МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему:

«Особливості використання колективного електронного підпису»

Завідувач
випускаючої кафедри Любчак О. В.

Керівник роботи Страх О. П.

Студент групи КБ –81 Щенякін Д. О.

СУМИ – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

Затверджую_____

Зав. кафедрою Любчак В. О.

«_____» _____2022 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи КБ-81 спеціальності «Кібербезпека» денної форми навчання Щенякіна Дениса Олеговича.

Тема: «Особливості використання колективного електронного підпису»

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Дата видачі завдання « _____ »_____ 2022 р.

Керівник випускної роботи _____ Страх О. П.

Завдання прийняв до виконання _____ Щенякін Д. О.

Зміст пояснювальної записки: 1) аналіз об’єктно-предметної області дослідження; 2) розкриття сутності колективного цифрового підпису; 3) вивчення сучасних сфер застосування колективних цифрових підписів; 4) розгляд можливості побудови власної моделі.

РЕФЕРАТ

Записка: 61 стор., 14 рис., 1 табл., 22 джерела.

Об'єкт дослідження — електронний цифровий підпис.

Предмет дослідження – особливості використання електронного цифрового підпису.

Мета роботи — провести аналіз особливостей використання електронного цифрового підпису.

Методи дослідження — метод аналітичного огляду, систематизація та узагальнення, дедуктивний метод отримання часткових результатів.

Результати — 1) опрацьовано відповідні джерела інформації з даного предмету дослідження; 2) охарактеризовано алгоритм побудови схем колективного цифрового підпису (КЦЕП); 3) вивчено області застосування КЦЕП; 4) побудовано приклад застосування КЦЕП.

ЦИФРОВИЙ ЕЛЕКТРОННИЙ ПІДПИС, КОЛЕКТИВНИЙ ЦИФРОВИЙ ЕЛЕКТРОННИЙ ПІДПИС, ІНТЕРНЕТ РЕЧЕЙ, ІНТЕРНЕТ ДРОНІВ, ТЕХНОЛОГІЯ БЛОКЧЕЙНУ.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

IoT – Інтернету речей (Internet of Things)

КЕК – криптографія з еліптичними кривими;

КЦЕП – колективний цифровий електронний підпис;

ЦЕП – цифровий електронний підпис;

ЦС – центр сертифікації.

ВСТУП

При обміні електронних даних між користувачами велике значення має захищення інформації від несанкціонованого доступу та зміни, а також надання юридичної сили для інформації. З іншого боку, з урахуванням розвитку технічного прогресу в умовах сьогодення майже кожне підприємство використовує технології та сучасну техніку для ведення перемовин та укладання договорів.

Одним з таких методів є цифровий електронний підпис (ЦЕП), який за допомогою спеціального програмного забезпечення дає змогу впевнитися в достовірності інформації документу, факт підписання певною особою та надійність реквізитів.

Цифровий підпис – це математична схема, яка використовується для перевірки автентичності цифрових повідомлень, та/або документів. Дійсний цифровий підпис за дотриманням певних умов дає одержувачу дуже високу впевненість у тому, що повідомлення було створено відомим відправником (автентичність), і що повідомлення не було змінено під час його передачі (цілісність) [15–17].

Цифрові підписи також є стандартним елементом більшості наборів криптографічних протоколів і зазвичай використовуються для розповсюдження програмного забезпечення, фінансових транзакцій, програмного забезпечення для управління контрактами та в інших випадках, коли важливо виявити підробку або втручання.

Цифрові підписи часто використовуються для реалізації електронних підписів, які включають будь-які електронні дані, які несуть намір підпису (підтвердження особи чи організації). ЦЕП мають юридичне значення в деяких країнах, включаючи Канаду, Південну Африку, США, Алжир, Туреччину, Індію, Бразилію, Індонезію, Мексику, Саудівську Аравію, Уругвай, Швейцарія, Чилі та країни Європейського Союзу [19]. На сьогодні в Україні юридична сила ЦЕП

також прирівняна до звичайного підпису. Тому підприємці все більше зацікавлені в отриманні цифрових електронних підписів для зручнішого електронного документообігу [21].

Але стрімкий розвиток новітніх ІТ-технологій, зокрема використання відкритих криптосистем, середовища Інтернету речей (IoT), технологій Інтернету дронів (IoD) потребує також перегляду існуючих методів застосування ЦЕП. Через це вже протягом останніх 10-15 років досліджується технологія використання так званих ***колективних цифрових електронних підписів*** (КЦЕП) [1–2, 4–5, 7–13]. Використання останніх можна знайти. Власне цьому виду цифрових електронних цифрових підписів і буде присвячена дана робота.

Тож **об'єктом дослідження** є цифровий електронний підпис, п**редметом дослідження** є особливості використання колективного цифрового електронного підпису. Відповідно було сформульовано мету роботи:

Мета роботи – провести аналіз особливостей використання електронного цифрового підпису.

Для реалізації поставленої мети було визначено такі ***завдання***: 1) опрацювати відповідні джерела інформації з даного предмету дослідження; 2) охарактеризувати алгоритм побудови схем колективного цифрового підпису (КЦЕП); 3) вивчити області застосування КЦЕП; 4) побудувати власний приклад застосування КЦЕП.

Структура кваліфікаційної бакалаврської роботи. Перший та другий розділ роботи містять теоретичну частину. В першому розділі описано основні відомості про ЦЕП, юридичні аспекти та алгоритм підписання електронного документу. В другому розділі описано використання КЦЕП в IoT та в IoD. Третій розділ містить практичну частину, а саме часткові результати щодо реалізації колективного цифрового підпису в технології блокчейну та безпеки нової схеми підпису.

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Основні відомості про ЦЕП

Цифрові електронні підписи (ЦЕП) зазвичай використовують асиметричну криптографію. У багатьох випадках вони забезпечують рівень перевірки та безпеки повідомлень, надісланих через незахищений канал: належним чином реалізований цифровий підпис дає одержувачу підстави вважати, що повідомлення було надіслано заявленим відправником.

Цифрові підписи багато в чому еквівалентні традиційним рукописним підписам, але правильно реалізовані цифрові підписи підробити складніше, ніж рукописні. Схеми цифрового підпису, у тому значенні, що обговорюються у нашій роботі, базуються на криптографії, і щоб бути ефективними, їх необхідно правильно реалізувати.

Повідомлення з цифровим підписом можуть бути будь-якими, представленими у вигляді бітового рядка: наприклад, електронна пошта, контракти або повідомлення, надіслані через якийсь інший криптографічний протокол. Найпростіша схема ж використання ЦЕП є такою

Степан підписує, адресоване Марті повідомлення «Привіт!» — додаючи до вихідного повідомлення копію, зашифровану її закритим ключем. Марта отримує і повідомлення, і підпис Степана. Для перевірки справжності повідомлення, використовуючи відкритий ключ Степана, Марта перевіряє, чи точно відповідає зашифрована копія, яка розшифровується за допомогою цього ключа, самому оригінальному повідомленню.

Схема цифрового підпису зазвичай складається з трьох стадій-алгоритмів:

- Алгоритм генерації ключів, який випадковим чином з набору можливих закритих ключів вибирає особистий ключ. Алгоритм виводить закритий ключ і відповідний відкритий ключ.

- Алгоритм підпису, який, надавши повідомлення та закритий ключ, створює підпис.
- Алгоритм перевірки підпису, який, враховуючи повідомлення, відкритий ключ і підпис, приймає або відхиляє запит на автентичність повідомлення.

Для ЦЕП необхідними є дві основні властивості [21]. По-перше, автентичність підпису, створеного з фіксованого повідомлення та фіксованого особистого ключа, можна перевірити за допомогою відповідного відкритого ключа. По-друге, генерувати дійсний підпис для сторони, не знаючи особистого ключа цієї сторони, має бути неможливо з обчислювальної точки зору. Цифровий підпис — це механізм автентифікації, який дає змогу відправнику прикріпити код, який діє як підпис. ***Алгоритм цифрового підпису (DSA), розроблений Національним інститутом стандартів і технологій*** **(США)**, є одним із багатьох прикладів алгоритму підпису.

Формально схема цифрового підпису являє собою трійку ймовірнісних поліноміальних алгоритмів часу (G, S, V), які позначають:

G (генератор ключа) генерує відкритий ключ (pk) і відповідний закритий ключ (sk) на вході 1^n, де n – параметр безпеки.

S (підпис) на входах повертає тег t: закритий ключ (sk) і рядок (x).

V (перевірка) виходи, прийняті або відхилені на входах: відкритий ключ (pk), рядок (x) і тег (t).

Для правильності S і V повинні задовольняти умову:

$$Pr\left[\left(pk,sk\right)\leftarrow G\left(1^n\right),\ V\left(pk,x,S\left(sk,x\right)\right)=прийнято\ \right]=1$$

Схема цифрового підпису є безпечною, якщо для кожного нерівномірного імовірнісного поліноміального часу зловмисника A, виконується:

$$Pr[\left(pk,\ sk\right)\leftarrow G\left(1^n\right),\ \left(x,t\right)\leftarrow AS\left(sk,\ \cdot\ \right)\left(pk,\ 1^n\right),$$

$$x \notin Q,\ V(pk,\ x,\ t) = прийнято] < negl(n),$$

де $AS(sk,\cdot)$ означає, що A має доступ до оракула $S(sk,\cdot)$, Q позначає набір запитів на S, зроблених A, який знає відкритий ключ pk і параметр безпеки n, а $x \notin Q$ означає, що зловмисник не може безпосередньо запитувати рядок x на S.

У 1976 році Вітфілд Діффі та Мартін Хеллман вперше описали поняття схеми цифрового підпису, хоча вони лише припустили, що такі схеми існували на основі функцій, які є односторонніми перестановками. Незабаром після цього Рональд Рівест, Аді Шамір і Лен Адлеман винайшли алгоритм RSA, який можна було використовувати для створення примітивних цифрових підписів (хоча лише як підтвердження концепції – «прості» підписи RSA не є безпечними). Першим широко розповсюдженим програмним пакетом, який пропонував цифровий підпис, був Lotus Notes 1.0, випущений у 1989 році, в якому використовувався алгоритм RSA.

Інші схеми цифрового підпису були розроблені незабаром після RSA, найпершими були підписи Лампорта, підписи Меркла (також відомі як «дерева Меркла» або просто «дерева гешів») та підписи Рабіна.

У 1988 році Шафі Голдвассер, Сільвіо Мікалі та Рональд Рівест стали першими, хто чітко визначив вимоги безпеки схем цифрового підпису. Вони описали ієрархію моделей атак для схем підпису, а також представили схему підпису GMR, першу, за якою можна було довести, що вона запобігає навіть екзистенційній підробці проти атаки обраного повідомлення, що є наразі прийнятим визначенням безпеки для схем підпису. Перша така схема, яка побудована не на функціях люка, а скоріше на сімействі функцій із набагато слабшою необхідною властивістю односторонньої перестановки, була представлена Моні Наором та Моті Юнгом.

У своїй основній статті Голдвассер, Мікалі та Рівест викладають ієрархію моделей атаки на цифрові підписи:

- Під час атаки лише за допомогою ключів зловмиснику надається лише відкритий ключ перевірки.
- В атаці з відомими повідомленнями зловмиснику надаються дійсні підписи для низки повідомлень, відомих зловмиснику, але не обраних ним.
- У атаці з адаптивним вибором повідомлення зловмисник спочатку дізнається підписи на довільних повідомленнях за його вибором.

Ієрархія результатів атаки тоді буде такою:

- Повний злам призводить до відновлення ключа підпису.
- ***Універсальна атака підробки*** призводить до можливості підробити підписи для будь-якого повідомлення.
- ***Вибіркова атака підробки*** призводить до підпису на повідомленні за вибором зловмисника.
- ***Екзистенційна підробка*** просто призводить до якоїсь дійсної пари повідомлення/підпис, яка ще не відома зловмиснику.

Таким чином, найсильнішим поняттям безпеки під час атаки на адаптивно обране повідомлення є захист від *екзистенційної підробки*.

До найпоширеніших алгоритмів, які використовують цифрові підписи, належать

1) алгоритм RSA;

2) алгоритм DSA;

3) алгоритм ECDSA;

4) алгоритм EdDSA;

5) алгоритм RSA з SHA;

6) алгоритм ECDSA з SHA.

7) схема підпису Ель-Гамала як попередника DSA та варіанти підпису Шнорра та алгоритму підпису Пуйнчеваля-Штерна;

8) Алгоритм підпису Рабіна;

9) схеми на основі криптографічного парування, такі як BLS;

10) NTRUSign є прикладом схеми цифрового підпису, заснованої на проблемах жорсткої решітки;

11) незаперечні підписи;

12) колективні підписи (Aggregate signature) – схеми підпису, які підтримують агрегацію: враховуючи n підписів на n повідомленнях від n користувачів, можна об'єднати всі ці підписи в один підпис, розмір якого є постійним у кількості користувачів. Цей єдиний підпис переконає перевіряючого, що n користувачів дійсно підписали n вихідних повідомлень. З біткойнами можна використовувати схему Мігіра Белларе та Грегорі Невена.

Далі наведемо спосіб створення ЦЕП за допомогою алгоритму RSA.

1.2. ЦЕП на основі алгоритму RSA

Ідея використання алгоритму шифрування RSA полягає в тому, що якщо Степан отримує повідомлення від Марти, він хоче переконатися, що воно насправді від Марти, а не від Зоряни. Якщо Зоряна може видавати себе за Марту та надсилати повідомлення Степану, прикидаючись Мартою, то це здається поганою справою.

На щастя, RSA визначає просту схему цифрового підпису. Це не обов'язково найкращий підхід до цифрових підписів, але він підходить, адже це свого роду доведення концепції, хоча й слабкі сторони у ньому також присутні. Для використання цифрових підписів RSA і Марта, і Степан повинні мати ключі RSA. Для них нам потрібно визначити функції шифрування та дешифровки. Припустимо, що функція шифрування повідомлень Марти — e_A, а її функція дешифровки — d_A. Щоб зашифрувати відкрите текстове повідомлення m для відправки Марті, Степан обчислює $c = e_A(m)$. Щоб розшифрувати c, Марта обчислює $d_A(c)$, так що $d_A(c) = d_A(e_A(m)) = m$. Аналогічно, нехай функція шифрування Степана буде e_B, а функція дешифровки — d_B.

Тепер припустимо, що Марта хоче надіслати повідомлення m Степану, і вона хоче ще підписати його, щоб Степан міг не тільки розшифрувати його, але й переконатися, що його надіслала Марта, а не Зоряна. Марта спочатку обчислює зашифрований текст $c = e_B(m)$. Потім вона обчислює $s = d_A(c)$, підписаний зашифрований текст, і надсилає s Степану. Цей алгоритм дій відомий як протокол ШПП (шифрування, а потім підпис).

