НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК 681.3.06

«До захисту допущено»
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2025 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Методи підвищення анонімності криптовалют на основі криптографічних порогових протоколів розподілу секрету»**

Виконав:
студент IV курсу, групи ФІ-13
Бондаренко Олександр Сергійович _____

Керівник:
професор кафедри ММЗІ, д.т.н., с.н.с
Кудін Антон Михайлович _____

Рецензент:
Доцент кафедри ММАД, к.ф-м.н.
Терещенко Іван Миколайович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ — 2025

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2025 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Бондаренко Олександр Сергійович

1. Тема роботи: «*Методи підвищення анонімності криптовалют на основі криптографічних порогових протоколів розподілу секрету*», науковий керівник дисертації: професор кафедри ММЗІ, д.т.н., с.н.с Кудін Антон Михайлович,

затверджені наказом по університету №__ від «__» _____ 2025 р.

2. Термін подання студентом роботи: «__» _____ 2025 р.

3. Об'єкт дослідження: *Анонімність у криптовалютах*

4. Предмет дослідження: *Схеми розподілу секрету для збереження анонімності у криптовалютах*

5. Перелік завдань: *дослідження наявних та побудова власного методу збереження анонімності при транзакціях пза допомогою схем розподілу секрету*

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: *Презентація доповіді*

7. Орієнтовний перелік публікацій: -

8. Дата видачі завдання: 10 вересня 2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2024 р.	Виконано
2	Пошук та аналіз наукових джерел, огляд літератури	Вересень-жовтень 2024 р.	Виконано
3	Аналіз існуючих методів деанонімізації в криптовалютах	16 - 30 листопада 2024 р.	Виконано
4	Аналіз методів збереження анонімності (кільцеві підписи, VPN, ZKP тощо)	грудень 2024 р.	Виконано
5	Дослідження та реалізація схем розподілу секрету (Шаміра, Фельдмана, Педерсона)	20-31 січня 2025р.	Виконано
6	Побудова власного протоколу на основі схеми Педерсона	лютий-квітень 2025р.	Виконано
7	Оцінка ефективності протоколу, порівняння з аналогами (ZKP тощо)	травень 2025р.	Виконано
8	Оформлення дипломної роботи	травень-червень 2025р.	Виконано

Студент _____ Олександр БОНДАРЕНКО

Керівник _____ Антон КУДІН

РЕФЕРАТ

Кваліфікаційна робота містить: 47 сторінки, 3 рисунки, 11 джерел.

Метою цієї роботи є покращення методів захисту анонімності у криптовалютах за допомогою схем порогового розподілу секрету, які дозволять витрачати меншу кількість ресурсів користувачів, що дозволить їх використання в більшій кількості рішень, ніж є наразі.

Об'єктом дослідження є анонімність у криптовалютах.

Предмет дослідження тут методи забезпечення анонімності у криптовалютах, криптовалюти, протоколи порогового розподілу секрету.

У цій роботі розглянуто атаки на анонімність, методи захисту анонімності такі, як кругові підписи та схеми порогового розподілу секрету, протоколи цибулевого підключення. Розглянуті дослідження повинні покращити розуміння нових напрацювань, які представленні в кінці, що і були задачею цієї роботи. Побудований новий протокол представлений в розділі 4.3.

Ключові слова: БЛОКЧЕЙН, АСИМЕТРИЧНА КРИПТОГРАФІЯ, СХЕМИ ПОРОГОВОГО РОЗПОДІЛУ СЕКРЕТУ, АНОНІМНІСТЬ

ABSTRACT

Qualification work contains: 47 pages, 3 figures, 11 sources.

The aim of this work is to improve the methods of anonymity protection in cryptocurrencies using verifiable secret sharing schemes that will allow users to spend less resources, which will allow their use in more solutions than currently available.

The object of research is anonymity in cryptocurrencies.

The subject of research here is methods of ensuring anonymity in cryptocurrencies, cryptocurrencies, verifiable secret sharing protocols.

This paper considers anonymity attacks, methods of protection anonymity protection methods such as circular signatures and threshold distribution schemes secret, and onion connection protocols. The reviewed research should improve the understanding of the new developments presented in the the end, which was the objective of this work. A new protocol was constructed is presented in section 4.3.

Keywords: BLOCKCHAIN, ASYMMETRIC CRYPTOGRAPHY, VERIFIABLE SECRET SHARING SCHEMES, ANONYMITY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ФТІ — Фізико-технічний інститут

РМТ — розріджене дерево Меркла

ZKP — zero-knowledge proof

БХП — Бірюков, Пустогаров, Пустогаров

СРС — схема розподілення секрету

ВСТУП

Актуальність дослідження. Актуальність данного дослідження обумовленна такими факторами, як зниження довіри користувачів до банків, а точніше банківської таємниці, що в свій час приводить до підвищення кількості бажаючих користуватися криптовалютами. Саме ці користувачі, як ми можемо зрозуміти зі слів вище, потребують покращенних методів забезпечення анонімності, що ми і розглядаємо в цій роботі.

Метою дослідження є побудова середовища, яке забезпечує анонімні перекази між користувачами, що буде досягатися за допомогою **задачі дослідження**, яка полягає у створення протоколу анонімізації на базі схеми розподілу секрету, який краще від існуючих за рахунок зменшення витрат ресурсів, які вимагаються від користувача та власника мережі. Для розв'язання задачі необхідно вирішити такі завдання:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) проаналізувати наявні методи;

3) поекспериментувати з цими методами;

4) на основі пункту 3 створити власне рішення, яке відмінне від наявних.

Об'єктом дослідження є Анонімність у криптовалютах.

Предметом дослідження є методи забезпечення анонімності у криптовалютах, криптовалюти, схеми порогового розподілу секрету.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: і тут коротенький перелік (наприклад, але не обмежуючись: теорії складності алгоритмів, методи програмування смарт-контрактів над блокчейном, аналізи графів зв'язків всередині блокчейну, аналіз зв'язку вхід-вихід, аналіз протоколу конфіденційності і пошук шляхів деаномізації за допомогою сторонніх джерел(реальне життя чи проблеми з реалізацією).

Наукова новизна отриманих результатів полягає у створенні нового

протоколу, який вперше поєднає підхід затирання реєстру блокчейна зі схемою Педерсона, що кардинально зменшить використання ресурсів на фоні наявних схем побудованих на основі технології Zero-knowledge proof.

Практичне значення результатів полягає у створенні простого та легкого у використанні протоколу переказів коштів, який зможе задовольнити пересічного користувача.

1 НАЯВНІ МЕТОДИ ДЕАНОНІМІЗАЦІЇ

Зазвичай поняття анонімності доволі розмите розуміння, хоч і звучить просто. Візьмемо за основу д0ля розгляду систему BitCoin з його поняттям адрес, про які поговоримо пізніше. Спробуємо формалізувати його нижче:

– аналітик не повинен мати можливість зв'язати всі можливі адреси користувача в одне ціле.

– аналітик не повинен мати можливість зв'язати попередній пункт з реальною особистістю користувача.

На основі цього можемо сказати:

Означення 1.1. Анонімність - це неможливість прив'язати знайти та прив'язати всю множину адрес до фізичної(чи юридичної) особи, яка володіє гаманцем.

Про методи втрати анонімності та їх формалізацію буде розглянути нижче, бо без розгляду вразливостей немає сенсу розповідати про захист.

1.1 Атаки на анонімність

Для початку розглянемо основу на яку будемо робити атаку, у нашому випадку - BitCoin. У ньому користувач має безліч адрес, які можна показати як $U = u_1, u_2, ...u_n$. Якщо ж ми знайдемо цю множину, то ми отримаємо повну історію транзакцій користувача з блокчейну. Деаномізацію провести нам вийде якщо ми зв'яжемо біткоїн-адрессу та детерміновану U, а також якщо ми її зможемо довести до реальної особистості, яка існує в нашому світі. Перший крок можна виконати двома шляхами: аналітика графа транзакцій і аналітика самої мережі Біткоїну. Перший шлях описано в роботі Бірюкова, Ховратовича та Пустогарова які показали, що при використанні відмінних біткоїн адрес

задача анономізації стає майже неможливою, бо в деяких випадках вузли графу можуть і не мати спільних ребер взагалі. Спробуємо розглянути атаку, яку вони побудували, нижче.

