}) + 8L(M_{rec})} - 1\right)\ln\left(\frac{1}{1-P}\right)}, \tag{4.7}$$

Тепер підрахуємо дійсну кількість товару, яку можна маркувати із зазначеною схемою. Для того введемо додаткову умову

$$L(M_{rec}) = L(\omega).$$

Із уточненнями рівняння буде мати вигляд:

$$\begin{cases} \omega(P, M_{rec}, n) = \sqrt{2\left(2^{8L(n) - 8L(M_{rec})} - 1\right)\ln\left(\frac{1}{1-P}\right)} \\ L(M_{rec}) = L(\omega) \end{cases} \tag{4.8}$$

Можна побачити, що довжина $M_{rec} = 37$ забезпечить найбільш повне використання простору значень нашої моделі, коли ймовірність колізії $P = 2^{-53}$, та $n = 163$. Якщо взяти 7 бітів як код групи товарів (приблизно 100 груп, із надлишковістю), то загальна кількість товарів, що може бути випущена у межах групи, дорівнюватиме $2^{30} = 1073741824$.

Підпис ECAO використовує декілька гешувань відновлюваної частини повідомлення. Одне з них включається до геш-токену, інше складається за модулем 2 із повідомленням. Таким чином, можна казати, що простір можливих значень об'єднується, де X – імовірність колізії; Y – кількість бітів корисної інформації, що може бути відновлено; Z – кількість підписів, що можна безпечно формувати.

Для підпису NR імовірність колізії суттєво менша, оскільки розмір n має не менше 2048 бітів.

Підпис ECPV не накладає обмежень на сумарну довжину повідомлення із геш-токеном. Таким чином, можна казати, що ECPV дозволяє встановити довільно малу ймовірність колізії геш-токену.

Стійкість до колізій невідновлюваної частини повідомлення

ЕЦП із відновленням повідомлення гарантує цілісність і неспростовність не тільки відновлюваної частини повідомлення, але й усього повідомлення взагалі. У [27–29] стверджується, що для підписів NR, ECNR, ECMR імовірність до колізії відкритої частини повідомлення така ж сама, як і для закритої (4.2), (4.3), (4.4). Для ECAO, ECPV імовірність колізії відкритої частини для ω підписів складе:

$$P(\omega, n) = 1 - e^{-\omega\frac{\omega-1}{2(2^{8L(n)}-1)}}. \tag{4.9}$$

4.14. АНАЛІЗ СКЛАДНОСТІ РЕАЛІЗАЦІЇ ЕЦП З ВІДНОВЛЕННЯМ ПОВІДОМЛЕННЯ

У таблиці 4.3 наведено значення числа операцій, що необхідно виконати для алгоритмів обчислення ЕЦП, які входять до [27, 186].

Таблиця 4.3. **Число операцій, що виконуються при обчисленні підпису**

	NR	ECRN	ECMR	ECAO	ECPV	ECKNR
Складання за модулем n	2	2	3	1	1	1
Множення за модулем n	1	1	2	1	1	1
Інверсія за модулем n	0	0	1	0	0	0
Скалярне множення в групі точок ЕК, або підведення до ступеня в скінченному полі	1	1	1	1	1	1
Сума за модулем 2	0	0	1	2	0	2
Гешування	1	1	1 (або 0)	2	1	0
MGF/KDF	0	0	0 (або 1)	1	1	2
БСШ	0	0	0	0	1	0

У таблиці 4.4 наведено значення числа операцій, що необхідно виконати для алгоритмів при перевірці ЕЦП.

Таблиця 4.4. **Число операцій, що виконуються при перевірці підпису**

	NR	ECRN	ECMR	ECAO	ECPV	ECKNR
Складання за модулем n	1	1	2	0	0	0
Множення за модулем n	0	0	2	0	0	0
Інверсія за модулем n	0	0	1	0	0	0
Складання на еліптичній кривій або множення у скінченному полі	1	1	1	1	1	1
Скалярне множення в групі точок ЕК, або підведення до ступеня в скінченному полі	2	2	2	2	2	2
Сума за модулем 2	0	0	1	2	0	2
Гешування	1	1	1 (або 0)	2	1	0
MGF/KDF	0	0	0 (або 1)	1	1	2
БСШ	0	0	0	0	1	0

У таблиці 4.5 наведено значення числа операцій, що необхідно виконати для алгоритмів додавання та подвоєння в різних базисах.

