НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК 519.8

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“15” травня 2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 113 «Прикладна математика»

на тему: Побудова ефективної системи електронного голосування з використанням ланцюгів блоків

Виконав (-ла): студент (-ка) 2 курсу, групи _____ФІ-63М_____
(шифр групи)

Олешко Костянтин Андрійович

Керівник к.в.-м.н. Фесенко А.В.

-

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент д.т.н. Качинський А.Б.

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК _____

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“15” травня 2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 113 «Прикладна математика»
(код і назва)

на тему: Побудова ефективної системи електронного голосування з використанням ланцюгів блоків

Виконав: студент 6 курсу, групи ФІ-63м
(шифр групи)

Олешко Костянтин Андрійович _ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник старший викладач кандидат фізико-математичних наук Фесенко А.В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)
_____
(підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти – другий (магістерський)

Спеціальність 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Олешко Костянтину Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації: на тему: Побудова ефективної системи електронного голосування з використанням ланцюгів блоків ,
науковий керівник дисертації: Фесенко Андрій Вячеславович, к.ф.-м.н. ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 02.07.2018р. № 1058-с

2. Термін подання студентом дисертації 16.05.2018р.

3. Об'єкт дослідження: системи електронного голосування, які використовують ланцюги блоків транзакцій.

4. Предмет дослідження: властивості систем електронного голосування з використанням ланцюгів блоків та їх узгодження з існуючими вимогами до систем голосування.

5. Перелік завдань, які потрібно розробити: 1) розглянути властивості технології ланцюгів блоків транзакцій 2) виконати порівняльний аналіз публічних і приватних ланцюгів блоків 3) дослідити властивості існуючих систем електронного голосування, побудованих на основі технології ланцюгів блоків 4) дослідити конструкцію платформи Stellar та використання нею

ланцюгів блоків транзакцій 5) побудувати систему електронного голосування на основі протоколу Stellar та проаналізувати її властивості

6. Орієнтовний перелік ілюстративного матеріалу: 61 сторінка, 2 рисунки, 32 джерела.

7. Орієнтовний перелік публікацій:

_____

_____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вибір теми наукового дослідження	01.09.17 – 30.09.17	
2	Дослідження кластичних систем електронного голосування	01.10.17 – 31.10.17	
3	Дослідження технології ланцюгів блоків та найбільш поширених протоколів консенсусу	01.11.17 – 31.12.17	
4	Дослідження платформи Waves та аналіз системи голосування побудованої на її основі	01.01.18 – 28.02.18	
5	Дослідження платформи Stellar та побудова системи електронного голосування на її основі	01.03.18 – 31.03.18	
6	Оформлення звіту	01.04.18 – 30.04.18	

Студент _____ К. А. Олешко
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ А. В. Фесенко
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

ABSTRACT

The work contains 61 pages, 2 pictures, 32 references.

The subject of research is the electronic voting based on chains of the transaction blocks

The subject of the study is the properties of electronic voting systems based on blockchain and their coordination with existing requirements for electronic voting systems.

The aim of this thesis is to research the existing electronic voting platforms based on blockchain, to build an electronic voting system based on blockchain, to analyze the properties of the system and to verify compliance with the requirements of the electronic voting mechanisms

Research task is to consider the properties chains of the transaction blocks technology; carry out a comparative analysis of public and private blockchains; research of the properties of the existing electronic voting systems based on blockchain technology; investigate design of the Stellar platform and its way of using blockchain technology; build e-voting system based on Stellar protocol and analyze its properties.

Key words: HASH-FUNCTION, CRYPTOCURRENCY, STELLAR, WAVES, ELECTRONIC VOTING.

РЕФЕРАТ

Об’єм роботи 61 сторінка, 2 рисунки, 32 джерела.

Об’єктом дослідження є системи електронного голосування, які використовують ланцюги блоків транзакцій.

Предметом дослідження є властивості систем електронного голосування з використанням ланцюгів блоків та їх узгодження з існуючими вимогами до систем голосування.