1.1.1 Атака БХП

Бірюков, Ховратович і Пустогаров запропонували атаку, яка демонструє, що навіть за умов псевдонімності, яку забезпечує протокол Bitcoin, мережевий рівень може стати джерелом інформації про реальні IP-адреси користувачів. Це суттєво підриває уявлення про анонімність у мережі Bitcoin та підкреслює необхідність забезпечення конфіденційності вже на рівні P2P-з'єднань.

Розглянемо алгоритм атаки покроково[2]:

1)Формування списків

На цьому етапі аналітик формує список вузлів, надсилаючи до всіх доступних вузлів мережі запити типу GETADDR. У відповідь надходять повідомлення з переліком адрес, кожну з яких можна перевірити на наявність активного з'єднання з Інтернетом шляхом встановлення TCP-з'єднання та надсилання повідомлення типу VERSION. Якщо відповідь отримано, відповідний вузол класифікується як сервер (peer P).

2)Складання списку деаномізації

На цьому етапі аналітик обирає набір вузлів, справжні IP-адреси яких він прагне ідентифікувати. Вихідні адреси можуть бути отримані з різних джерел. Зокрема, це можуть бути діапазони IP-адрес основних інтернет-провайдерів, адреси, які вже були оприлюднені в мережі Bitcoin, або ж записи, зібрані під час попереднього етапу аналізу.

3)Відображення клієнтів з їхніми вхідними пірами.

Аналітик ідентифікує вхідні вузли, з якими встановлюють з'єднання клієнти мережі. Для кожного клієнта аналізується множина таких

вузлів, з якими він взаємодіє. Якщо клієнт підключений щонайменше до трьох вхідних вузлів, цього, як правило, достатньо для його унікальної ідентифікації, оскільки ймовірність збігу наборів вхідних вузлів між різними клієнтами є вкрай низькою. Зібрана інформація додається до бази даних зловмисника для подальшого аналізу.

4)Відображення транзакцій на вхідні піри

На цьому етапі аналітик намагається зіставити транзакції, які з'являються в мережі, із вузлами входу, отриманими на попередніх етапах. Він відстежує повідомлення INVENTORY із хешами транзакцій, що пересилаються через його підключення, і збирає перші 10 адрес серверів, які передали ці повідомлення(набір R_T). Далі порівнюють $E_P^{'}$ (входи, які отримують транзакцію від клієнта під час її трансляції) з R_T, шукаючи збіги. Якщо збіг є, то формується пара. Якщо відсутні - то дивляться пари, і далі просто піри поодинці.

На жаль, оригінальні автори не надали формалізованої оцінки обчислювальної складності запропонованої атаки. У зв'язку з цим спробуємо самостійно здійснити відповідну оцінку. Орієнтовну складність атаки можна подати у вигляді $O(n \cdot m \cdot k^2 \cdot p)$, де:

· n - кількість вузлів до яких потрібно підключитися під час формування списків та складання списку деаномізації.

· m - кількість транзакцій, які потрібно проаналізувати.

· k - кількість можливих вузлів входу

· p - кількість транзакцій, які потрібно відобразити на вузли входу

Захистити нас від цієї атаки допомогає протокол Dandelion++, використання Tor та VPN сервісів для цибулевого з'єднання.

1.1.2 Атака Сибіли

Атаки типу "Сибіла"(Sybil attacks) полягають у тому, що зловмисник у мережі з архітектурою peer-to-peer створює та контролює

велику кількість фіктивних ідентичностей без необхідності отримання дозволу від інших учасників. Такий підхід дозволяє йому отримати непропорційний вплив на функціонування мережі. Ця проблема є актуальною для багатьох P2P-протоколів, зокрема BitTorrent, Tor, а також криптовалютних мереж.

У контексті криптовалют атаки Сибіли можуть бути використані для зв'язування транзакцій із конкретними користувачами, що фактично порушує їхню анонімність відповідно до раніше наведеного визначення. Крім того, шляхом створення великої кількості підроблених вузлів, зловмисник здатен перевантажити мережу, спричинити затримки або навіть збої в обробці транзакцій.

У найгіршому сценарії, аналітик, використовуючи такі атаки, може нанести суттєву шкоду репутації та економічній стабільності мережі. Отримавши змогу підробляти значну кількість даних або контролювати більшість обчислювальної потужності, він може втручатися у функціонування мережі, зокрема дестабілізувати ринкові процеси на торгових платформах.

1) Створення фальшивих ідентифікацій

На цьому кроці аналітик створює велику кількість фальшивих ідентифікацій або аккаунтів. У контексті криптовалют це можуть бути фальшиві адреси або вузли в мережі.

2)Захоплення контролю над мережею

Ці фальшиві ідентифікації використовуються для отримання контролю над значною частиною мережі. Наприклад, у децентралізованих міксерах [8] зловмисник може створити багато підроблених адрес і брати участь у міксуванні, щоб відстежувати транзакції інших користувачів

3)Вплив на консенсус

Зловмисник може використовувати ці фальшиві ідентифікації для маніпуляції механізмами консенсусу, що може призвести до

фальсифікації даних або навіть до атаки 51%, коли зловмисник отримує контроль над більшістю обчислювальної потужності мережі.

4) Порушення роботи мереж

Робота порушується, довіра втрачається, можлива втрата користувачів.

5)Деанонімізація користувачів

Зв'язування транзакцій з конкретними користувачами, що знижує рівень анонімності.

Відповідної оцінки складності для атаки типу Сибіла в оригінальних джерелах також не було надано, тому спробуємо здійснити її самостійно. Орієнтовну обчислювальну складність такої атаки можна оцінити як:

$$O(D \cdot N)$$

де:

· D — ступінь децентралізації мережі, що може набувати значень у межах $0 \leqslant D \leqslant 1$. Значення $D = 0$ відповідає повністю централізованій системі (наприклад, коли існує лише один контролюючий вузол або один вузол монополізує всі з'єднання). У такому випадку здійснення атаки Сибіла є тривіальним. Натомість $D = 1$ означає ідеальну децентралізацію, за якої всі вузли мають рівний доступ до ресурсів мережі та жоден з них не домінує над іншими. У такій системі здійснення атаки суттєво ускладнюється через велику кількість учасників і з'єднань, однак залишається теоретично можливою.

· N — кількість вузлів, які необхідно контролювати для досягнення бажаного впливу на мережу.

Як видно з виразу, складність атаки зростає лінійно зі зростанням рівня децентралізації та кількості контрольованих вузлів, що робить її технічно простішою для централізованих систем і значно складнішою для децентралізованих мереж.

2 НАЯВНІ МЕТОДИ ЗБЕРЕЖЕННЯ АНОНІМНОСТІ

Основний акцент даної роботи зосереджено на засобах захисту, а не на методах атаки, тому подальший розгляд стосуватиметься саме механізмів забезпечення анонімності. На сьогодні існує кілька криптовалют, які вважаються стійкими в контексті анонімності (хоча варто зазначити, що така оцінка є умовною та не може вважатися абсолютною). До них належать:

– Monero (XMR)

– Zcash (ZEC)

– Dash (DASH)

– Verge (XVG)

– та інші

Ці криптовалюти використовують різноманітні асиметричні криптографічні примітиви, які інтегруються у протоколи та формують основний захисний рівень для користувача, аналогічно до Bitcoin. Проте захист даних не обмежується лише математичними механізмами. Значну роль у забезпеченні анонімності відіграє і захист на рівні мережевої взаємодії.

Зокрема, в P2P-мережах або overlay-мережах застосовуються додаткові інструменти для забезпечення знеособлення учасників. Серед таких засобів можна відзначити:

– кільцеві підписи,

– багатошарові VPN,

– та інші мережеві технології

2.1 Кільцеві підписи

Існують цікаві історичні паралелі до сучасних криптографічних механізмів. Наприклад, прообразом кільцевого підпису можна вважати практику, що використовувалася японськими селянами у зверненнях до свого господаря. Для того щоб приховати автора ініціативи та уникнути покарання, підписанти розміщували свої підписи у вигляді кола, що унеможливлювало ідентифікацію ініціатора.

Цей принцип, заснований на анонімності в межах групи, має очевидні концептуальні подібності з кільцевими підписами, які застосовуються в сучасних анонімних криптовалютах. Далі буде вже наведено формальне означення кільцевого підпису.