Щоб розшифрувати повідомлення, Степан спочатку обчислює $c = e_A(s)$, а потім обчислює $m = d_B(c)$. Після того, як він виконає ці дві розшифровки, він отримує вихідне повідомлення

$$d_B(e_A(d_A(e_B(m)))) = d_B(e_B(m)) = m.$$

Крім того, він буде знати, що повідомлення від Марти, оскільки Марта єдина, хто знає d_A: це її особистий ключ! Якщо Зоряна спробує підробити повідомлення від Марти, вона не зможе використовувати d_A; замість цього їй доведеться спробувати свою власну версію, скажімо, d'_A. Але це не вдасться: *замість того, щоб повернути оригінальне повідомлення, Степан отримає спотворену нісенітницю.*

Зауваження 1.1. Марта і Степан можуть погодитися на інший протокол цифрового підпису, а саме на протокол ППШ («підписати, а потім шифрувати»): Марта могла спочатку обчислити $t = d_A(m)$, а потім $u = e_B(t)$, після чого Степан міг би відновити m шляхом обчислення $t = d_B(u)$, а потім $m = e_A(t)$. Що стосується відновлення повідомлення, то це нормально.

Однак, з криптографічної точки зору, це ППШ – гірший спосіб.

Зауваження 1.2. Типовими повідомленнями, які Марта може надсилати Степану, є числа за модулем n_B, де n_B – частина модуля відкритого ключа Степана. Отже, m і c мають бути числами за модулем n_B. З іншого боку, щоб застосувати d_A до c, c має бути числом за модулем n_A, частиною модуля відкритого ключа Марти. Щоб обійти цю проблему, можна припустити, що повідомлення Марти складається з послідовності частин $m_1,\ldots, m_k$ чисел за модулем n_B, і ми

розглядаємо їх разом, щоб утворити k-цифрове число за основою n_B. Аналогічно, зашифрований текст являє собою послідовність $c_1,...,c_k$ чисел за модулем n_B, яку ми розглядаємо як k-розрядне число за основою n_B. Щоб застосувати d_A до зашифрованого тексту, Марта повинна спочатку перетворити це число у число за основою n_A, перш ніж застосовувати d_A. Це може змінити кількість цифр/рядків у повідомленні, яке вона надсилає, але все ж не викличе проблем із їх розшифровкою.

Абстрагуючись від примітивного приклада роботи відкритих і закритих ключів RSA, можна стверджувати, що якщо у нас є будь-які дві пари функцій шифрування та дешифровки e і d, причому суперпозиції $e \circ d$ і $d \circ e$ обидві рівні ідентичності, одна для Марти і одна для Степана, то працює той самий протокол.

Давайте тепер проаналізуємо, що ми хочемо від схеми підпису.

По-перше, якщо Степан отримує повідомлення з цифровим підписом Марти, він повинен мати можливість довіряти, що Марта дійсно надіслала повідомлення, а не Зоряна. Тобто ми не хочемо, щоб Зоряна могла підробити цифровий підпис Марти. Під цим ми можемо мати на увазі кілька рівнів підробленості підпису:

Повністю підроблений: Зоряна може підробити підпис Марти на будь-якому повідомленні.

Вибірково підроблений: Зоряна може підробити підпис Марти на розумній частині повідомлень, якою вона могла б зацікавитися.

Екзистенційно підроблений: є повідомлення, на якому Зоряна може створити дійсну версію підпису Марти.

За допомогою цієї схеми Зоряні дуже легко створити екзистенційну підробку. Давайте напишемо повідомлення, яке Зоряна хоче підписати, так що підписаною версією буде $d_A(m)$. Для екзистенційної підробки Зоряні потрібно лише вміти сконструювати деяку пару $(m,dA(m))$. (тут припускається, що вона не знає функції d_A, вона просто включає особистий ключ Марти). З іншого боку, Зоряна може змінити цей процес: почати з якогось уже підписаного повідомлення

m' і зашифрувати його відкритим ключем Марти, щоб отримати $e_A(m')$. Оскільки $d_A(e_A(m')) = m'$, Зоряна може побудуйте пару $(e_A(m'), m')$, яка є дійсним підписом. Іншими словами, Зоряна може створити якесь повідомлення, на якому вона зможе підробити підпис Марти.

Швидше за все, повідомлення, яке Зоряна може підписати, є брехнею, тому підписувати таке повідомлення для неї не буде дуже цінно: навіть якби Степан підозрював, що повідомлення надійшло від Марти, для нього не було б жодного способу перевірки. Але що, якщо повідомлення написано не англійською мовою і, як правило, не призначене для читання людиною? Що, якщо повідомлення – це просто випадково сконструйовані двійкові рядки для використання в іншому контексті? Або випадково згенеровані ключі шифрування? Тоді той факт, що Зоряна вміє підписувати певні повідомлення, навіть безглузді, може стати серйозною проблемою.

Ще одна атака на цифрові підписи RSA походить від піддатливості. Якщо Зоряна знайде дві пари з підписом $(m_1, d_A(m_1))$ і $(m_2, d_A(m_2))$, то вона може побудувати нове повідомлення: $(m_1 m_2, d_A(m_1 m_2))$. Це може бути проблемою якщо продукт двох значущих повідомлень знову значущий, і це залежить від контексту. Швидше за все, добуток двох значущих повідомлень – знову тарабарщина: припустимо це фрази англійською мовою, перетворені на числа. Коли ми перемножуєте ці два числа, отримане число не є числовою формою змістовної англійської фрази. Тим не менш, добуток двох значущих повідомлень, швидше за все, буде значущим повідомленням, ніж випадкове повідомлення, тому ця атака принаймні трохи занепокоює.

З іншого боку, якщо Зоряна може вимагати підписання вибраних повідомлень, то вона може створювати підроблені підписи для значущих повідомлень: якщо вона хоче підписати повідомлення m, то вона може вибрати m_1 і m_2 з $m_1 m_2 = m$ і змусити Марту підписати m_1 та m_2. Це дає їй достатньо інформації для підпису m.

Щоб обійти деякі з цих проблем, звичайним рішенням є не підписання безпосередньо повідомлення m, а гешована версія m. Враховуючи повідомлення m, ми хочемо створити новий рядок $h(m)$ з такими властивостями:

• $h(m)$ легко обчислити.

• Враховуючи $h(m)$, обчислювально неможливо відновити m.

• Загалом, якщо врахувати деякий рядок h, обчислювально неможливо знайти будь-яке повідомлення m' таке, що $h(m') = h$, або навіть визначити, чи воно існує.

• Важко знайти два повідомлення m та m' такі, що $h(m) = h(m')$.

Ми називаємо таку функцію h ***криптографічною геш-функцією***. Майже будь-яка досить складна (але все ще легко обчислювана) функція, яка приймає значення в достатньо великому діапазоні, є криптографічною геш-функцією.

Замість того, щоб підписувати безпосередньо m, обчислюючи $d_A(m)$, ми спочатку гешуємо його, а потім підписуємо гешовану версію:$d_A(h(m))$. Це вирішує деякі проблеми, які ми обговорювали вище зі схемою цифрового підпису RSA. Наприклад, що станеться, якщо Зоряна спробує використати відкритий ключ Марти для створення підписаного повідомлення, починаючи з підпису, а потім відновлюючи повідомлення? Їй потрібно було б побудувати пару $(m,d_A(h(m)))$. Якщо вона починає з підписаної версії $m' = d_A(h(m))$, то вона може застосувати e_A, щоб отримати $e_A(m') = h(m)$, але на основі визначення криптографічної ueш-функції важко відновити m або будь-яке m_1 з $h(m) = h(m_1)$. Таким чином, Зоряна приречена на поразку. Інші атаки, засновані на піддатливості, також зазнають невдачі, оскільки криптографічна геш-функція не піддається у будь-який спосіб.

1.3. Юридичні аспекти щодо використання цифрового електронного підпису

Щоб пришвидшити та спростити процес функціонування справочинних функцій, часто використовують документаційно-інформаційну базу в електронному вигляді. Електронний документ, який підтверджується електронним цифровим підписом, має таку ж юридичну силу, як і всім нам звичний паперовий, котрий власноруч підписала відповідальна особа.

В Україні загальні питання щодо впровадження цифрових електронних підписів підтвердження та електронного документообігу (далі — ЦЕП) регулюються відповідними законодавчими актами, зокрема **«Закон України про електронні документи та електронний документообіг» № 851-IV від 22 травня 2003 року [22] (далі — Закон № 851)** і **«Закон України про цифровий електронний підпис» № 852-IV (далі — Закон № 852)**.

У США діє «Стандарт цифрового підпису» (DSS). Це федеральний стандарт обробки інформації (FIPS), який визначає алгоритми, які використовуються для створення цифрових підписів за допомогою алгоритму безпечного гешування (SHA) для автентифікації електронних документів. DSS надає нам лише функцію цифрового підпису (див. рис. 1.1), а не будь-які стратегії шифрування чи обміну ключами.

Сторона відправника

У підході DSS, геш-код генерується з повідомлення, і функції підпису надаються наступні вхідні дані:

1. Геш-код.
2. Випадкове число «*k*», згенероване для цього конкретного підпису.
3. Закритий ключ відправника, тобто $PR(a)$.
4. Глобальний відкритий ключ (який є набором параметрів для принципів зв'язку), тобто $PU(g)$.

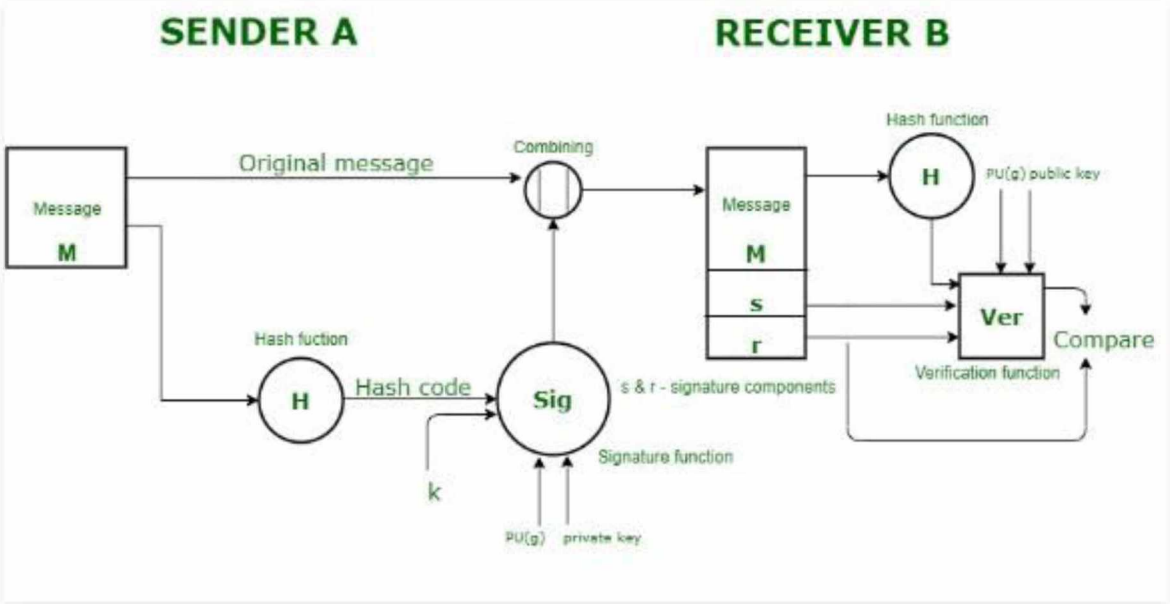


Рис. 1.1. Схема цифрового підпису за стандартом DSS

Ці вхідні дані для функції забезпечать вихідний підпис, що містить два компоненти – «*s*» та «*r*». Таким чином, вихідне повідомлення, об'єднане з підписом, надсилається одержувачу.

Сторона одержувача:

На стороні одержувача виконується перевірка відправника. Генерується геш-код надісланого повідомлення. Існує функція перевірки, яка приймає такі вхідні дані:

1. Геш-код, згенерований приймачем.
2. Компоненти підпису «*s*» і «*r*».
3. Відкритий ключ відправника.
4. Глобальний відкритий ключ.

Вихід функції перевірки порівнюється з компонентом підпису «*r*». Обидва значення будуть збігатися, якщо надісланий підпис дійсний, оскільки тільки відправник за допомогою його приватного ключа може створити дійсний підпис.

1.3.1. Детальна інформація щодо законодавчих актів України про використання ЦЕП

Так, згідно з Закону № 852, ЦЕП — це вид електронного підпису, отриманого шляхом застосування криптографічного перетворення набору відповідних електронних даних. Цей підпис, згідно визначення, має входити як складова до згаданого набору даних або логічно з ним поєднуватися. Функціональним же призначенням підпису є надання змоги підтвердити (з його допомогою) цілісність документа та ідентифікувати особу, яка підписує цей документ.

ЦЕП створюють шляхом використання особистого (таємного) ключа, а перевіряють – з допомогою відкритого (публічного) ключа.

Із загальної точки зору електронний підпис законодавчо характеризується як електронній дані, які доповнюють інші важливі дані в електронній формі, та /або мають із цими даними логічний зв'язок. Безпосереднім же призначенням цих даних є ідентифікація особи, яка підписує важливі дані (документ). Система цифрового електронного підпису має будуватися та принципі, що кожен користувач мережі має свій власний (особистий) ключ, який зберігається в таємниці та використовується для формування підпису, але також в системі зосереджено визначений публічно набір даних – відкриті ключі, кожен із яких слугує парою до відповідного особистого ключа: на відміну від останнього, відомий кожному користувачу мережі та призначений для перевірки автентичності підпису.

Всі відкриті ключі повинні мати відповідні сертифікати. Як зазначено в Законі № 852, сертифікати особистих ключів (далі – просто сертифікати ключів) – це документи, видані центром сертифікації ключів [14], який засвідчує чинність і належність відкритих ключів відповідним підписувачам. Такі сертифікати ключів можуть бути визначені (підтверджені) як в електронному, так і в паперовому вигляді (їх використання для ідентифікації осіб–підписувачів.

Юридичні чи фізичні особи – суб'єкти підтверджуючого справочинства щодо обігу електронних документів – використовують ЦЕП для автентифікації підписаних документів (підтвердження цілісності даних в електронній формі) та ідентифікації самих осіб, які підписують документ.

У Законі № 852 [18] також говориться про те, що за правовим статусом можлива рівноправність ЦЕП та авторського (власноручного) підпису з печаткою, якщо виконані такі вимоги:

- за допомоги надійних інструментів електронного підпису відповідний ЦЕП є підтвердженим з застосування розширеного сертифіката ключа; іншими словами ЦЕП отриманий на вимогу в одному з Акредитованих місць сертифікації ключів;
- на момент автентифікації ЦЕП використовувався посилений сертифікат ключа, який був чинним на момент його накладання;
- особистий ключ особи, яка підписувала документ, відповідає зазначеному в сертифікаті відкритому ключу.

Застосування ЦЕП не змінює послідовність підписання договорів, документів і т.д., для здійснення угод у письмовій формі встановленого законом.