Таблиця 4.5. **Складність додавання та подвоєння в різних базисах**

Координати	Додавання точок	Подвоєння точок
Афінні	$t(A+A) = I + 2M + S$	$t(2A) = I + 2M + 2S$
Проективні	$t(P+P) = 12M + 2S$	$t(2P) = 7M + 5S$
Якобіанові	$t(y+y) = 12M + 4S$	$t(2I) = 4M + 6S$
Чудновського	$t(y^c + y^c) = 11M + 3S$	$t(2I^c) = 5M + 6S$
Модифіковані якобіанові	$t(y^m + y^m) = 13M + 6S$	$t(2I^m) = 4M + 4S$

Необхідно враховувати, що швидкість функції гешування приблизно в десять разів більша, ніж інверсії.

4.15. ЧИСЛО ГЕШУВАНЬ MGF

Стандарт ISO/IEC 9796-3 визначає алгоритми формування гамми MGF_1 та MGF_2, як:

$$MGF_1(x, l) = [Hash(x \| I2OSP(0,4)) \| \| Hash(x \| I2OSP(1,4)) \| \| \ldots \| \| Hash(x \| I2OSP(k-1,4))]^{8l}; \quad (4.10)$$

$$MGF_2(x, l) = [Hash(x \| I2OSP(1,4)) \| \| Hash(x \| I2OSP(2,4)) \| \| \ldots \| \| Hash(x \| I2OSP(k,4))]^{8l}, \quad (4.11)$$

де $k = l/L_{Hash}$.

Таким чином, для кожного виконання функцій MGF потрібне обчислення k функцій гешування.

У таблиці 4.6 наведено експериментально отримані значення швидкості підписів алгоритмів ЕЦП стандарту ISO/IEC 9796-3 [29].

Таблиця 4.6. **Швидкість підписів для алгоритмів стандарту ISO/IEC 9796-3**

	Підпис (100/с)	Перевірка (100/с)
NR	3,78	7,78
ECNR	10,61	21,54
ECMR	11,13	20,51
ECAO	10,91	21,25
ECPV	9,61	17,54
ECKNR	12,61	20,54

У таблиці 4.7 наведені дані щодо обсягу пам'яті, що необхідна для реалізації алгоритмів ISO/IEC 9796-3.

Таблиця 4.7. **Обсяг пам'яті, що необхідний для реалізації алгоритмів ISO/IEC 9796-3**

	Підпис/100	Перевірка/100
NR	170514864	190513864
ECNR	1274718104	2610783160
ECMR	1282372192	2506140400
ECAO	1284437920	2603164096
ECPV	1074718104	1610783160
ECKNR	1274718204	1810783460

Висновки та рекомендації

1. Існують два типи підписів: з відновленням повідомлення та додатком. На цей час абсолютно поширеним типом є підпис з додатком повідомлення.

2. Сьогодні в усіх сферах діяльності широкого розповсюдження набули мобільні обчислювальні та комунікаційні системи, пристрої особистої ідентифікації. Такі системи характеризуються низькою вартістю виготовлення, невеликою обчислювальною здатністю та невеликими об'ємами пам'яті. Для забезпечення цілісності й достовірності інформації за допомогою цих систем доцільно використовувати спеціалізовані криптографічні алгоритми, що визначені в ISO/IEC 9796-3.

3. Підпис з відновленням повідомлення, порівняно з підписом з додатком, надає додаткову послугу безпеки – конфіденційність. Також для невеликих обсягів повідомлення можливо зробити таємною всю інформацію, що передається, у самому підписі.

4. Підписи з відновленням повідомлення стандартизовані в міжнародних стандартах ISO/IEC 15946-4, а потім ISO/IEC 9796-3. Стандарт ISO/IEC 9796-3 поширює й уточнює алгоритми, що вказані в ISO/IEC 15946-4, та з 2008 року є основним стандартом підписів з відновленням повідомлення.

5. Стандарт ISO/IEC 9796-3 містить 5 підписів у групі точок ЕК, та 1 у скінченному полі. Підписи мають спільну загальну схему Ніберга-Рюпеля, але в них використовують для оптимального використання r-компоненти модифікованого алгоритму передпідпису.

6. Найбільш перспективними є підписи ECPV та ECNR. ECNR з модифікаціями, по суті, є національним стандартом України ДСТУ 4145:2002. ECPV є перспективним підписом, що використовує симетричне шифрування для включення інформації до підпису і не накладає обмежень на кількість інформації, що може бути відновлена. ECPV також є підписом з найменшою довжиною.

7. На міжнародному рівні розглядається можливість використання ECPV та ECNR у RFID-чипах для захисту товарів від підробок і для маркувань медикаментів у Індії.

8. Особливо необхідно відзначити, що, як випливає з таблиці 4.6, усі підписи з відновленням повідомлення є асиметричними з точки зору складності обчислення й перевіряння ЕЦП. Указане має бути враховано при обчисленні та перевірянні ЕЦП в реальному масштабі часу.