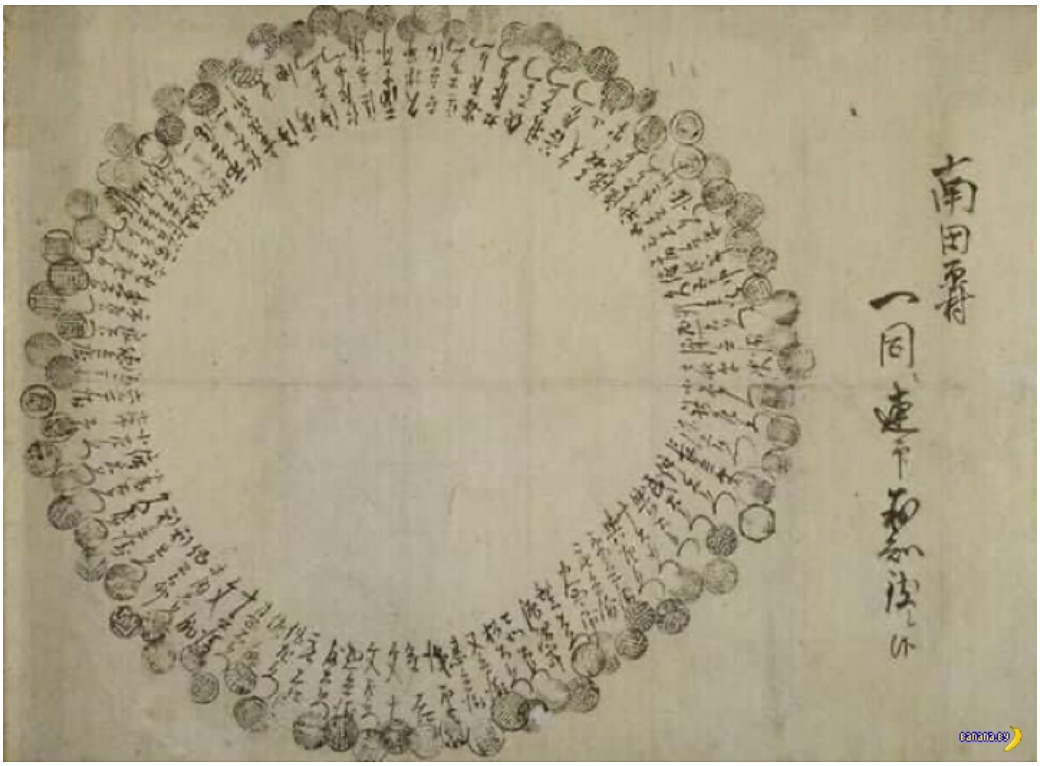

Рисунок 2.1 – приклад такого підпису

Якщо говорити загально, кільцевий підпис - це тип цифрового підпису, який може виконувати будь-який член набору користувачів, кожен з яких має ключі. Таким чином, повідомлення, підписане

кільцевим підписом, сприймається кимось із певної групи людей. Однією з властивостей безпеки кільцевого підпису є те, що має бути обчислювально неможливим визначити, який з ключів членів набору був використаний для створення підпису. Для кращого розуміння буде розглянуто формалізований вигляд, який представили Рон Рівест, Аді Шамір і Яель Тауман Калай на ASIACRYPT у 2001 році[9].

Схема кільцевого підпису визначається двома процедурами:

1) Кільцевий підпис $(m, P_1, P_2, \ldots, P_r, s, S_s)$, який виробляє кільцевий підпис σ для повідомлення m, заданого відкритими ключами $P_1, P_2, \ldots, P_r$ елементів r-члена кільця, разом із секретним ключем S_s s-го члена (який є фактичним підписувачем).

2) Кільцева перевірка (m, σ), яка приймає повідомлення m і σ підпису (що включає відкриті ключі всіх можливих підписантів), і виводить `true` або `false`.

Схема кільцевого підпису реалізується просто: підписувачу не потрібні знання, згода або допомога інших учасників кільця, щоб включити їх у кільце — все, що йому потрібно, це знання їх звичайних відкритих ключів. Різні учасники можуть використовувати різні незалежні схеми підпису з відкритим ключем, з різними розмірами ключів і підписів. Верифікація повинна задовольняти звичайним умовам обґрунтованості та повноти. Як зазначили автори, їх схема особливо стійка при використанні з підписами Рабіна та RSA. У роботі[9] зазначені такі причини такого висловлювання:

– Підписання вимагає одного модульного піднесення до степеня, також одного або двох модульних множень для кожного, хто підписує.

– Верифікація вимагає одного або двох модульних множень для кожного члена кільця.

Для повідомлення m, яке потрібно підписати, його секретним ключем S_s та послідовністю відкритих ключів $P_1, P_2, \ldots, P_r$ всіх членів кільця, схема в руках підписувача працює таким чином:

2.1.1 Алгоритм формування кільцевого підпису

Процедура формування кільцевого підпису складається з наступних кроків:

1)Вибір ключів

Підписувач спочатку обчислює симетричний ключ k як хеш повідомлення m, яке потрібно підписати:

$$k = h(m).$$

2) Формування кільця:

Підписувач вибирає значення ініціалізації (або «склейку») v рівномірно випадковим чином з множини $(0,1)^b$.

3) Генерація випадкових значень

Підписувач вибирає випадкові значення x_i для всіх інших членів кільця, $1 \leqslant i \leqslant r, i \neq s$, рівномірно і незалежно з $(0,1)^b$, та обчислює

$$y_i = g_i(x_i).$$

4) Обчислення значень

Підписувач розв'язує наступне кільцеве рівняння для y_s:

$$C_{k,v}(y_1, y_2, \ldots, y_r) = v.$$

Припускається, що для заданих довільних значень інших аргументів існує єдине значення y_s, яке задовольняє рівнянню і може бути ефективно обчислене.

5)Інверсія перестановки трепдору підписувача

Підписувач використовує свої знання про свій секретний ключ, щоб

інвертувати g_s на y_s і отримати

$$x_s = g_s^{-1}(y_s).$$

6) Формування підпису

Підпис для повідомлення m визначається як кортеж довжини $2r + 1$:

$$(P_1, P_2, \ldots, P_r; v; x_1, x_2, \ldots, x_r).$$

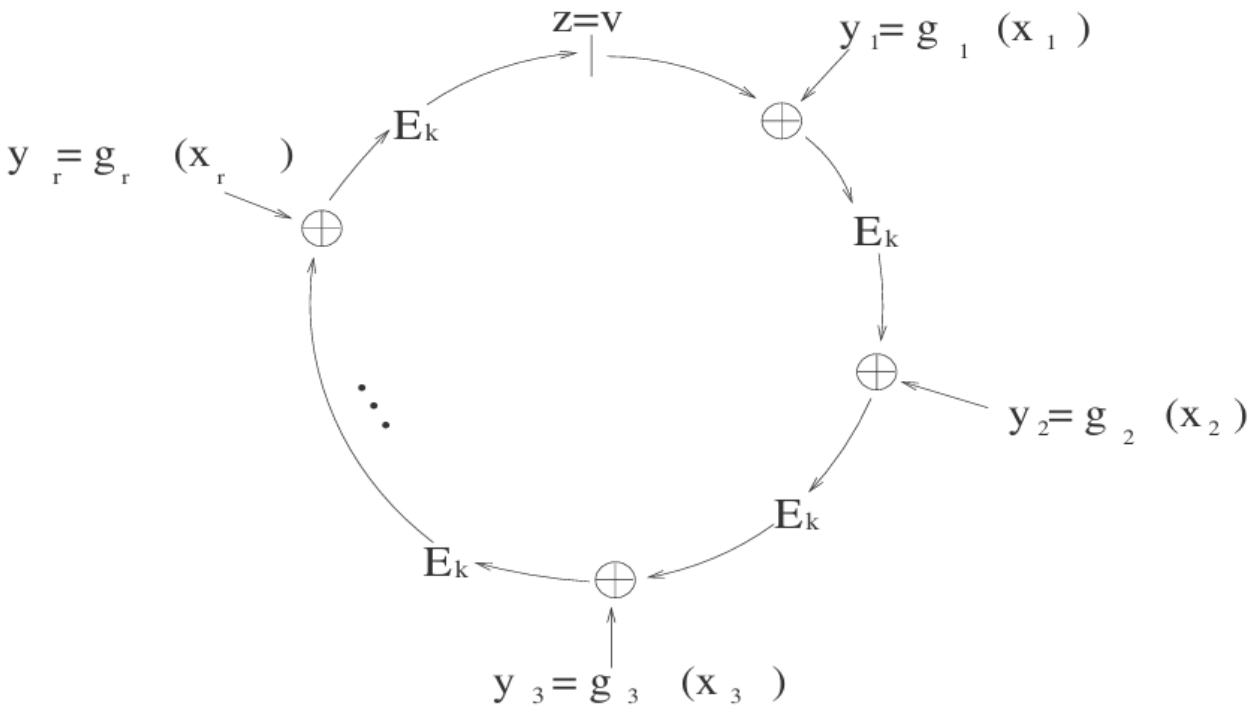

Рисунок 2.2 – приклад такого підпису

Перевірка кільцевого підпису складається з наступних кроків:

1) Застосування перестановки-трепдору

Перевіряючий обчислює для кожного $i = 1, 2, \ldots, r$:

$$y_i = g_i(x_i).$$

2) Отримання ключа

Перевіряючий хешує повідомлення m для отримання ключа k:

$$k = h(m).$$

3) Перевірка рівняння кільця

Перевіряючий перевіряє, що отримані y_i задовольняють фундаментальне рівняння:

$$C_{k,v}(y_1, y_2, \ldots, y_r) = v.$$

Висновки до розділу 2

Щодо застосування, найкращими прикладами є Monero[11] та Zcoin, які були згадані на початку розділу. Вище наведено схему, що ілюструє структуру такого підпису. Важливо відзначити схожість цієї цифрової схеми з історичним прикладом, наведеним раніше.

3 СХЕМИ РОЗПОДІЛЕННЯ СЕКРЕТУ ДЛЯ ЗАХИСТУ АНОНІМНОСТІ

У цьому розділі розглядатимуться доволі прості по вимогам ресурсів схеми порогового розподілення секрету, а точніше їх батько(один з) схема Шаміра, схема Фельдмана і схема Педерсона.

3.1 Схема розподіленного секрету Шаміра

Ця схема є однією з найпопулярніших через легкомасштабованість, що призводить до легкої підтримки систем на її основі. Загалом, метою цього алгоритму можна вважати розділення секретних даних D на n частин так, щоб будь-які k з них дозволяли відновити D, але будь-які k-1 або менше - не давали жодної інформації про D.

Сам алгоритм[10] формулюється так:

Алгоритм 3.1. Проста (k, n) схема розподілу секрету:

1. Вибір поля

Обираємо просте число p, таке що $p > \max(D, n)$. Робота алгоритму відбувається в полі $\mathbb{Z}_p$.

2. Генерація полінома

Створюється випадковий поліном степеня $k-1$:

$$q(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1},$$

де $a_0 = D$, а $a_1, \ldots, a_{k-1}$ — випадково вибрані коефіцієнти з $\mathbb{Z}_p$.

3. Створення частин

Обчислюємо n частин:

$$D_i = q(i) \mod p, \quad \text{для } i = 1, 2, \ldots, n.$$

Кожна частина — це пара (i, D_i).

Алгоритм 3.2. Відновлення секрету[10]

Будь-які k частин дозволяють побудувати поліном $q(x)$ методом інтерполяції (наприклад, через інтерполяцію Лагранжа), і обчислити секрет:

$$D = q(0).$$

Отже, (t,n)-порогова СРС Шаміра дозволяє однозначно відновлювати секрет будь-якій групі з t учасників схеми і забезпечує досконалу секретність (теоретико-інформаційну стійкість) проти спроби обчислення секрету будь-якою групою з j учасників, що володіють необмеженою обчислювальною потужністю (j<t).

При цьому, у СРС Шаміра нечесний дилер D може роздати учасникам P_1,.....P_n несумісні частки, з яких вони ніколи не відновлять секретний ключ K. Необхідно запропонувати таку схему, у якій можна було б перевірити сумісність часток секрету породила дві наступні схеми, які будуть розлядатись в цьому розділі.

3.2 Схема розподіленного секрету Фельдмана

Класичні інтерактивні схеми порогового розподілення секрету вимагають обміну повідомленнями між учасниками для перевірки коректності отриманих часток секрету. Фельдман запропонував неінтерактивний підхід, уякому кожен учасник може самостійно

переконатися в достовірності своєї частки, використовуючи гомоморфні властивості криптографічного шифрування.

Основою даної схеми є дискретне логарифмування у скінченних групах або еліптичних кривих, яке забезпечує достовірність отриманих часток без потреби в додатковій взаємодії між учасниками.

Алгоритм 3.3. Алгоритм розподілу секрету[5]

1. Дилер генерує поліном $q(x)$ ступеня t:

$$q(x) = a_0 + a_1 x + \cdots + a_t x^t, \quad \text{де } a_0 = s.$$

2. Дилер обчислює n частин:

$$s_i = q(i), \quad \text{для } i = 1, 2, \ldots, n.$$

3. Дилер публікує $t + 1$ підтверджень:

$$c_j = g^{a_j} \mod p, \quad \text{для } j = 0, \ldots, t.$$

4. Дилер надсилає кожному учаснику i приватно його частину s_i.

Алгоритм 3.4. Алгоритм перевірки секрету[5]

Кожен учасник i перевіряє правильність своєї частини s_i за допомогою перевірки:

$$g^{s_i} \stackrel{?}{=} \prod_{j=0}^{t} c_j^{i^j} \mod p.$$

Ця перевірка гарантує, що $s_i = q(i)$, оскільки:

$$g^{q(i)} = \prod_{j=0}^{t} g^{a_j i^j} = \prod_{j=0}^{t} c_j^{i^j}.$$

Алгоритм 3.5. Алгоритм відновлення секрету[5]

Будь-яка підмножина з $t+1$ учасників, що мають коректні частини (i_k, s_{i_k}), може відновити секрет s шляхом інтерполяції полінома $q(x)$ над $\mathbb{Z}_p$ та обчислення:

$$s = q(0).$$

На основі цього можна сказати, що у схемі перевіреного поділу секрету кожен може перевірити тільки свою частку, але не чужу - для цього потрібні схеми публічно перевіряємого поділу секрету.

3.3 Схема розподіленного секрету Педерсона

Схема Педерсона дозволяє одному учаснику (розподільнику) секретно та верифіковано розподілити секрет серед кількох учасників так, щоб будь-яка кворумна підмножина (наприклад, t із n учасників) могла відновити секрет. Вона побудована на основі схеми розподіленного секрету Шаміра. Також кожен учасник може перевірити правильність своєї частки без розкриття самого секрету.

Головна особливість схеми Педерсона є забезпечення теоретико-інформаційної секретності, яка в свою чергу побудована на тому, що якщо у нас є противник з необмеженними ресурсами і неймовірною іноземною технологією яка може вирішувати задачу дискретного логарифмування на притомний час, то навіть так він не зможе дізнатись з результату перевірочного рівняння нічого про секрет, який знаходиться в степені елементу порядку простого великого числа в группі Z_p^*.

У своїй роботі[7] Педерсон представив схему так:

Параметри на вхід:

– q — велике просте число.

– $g, h \in \mathbb{Z}_q^*$ — генератори підгрупи порядку q, такі що $\log_g h$ невідоме.

– n — кількість учасників.

– t — поріг відновлення (тобто потрібно принаймні t часток для реконструкції секрету).

– Всі операції виконуються в $\mathbb{Z}_q$.

Алгоритм 3.6. Ініціалізація

1) Розподілювач обирає секрет $s \in \mathbb{Z}_q$.

2) Випадковим чином обирає коефіцієнти полінома ступеня $t-1$:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}, \quad \text{де } a_0 = s$$

3)Випадково обирає ще один поліном такого ж ступеня:

$$f'(x) = b_0 + b_1 x + \cdots + b_{t-1} x^{t-1}$$

4) Обчислює комітменти до коефіцієнтів:

$$c_j = g^{a_j} h^{b_j}, \quad \text{для } j = 0, \ldots, t-1$$

Алгоритм 3.7. Розподіл часток (учасникам)

Для кожного учасника з ідентифікатором $i = 1, \ldots, n$:

1) Обчислює частки:

$$s_i = f(i), \quad r_i = f'(i)$$

2) Надсилає учаснику i значення (s_i, r_i).