Також Закон № 852 передбачає, що підписант повинен тримати свій особистий ключ ЦЕП в таємниці. Тобто особа повинна накладати підпис на документи самостійно. Іншій особі передавати особистий ключ ЦЕП заборонено.

Як ми вже знаємо, під час застосування ЦЕП потрібно зберігати особистий власний ключ у таємниці, інакше він буде вважатись скомпрометованим, таким, що застосовують незаконно. Згідно с Законом України № 852 передбачена можливість зберігання на захищених носіях особистих ключів ЦЕП. Захищений носій особистих ключів – один з надійних засобів цифрового електронного підпису, потрібний для збереження особистого ключа, котрий вміщує вбудовані апаратно-програмні засоби, що гарантують захист даних від несанкціонованого

доступу (далі – НСД), від прямого перегляду значень параметрів особистих ключів та їх відтворення.

Останні зміни до Постанови Кабінету Міністрів України (далі – КМУ) від 28 жовтня 2004 року №1452 «Про затвердження Порядку застосування цифрового електронного підпису органами державної влади, органами місцевого самоврядування, підприємствами, установами та організаціями державної форми власності» [20] передбачають, що для здійснення інформаційного обміну, угод, надання адміністративних та інших послуг в електронній формі з другими юридичними особами органи місцевого самоврядування, державної влади та підприємства державної форми власності використовують тільки захищені носії.

Захищені носії особистих ключів зобов'язані застосовувати: державні реєстратори юридичних та фізичних осіб, державні реєстратори прав на нерухоме майно.

Слід завважити, що в листопаді 2018 року набув чинності Закон України «Про електронні довірчі послуги». За його основи розроблено проект постанови КМУ «Про Порядок застосування електронних довірчих послуг в органах державної влади, органах місцевого самоврядування, підприємствах, установах та організаціях державної форми власності». Для завірення чинності відкритого ключа державні установи застосовують тільки його кваліфікований сертифікат, а для реалізації повноважень, спрямованих на набуття, зміну чи припинення прав та (або) обов'язків фізичної чи юридичної особи тільки захищені носії особистих ключів.

1.3.2. Апаратні інструменти генерування ЦЕП

Розглянемо деякі апаратні інструменти захисту інформації, які дозволяють генерувати ЦЕП [20].

Електронний ключ «Кристал-1», виготовлений у вигляді невеликого знімного USB-пристрою з програмним інтерфейсом CCID та електронним флеш-диском (протокол USB Mass Storage);

електронний ключ «Алмаз-1К»; виготовлений у вигляді невеликого знімного USB-пристрою, з програмним CCID-інтерфейсом;

електронний ключ «Кристал-1Д»; виконаний у вигляді невеликого знімного USB-пристрою. Цей інструмент призначений для захисту службової інформації.

Ці ключі виконують такі функції:

«Кристал-1»

– автентифікація оператора електронно-обчислювального пристрою при доступі до ключа;

– створення особистих та відкритих ключів для алгоритму ЦЕП;

– створення особистих і відкритих ключів для протоколу розподілу ключів;

– створення ключів для алгоритмів шифрування та створення випадкових послідовностей на основі обладнання;

– зберігання у внутрішній пам’яті та захист особистих ключів від НСД;

– перевірка та створення ЦЕП;

– обчислення геш-значень;

– розділення ключових даних на основі протоколу асиметричного розподілу;

–зберігання та захист від НСД довільних даних у внутрішній пам’яті;

– збереження та шифрування файлів у прилеглому електронному flash-диску;

– контроль працездатності та цілісності вбудованого програмного забезпечення та ін.

«Алмаз-1К»

– автентифікація користувача при доступі до ключа;

– створення відкритих та особистих ключів для алгоритму ЦЕП;

– створення відкритих та особистих ключів для протоколу розподілу ключів;

– створення ключів для шифрувального алгоритму та генерація випадкових послідовностей на основі апаратного генератора;

– збереження та захист від НСД особистих ключів у внутрішній пам'яті;

перевірка та формування ЦЕП;

– розрахунок геш-значень;

– розподіл ключових даних згідно з протоколом асиметричного розподілу;

– зберігання та захист від НСД у внутрішній пам'яті довільних даних;

– контроль працездатності та цілісності вбудованого програмного забезпечення та ін.

«Кристал-1Д»

– автентифікація користувача обладнанням при доступі до ключа;

– створення ключів;

– зберігання та захист від НСД ключів у внутрішній пам'яті;

–перевірка та формування ЦЕП;

– розподіл та шифрування найважливіших даних;

– зберігання та захист від НСД довільних даних у внутрішній пам'яті;

– контроль цілісності й працездатності вбудованого програмного забезпечення та ін.

Для всіх зазначених засобів передбачено, що апаратна реалізація забезпечує захищеність процесу виконання криптографічних перетворень та унеможливлює доступ до особистих власних ключів із боку апаратного-програмного середовища.

Електронні ключі реалізують такі криптографічні алгоритми та протоколи:

1. шифрування за ДСТУ 28147:2009 (режим простої заміни та режим вироблення імітовставки);
2. ЦЕП за ДСТУ 4145-2002 (усі довжини ключів передбачає стандарт);

3. гешування за ГОСТ 34.311-95;
4. протокол розподілу ключових даних Діффі-Хеллмана в групі точок еліптичної кривої (довжина ключа до 571 біту).

Особисті ключі генерують, зберігають і застосовують лише всередині електронного ключа, вони жодним чином не потрапляють за його межі.

Зберігання особистих власних ключів та інших ключових даних здійснюється у внутрішньому постійному пристрої запам'ятовування електронного ключа.

1.3.3. Види ЦЕП з юридичною силою

Існуючі ЦЕП можна поділити на три види:

- простий ЦЕП;
- посилений некваліфікований ЦЕП;
- посилений кваліфікований ЦЕП.

У випадку ***простої системи електронного цифрового підпису*** для створення використовуються паролі, коди і інші засоби. Простий електронний цифровий підпис - інструмент підтвердження достовірності електронних даних відправником. Він вважається дійсним при дотриманні наступних умов:

- електронний документ підписаний ЦЕП;

- ключ електронного підпису створений відповідно до вимог інформаційної системи, за допомогою якої проводилася засвідчення і відправка електронних повідомлень відправником.

У нормативних і правових документах, а також у договорах учасників в обов'язковому порядку визначають основні правила застосування простого електронного цифрового підпису:

- механізм ідентифікації автора підпису в електронному документі;
- обов'язкове дотримання вимог конфіденційності при використанні електронного підпису відповідальними особами;

- виконання вимог закону №852 щодо використання простого електронного цифрового підпису;
- неможливість застосування ЦЕП до секретних державних документів.

Посилений некваліфікований ЕПЦ. Для створення такого підпису використовується криптографічна програма, що працює на основі ключа електронного цифрового підпису. Посилений некваліфікований підпис дозволяє визначити укладача документа, який його поставив, і наявність змін в листі після його підписання. Застосування некваліфікованої ЕП дозволяє не використовувати сертифікат ключа електронного цифрового підпису (за умови дотримання вимог законодавства, інших нормативних документів та договорів між відправником і адресатом).

Посилений кваліфікований ЦЕП. Особливість цього різновиду електронного цифрового підпису є наявність спеціального ключа перевірки, що міститься в кваліфікованому сертифікаті. Формування та перевірка посиленим кваліфікованим ЦЕП відбувається за допомогою спеціальних засобів електронного підпису, які відповідають вимогам закону України №852. Паперові документи з рукописним підписом і електронні документи, на яких стоїть посилений кваліфікована підпис, мають однакову юридичну силу (крім випадків, які визнають виключно рукописний підпис, передбачених в законодавстві). Також законом допускається встановлення в нормативних актах і договорах між відправником і отримувачем додаткових вимог до електронних документів, підписаних посиленим кваліфікованим підписом.

Порівняємо розглянуті види електронного цифрового підпису за аналогією зі знайомими фізичними засобами ідентифікації особистості: Простий ЕПЦ схожий на бейдж – будь-який сторонній чоловік може ним скористатися, тому відповідальність за збереження даних лежить на власнику підпису. Некваліфікований ЕПЦ аналогічний пропуску в компанії, при цьому між сторонами угоди є певний рівень довіри. Кваліфікований ЕПЦ як паспорт –

найважливіший інструмент для ідентифікації, який надає можливість користуватися всіма послугами. Відповідно до закону «Про електронний підпис», ЦЕП в Україні створений за закордонними стандартами. Видача сертифіката ключа в іншій державі не може бути причиною невизнання юридичної сили документа, на якому стоїть такий підпис.

В табл. 1.1 наведено також класифікацію ЦЕП за їх відкритістю.

Таблиця 1.1

Види електронних підписів та їх характеристика

Вид	Характеристика
Електронний підпис з одноразовим ідентифікатором	Під час використання даних в електронній формі подаються у вигляді алфавітно-цифрової послідовності, додаються до інших електронних даних особою, яка прийняла пропозицію (оферту) укласти електронний договір, а також надсилаються іншій стороні цього договору.
Відкритий цифровий електронний підпис	Наявний параметр криптографічного алгоритму для автентифікації цифрового електронного підпису щодо його доступності для суб'єктів відносин у будь-якій сфері використання такого підпису.
Закритий (особистий) цифровий електронний підпис	Закритий підпис є унікальною послідовністю символів, яка призначена для створення цифрового електронного підпису в електронних документам. Параметр криптографічного алгоритму формування цифрового електронного підпису доступний тільки підписанту, підпис працює з особистими ключами тільки в парі з відкритим ключем.

1.3.4. Алгоритм підписання електронного документа

При підписанні електронного документу його зміст не змінюється, а прикладається блок даних, – цифровий електронний підпис. Є два етапи отримання цього блоку:

На першому етапі обчислюється так званий «відбиток повідомлення» [19] (message digest) за допомогою ПЗ і потрібної математичної функції.

Цей відбиток володіє такими властивостями:

— фіксована довжина, незалежно від довжини повідомлення;

— унікальність відбитку для всіх повідомлень;

— неможливість відновлення повідомлення по його відбитку.

Під час другого етапу відбиток документу шифрується с використанням ПЗ і особистого ключа автора. Тільки при допомозі відкритого ключа автора можна дешифрувати ЦЕП і отримати початковий відбиток, згідний з документом. Шифрування особистим ключем автора підтверджує авторство документу, а обчислення відбитку документу захищає його від модифікації іншими особами після підписання.

Дуже важливо, щоб підпис використовувався власником, задля цього використання підпису відбувається під пасивним спостереженням провайдера, котрий надає послугу [16].

Для використання електронного підпису, користувач завантажує документ на сайт провайдера, додає підпис та відсилає його на електронну адресу другого контрагента. Також інший користувач може продивитись ким документ був підписаний і додати свій підпис. Потім у користувачів буде відображатись документ про осіб, які підписали його.

Для використання цієї технології користувачі повинні зареєструватись в одного провайдера ЦЕП. Система дійсно перевіряє, чи належать користувачеві

ідентифікатори, які він надав реєструючись. Всі копії підписів осіб які отримали доступ до документу зберігаються на сервері постачальника послуги.

Якщо вам потрібен більш захищений варіант, з можливістю шифрування даних, та з кращим захистом, то цифровий електронний підпис один із кращих варіантів [17].

Наведемо наочний приклад використання ЦЕП для підпису документа Word. Для цього скористаємося послугою «центрального засвідчувального органу» який знаходиться за посиланням https://czo.gov.ua/ [20] (рис. 1.2).

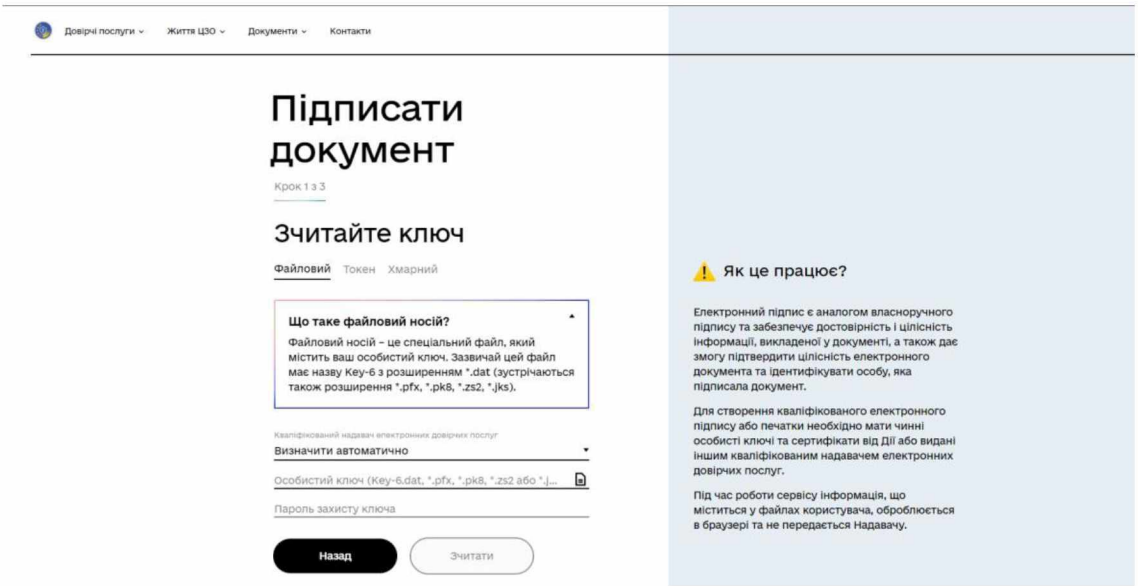


Рис. 1.2. Використання ЦЕП через послугу «центрального засвідчувального органу». Етап 1.

Після внесення даних натискаємо на «Зчитати» (рис. 1.3).

Після цього ми перевіряємо дані та тиснемо «Далі» (рис. 1.4). Обираємо потрібний документ та натискаємо «Далі» (рис. 1.5).

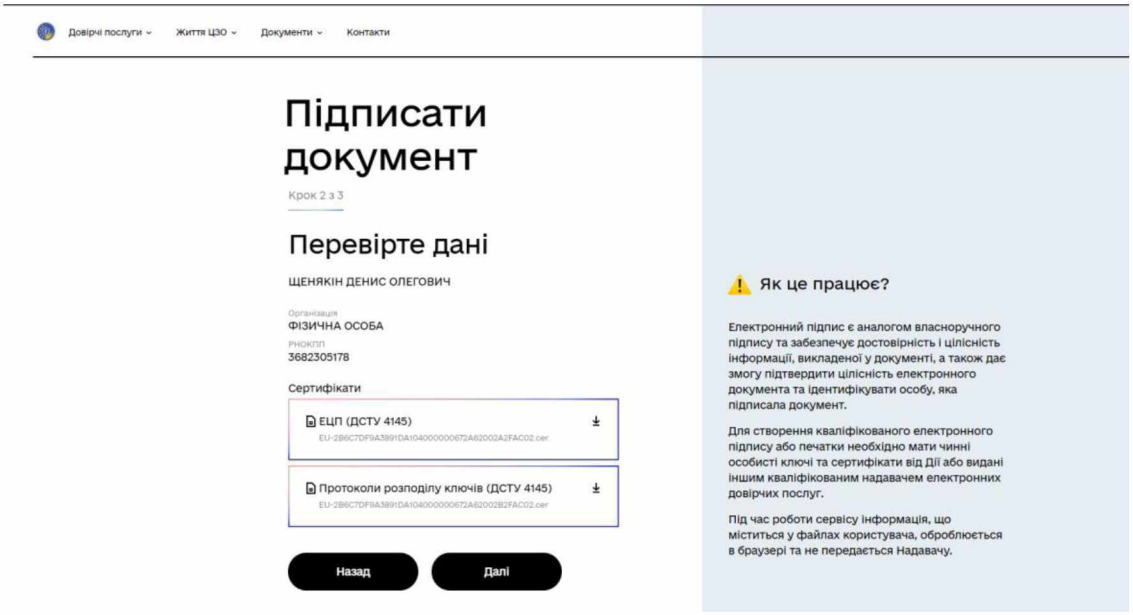


Рис. 1.3. Використання ЦЕП через послугу «центрального засвідчувального органу». Етап 2.