Метою даної дипломної роботи є дослідження існуючих платформ для електронного голосування на основі ланцюгів блоків та побудова системи електронного голосування з використанням ланцюгів блоків, а також дослідження властивостей системи і перевірка на відповідність вимогам до механізмів електронного голосування.

Завдання дослідження – розглянути властивості технології ланцюгів блоків транзакцій; виконати порівняльний аналіз публічних і приватних ланцюгів блоків; дослідити властивості існуючих систем електронного голосування, побудованих на основі технології ланцюгів блоків; дослідити конструкцію платформи Stellar та використання нею ланцюгів блоків транзакцій; побудувати систему електронного голосування на основі протоколу Stellar та проаналізувати її властивості.

Ключові слова: ГЕШ-ФУНКЦІЯ, КРИПТОВАЛЮТА, STELLAR, WAVES, ЕЛЕКТРОННЕ ГОЛОСУВАННЯ.

ЗМІСТ

ВСТУП 7

1 ОСНОВНІ ПОЛОЖЕННЯ СИСТЕМ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ 9

1.1 Основні відомості 9

1.2 Механізми електронного голосування 13

1.3 Опис технології ланцюгів блоків 24

1.4. Порівняння публічних і приватних платформ, побудованих на основі технології ланцюгів блоків 29

Висновки до розділу 1 30

2 СИСТЕМА ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ ПЛАТФОРМИ WAVES 32

2.1 Особливості платформи Waves 32

2.2 Властивості системи голосування, побудованої на основі платформи Waves 33

2.3 Перевірка на відповідність вимогам до систем електронного голосування 36

Висновки до розділу 2 37

3 СИСТЕМА ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ ПЛАТФОРМИ STELLAR 38

3.1 Побудова системи електронного голосування 38

3.2 Перевірка на відповідність вимогам до систем електронного голосування 53

Висновки до розділу 3 56

ВИСНОВКИ 57

ПЕРЕЛІК ПОСИЛАНЬ 59

ВСТУП

Актуальність роботи: після створення криптовалюти Bitcoin та подальшого її дослідження, складові частини протоколу почали використовувати для вдосконалення існуючих та створення нових систем в різних галузях життєдіяльності, в яких є необхідність децентралізованого збереження і обробки даних.

Великого поширення набула технологія ланцюгів блоків, яка використовується в Bitcoin для запису транзакцій в базу даних. Також, ланцюги блоків слугують для перевірки факту здійснення транзакції, так як весь ланцюг блоків є загальнодоступним, та будь-хто може перевірити вхідні та вихідні дані транзакції. На даний момент часу загальна капіталізація ринку криптовалют складає понад 350 мільярдів доларів. Існує понад двох тисяч проектів, які пропонують рішення на базі технології ланцюгів блоків. Наприклад, Hyperledger Fabric – платформа для розподілених рішень ,побудованих на основі технології ланцюгів блоків, яка забезпечує високий рівень конфіденційності, стійкості, гнучкості та масштабованості. Наступним прикладом є Corda, яка представляє собою платформу на основі ланцюгу блоків Ethereum з обмеженим доступом, що використовує технологію розумних контрактів та призначена для використання виключно фінансовими установами.

Технологія ланцюгів блоків активно впроваджується в фінансовий сектор. Яскравим прикладом такого впровадження є платформи Stellar та Ripple, які дозволяють фінансовим організаціям здійснювати валютні операції протягом декількох секунд, та з дуже низьким значенням комісії. Окрім цього на основі платформи Stellar можуть бути створені та випущені нові криптовалюти, правила обігу яких будуть повністю контролюватись емітентом.

Одним із можливих напрямків застосування ланцюгів блоків є впровадження їх у системи електронного голосування. Це дасть змогу дозволить будувати системи голосування з різними необхідними властивостями, можливо, навіть новими, але для цього необхідно проводити дослідження та бути впевненим у їх достовірності. На даний момент вже існують платформи для проведення електронного голосування на основі платформ Ethereum, Zcash і Waves.

Об'єкт дослідження: системи електронного голосування, які використовують ланцюги блоків транзакцій.

Предмет дослідження: властивості систем електронного голосування з використанням ланцюгів блоків та їх узгодження з існуючими вимогами до систем голосування.