Алгоритм 3.8. Верифікація частки (учасником)

Учасник перевіряє:

$$g^{s_i} h^{r_i} \stackrel{?}{=} \prod_{j=0}^{t-1} c_j^{i^j}$$

Якщо рівність виконується, частка вважається коректною.

Алгоритм 3.9. Відновлення секрету

Зібравши будь-які t коректних часток (i_k, s_{i_k}), обчислюється:

$$s = f(0) = \sum_{k=1}^{w} s_{i_k} \cdot \lambda_k, \quad \text{де } \lambda_k = \prod_{j \neq k} \frac{i_j}{i_j - i_k}$$

Це дає змогу відновити секрет $s = a_0$ за допомогою інтерполяції Лагранжа.

Висновки до розділу 3

Схема Педерсона є надійною криптографічною схемою розподілу секрету, яка гарантує як конфіденційність, так і перевірюваність часток, що передаються учасникам. Вона базується на схемі Шаміра, яка в свою чергу дозволяє уникати проблем від нечестного дилера, але розширена додаванням засобів перевірки часток через комітменти, не розкриваючи сам секрет. Головною перевагою цієї схеми є теоретико-інформаційна безпека, яка не залежить від обчислювальної складності, навіть якщо противник має доступ до ресурсів, здатних вирішувати задачу дискретного логарифмування.

Кожен учасник може впевнено перевірити, що отримана частка є коректною, без необхідності взаємодії з іншими учасниками або розкриття додаткової інформації. На відміну від схеми Фельдмана, яка також базується на схемі Шаміра та дозволяє перевірку часток за допомогою відкритих комітментів, схема Педерсона є сильнішою у плані безпеки, оскільки забезпечує інформаційно-теоретичну (а не лише обчислювальну) секретність. Через що була обрана схема Педерсона для власного рішення.

4 ПРОТОКОЛИ ЗАХИСТУ АНОНІМНОСТІ

У цьому розділі розглянуто деякі з варіантів збереження анонімності, зокрема протоколи Atlantis від Distributed Labs, сімейство протоколів Dandalion і також рішення покладене в основу дипломної роботи, звісно з прикладами реалізації рішень, що відповідають вимогам анонімності.

4.1 Протокол Atlantis

Першою буде розглянуто роботу представлену взимку 2024 року компанією Distributed Labs.

Для початку розглянемо, що він таке і як з ним працювати. Цей протокол дуже сильно надихався міксером Tornado Cash, що видно як в його офіційному представленні, так і в самому рішенні. Сам протокол являє собою набір операцій над розрідженним деревом Меркла, яке є своєрідним серцем цього підходу. Набір операцій такий:

1) депозит токенів в дерево (дозволяє конвертувати монети у комітмент, що зберігається у PMT)

2) трансфер токенів всередині дерева (витрачає існуючі комітменти та створює нові невитрачені комітменти)

3) вивід токенів з дерева (дозволяє конвертувати комітменти у публічну кількість монет і виводити їх на рахунок користувача)

За словами розробників такий підхід гарантує анонімність, невідстежуванність, а також безмежність та гетерогенність транзакцій.

Власне кажучи, основою цієї конструкції є не просто кажучи PMT, а саме комітменти в ньому, які представлені точками на еліптичних кривих.

Означення 4.1. Комітмент - це зобов'язання, яке використовується для представлення балансу користувача. Він створюється за допомогою

маніпуляцій з над генераторами групи та секретним ключем користувача(приклад цього як раз наведений нижче).

Комітменти представлені там такою формулою:

$$C = aH + skG$$

α — кількість монет у комітменті,

sk - секретний ключ користувача,

H,G - генератори групи

Якщо користувач хоче, щоб цей комітмент підтримував ще і кількість активів, то він робить суму комітментів по n, де i буде індентифікатором, автор запропонував адресу цього комітмента, а кожна i-та кількість монет буде множитись на i-тий генератор того типу монет, який ми використовуємо.

4.1.1 Депозит

Операція депозиту доволі проста і ресурсно проста, тому що цей смарт-контракт виконується на основі публічних данних, які надаються самим ініціатором операції. Тобто, на вхід ми подаємо такі значення:

– кількість депонованих монет для різних типів активів

– генератори типів монет,

– генератор власності

– публічний ключ депозиту(секретний ключ користувача з генератором властника)

Саме на основі цих данних користувач можем побудувати коммітмент в PMT.

Алгоритм депозиту має такий вигляд[6]:

1) користувач здійснює депозит і передає значення публічного ключа до контракту.

2) контракт дістає кількість та генератори типів депонованих монет

з данних депозиту

3) користувач виконує обчислення комітменту

4) в дереві зберігається комітмент за індексом рівним хешу комітмента

Головною особливістю такого підходу можна вважати можливість робити більше одного депозиту на користувача.

4.1.2 Трансфер

Трансфер витрачає список існуючих комітментів і створює нові. При цьому відравник та одержувач коштів повинні вже мати можливість разом створити транзакцію, що доволі важливо. Коротко кажучи, це можна описати так[6]:

– Відправник обчислює список нуліфікаторів(щоб запобігти повторне витрачання комітменту) для комітментів, які потрібно витратити, і передає їх одержувачу

– одержувач формую на основі отриманних данних свій комітмент, додатково він генерує доказ (ZKP) діапазону для суми у своєму комітменті.

– Відправник створює свої підписи, агрегує їх із підписами одержувача та генерує фінальний доказ (включаючи комітменти для здачі, якщо потрібно).

Звісно це в загальному вигляді, сам алгоритм виглядає страшніше, тому чому би не глянути на його повну версію тут? Як видно вище, він складається з трьох етапів[6]: ініціації, підготовки та самого трансферу

Тепер розпишемо те, як вони працюють один за одним і почнемо з ініціації.

1) Відправник маючи список невитрачених комітментів з секретними ключами, обчислює список нуліфікаторів на його основі

2) Відправник надсилає цей список нуліфаєрів

Тепер розглянемо підготовку[6].

1) Одержувач генерує комітмент на основі своїх власних ключа і

генератора власності

2) Одержувач підписує ключ з нуліфаєром (як відомо, він робить з обчисленного комітмента)

3) Одержувач генерує Zero Knowlage proof(далі ZKP) для такого відношення:

$$R_{op} = \{a_i^{(n)}, sk_r; G, H_i^{(n)}, C_r \mid \forall i \in [0,n], a_i \in [0, 2^{128}), C_r = \sum_{i=0}^{n} a_i H_i + sk_r G\}$$

4) Одержувач відправляє підпис і ZKP відправнику

І тепер про найскладніше і найважливіше: сам трансфер[6].

1)Відправник генерує список підписів для всіх вхідних комітментів

2)Відправник генерує публічний підпис комітмента та його самого для здачі

3)Відправник генерує підпис на основі згенерованного підпису і його нуліфаєру

4)Відправник генерує ZKP, щоб змінити комітмент

5)Відправник обчислює агрегований підпис для всіх вхідних та вихідних комітментів

6)Відправник генерує фінальний платіжний ZKP для відношення:

$$R_p = \{sk_s^{(n)}, \text{sigagg}, C_s^{(n)}, \pi_t^{(n)}; \text{null}_c^{(n)}, C_r, C_c, t :$$

$$\text{sigVer}(\text{sigagg}, \sum_{i=0}^{n} C_{s_i} - C_r - C_c, \text{null}_s^{(n)}) \to \text{true},$$

$$\forall i \in [0,n], \text{null}_{s_i} \leftarrow \text{hash}(sk_{s_i}),$$

$$\forall i \in [0,n], \text{MBVer}(C_{s_i}, \pi_{t_i}, t) \to \text{true}\}$$

7)Відправник передає ZKP, які відповідають за доказ діапазону значення переказу, за теж саме для решти і сам фінальний доказ. Додає комітменти решти і основного платежу в PMT, а їх нуліфаєр у список витраченних комітментів.

4.1.3 Вивід

На виводі користувач повинен довести, що існує такий невитрачений комітмент з його токенами і опублікувати нуліфаєр, пов'язаний з секретним ключем(який лежить в основі публічного ключа комітменту). Для цього потрібно буде згенерувати ZKP

$$R_w = \{sk, C, \pi_t; a_i^{(n)}, G, H_i^{(n)}, \text{null} \mid \text{null} \leftarrow \text{hash}(sk),$$

$$\text{MBVer}(C, \pi_t, t) \rightarrow \text{true}, C = \sum_{i=0}^{n} a_i H_i + skG\}.$$

Після генерації доказу контракт перевіряє його та при відсутності в ньому нуліфаєрів, які належать списку витрачених комітментів, користувач може вивести ці токени на свій публічний рахунок.