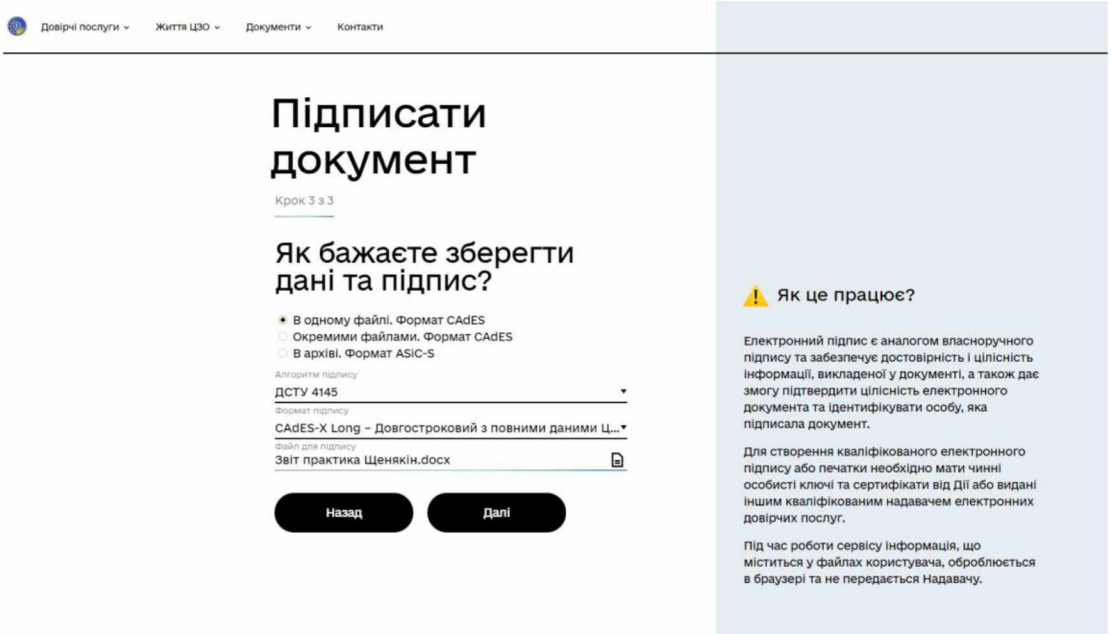


Рис. 1.4. Використання ЦЕП через послугу «центрального засвідчувального органу». Етап 3.

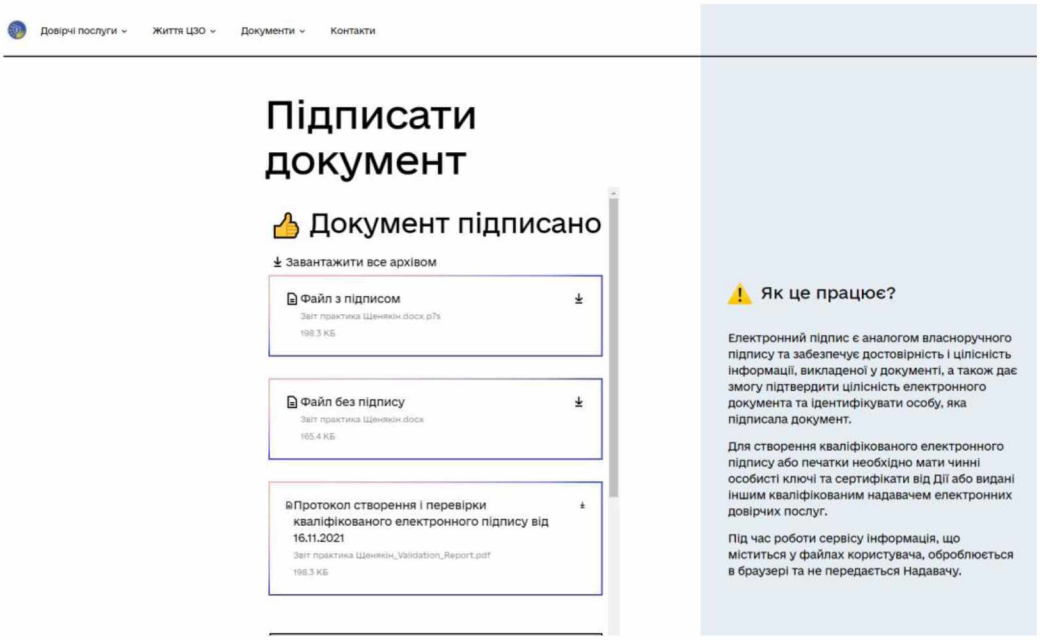


Рис. 1.5. Використання ЦЕП через послугу «центрального засвідчувального органу». Етап 4.

Тепер обираємо яким чином отримати підписаний документ та завантажуємо його. Далі за допомогою програми «Користувач КНЕДП (ЦСК) ІДД ДПС» перевіряємо файл на наявність підпису (рис. 1.6).

Рис. 1.6. Використання ЦЕП через послугу «центрального засвідчувального органу». Етап 5.

1.4. Колективні цифрові підписи

Тут власне і підходимо до предмету нашого дослідження – колективних цифрових електронних підписів.

Як зазначалося у пункті 1.1. під ***колективним цифровим електронним підписом*** **(КЦЕП)** розуміється схема підпису, яка підтримує підписання повідомлень декількома користувачами, об'єднуючи всі ці підписи в один підпис, розмір якого є сталим і не залежить від кількості користувачів [2].

Тут не слід плутати колективний підпис із множинним, за яким відбувається *n*-кратне підписання одного і того ж повідомлення.

Розглядаючи відповідні дослідження щодо використання КЦЕП, можна стверджувати, що це переслідує 3 основні мети:

1) забезпечення кращого захисту інформації, її цілісності та доступності;

2) забезпечення колективної відповідальності при укладанні угод через підтвердження автентичності;

3) забезпечення більшої пропускної здатності системи та зменшення витрат.

Серед раніше вивчених методів колективного підпису є багато методів, які генерували колективні підписи за допомогою операцій сполучення. Це було неефективно. Обчислювальне навантаження збільшувалося зі збільшенням кількості підписувачів. Лише після створення систем шифрування з відкритим ключем (алгоритмів Діффі–Хеллмана, чи алгоритмів RSA) почали розроблятися методи колективного підпису, які крім згаданих вище алгоритмів шифрування використовували криптографію на основі ідентифікації (**IBC**), підписання на основі ідентифікації (**IBS**), колективні підписи на основі ідентифікації (**IBAS**) та криптографію без сертифікатів (**CLC**) [4, 5, 8].

Розглянемо найпростішу схему колективного цифрового підпису.

На рисунку 1.7 наведена схема [3], яка може, наприклад використовувати стандартний алгоритм шифрування RSA. Так, обирається велике число $N = pq$, яке є добутком двох простих чисел p та q, обираються довільні числа $r_i < N$ і обчислюються значення $t_i = r_i^e \bmod N$, де e – відкритий ключ заданого алгоритму шифрування RSA, обираються дві довільні геш-функції $H_1 : \{0,1\}^* \to \{0,1\}^*$ та

$H_2 : \{0,1\}^* \to Z_N^*$ та обчислюються відкриті та таємні ключі користувачів $H_2(ID_i)$ і $g_i = H_2(ID_i)^d$ відповідно. Тоді підписання повідомлень $m_1, m_2, ..., m_n$ можна визначити через обчислення таких значень

$$s_i = s_i \cdot r_i \cdot g_i^{H_2(t_1||...||t_i, ID_1||...||ID_i, m_1||...||m_i)} \bmod N$$

– підписи користувачів. Колективним же підписом буде значення:

$$s_n = s_{n-1} \cdot r_n \cdot g_n^{H_2(t_1||...||t_n, ID_1||...||ID_n, m_1||...||m_n)} \bmod N.$$

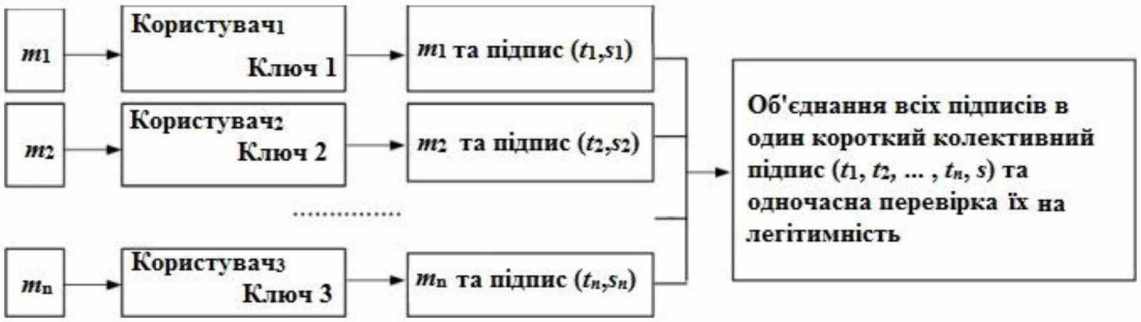


Рис. 1.7. Найпростіша схема КЦЕП

Аналогічно, як і підтвердження будь-якого звичайного підпису, ми можемо отримати підтвердження всього колективного підпису. Підтвердження буде тут означати, що всі користувачі підписали зазначені повідомлення і ці повідомлення не були змінені. Для такої верифікації достатньо перевірити умову

$$s_n^e = \left(\prod_i^n t_i \cdot H_2(ID_i)^{H_2(t_1||...||t_n, ID_1||...||ID_n, m_1||...||m_n)} \right) = 1 \bmod N.$$

Однак згадані вище методи IBC, і CLC створюють проблеми з депонуванням та/або розподілом ключів. Щоб вирішити такі проблеми, були запропоновані підписи на основі сертифікатів (**CBS**) і сукупні підписи на основі сертифікатів (**CB-AS**), для яких і на сьогодні проводяться дослідження, щоб переконатися, що вони відповідають ряду вимог безпеки, таким як цілісність даних, невідмовність, а також протидія підробці підписів [11].

Великого значення на сьогодні набуває і застосування схем КЦЕП. На рис. 1.8. наведено приклад використання колективного підпису для верифікованої агрегації даних з датчиків, які підписують їх та передають через шлюз у хмару.

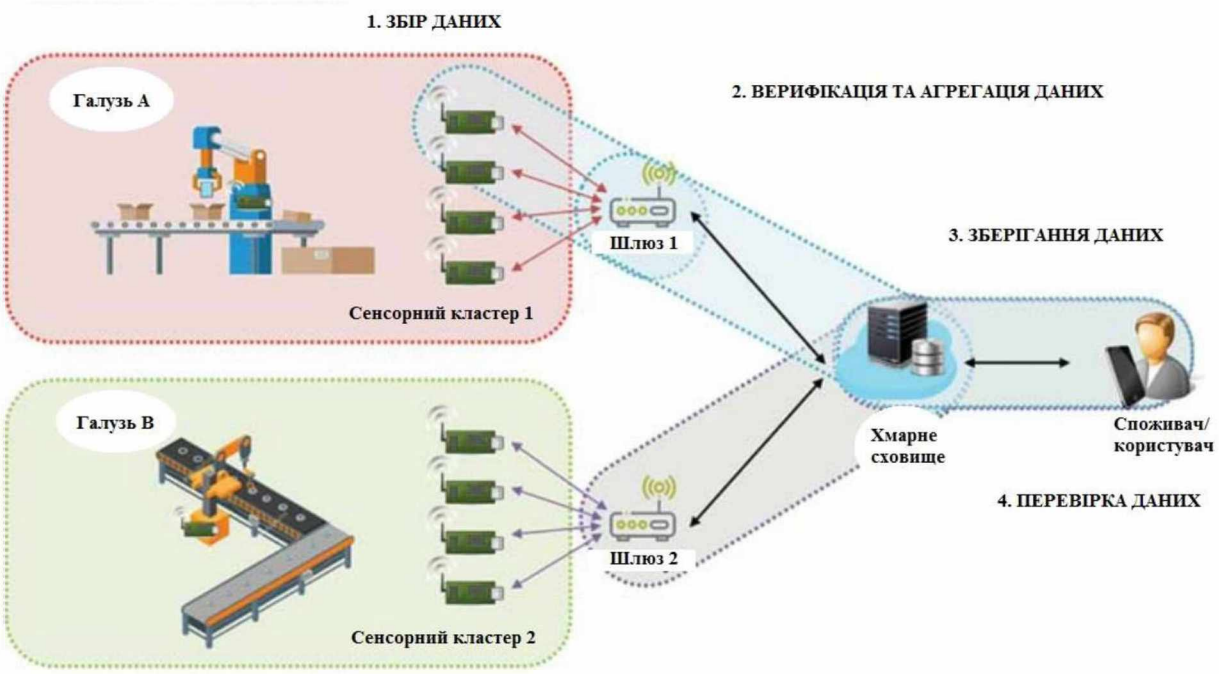


Рис. 1.8. Схема збору, перевірки та зберігання даних у середовищі IoT, які передбачають використання колективного підпису

РОЗДІЛ 2. РЕАЛІЗАЦІЇ КОЛЕКТИВНОГО ЦИФРОВОГО ПІДПИСУ

У цьому розділі спробуємо розглянути найбільш цікаві моменти щодо реалізацій КЦЕП в різних технологіях застосування.

2.1. Використання КЦЕП в середовищах IoT

Останні розробки в області інформаційно-комунікаційних технологій (ІКТ), Інтернету речей (IoT) сприяли промисловій «смартизації»; розумні фабрики та розумні галузі, які пов'язують реальний і віртуальний світи через кіберфізичні системи (CPS). CPS обробляє завдання та інформацію фізичного світу у віртуальному просторі за допомогою IoT та інших мереж і постійно адаптується до змін без втручання людини. Щоб CPS функціонував добре, природа середовища IoT має важливе значення, оскільки багато фізичних речей повинні бути підключені до датчиків.