4.2 Сімейство протоколів Dandalion

Протокол Dandelion — це мережевий протокол, запропонований у 2017 році з метою підвищення анонімності в мережі Bitcoin (у відповідь на атаку типу BHP, описану в розділі про вразливості). Його основна ідея полягає в поділі процесу передачі транзакцій на дві фази: фазу анонімності та фазу розповсюдження.

У фазі анонімності транзакція передається випадковим чином через кілька вузлів, що ускладнює ідентифікацію її джерела. Після цього транзакція переходить у фазу розповсюдження, де поширюється по всій мережі за допомогою механізму дифузії.

Такий підхід забезпечує квазіоптимальний рівень анонімності при незначному впливі на продуктивність мережі. Протокол було пізніше вдосконалено до версії Dandelion++, яка усуває деякі недоліки оригінальної версії та була впроваджена в Bitcoin, Monero, Zcoin та інші проєкти. Переваги й недоліки цього підходу розглядаються нижче.

4.2.1 Dandalion

Як зазначалося раніше, протокол Dandelion складається з двох фаз: фази анонімності (стебла) та фази розповсюдження (пухнастості).

У фазі анонімності транзакція передається випадковим чином через кілька вузлів. Це ускладнює визначення джерела транзакції. Кожен вузол передає транзакцію одному випадково обраному сусідньому вузлу. Процес триває протягом випадкової кількості кроків, що забезпечує додатковий рівень анонімності. Важливо, що всі транзакції від усіх джерел проходять через один і той самий шлях, причому противник не повинен мати можливості дізнатися структуру цього шляху.

У фазі розповсюдження транзакція поширюється по всій мережі за допомогою дифузії. Це забезпечує швидке поширення транзакції, дозволяючи всім вузлам отримати її.

Щодо недоліків та переваг протоколу, варто спочатку розглянути недоліки, які призвели до необхідності модифікації алгоритму. Такими недоліками є:

– погана масштабованість алгоритму;

– хоч і присутній, але недостатньо якісний захист від супернод атаки;

– затримки в передачі через двофазність;

– середня складність реалізації алгоритму.

Останнє можна вважати як за перевагу, так і за недолік цього протоколу. Перевагами же є:

– мінімальні витрати на продуктивність, тобто користувач не очікує тривалий час поки транзакція пройде;

– квазіоптимальна анонімність, яка забезпечує захист від атаки БХП.

Головною особливістю є фаза розповсюдження, яка працює за таким алгоритмом :

Алгоритм 4.1. Протокол розповсюдження Dandelion [4]

ВХІД:повідомлення X_v, джерело v, граф анонімності G, граф

розповсюдження H, параметр $q \in (0,1)$

1. Спочатку алгоритм знаходиться у фазі анонімності (позначено як anonPhase = True). Початковий вузол (head) встановлюється як джерело v. Множина отримувачів (recipients) ініціалізується повідомленням від джерела.

2. Поки триває фаза анонімності

– Вибирається випадковий сусідній вузол (target) із вихідних сусідів вузла head у графі анонімності G.

– Повідомлення передається від поточного вузла head до target, і множина отримувачів доповнюється.

– Змінна head оновлюється на вузол target

– Генерується випадкове число u з рівномірного розподілу на відрізку [0,1].

– Якщо $u \leqslant q$, то фаза анонімності завершується (anonPhase = False).

3. Після завершення фази анонімності запускається фаза дифузії, під час якої повідомлення поширюється по графу поширення H, починаючи з останнього вузла head.

Недоліки алгоритму не сприяли його прийняттю в той час криптовалютами, що призвело до розробки алгоритму Dandelion++, який буде розглянутий далі.

4.2.2 Dandalion++

Перш за все, головна особливість цього протоколу - він прийнятий на відміну від свого попередника. Мало того - він вважається гарантовано анонімним. Він явно покращує ідеалістичні припущення оригінальної пропозиції Dandelion і відрізняється від більшості протоколів анонімності широкомовного зв'язку своїм підходом до цілей використання та метрик аналізу.

У Bitcoin, коли користувач транслює транзакцію з вузла, вона

поширюється на вузли, підключені до цього конкретного вузла, відомі як його вузли. Потім повідомлення про транзакцію згодом поширюється в результаті ланцюгової реакції, де кожен вузол додатково поширює повідомлення серед вузлів, до яких він підключений. Це називається протоколом пліток Bitcoin, і саме завдяки цьому транзакції можуть дуже швидко досягати більшості вузлів мережі.

Тепер Bitcoin реалізує форму трансляції, відому як дифузія, коли кожен вузол розподіляє транзакції з експоненціальними та незалежними затримками серед своїх сусідів, щоб пом'якшити деанонімізацію IP-адреси користувача. Незважаючи на свою ефективність, нещодавно було доведено в кількох дослідженнях, дифузія не забезпечує належного захисту анонімності. Походження повідомлення про транзакцію та її IP-адреса (яка не включена в повідомлення про транзакцію Bitcoin) можуть бути відображені сторонніми спостерігачами, якщо вони контролюють достатню кількість вузлів або використовують супервузол, який підключений до значної кількості вузлів. Вони можуть ефективно відображати вихідну адресу, спостерігаючи, які вузли бачать транзакцію першими. Пов'язуючи IP-адресу з псевдонімом відправника, третя сторона може деанонімізувати користувачів і пов'язати подальші транзакції, навіть якщо для кожної транзакції використовується новий публічний ключ. Спочатку Dandelion була запропонована для пом'якшення цих вразливостей, але покладалася на теоретичні гарантії, які не виправдалися на практиці. Оригінальна пропозиція «Dandalion» містила такі припущення:

– усі вузли контролюються нами

– 1 вузол = 1 транзакція

– усі вузли Bitcoin використовують Dandelion

На практиці все виявилось не так добре, як хотілось, тому і прийшлось прибігати до покращень вже в Dandelion++. Основні проблеми з оригінальним протоколом Dandelion пов'язані з його недооцінкою конкретних типів супротивників через припущення про їхні обмежені

знання. Dandelion++ особливо зосереджується на внесенні незначних змін у варіанти реалізації Dandelion, такі як топологія графіка та механізми пересилання повідомлень. В результаті, ці невеликі зміни в алгоритмі експоненціально збільшують простір проблем для аналізу анонімності. Dandelion++ покладається на збільшення обсягу інформації, яку зловмисники повинні вивчити для деанонімізації користувачів.

Dandelion++ помітно відрізняється від Dandelion своєю фазою стебла, коли він пропускає транзакції через переплетені шляхи, відомі як кабелі, перш ніж розсіяти повідомлення транзакції в мережу. Кабелі можуть бути фрагментовані, але його інтуїція у виборі вузла для поширення все ще обмежена його місцевим районом. Це важливий фактор при порівнянні рішень для анонімності на рівні мережі, таких як Tor, який є протоколом цибулевої маршрутизації, де клієнтам потрібна глобальна поточна інформація мережі для визначення шляхів транзакцій.

Все сімейство Dandalion використовує асинхроннні цикли. Кожен вузол йде вперед коли його внутрішній ітератор досягає певного порогу. Граф анонімності використовує випадковий 4-регулярний граф, а не лінійний граф для фази анонімності. Вибір ретрансляторів Dandelion++ за вузлами не залежить від того, чи підтримують їхні вихідні сусіди Dandelion++.

Теорема 4.1. *[4] Максимальна очікувана точність на випадковому 4-регулярному графіку з невідомим супротивнику графіком обмежена*

$$\mathrm{DOP}_T \leqslant 8 \cdot \mathrm{DF}_S + 6p^2 + O(p^3),$$

де DOP_T *і* DF_S *позначають очікувану точність при оптимальній і першій шпигунській оцінках відповідно.*

Алгоритм 4.2. Алгоритм побудови приблизного 4-регулярного графа

ВХІД: множина вузлів $V = \{v_1, v_2, \ldots, v_n\}$.

ВИХІД: орієнтований графа $H(V, E)$, у якому кожен вузол матиме

в середньому 4 вихідних ребра, а також граф буде зв'язним.

1. Для кожного вузла $v \in V$ виконуємо такі кроки:

1) Випадково вибираємо два різних вузли зі множини V, окрім самого v:

$$u_1 \sim \text{Unif}(V \setminus \{v\}), \quad u_2 \sim \text{Unif}(V \setminus \{v, u_1\})$$

2) Додаємо спрямовані ребра від v до u_1 та u_2:

$$E \leftarrow E \cup \{(v \to u_1), (v \to u_2)\}$$

2. Після виконання цього процесу для всіх $v \in V$, отримуємо граф $H(V, E)$ з середнім ступенем виходу близько 2 на вузол, що відповідає приблизно 4-регулярному графу в орієнтованому вигляді.

3. Цей граф використовується як анонімний граф для подальших операцій протоколу.

4.3 Власна розробка

Минулі рішення мали великі вимоги до ресурсної бази користувачів доволі грубе виконання поняття анонісності, яке реалізоване тим, що шляхи(в схожих рішеннях) були передобчислені, що призводило до того, що за наявністю часу та ресурсів у оппонента, їх можна було перебрати за притомний час. Саме такі проблеми покликаний вирішити цей протокол: велика кількість ресурсів для роботи і покращення забезпечення анонімності користувача.

Також буде розглянута робота самостійного та нового протоколу, який дозволяв би анонімно проводити дії з токенами. Він об'єднує підходи таких протоколів анонімізації, як Zerocoin, CoinJoin, а також увібрав напрацювання авторів Monero.

Сенс полягає в тому, що кожен з цих підходів має вразливості, які передбачається усунути за рахунок поєднання їхніх найкращих властивостей. Для прикладу, основна складність CoinJoin, в тому, що ми

повинні гарантовано мати оффчейн канал, який забезпечуватиме постачання вхідних них. На основі цього цей протокол може стверджувати, що стороній спостерігач не зможе відрізнити два виходи й визначити, який з них був вхідним. Це було наведено в прикладі[3], який вставлено сюди для зручності представлення.

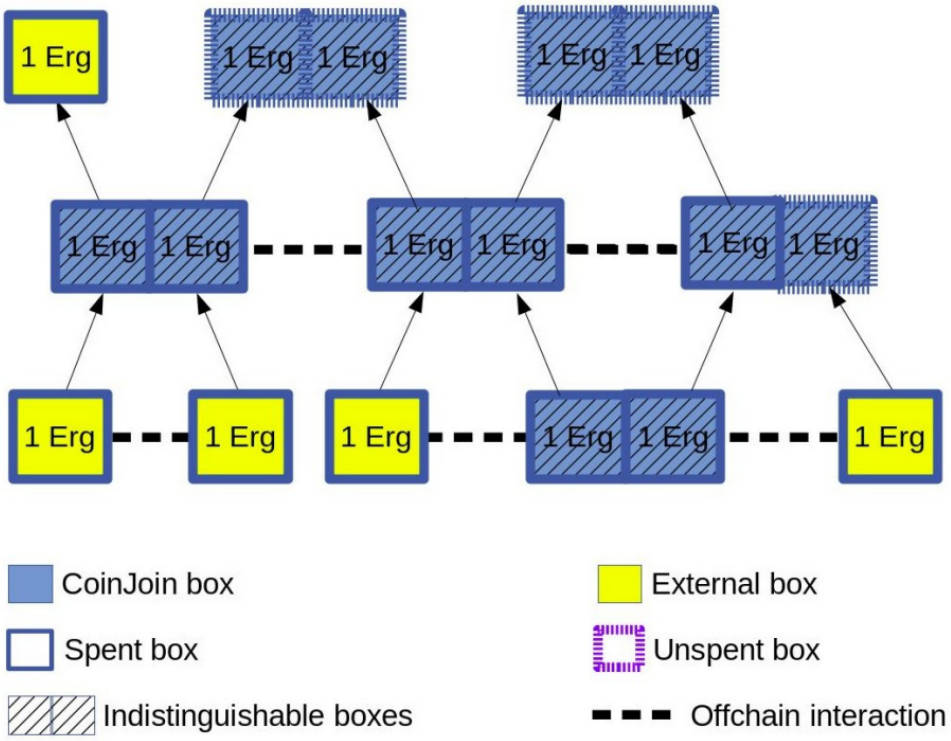

Рисунок 4.1 – приклад роботи CoinJoin

Цей підхід дозволяє уникати повного зв'язування, бо кожна транзакція має 50% незалежності. Плюсом цього підходу може бути те, що вже в наступній транзакції вихід може бути використаний як вхід, тому процес може бути повторений, щоб підвищити незалежність до будь-якого бажаного рівня.

Подібний підхід застосовується в Monero і має назву «хмара анонімності». Його ключова особливість полягає в тому, що шлях між відправником і одержувачем прихований, що унеможливлює встановлення зв'язку між користувачами та їхніми транзакціями. Суть цього методу — у затиранні групи транзакцій, що фактично нівелює її

для системи і не дозволяє стороннім особам визначити, хто саме здійснив переказ до отримувача

Сам алгоритм матиме таку структуру:

Алгоритм 4.3. Алгоритм елементу транзакції:

1. Користувач обчислює сід за допомогою KMAC(Keccak Message Authentication Code), який в свою чергу використовуватиме конкатенацію транзакції та середнього часу натискання клавіш за останні 10 символів,

2. Із згенерованого зерна генеруємо псевдовипадкову послідовність за допомогою ChaCha20

3. Із заданої послідовності будуємо квазівипадковий(через те, що ми намагаємось розподілити всі шляхи так, щоб вони відповідали рівнорозподіленному розподілу) шлях на системним гаманцям, по якому ми відправляємо вже токени

Щоб забезпечити цілістність та збереження секрету пропонується використовувати порогову схему розподіленного секрету Педерсона через особливості описані в розділі протоколів розподілу секрету. Все разом повинно мати такий вигляд:

Алгоритм 4.4. Повний протокол переказу коштів:

1) Обчислюємо зерно згідно алгоритму 3.1,

$$seed = KMAC(key1, message, lenght),$$

де:

· message - $hash(userTransac||avg(T))$

· key1 - згенерований користувачем згідно правил описаних в [1]

· lenght - $w \cdot 8$,

2) на цьому кроці ми ініціалізовуємо ChaCha20,

$$S = ChaCha20_{Encrypt}(seed, key, nonce),$$

де:

· seed - обчислено на кроці 1

· key2 - згенерований користувачем згідно правил описаних в [1]

· nonce - $trunc_{96}(hash_{userTransac})$

3) Формування шляху, розбиваючи S на множину з W частин 8 бітових елементів, формально описуючи це виглядає так:

$$S = \{s_1, s_2, ...s_w | i = [1, w], s_i \in Z_q\}$$

4) Створюємо комітмент, де $r \in Z_q$:

$$Com(S,r) = \prod_{i=0}^{w-1} g_i^{s_i} \cdot h^r$$

5) Розподіляємо $s_i = f(i), r_i = f^{'}(i)$ між участниками як пару (s_i, r_i)

Веріфікація буде проводитись відповідно до описаного у відповідному розділі зі схемою Педерсона, а саме так:

$$g^{s_i} h^{r_i} \stackrel{?}{=} \prod_{j=0}^{w-1} c_j^{i^j}$$

Якщо рівність виконується, то верифіковація вважається успішною. З відновлення історія ідентична,

$$s = f(0) = \sum_{k=1}^{t} s_{i_k} \cdot \lambda_k, \quad \text{де } \lambda_k = \prod_{j \neq k} \frac{i_j}{i_j - i_k}$$

Системні особливості:

– Передньо сформовані системні гаманці, яких повинно бути $q = w^n$ штук, де w - кількість гаманців на одну людину, (рекомендується брати значення цієї змінної вище 11, тому що як відомо до 11 включно входів відслідковується під час атаки Судоку), а n - кількість користувачів в мережі. Наприклад, якщо візьмемо кількість гамантів на людину 15, а мережа у нас на 2 людей, то ми задієюмо 225 гаманців.

– Обрано ChaCha20 через рекомендацію eSTREAM, а також його швидкодію на фоні аналогів.