У хмарному середовищі IoT дані збираються з пристроїв IoT і безпечно зберігаються в хмарі, завдяки чому законні користувачі можуть отримати доступ до хмари та перевірити дані. Більш того, пристрої IoT використовується також і в промислових середовищах Інтернету речей (Industrial Internet of Things, IIoT), таких як виробництво, транспортування та енергетика, а також у медичних середовищах і розумних будинках. Однак, якщо датчики передають звичайний текст, можливі атаки підробки даних та повторного відтворення. Зокрема, якщо комунікації у великих мережевих системах, таких як середовища IIoT, є незахищеними, то успішна атака на мережу може призвести до збитків на мільйони доларів.

Щоб вирішити цю проблему, необхідна легка технологія криптографії для забезпечення конфіденційності даних, а також технологія цифрового підпису, яка забезпечує цілісність даних, згенерованих сенсорним пристроєм у середовищі IoT.

Користувач перевіряє підпис, забезпечуючи цілісність повідомлення. У середовищі IoT користувачі можуть отримувати безпечні послуги лише за умови забезпечення цілісності даних, зібраних з усіх пристроїв. Крім того, в середовищі, де збираються масштабні дані, такому як IIoT, концепція агрегації, колективного підпису, важлива для ефективного розподілу даних. Так, у статті [] було реалізовано модель колективного підпису CB-AS та запропоновано нову легшу схему CB-AS для хмарних середовищ IoT. Цей запропонований метод є ефективним методом CB-AS, який забезпечує ізоляцію ключа.

2.1.1. КЕК та проблема дискретного логарифма

Криптографія з еліптичними кривими (КЕК) — це метод шифрування з відкритим ключем, який використовує той факт, що проблема дискретного логарифма на еліптичній кривій є складною. Порівняно зі звичайною криптографією з відкритим ключем (RSA, Діффі–Хеллман), такий самий захист можна отримати за допомогою меншої кількості бітів, шифрування обробляється з високою швидкістю, а керування ключами просте, оскільки використовується короткий ключ. Тому він широко використовується в Інтернеті речей та інших середовищах.

Еліптичною кривою у КЕК визначають множину розв'язків (x, y) рівняння $y^2 = x^3 + ax + b \pmod p$, визначених для довільних цілих чисел a та b. Той факт, що точка $P = (x, y)$ знаходиться на еліптичній кривій, означає, що наведене вище рівняння із такими значеннями (x, y) виконується (перетворюється на тотожність). Також для еліптичної кривої можна визначити операцію множення точки на число і $Q = t \cdot P$ існує для будь-якого цілого t для кожної точки P. Знаходження ж значення t за даними двома точками P і $Q = t \cdot P$ є задачею дискретного логарифма еліптичних кривих і не може бути розв'язаною в загальному випадку. Іншими словами, легко знайти Q як $Q = t \cdot P$, але дуже важко зробити висновок про значення t, навіть якщо ми знаємо Q і P.

2.1.2. Ізоляція ключів у КЦЕП

Щодо використання колективного цифрового підпису після ряду розроблених моделей все ж залишалася проблема депонування ключів. Щоб вирішити цю проблему, було розроблено схему (CBC), яка поєднує переваги PKC та IBC. Користувач створює пару відкритих і особистих ключів і отримує сертифікат із ідентифікатором та відкритим ключем від довіреного центру сертифікації (ЦС). Сертифікат CBC служить особистим (секретним) ключем для користувача. Підпис і розшифровка виконуються за допомогою сертифіката користувача та закритого ключа одночасно. Таким чином, вирішується проблема депонування ключів, спричинена тим, що центр генерації ключів (KGC) видає ключі кожному власнику IBS, і спрощується керування сертифікатами (як це було у попередніх схемах PKC). Крім того, усувається перевантаження перевірки відкритого ключа. Існують методи CBS та CB-AS з використанням CBC.

У методах на основі підпису приватний ключ, який використовується при підписанні, має бути абсолютно захищеним. Якщо цей ключ розкритий, існує кілька загроз безпеки. Припускаючи, що зловмисник викрив ключ, фізично атакуючи датчик на заводі, зловмисник може використовувати цей ключ, щоб підробити повідомлення та підпис із датчика. Це може призвести до помилок у виробництві, і вся система може зупинитися. Це фатально у великому взаємопов'язаному середовищі, як-от розумна фабрика чи інший приклад Інтернету речей. Важливо безпечно керувати та зберігати особисті ключі, і серед різних методів для цього є технологія ізоляції ключів.

Ізоляція ключів була представлена Додісом у 2002 році. Користувачі оновлюють свої закриті ключі за допомогою фізично захищеного пристрою, який називається Helper (рис. 2.1). Кожен користувач створює відкритий ключ, приватний ключ і тимчасовий секретний ключ для початкового підпису. Після цього помічник бере відкритий ключ користувача, створює ключ, відомий як ключ

оновлення, який можна використовувати для періоду в *t* діб, і надсилається користувачеві.

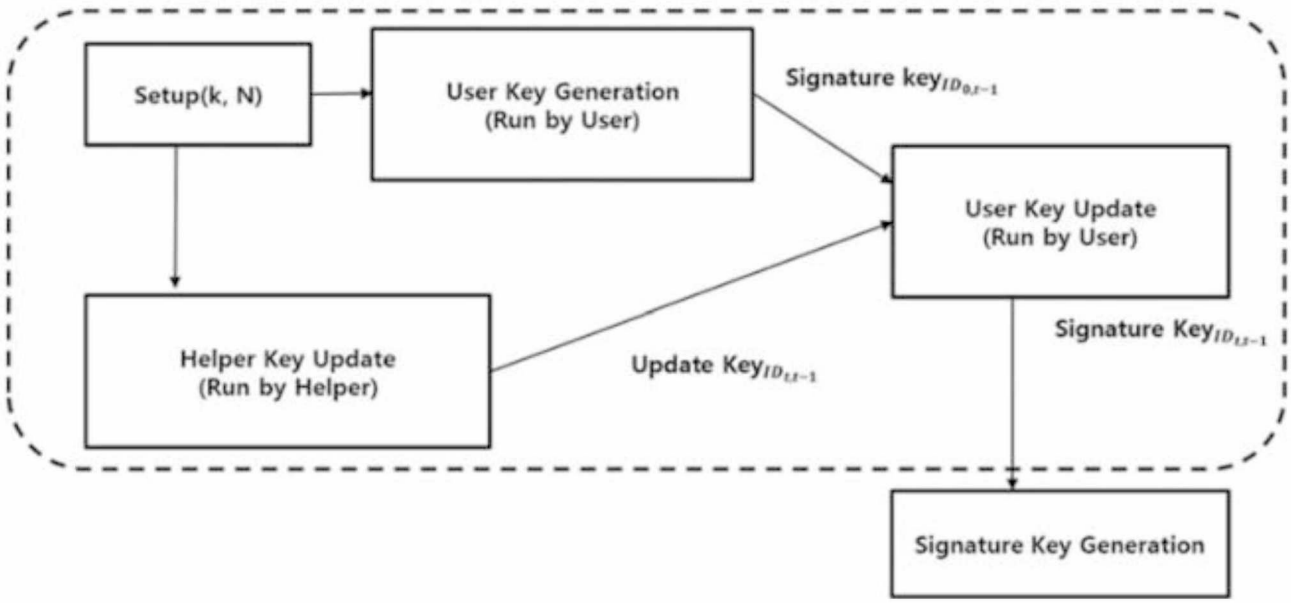


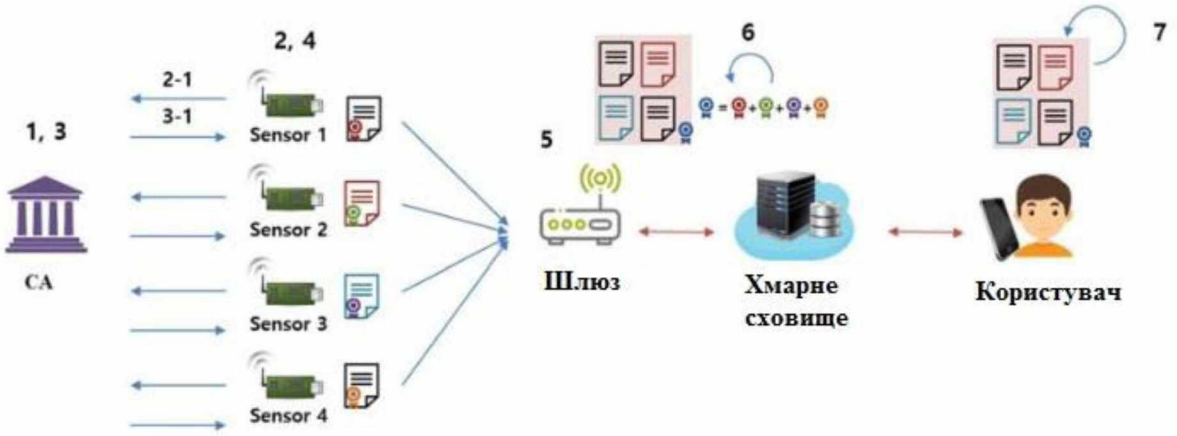
Рис. 2.1. Схема Helper для генерації ключа підпису з використанням ізоляції ключа

За допомогою отриманого ключа оновлення користувач оновлює наявний ключ підпису ключем підпису, який можна використовувати протягом періоду t. Потім Helper видає новий ключ для використання протягом наступного періоду в *t* діб. Методи IBC, CBC та CLS використовують ключові методи ізоляції для вирішення проблеми опромінення ключа.

CBS – це підпис для одного повідомлення. Метод CB-AS збирає кілька повідомлень і створює колективний підпис. У CBS, якщо кілька відправників передають дані, для кожного повідомлення створюється підпис. Таким чином, кількість підписів збільшується зі збільшенням кількості підписантів, і перевіряючий повинен перевірити всі підписи. CB-AS об'єднує кілька підписів в один.

Для перевірки використовуються відкриті ключі відправника, але всі підписи перевіряються разом. Це зменшує витрати на перевірку та пам'ять,

витрати на обчислення та пропускну здатність. Багато методів CB-AS, які не базуються на криптографічному паруванні, були застосовані в середовищах IIoT, таких як розумні фабрики.



1. Етап налаштування 2. Етап генерації ключів 2-1. Запит сертифікату
3. Етап генерації сертифікату 3.1. Передача сертифікату 4. Етап підписання
5. Етап перевірки підпису 6. Етап формування колективного підпису
7. Етап перевірки колективного підпису

Рис. 2.2: Структура сукупного підпису на основі сертифіката

2.2.3. Модель атаки в CB-AS

Схема на рисунку 2.2 показує основну структуру CB-AS. Метод CB-AS містить CBS, який спочатку реєструє датчик у ЦС, а потім видає сертифікат. Загалом, методика складається з семи етапів. Етапи налаштування та створення сертифікатів виконуються в ЦС. Під час налаштування створюються загальнодоступні параметри та головний ключ ЦС. На цьому етапі, коли користувач виконує запит, генерується та передається сертифікат, що відповідає ідентифікатору користувача та його відкритому ключу. На етапі генерування ключа користувач створює пару відкритих/особистих ключів, а потім підпис для повідомлень за допомогою сертифіката, відкритого та закритого ключів і його/її особи. На етапі підпису користувач підписує повідомлення та надсилає його на шлюз, який перевіряє повідомлення та підпис. Повідомлення та підписи від

кількох підписантів об'єднуються в одне шлюзом і передаються до хмарного сховища. Верифікатор перевіряє всі повідомлення та агрегований підпис.

Метод CB-AS вразливий до підробки повідомлень і підписів. Відкритий ключ CB-PKC можна автентифікувати за допомогою сертифіката, але зловмисник може замінити відкритий ключ іншого користувача. Атака із заміною відкритого ключа на CBS може підробити підпис, який пристрій A надсилає на пристрій B, і замінити відкритий ключ пристрою A (відкритий для перевірки підпису) відкритим ключем, згенерованим зловмисником. Така атака можлива, оскільки замінений відкритий ключ зловмисника, який може обійти перевірку підпису, створеного за допомогою приватного ключа пристрою A, не може бути автентифікований як відкритий ключ пристрою A. Крім того, ЦС може підробити підпис пристрою A за допомогою сертифіката.

Модель безпеки CB-AS повинна протидіяти двом типам атак. Ці дві моделі схожі на ігри, в яких компетентні зловмисники A_1 та A_2 спілкуються з Challenger (C), щоб успішно підробляти підписи. У моделі оракула Challenger (C) відповідає за обчислення та виконання значення, коли його запитує користувач. A_1 виконує роль стороннього, який може довільно замінити відкритий ключ законного користувача новим ключем, але не знає сертифіката чи головного ключа. A_2 – це зловмисник, який може діяти як зловмисний ЦС або контролювати головний ключ ЦС, але не може замінити відкриті ключі користувачів. Якщо A_1 або A_2 хочуть підробити підписи, підпис можна підробити шляхом багаторазового виконання ряду запитів із типами атаки 1 і 2 (записані нижче). Загрози безпеці, згадані вище, виникають не тільки з CB-AS, але і з CB-KIAS, де забезпечується ізоляція ключа. Якщо порівнювати CB-KIAS і CB-AS, він включає етап оновлення ключа та етап генерації ключа підписання.

Атака безпеки типу I, зловмисник A_1

Атака виконується на основі моделі CB-KIAS, яка забезпечує функцію ізоляції ключа.

Налаштування: Challenger (C) створює головний відкритий/особистий ключ у ЦС та системні параметри шляхом виконання цього етапу.

Генерація ключа: шпигун (зловмисник) надсилає свою особу, щоб отримати ключ. Challenger (C) отримує ключ, створює відкритий ключ PU_i і передає його зловмиснику.

Підміна відкритого ключа: зловмисник може замінити відкритий ключ користувача на PUi. Немає необхідності отримувати закритий ключ користувача. Зловмисник може повторити цю фазу.

Генерація сертифіката: зловмисник запитує автентифікацію для (ID,PU_i), а Challenger (C) надсилає зловмиснику сертифікат, отриманий шляхом виконання цього запиту.

Підписання ключа: зловмисник робить запит на отримання підпису для ключа, а Challenger (C) створює тимчасовий ключ для підпису $TskID_i$,0 і надсилає його зловмиснику A_1.

Створення підпису: зловмисник робить запит на підпис (ID,m), а Challenger (C) виконує генерацію підпису, щоб отримати цей підпис. Потім Challenger (C) надсилає підпис шпигуну та записує відповідь.

Оракул можна використовувати для запитів всіх етапів атаки (все ж залежно від компетенції A_1). Тоді зловмисник може вивести $(ID', PU_i', \sigma', m', t')$.

– A_1 ніколи не робив запит на ідентифікатор через оракул ключа підпису.

– A_1 ніколи не робив запит на (ID', PU_i') через оракул ключа підпису.

– (σ', m', t') є законною парою підробленого повідомлення та підпису, але A_1 ніколи не робив запит на (ID', m', t') через оракул генерації ключа.