– Під час першого етапу нам потребується доступ до пристрою вводу користувача, а точніше клавіатури, що може бути по-перше вразливістю, бо якщо ми можемо зчитати інформацію з клавіатури, то треба попіклуватись про те, щоб USB порти на клавіатурі не мали доступу до зчитування інформації застосунком. Також, це може викликати побоювання щодо безпеки, адже доступ до клавіатури може створити потенційні ризики стороннього зчитування(тобто побоювання щодо клептографічного елементу всередині), чого хотілось би уникнути.

Взаємодія між користувачами відповідає стандартній взаємодії користувачі схеми Педерсона, що наведена в [7]

Порівняння з представленними аналогами представлено нижче в таблиці, на основі якої можна сказати, що представлене в роботі рішення є кращим та доступнішим для більшої кількості користувачів.

Критерій	**Atlantis**	**Dandelion / Dandelion++**	**Власний протокол (на основі Педерсона)**
Тип протоколу	Смарт-контракт із використанням RMT та zkSNARK	Мережевий протокол маршрутизації транзакцій	Криптографічний протокол з використанням порогової схеми Педерсона
Метод анонімізації	Комітменти та ZKP при трансфері та виводі	Приховування першого вузла через фазу «стебла» та «квітки»	Пороговий поділ секрету і перевіряльність часток
Рівень анонімності	Високий (завдяки zkSNARK та комітментам)	Середній (мережева анонімність, але не захищає вміст транзакції)	Високий (теоретико-інформаційна секретність, неможливість деанонімізації навіть з необмеженими ресурсами)
Ресурсомісткість	Висока — генерація zk-доказів обчислювально дорога	Низька — переважно мережеві затрати	Низька
Стійкість до атак БХП / Сибіли	Частково (залежить від застосування мережевих обгорток)	Спеціально створений для протидії БХП / Сибіли	Стійкий на рівні даних (при використанні анонімних каналів)
Верифікація часток	zk-докази дозволяють переконатись у коректності	Немає, не працює з частками, а лише з маршрутами	Публічне підтвердження кожної частки через комітменти
Гнучкість / масштабованість	Висока, але ускладнена zk-доказами	Висока на рівні P2P-мереж	Висока, завдяки відсутності потреби у складній інфраструктурі

Висновки до розділу 4

Під час розробки протоколу Атлантіс виникли труднощі, пов'язані з реалізацією ZKP у механізмі трансферу, зокрема при використанні zkSNARK та Circom. Час від часу схема Circom не розпізнавалась контрактом і не компілювалася у верифікатор. Такий підхід потребує значних обчислювальних ресурсів, що може негативно вплинути на масштабованість рішення. У випадку, якщо генерація доказу покладається на користувача, це суттєво обмежує потенційну аудиторію через високі технічні вимоги. Якщо ж обчислення здійснюються на стороні мережі або розробника, це знижує рівень децентралізації, що може викликати недовіру до рішення та потребуватиме додаткових витрат на інфраструктуру.

В той час, як запропонований в цьому розділі протокол об'єднує найкращі риси CoinJoin, Zerocoin і Monero для забезпечення анонімності транзакцій, усуваючи їхні вразливості. Завдяки використанню псевдовипадкової маршрутизації, AES-256 та порогового розподілу секретів, протокол гарантує приватність, цілісність і стійкість до відстеження. Він є перспективним рішенням для анонімного обігу токенів у децентралізованих системах.

Протокол Dandelion спрямований переважно на забезпечення анонімності на мережевому рівні, зокрема під час розповсюдження транзакцій у мережі однорангових вузлів. Його механізм, що включає фази стебла і пухнастості, дозволяє частково приховати джерело транзакції шляхом її попередньої передачі обмеженим маршрутом перед широкомовною трансляцією. Однак, цей підхід не забезпечує криптографічного захисту самих транзакцій, залишаючи відкритими для аналізу інформацію про зв'язки між входами і виходами, а також вимагає високого рівня довіри до мережевого середовища.

На відміну від цього, запропонований у роботі протокол ґрунтується

на поєднанні криптографічних методів з підходами, запозиченими з протоколів CoinJoin, Zerocoin та Monero. Такий підхід дозволяє досягти значно вищого рівня анонімності та захисту від аналізу транзакцій, оскільки забезпечує маскування логічних зв'язків між відправником і одержувачем, використовуючи псевдовипадкові маршрути переказу токенів, формування комітментів, а також порогову схему розподілу секрету Педерсона, яка знижує вимоги до рівня довіри дилеру через її інформаційно-теоретичну стійкість.

ВИСНОВКИ

У межах виконання дипломної роботи було досягнуто основної мети дослідження — розробки протоколу забезпечення анонімності у криптовалютних системах на основі криптографічних порогових протоколів розподілу секрету. У процесі дослідження здійснено огляд сучасного стану проблеми збереження анонімності в криптовалютах, проаналізовано відомі підходи до її забезпечення, зокрема механізми, що реалізуються у криптовалютах Monero, Zcash, а також протоколах Dandelion, Atlantis. Проведено аналіз атак на анонімність, таких як атака БХП та атака Сибіли. Це дозволило виявити ключові недоліки існуючих рішень, зокрема високу обчислювальну складність, технічну складність впровадження та залежність від сторонніх сервісів.

Подальше вивчення криптографічних інструментів дало змогу зосередити увагу на схемах розподілу секрету, зокрема схемі Педерсона, яка забезпечує не лише конфіденційність переданої інформації, але й її перевірку без розкриття вмісту, що є критично важливим для досягнення анонімності. На основі проведеного аналізу та експериментального вивчення зазначених механізмів було запропоновано власний протокол, у якому реалізовано концепцію затирання слідів транзакцій шляхом поєднання схеми Педерсона з криптографічними комітментами та механізмами маршрутизації, подібними до Dandelion++. Особливістю запропонованого рішення є зменшення обчислювальних витрат у порівнянні з класичними протоколами на основі доказів з нульовим розголошенням, що дозволяє адаптувати систему для використання пересічними користувачами без втрати рівня конфіденційності.

Таким чином, результати роботи підтверджують доцільність використання порогових схем розподілу секрету для побудови анонімних протоколів у криптовалютних системах. Розроблений підхід демонструє як теоретичну, так і практичну ефективність, відкриваючи перспективи

для подальших досліджень у напрямку його оптимізації, формалізації моделі безпеки, а також розширення функціоналу шляхом підтримки мультисигнатурних транзакцій і інтеграції з мобільними або розподіленими середовищами.

ПЕРЕЛІК ПОСИЛАНЬ

[1] . *Recommendation for Key Derivation Using Pseudorandom Functions*. Англ. 2022, с. 34 с. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf.

[2] Alex Biryukov, Dmitry Khovratovich та Ivan Pustogarov. *Deanonymisation of clients in Bitcoin P2P network*. 2014. arXiv: 1405.7418 [cs.CR]. URL: https://arxiv.org/abs/1405.7418.

[3] Alexander Chepurnoy та Amitabh Saxena. *Zerojoin: Combining Zerocoin and CoinJoin*. 2020. URL: https://eprint.iacr.org/2020/560.

[4] Giulia Fanti та ін. *Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees*. 2018. arXiv: 1805.11060 [cs.CR]. URL: https://arxiv.org/abs/1805.11060.

[5] Paul Feldman. «A Practical Scheme for Non-interactive Verifiable Secret Sharing». Англ. В: (1987), с. 11.

[6] Oleksandr Kurbatov, Kyrylo Riabov та Mykhailo Velykodnyi. *Atlantis Protocol*. Англ. 2024. URL: https://arxiv.org/abs/2412.02585.

[7] Torben Pryds Pedersen. *Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing*. Англ. 1991.

[8] Mikerah Quintyne-Collins. *Short Paper: Towards Characterizing Sybil Attacks in Cryptocurrency Mixers*. Cryptology ePrint Archive, Paper 2019/1111. 2019. URL: https://eprint.iacr.org/2019/1111.

[9] Ronald L. Rivest, Adi Shamir та Yael Tauman. «How to Leak a Secret». В: *Advances in Cryptology — ASIACRYPT 2001*. За ред. Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, с. 552—565.

[10] Adi Shamir. «How to share a secret». В: *Commun. ACM* 22.11 (листоп. 1979), 612–613. ISSN: 0001-0782. DOI: 10.1145/359168.359176. URL: https://doi.org/10.1145/359168.359176.

[11] *What is Monero (XMR)?* URL: https://www.getmonero.org/get-started/what-is-monero/.