Означення 2.1. У типі атаки CB-AS (I), якщо немає зловмисника A_1, який може виграти з достатньою ймовірністю протягом імовірнісного поліноміального часу, екзистенційно неможливо займатися підробкою.

Атака безпеки типу II, зловмисник A_2

Попередні етапи атак для A_1 залишаються і для A_2.

Ключ із Helper: коли зловмисник отримує запит на ключ із Helper, Challenger(C) генерує особистий ключ і відкритий ключ Helper (hsk_1,hsk_2,hpk_1,hpk_2) і надсилає їх підробнику A_2. Так, як A_2 знає головний ключ ЦС, він може здійснити атаку підробки та виконати запити оракула для всіх етапів атаки як для A_1. Потім зловмисник може вивести $(ID', PU_i', \sigma', m')$.

– A_2 ніколи не робив запит на ідентифікатор через оракул ключа підпису.

– (σ', m', t) є допустимою парою підробленого повідомлення та підпису, але A_2 ніколи не робив запит на (ID', m', t) через оракул генератора підпису.

Означення 2.2. У типі атаки CB-KIAS (II) екзистенційно неможливо займатися підробкою за відсутності зловмисника A_2, який може виграти з достатньою ймовірністю в межах імовірнісного поліноміального часу.

2.1.4. Вимоги безпеки

Цілісність та надійність даних: важливо забезпечити цілісність та надійність даних (повідомлень), що передаються до/від датчиків у середовищі Інтернету речей. У існуючих системах CB-AS шлюз окремо перевіряє дані, передані кількома датчиками, а також об'єднує та передає цю інформацію. Користувач, який бажає перевірити дані, повинен перевірити остаточний колективний підпис.

Така перевірка є важливою для забезпечення цілісності/невідмовності повідомлень та надійності пристроїв, які передають та отримують повідомлення.

Непідробність: як описано у попередньому пункті, підробка підпису може статися через атаку підміни відкритого ключа, ініційовану зловмисником A_1, або коли A_2 використовує головний ключ ЦС для створення сертифіката користувача.

A_1 не повинен мати можливість генерувати легітимний підпис навіть за допомогою заміни відкритого ключа, використовуючи дійсний ідентифікатор

користувача. A_2 не повинен мати можливість підробити підпис, навіть якщо цей підпис створено за допомогою ключа та сертифіката підписувача. Тому верифікатор не повинен мати можливість перевірити підроблений підпис.

Запобігання впливу ключа на бічний канал: повідомлення підписуються для забезпечення цілісності та надійності датчика. Тобто ключ підпису підписувача (сенсорного пристрою) не повинен витікати назовні або витягуватися через загальнодоступне значення. Якщо зловмисник може вивести або вкрасти ключ підпису СМС, за допомогою фізичної атаки, наприклад, атаки на бічний канал, зловмисник може підробити підпис згенерованих повідомлень. Це знижує надійність сенсорних пристроїв IoT, і зловмисник може підробити та передати будь-яку кількість повідомлень за допомогою отриманого значення ключа підпису. Тому, застосовуючи такий метод, як ізоляція ключа, ключ підпису підписувача буде постійно оновлюватися.

2.2. Застосування колективного цифрового підпису в середовищах IoD

Ще однією сферою застосуванням КЦЕП є Інтернет дронів (Internet of Drons, IoD) – концептуально новий вид мережевих технологій, який виник в результаті широкого спектру застосувань дронів у різних сферах: військовій сфері, сільському господарстві, управлінні транспортним рухом, природних спостереженнях тощо. Мережі IoD, таким чином, засновані на використанні кількох малих дронів, підключених через інтернет. Така мережа має всі технологічні ресурси, необхідні для автономного виконання поставленого завдання, включаючи комунікаційний модуль для передачі та отримання даних, датчики для збору даних, пам’ять для зберігання даних датчиків і процесори для обчислень.

Мережі IoD зазвичай використовують для програм, які вимагають від користувачів отримувати дані з дронів в режимі реального часу. І тут існує велика

ймовірність того, що через достатню кількість бездротових з'єднань між дронами зловмисник може перехоплювати керування деякими з них, або ж здійснювати атаки на їх мережу, маскуючи свою присутність в мережі за іншою особою (проблеми безпеки та конфіденційності рідко враховуються при розробці малих дронів). Зловмисники, які мають намір порушити заходи безпеки та конфіденційності мережі IoD, мають декілька варіантів здійснення своїх зловмисних намірів. Вони можуть, наприклад, передавати велику кількість запитів на резервування, підслуховувати контрольні повідомлення та/або підробляти обмін інформацією. Для вирішення цієї проблеми потрібна легка криптографічна схема для забезпечення конфіденційності даних, а також схема цифрового підпису для забезпечення цілісності даних, згенерованих дроном у середовищі IoD. Для підвищення ефективності розподілу даних, коли для збору даних із визначеної зони задіяно багато дронів, важливого значення набуває саме колективний підпис, який дозволяє стиснути кілька повідомлень від різних користувачів в один.

Замість того, щоб перевіряти всі окремі підписи, версифікатору КЦЕП потрібно просто перевірити сукупний підпис, що призведе до значного зменшення загальної довжини підписів. В результаті цього, навантаження на мережеву передачу можна мінімізувати, а ефективність перевірки множини підписів можна підвищити, використовуючи схему КЦЕП.

Криптографія на основі гіпереліптичної кривої (КГЕК) з малим розміром ключа забезпечує такий самий рівень безпеки, як і білінійне парування і криптографія з еліптичною кривою (КЕК). Ключові внески запропонованої схеми підсумовані таким чином:

У роботі [6] запропонована модель системи CB-AS (зображена на малюнку 2.3). Дрони-члени (M-дрони), дрони-агрегатори (AGT-дрони), центр сертифікації (ЦС) і базова станція (БС) – це чотири категорії об'єктів у пропонованій системі. M-Дрони забезпечують моніторинг певної зони, а AGT-дрон служить головкою кластера для групи M-дронів, які безпосередньо приєднані до неї.

ЦС відповідає за налаштування та створення сертифікатів. БС, з іншого боку, виконує взаємну автентифікацію перед призначенням завдань обома типам дронів (AGT-донів та М-Дронів). Процес автентифікації запускається БС, що дозволяє дрону-агрегатору перевіряти, засвідчувати та розповсюджувати запити автентифікації на свої М-дрони. Тож, AGT-дрон служить мостом між БС та М-дронами, забезпечуючи обчислювальні та комунікаційні можливості для керування своїм М-дронів в кластері.

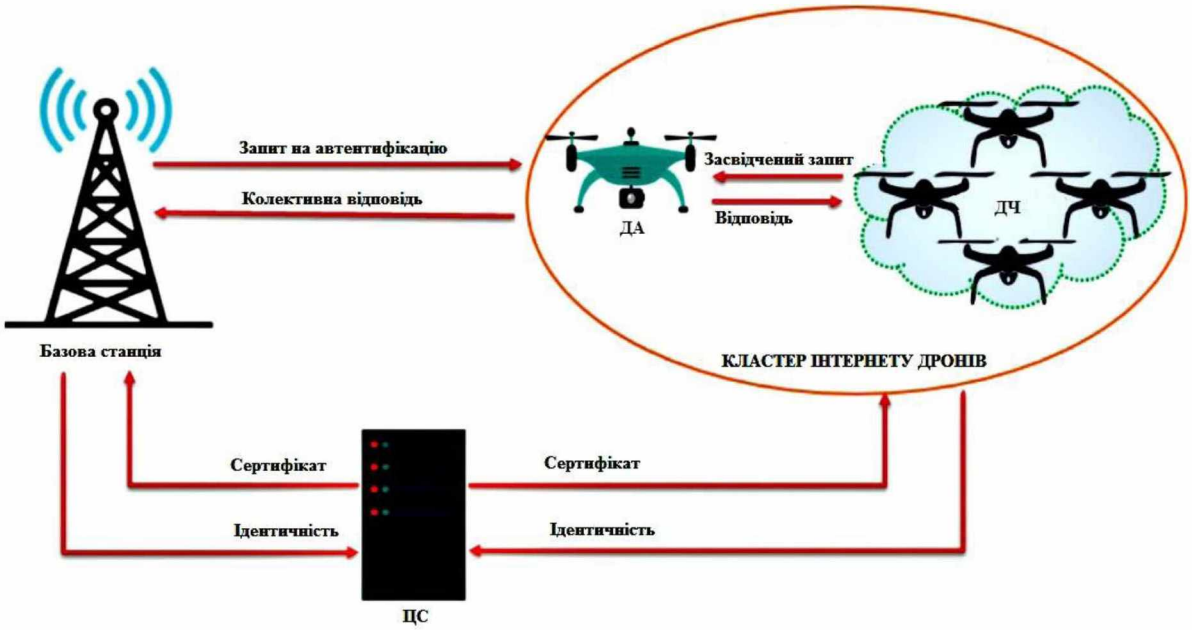


Рис. 2.3. Модель CB-AS системи середовища IoD

РОЗДІЛ 3. ОСОБЛИВОСТІ НОВОЇ РЕАЛІЗАЦІЇ КЦЕП У ТЕХНОЛОГІЇ БЛОКЧЕЙНУ

У цьому розділі, скориставшись певними базовими результатами досліджень, розглянемо приклад нового використання КЦЕП у технології блокчейну.

3.1. Попередні дослідження та результати

З моменту появи криптосистем, зокрема біткойна, блокчейн як основна технологія біткойна привертає все більше уваги.

Як комбінація різноманітних технологій, таких як розподілене зберігання даних, однорангова мережа, механізм консенсусу та криптографічний алгоритм, блокчейн має широкі перспективи у застосуваннях.

Ще до досягнення характеристик блокчейну збереження конфіденційності є центром академічних досліджень. Зокрема, репрезентативними проєктами щодо подібних досліджень є використання так званого кільцевого підпису та інші криптографічні технології (проєкти Monero і Zcach). Крім того, досягнення швидкої торгівлі для задоволення реальних економічних вимог є ще однією проблемою, з якою стикається блокчейн (широко відома мережа Lightning).

Все ж, ще існує багато недоліків, пов'язаних з криптосистемами. В деякій мірі ідеальною технологією для вирішення цих недоліків вважався блокчейн, до якої приєднують різні моделі протоколів захисту [13].

Dash використовує техніку, відому як CoinJoin. В двох словах, Coin Join змішує кілька транзакцій різних користувачів в одну транзакцію через деякі головні вузли.

У Dash кожен користувач вибирає адресу, а потім надсилає її на головний вузол для змішування з іншими адресами. Трансакції можуть здійснюватися лише

з сумою 0,1, 1, 10 і 100, що збільшує складність для зловмисників вгадати релевантність транзакцій за сумою транзакцій.

У Dash все ще існує ризик того, що головні вузли можуть контролюються зловмисниками, що може призвести до розкриття конфіденційності користувачів. Для вирішення цієї проблеми в Monero була запропонована гібридна криптографічна схема, яка не залежить від центральних вузлів. У Monro є дві технології: одна називається прихованою адресою, а інша — кільцевим підписом.

Прихована адреса – це вирішення проблеми релевантності вхідних і вихідних адрес. Кожного разу, коли відправник здійснює транзакцію, буде обчислюватися одноразовий відкритий ключ із використанням КЕК через адресу одержувача. Потім відправник надсилає цей відкритий ключ разом із додатковим повідомленням у блокчейні. І одержувачі можуть виявити кожну транзакцію на основі власного таємного ключа, щоб визначити, чи відправник уже виконав транзакцію.

Коли одержувач хоче використати транзакцію, він може обчислити таємний ключ підпису на основі свого таємного ключа та інформації про транзакцію.

Крім того, Монро запропонувала схему кільцевого підпису.

Щоразу, коли відправник хоче здійснити транзакцію, транзакція буде підписана закритим ключем відправника та відкритими ключами інших користувачів, вибраними випадковим чином. Під час перевірки підпису потрібні відкриті ключі інших користувачів і параметри в підписі.

У Zcash була запропонована нова схема з доказом нульового знання, яка дозволяє користувачам приховувати інформацію про транзакції лише шляхом взаємодії з самим криптографічним алгоритмом, так що всі транзакції створюються однаково.

У Zcash було використано неінтерактивне доведення, яке називається zk-SNARK. Давайте обговоримо найпростіший випадок, припустивши, що сума в Zcash фіксована, наприклад 1 BTC (1 біткоїн).

Тоді процес карбування монет еквівалентний тому факту, що користувач вливає 1 BTC в пул умовного депонування, а потім записує зобов'язання, яке можна розрахувати за серійним номером і приватним ключем користувача до списку. Коли користувач хоче витратити гроші, йому потрібно зробити два кроки:

1) Вказати серійний номер.

2) Задіяти zk-SNARK для доведення того, що він має приватний ключ користувача для створення цього зобов'язання [13].

3.2. Реалізація колективного цифрового підпису в технології блокчейну

Нехай G_1 та G_2 – дві мультиплікативні циклічні групи простого порядку p, а g_1 та g_2 – відповідні їм твірні елементи. Нехай також визначено обчислювальний ізоморфізм φ групи G_2 на групу G_1 як $\varphi(g_2)=g_1$. Криптографічне парування визначене як білінійне перетворення $e: G_1 \times G_2 \to G_T$, де G_T – мультиплікативна група того ж порядку p, яке задовольняє наступні властивості:

1) $\forall u \in G_1 \forall v \in G_2 \forall a,b \in Z_p:\ e\left(u^a, v^b\right) = e(u,v)^{ab}$;

2) $\exists u \in G_1 \exists v \in G_2:\ e(u,v) \neq o$, де o – одиничний елемент групи G_2;

3) $\forall u \in G_1 \forall v \in G_2$ існує ефективний алгоритм для обчислення $e(u,v)$.

Нехай U – множина користувачів, кожен $u \in U$ з яких має пару ключів для підпису (PK_u, SK_u), а $U_1 \subseteq U$ визначає множину користувачів, чиї підписи будуть утворювати ***колективний підпис*** (будуть агреговані). Всі користувачі $u \in U_1$ генерують підписи σ_u для обраного ними повідомлення M_u, а потім ці підписи перегруповуються в єдиний підпис колективною спільнотою, яка не може належати множині U_1, чи якій недовіряють користувачі з U.

Результатом колективного підпису є підпис σ, довжина якого така ж, як і будь-який одиничний підпис кожного користувача. Колективні підписи мають властивість, що верифікатор може переконатися, що кожен користувач підписав відповідні повідомлення, якщо отримано кожне з цих повідомлень та підпис σ.

Нехай дано поле цілих чисел F_q з характеристикою, більшою за 3. Тоді *Еліптичною кривою* E над полем F_q називається множина всіх розв'язків $(x,y)\in F_q\times F_q$ рівняння $y^2=x^3+ax+b$, де $a,b\in F_q$ є такими, що $4a^2+27b^2\neq 0$, разом зі спеціальною точкою ∞, яка називається точкою на нескінченності. Відомо, що E — абелева група, одиничним елементом якої є точка ∞. Правила групового додавання подамо нижче

1) якщо точка P еліптичної кривої E має координати (x_1,y_1); то протилежна їй точка $-P$ має координати $(x_1,-y_1)$;

2) якщо разом з точкою $P(x_1,y_1)$ цій же еліптичній кривій E належить точка $Q(x_2,y_2)$ така, що $Q\neq -P$, то $P+Q=(x_3,y_3)$, де

$$\begin{aligned} x_3&=\lambda^2-x_1-x_2,\\ y_3&=\lambda(x_1-x_3)-y_1, \qquad ,\\ \lambda&=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1}, \text{якщо } P\neq Q,\\ \dfrac{3x_1^2+a}{2y_1}, \text{якщо } P=Q\end{cases}\end{aligned} \tag{3.1}$$

Якщо F_q є полем характеристики 2, то *еліптичною кривою E нульового j-інваріанту* над F_q називається множина всіх розв'язків $(x,y)\in\overline{F_q}\times\overline{F_q}$ рівняння $y^2+cy=x^3+ax+b$, де $a,b,c\in F_q$, причому $c\neq 0$, разом із точкою ∞ на нескінченності. Відомо, що E знову э абелевою групою з такими правилами групового додавання:

1) якщо точка P еліптичної кривої E має координати (x_1,y_1); то протилежна їй точка $-P$ має координати (x_1,y_1+c);

2) якщо разом з точкою $P(x_1,y_1)$ цій же еліптичній кривій E належить точка $Q(x_2,y_2)$ така, що $Q\neq -P$, то $P+Q=(x_3,y_3)$, де

$$x_3 = \begin{cases} \left(\dfrac{y_2 - y_2}{x_1 + x_2}\right)^2 + x_1 + x_2, \text{якщо } P \neq Q, \\ \dfrac{x_1^4 + a^2}{c^2}, \text{якщо } P = Q; \end{cases}$$

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + y_1 + c, \text{якщо } P \neq Q, \\ \left(\dfrac{x_1^2 + a}{c}\right)(x_1 + x_3) + y_1 + c, \text{якщо } P = Q. \end{cases} \quad (3.2)$$

Якщо F_q є полем характеристики 2, то *еліптичною кривою* E *ненульового* j*-інваріанту* над F_q називається множина всіх розв'язків $(x, y) \in \overline{F_q} \times \overline{F_q}$ рівняння $y^2 + xy = x^3 + ax + b$, де $a, b, c \in F_q$, причому $c \neq 0$, разом із точкою ∞ на нескінченності. Відомо, що E знову э абелевою групою з такими правилами групового додавання:

1) якщо точка P еліптичної кривої E має координати (x_1, y_1); то протилежна їй точка $-P$ має координати $(x_1, y_1 + x_1)$;

2) якщо разом з точкою $P(x_1, y_1)$ цій же еліптичній кривій E належить точка $Q(x_2, y_2)$ така, що $Q \neq -P$, то $P + Q = (x_3, y_3)$, де

$$x_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a, \text{якщо } P \neq Q, \\ x_1^2 + \dfrac{b}{x_1^2}, \text{якщо } P = Q; \end{cases}$$

$$y_3 = \begin{cases} \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + y_1 + x_3, \text{якщо } P \neq Q, \\ x_1^2 + \left(x_1 + \dfrac{y_1}{x_1}\right)x_3 + x_3, \text{якщо } P = Q. \end{cases} \quad (3.3)$$

Коли транзакції генеруються на блокчейні, криптографічні підписи використовуються для оцінки законності транзакцій та ідентифікації відправників. Крім того, алгоритми підпису спрямовані на збереження конфіденційності транзакцій, включаючи адреси обох сторін та суму транзакції. Наприклад, у біткойнах для підписів транзакцій використовуються алгоритми ECDSA, RIPEMD та SHA256.

Без втрати загальності, нехай ми маємо справу з однією транзакцією, яка розділена на входи та виходи (див. рис. 3.1.) [13].

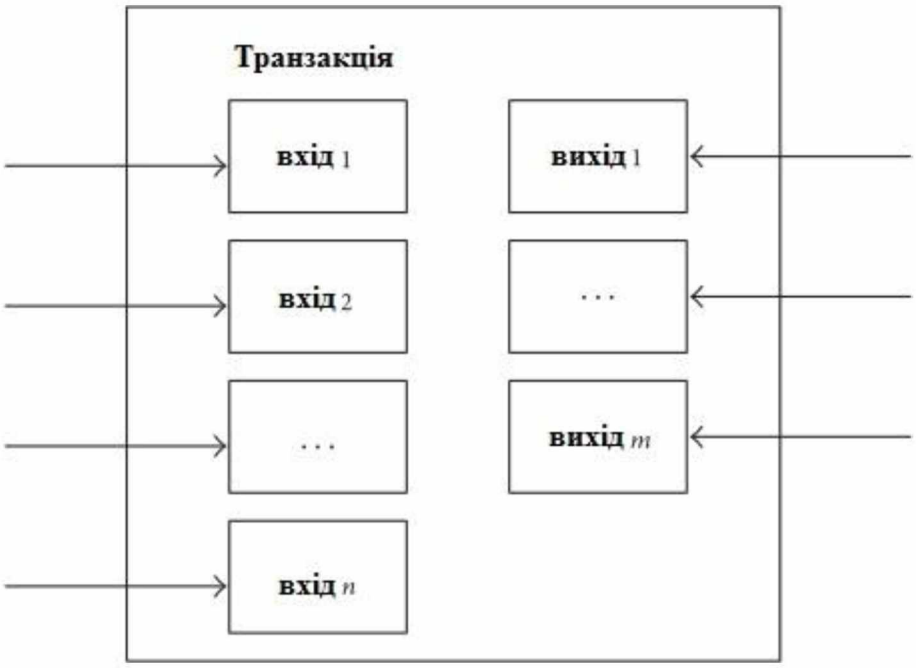


Рис. 3.1. Модель єдиної транзакції

Як показано на рис. 3.1, транзакція містить n входів та m виходів. Нехай відповідні обсяги входів транзакції дорівнюють d_i $(1 \le i \le n)$, а відповідні обсяги її виходів дорівнюють s_j $(1 \le j \le m)$, тобто має виконуватися рівність $\sum_{i=1}^{n} d_i = \sum_{j=1}^{m} s_j$.

Для того, щоб приховати значення d_i та s_j для кожних i та j, документ транзакції використовує КЕК. У якості твірного елемента поля F_p виберемо

значення G, а за форми передачі відповідних входів та виходів візьмемо значення $I_i = d_i G$, $O_j = s_j G$. За правилами виконання операцій з еліптичною кривою (2.1)–(2.3) справедливі такі рівності:

$$\begin{aligned} \sum_{i=1}^{n} d_i G = \sum_{i=1}^{n} I_i = \left(\sum_{i=1}^{n} d_i \right) \cdot G, \\ \sum_{j=1}^{m} s_j G = \sum_{j=1}^{m} O_j = \left(\sum_{j=1}^{m} s_j \right) \cdot G \end{aligned} \qquad (2.4)$$

Оскільки зловмисники не можуть отримати значення входів d_i та виходів s_i, маючи лише значення добутків за еліптичною кривою (2.4) (неможливо знайти невідомий числовий множник), сума транзакції може бути прихована за допомогою цієї схеми. Але ця схема має недолік: базова схема задовольняє адитивний гомоморфізм, який є важливою властивість для оцінки безпеки алгоритму, особливо зважаючи на швидкий розвиток квантових комп'ютерів.

У системі біткойн існують зрілі алгоритми атак, такі як *егоїстична майнінгова атака*, *атака затемнення* чи *атака на впертий майнінг*. Подібні недоліки є і в розглянутій базовій схемі, яку розглянемо далі.

Нехай еліптична крива на скінченній групі F_p задається набором (p,a,b,G,n), де $G = \{g_1, g_2\}$ – твірний елемент поля F_p, $n \cdot G = o$. Модифікована схема буде виконуватися за таким алгоритмом:

1) Обчислюємо значення $I_i = d_i \cdot G, i = \overline{1,n}$, $O_j = s_j \cdot G, j = \overline{1,m}$.

2) Для кожного $i \left(i = \overline{1,n} \right)$ випадковим чином вибираємо $\tilde{d}_i \in Z_p$ і обчислюємо $D_i = \tilde{d}_i \cdot G$, $h_i^{(1)} = H\left(D_i \| d_i\right)$ (тут H – деяка оборотна геш-функція від результату дії звичайного логічного оператора «OR») та $\tilde{D}_i = \tilde{d}_i \cdot h_i^{(1)} + d_i$. Аналогічно випадковим чином обираємо $\tilde{s}_j \in Z_p$ і обчислюємо $S_j = \tilde{s}_i \cdot G$,

$h_j^{(2)} = H\left(S_j \| s_j\right)$ та $\tilde{S}_j = \tilde{s}_j \cdot h_j^{(2)} + s_j$. Тоді формами передачі входів і виходів будуть значення $\sum_{i=1}^{n} \tilde{D}_i$ та $\sum_{j=1}^{m} \tilde{S}_j$ відповідно.

Доцільність модифікованої схеми отримується з обчислювальних значень, за якими, з урахуванням форм передачі, ми маємо таку справедливу рівність:

$$\sum_{i=1}^{n} h_i^{(1)} D_i - \sum_{i=1}^{n} \tilde{D}_i G = \sum_{j=1}^{m} h_j^{(2)} S_j - \sum_{j=1}^{m} \tilde{S}_j G.$$

Таким чином, ми запропонували нову схему, яка спрямована на приховування суми транзакцій на блокчейні, які містять кілька входів і виходів. Виходячи з цього, можна розробити нову схему підпису, яка може захистити транзакції та зберегти розмір підписів незмінним незалежно від кількості входів і виходів. Нагадаємо, що еліптична крива E над скінченним полем F_p задається набором (p,a,b,G,n). Базисними групами поля є групи G_1 та G_2 з відповідними твірними елементами g_1 та g_2, обчислюваний ізоморфізм φ діє з групи G_2 на групу G_1, а білінійне відображення – $e: G_1 \times G_2 \to G_T$ з цільовою групою G_T.

Нехай $H_s : \{0,1\}^* \to F_q, H_p : E\left(F_q\right) \to E\left(F_q\right)$.

Опишемо нову схему ЦЕП:

Генерація ключів. Конкретний користувач вибирає довільним чином $x \overset{R}{\leftarrow} Z_p, a \in E$ і обчислює $v = g_2^x, A = aG$. Відкритим ключем підпису користувача та закритим ключем підпису є $v \in G_2$ та $x \in Z_p$ відповідно. Для здійснення ж транзакції відкритим ключем та закритий ключем будуть $A \in E$ та $a \in E$.

Підписання. Ми припускаємо, що відправник по відношенню до конкретного одержувача хоче здійснити транзакцію (наприклад відправити платіж), відкритим ключем транзакції якого є B. Відправник генерує випадковий $r \in [1, n-1]$ і обчислює одноразовий відкритий ключ $P = H_s\left(rB\right)G + A$, а потім

обчислює $\sigma = P^x$. Підписом буде $\sigma \in G_1$. Значення $R = rG$ також буде розміщено десь у транзакції.

Перевірка. Враховуючи відкритий ключ транзакції v відправника та його підпис σ, одержувач обчислює $P' = H_s(b \cdot R) \cdot G + A$, а потім приймає, якщо має місце рівність $e(\sigma, g_2) = e(P', v)$.

Ми знаємо, що $b \cdot R = b \cdot r \cdot G = r \cdot B$; тоді ж $P' = P$. І за правилами білінійних відображень отримуємо, що $e(\sigma, g_2) = e(P^x, g_2) = e(P, g_2^x) = e(P', v)$. На рис. 3.2 представлена структура описаної вище базової схеми підпису [13].

Рис. 3.2. Базова розширена схема ЦЕП для транзакцій

Щоб досягти мети покращення ефективності схеми підпису, тепер поєднаємо колективний підпис з нашою базовою схемою підпису та запропонуємо модифіковану схему підпису.

Генерація ключів. Для колективної підмножини користувачів $U_1 \subseteq U$ кожному користувачеві призначаємо індекс i з діапазону від 1 до $k = |U_1|$. Кожен користувач довільним чином вибирає $x_i \xleftarrow{R} Z_p$, $a_i \in E$ та обчислює $v_i = g_2^{x_i}, A_i = a_i \cdot G$. Відкритим ключем підпису та закритим ключем підпису

користувача u_i є відповідно значення $v_i \in G_2$ та $x_i \in Z_p$. Відкритий ключ транзакції та закритий ключ транзакції користувача u_i – це значення $A_i \in E$ та $a_i \in E$ відповідно.

Підписання. Для кожного $i\left(i=\overline{1,k}\right)$ ми припускаємо, що користувач u_i по відношенню до конкретного одержувача хоче здійснити транзакцію a, відкритим ключем транзакції якого є B_i. Потім u_i генерує випадкове $r_i \in [1, n-1]$ та обчислює одноразовий відкритий ключ $P_i = H_s\left(r_i B_i\right)G + A_i$, а потім обчислює $\sigma_i = P_i^{x_i}$. Підписом і буде значення $\sigma_i \in G_i$, причому значення $R_i = r_i \cdot G$ також буде поміщене десь у транзакції.

Агрегація. Обчислюємо $\sigma \leftarrow \prod_{i=1}^{k} \sigma_i$; колективний підпис і буде виражатися числом $\sigma \in G_1$.

Колективна перевірка. Ми отримуємо колективний підпис $\sigma \in G_1$ для агрегованої проіндексованої підмножини користувачів $U_1 \subseteq U$ та отримуємо оригінальний $P_i = H_s\left(r_i B_i\right)G + A_i$ і відкритий ключі для всіх користувачів $u \in U_1$. Для того, щоб перевірити колективний підпис σ, нам потрібно обчислити $P_i' = H_s\left(b_i \cdot R_i\right) \cdot G + A_i\left(i=\overline{1,k}\right)$ та прийняти його, якщо виконується рівність

$$e\left(\sigma, g_2\right) = \prod_{i=1}^{k} e\left(P_i', v_i\right). \qquad (2.10)$$

Використовуючи властивості білінійного відображення, ліву частину рівності (2.10) можна певним чином перетворити. Так, матимемо:

$$e\left(\sigma, g_2\right) = \prod_{i=1}^{k} e\left(\prod_{i=1}^{k} \sigma_i, g_2\right) = \prod_{i=1}^{k} e\left(\prod_{i=1}^{k} P_i^{x_i}, g_2\right) = \prod_{i=1}^{k} e\left(P_i^{x_i}, g_2\right) =$$

$$= \prod_{i=1}^{k} e\left(P_i', g_2^{x_i}\right) = \prod_{i=1}^{k} e\left(P_i', v_i\right) \qquad (2.11)$$

На малюнку 3.3 представлена структура нашої моделі транзакції з колективним підписом.

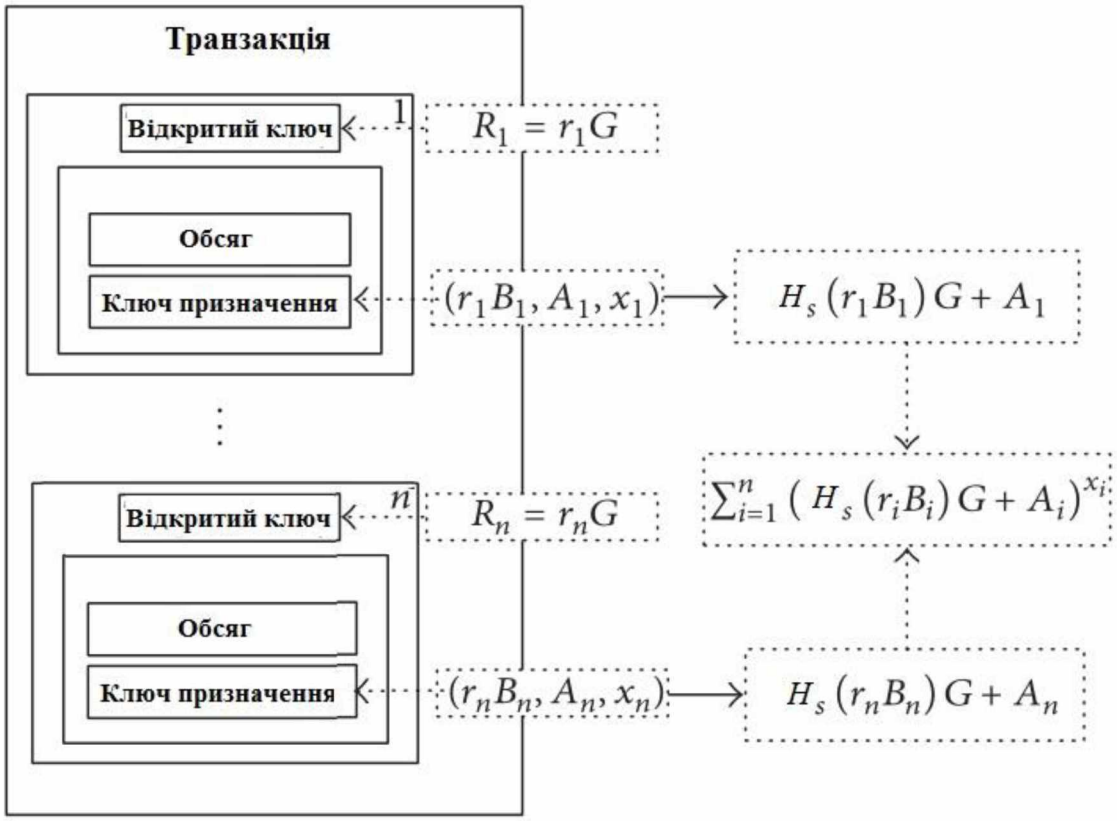


Рис. 3.3. Модель транзакції з колективним підписом

Як показано на малюнку 3.3, підпис залишається незмінним незалежно від кількості входів і виходів, які містить транзакція.

3.3. Безпека нової схеми підпису.

Легко показати, що безпека нашої нової схеми підпису еквівалентна традиційному білінійному колективному підпису. Як сукупна модель безпеки вибраного ключа, яка була запропонована в [13], безпека сукупних схем підпису еквівалентна неспроможністю зловмисника екзистенційно підробити колективний підпис. *Екзистенційна підробка* тут означає, що зловмисник намагається підробити колективний підпис на субтранзакціях за його вибором за допомогою

інших субтранзакцій у конкретній транзакції. Зловмиснику A надається лише відкритий ключ. Його мета – екзистенційна підробка колективного підпису, але йому надано право вибирати всі відкриті ключі, крім відкритого ключа запиту. Супротивнику також надається доступ до оракула підпису на ключі виклику. Його перевага Π_A визначається як ймовірність успіху в наступній грі.

Налаштування. Підроблювачу A колективного підпису надається відкритий ключ PK_1, згенерований випадковим чином.

Запити. Використовуючи ключ PK_1, A продовжує виконувати адаптивні запити на підписи у субтранзакціях за його вибором.

Відповідь. A виводить $k-1$ додатковий відкритий ключ PK_1, …, PK_k. Ці ключі разом із початковим ключем будуть включені до підробленого колективного підпису A. A також виводить субтранзакції $T_1,...,T_k$, і відповідний колективний підпис σ користувачів k, кожен для своєї відповідної субтранзакції.

Зловмисник виграє, якщо колективний підпис σ є дійсним колективним підписом для субтранзакцій $T_1,...,T_k$ під відповідними ключами PK_1, …, PK_k, і σ є нетривіальним.

Означення 3.1. Підробник колективного підпису $A(t,q_H,q_S,N,\varepsilon)$ порушує схему колективного підпису N-того користувача в моделі колективного обраного ключа, якщо виконуються такі умови:

1) t – найбільший час виконання A;

2) q_H – найбільша кількість запитів A до геш-функції, а q_S – найбільша кількість запитів A до підписувального оракула;

3) ε – найменша ймовірність порушення безпеки схеми колективного підпису;

4) підроблений колективний підпис, отриманий щонайбільше для N користувачів.

Схема підпису є $(t,q_H,q_S,N,\varepsilon)$-безпечною щодо екзистенційної підробки в моделі колективного вибраного ключа, якщо жодна підробка $(t,q_H,q_S,N,\varepsilon)$ її не порушує. Також має місце наступна теорема, яка показує, що для підтвердження безпеки в моделі з вибраним колективним ключовим підписом цього достатньо врахувати просте обмеження.

Теорема 3.1. *Нехай* (G_1,G_2) *є* (t',ε')*-білінійна групова пара з умовами Діффі-Хеллмана, кожна група в якій має порядок* p*, та відповідні твірні елементи* g_1 *і* g_2*, з обчислювальним ізоморфізмом групи* G_2 *на групу* G_1 *та з білінійним відображенням* $f:G_1\times G_2\to G_T$*. Тоді білінійна схема колективного підпису на* (G_1,G_2) *є* $(t,q_H,q_S,N,\varepsilon)$*-захищеною від екзистенційної підробки в моделі колективного вибраного ключа для тих і тільки тих значень параметрів* t *та* ε*, які задовольняють умови*:

1) $\varepsilon\geq e(q_S+N)\cdot\varepsilon'$;

2) $t\leq t'-c_{G_1}(q_H+2q_S+N+4)-(N-1)$.

Тут e – це математична стала.

Враховуючи наведені на початку пункту доведення безпеки схеми, яка використовується для приховування суми транзакцій, ми можемо отримати, що наша схема підпису задовольняє властивість непідробленості та інші властивості безпеки.

3.4. Додаткові характеристики у запропонованій схемі

Час колективного підписання. В одному підписі реалізовано одну операцію гешування, одне модульне множення потужності та одну операцію множення. Нехай σ – колективний підпис складений з n підписів користувачів

$\sigma_1,...,\sigma_n$. Час перевірки цього підпису σ є лінійно залежним від n. І реалізовано одне множення з агрегацією.

Загальний час перевірки. За одну перевірку реалізовано k операцій гешування та $n+1$ операція з білінійними відображеннями. Якщо σ – колективний підпис складений з n підписів користувачів $\sigma_1,...,\sigma_n$. Час перевірки цього підпису σ є лінійно залежним від n.

Простір підписів. Нехай знову σ – колективний підпис складений з n підписів користувачів $\sigma_1,...,\sigma_n$. Простір підписів становитиме $1/n$ від нормального підпису.

ВИСНОВКИ

На сьогодні існує величезна кількість різних потужних криптографічних методів та засобів захисту переданої інформації. Але якщо вам потрібен ще більш захищений варіант, з можливістю аналогічного шифрування даних, та з кращим захистом, то цифровий електронний підпис один із кращих варіантів

До переваг використання ЦЕП можна віднести:

- Удосконалення бізнес-процесів на підприємстві та економія ресурсів. Достатньо сильно зменшує обсяги паперової документації. Зменшує затрати підприємства та час робітників, зв'язані з укладанням договорів, оформленням платіжних документів.
- Використання ЦЕП дуже просте і доступне кожній людині незалежно від рівня користування ПК, освіти та роду діяльності.
- Відсутність можливості доступу до неї будь-кого, хто не володіє секретним кодом завдяки надійним криптографічним перетворенням.
- Безпечність використання ЦЕП обумовлюється тим, що надійні засоби, які використовуються для роботи з ЦЕП, перевіряються в Державній службі спеціального зв'язку та захисту інформації України.
- Можливість ведення документообігу з державними структурами без безпосередньої присутності. Зручність в обміні даними з усіма гілками влади, при передачі звітності у всі контролюючі органи, що займаються звітами в електронній формі.
- Ведення ділових відносин без додаткових зустрічей і багатогодинних переговорів.

Зростання популярності використання ЦЕП, тільки за даними Міністерства цифрової трансформації України за 2020 рік кількість осіб яким було видано ЦЕП зросла на 69% в порівнянні з 2019, а саме 7.2млн відповідно до 4,3м.

Виконуючи поставлені завдання, було проведено аналіз колективного цифрового підпису та особливостей його використання, зокрема показано, що використання КЦЕП

- забезпечує кращий захист інформації, її цілісність та доступність;
- частково вирішує проблеми з депонуванням та розподілом ключів користувачів;
- дозволяє отримати більшу пропускну та обчислювальну здатність систем та зменшення витрат.

Самостійним внеском у цій роботі можна вважати:

- систематизація знань про цифрові електроні підписи та їх використання, зокрема юридичні аспекти ЦЕП;
- вивчення наукових праць щодо використання КЦЕП;
- отримання прикладу моделі застосування КЦЕП та її теоретичної перевірки на надійність.

ЛІТЕРАТУРА

1. European patent application [Електронний ресурс] – Режим доступу: https://worldwide.espacenet.com/patent/search/family/039156776/publication/EP2680046A1?q=13185533.0
2. Goldwasser Sh. A digital signature scheme secure against adaptive chosen-message attacks / Sh. Goldwasser, S. Micali, R. Rivest // SIAM Journal on Computing, 17(2), P. 281–308, Apr. 1988.
3. Hohenberger S. Synchronized Aggregate Signatures from the RSA Assumption / Susan Hohenberger, Brent Waters // [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/324109825_Synchronized_Aggregate_Signatures_from_the_RSA_Assumption.
4. Hohenberger S. Universal Signature Aggregators / S. Hohenberger, V. Koppula, B. Waters // Lecture Notes in Computer Science book series – Vol. 9057 – 2015.
5. Ishida A. Proper Usage of the Group Signature Scheme in ISO/IEC 20008-2 / A. Ishida, Yu. Sakai, K. Emura, G. Hanaoka, K. Tanaka // 5 July 2019, (https://www.researchgate.net/publication/334331763_Proper_Usage_of_the_Group_Signature_Scheme_in_ISOIEC_20008-2)
6. Khan M. A. An efficient certificate-based aggregate signature scheme for Internet of Drones / M. A. Khan, Ullah I., Alsharif M. H., Alghtani A. H., Aly A. A., Chen Ch.-M. // Security and Communication Networks – Vol. 2022, Article ID 9718580, 9 p. (https://doi.org/10.1155/2022/9718580).
7. Kuchta V. Distributed Protocols for Digital Signatures and Public Key Encryption // [Електронний ресурс] – Режим доступу: https://openresearch.surrey.ac.uk/esploro/outputs/doctoral/Distributed-protocols-for-digital-signatures-and-public-key-encryption/99514478502346.
8. Lu S. Sequential Aggregate Signatures, Multisignatures, and Verifiably Encrypted Signature Without Random Oracles / S. Lu, R. Ostrovsky, A. Sahai //

[Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/257334316_Sequential_Aggregate_Signatures_Multisignatures_and_Verifiably_Encrypted_Signatures_Without_Random_Oracles.

9. Mykletun. E. Signature Bouquets: Immutability for Aggregated/Condensed Signatures / E. Mykletun, M. Narasimha, G. Tsudik // Lecture Notes in Computer Science book series, Vol. 3193 (doi: 10.1007/978-3-540-30108-0_10).
10. Selvi S. Sh. D. Security Analysis of Aggregate signature and Batch verification signature schemes / S. Sh. D. Selvi, S. S. Vivek, J. Shriram, S. Kalaivani, C. P. Rangan // [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/220333287_Security_Analysis_of_Aggregate_signature_and_Batch_verification_signature_schemes.
11. Shin W. Flexible video authentication based on aggregate signature / W. Shin, Y.-J. Hong, W.-Y. Lee, K.-H. Rhee // Journal of Korea Multimedia Society – Vol. 12, No 6, – 2009, pp. 833-841.
12. Wu. Y. Aggregating signatures of MPEG-4 elementary streams // [Електронний ресурс] – Режим доступу: https://ieeexplore.ieee.org/document/1521390.
13. Yuan Ch. Research on signature scheme on blockchain / Ch. Yuan, M. Xu, X. Si // Security and Communication Networks – Vol. 2017, Article ID 4746586, 10 p. (doi:10.1155/2017/4746586).
14. Верховна рада України [Електронний ресурс] – Режим доступу: https://zakon.rada.gov.ua/laws/show/852-15#Text
15. Застосування електронного підпису [Електронний ресурс] – Режим доступу: https://pidruchniki.com/1122121360612/dokumentoznavstvo/zastosuvannya_elektronnogo_pidpisu.
16. Кольбашенко Д. А. Цифровий електронний підпис / Д. А. Кольбашенко [Електронний ресурс] // Вісник ЛНУ імені Тараса Шевченка. – 2014. – № 2. – С. 234. Режим доступу: http://nbuv.gov.ua/node/554

17. Линник О. В. Виявлення підробки цифрового електронного підпису для встановлення змін у документі / О. В. Линник [Електронний ресурс] // Юридичний науковий електронний журнал. – 2015. – № 2. – С. 209–211. Режим доступу: http://www.lsej.org.ua/2_2015/59.pdf
18. Сеанс телефонного зв'язку «гаряча лінія» на тему: «Удосконалення умов та порядок видачі ключів ЦЕП» [Електронний ресурс] – Режим доступу:http://dp.sfs.gov.ua/media-ark/news-ark/print-130990.html
19. Цифровий електронний підпис [Електронний ресурс] // 2010 – Режим доступу: https://wiki.tntu.edu.ua
20. Цифровий електронний підпис: що треба про нього знати [Електронний ресурс] // 2016 – Режим доступу: https://bcp.org.ua/novosti/elektronnij-tsifrovij-pidpis-shho-treba-pro-nogo-znati
21. Шматко О. В. Застосування криптографії в блокчейн [Електронний ресурс] – Режим доступу: https://pns.hneu.edu.ua/course/view.php?id=5681#section-3
22. Юридичні аспекти використання цифрового електронного підпису // [Електронний ресурс] – Режим доступу: https://dealssign.com/blog/yuridichni-aspekti-vikoristannya-elektronnogo-cifrovogo-pidpisu/