Метою роботи є: дослідження існуючих платформ для електронного голосування на основі ланцюгів блоків та побудова системи електронного голосування з використанням ланцюгів блоків, а також дослідження властивостей системи і перевірка на відповідність вимогам до механізмів електронного голосування.

Завдання дослідження: розглянути властивості технології ланцюгів блоків транзакцій; виконати порівняльний аналіз публічних і приватних ланцюгів блоків; дослідити властивості існуючих систем електронного голосування, побудованих на основі технології ланцюгів блоків; дослідити конструкцію платформи Stellar та використання нею ланцюгів блоків транзакцій; побудувати систему електронного голосування на основі протоколу Stellar та проаналізувати її властивості.

1 ОСНОВНІ ПОЛОЖЕННЯ СИСТЕМ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ

В даному розділі будуть розглянуть класичні системи електронного голосування, а також найбільш поширені алгоритми консенсусу на платформах що побудовані на основі ланцюгів блоків.

1.1 Основні відомості

Для розгляду систем електронного голосування необхідно ввести означення системи електронного голосування.

Означення 1.1 Електронне голосування є процесом, в якому виконуються завдання щодо збору та підрахунку голосів з використанням електронних засобів. Системи електронного голосування, на відміну від звичайного голосування, дозволяють людям голосувати з будь-якого місця, якщо вони мають доступ до мережі Інтернет.

При побудові системи електронного голосування буде використовуватись технологія ланцюгів блоків. Для розгляду технології ланцюгів блоків наведемо означення геш-функції та криптографічної геш функції. Адже з їх допомогою формується ланцюг блоків.

Означення 1.2 [1] Геш-функція h це обчислювально ефективна функція виду $h: \{0,1\}^* \to \{0,1\}^n$

Будь-яка функція, що задовольняє нижче наведеним умовам, може називатись геш-функцією:

1) Стиснення $-$ h відображає вхідне значення x певної скінченної довжини в вихідне значення $h(x)$, що має зазначену довжину n.

2) Простоту обчислення $-$ дане h для будь-якого вхідного значення

x таке, що $h(x)$ легко обчислюється.

Для використання у криптографії геш-функція має задовольняти більшій кількості умов.

Означення 1.3 [1] Геш-функція (без ключа) h називається криптографічною геш-функцією, якщо вона задовольняє таким умовам (x, x' – вхідні значення, y, y' – вихідні):

1) Стійкість до пошуку першого прообразу – для всіх вихідних значень практично неможливо знайти таке вхідне значення, яке дає таке геш-значення. Тобто знаходження прообразу x' такого, що $h(x') = y$ коли відоме тільки геш-значення y, а значення відповідного входу невідоме матиме ймовірність не більшу за ε:

$$P(h(x') = y) < \varepsilon$$

2) Стійкість до пошуку другого прообразу – практично неможливо знайти будь-яке інше вхідне значення, що дає таке саме геш-значення, що й задане. Тобто імовірність для даного x знайти другий прообраз $x' \neq x$ такий, що $h(x') = h(x)$ буде меншою за ε:

$$P(h(x') = h(x)\ ) < \varepsilon$$

3) Стійкість до пошуку колізій – практично неможливо знайти два відмінні вхідні значення, що дають однаковий геш. Тобто імовірність знайти такі $x' \neq x$, що $h(x') = h(x)$ буде меншою певного заданого ε:

$$P(h(x') = h(x)\ ) < \varepsilon$$

Означення 1.4 [1] Геш-функція, що задовольняє умовам стійкості до пошуку першого, другого прообразу і колізій називається стійкою або надійною/

Для розгляду протоколів консенсусу і побудови системи електронного голосування введемо означення ланцюга блоків.

Означення 1.5 [2] Ланцюг блоків – розподілена база даних, яка складається з блоків даних, криптографічно пов'язаних між собою [2]. Кожен блок містить узгоджений набір транзакцій, що пройшли перевірку відповідно до протоколу консенсусу.

Більшість платформ, побудованих на основі технології ланцюгів блоків, використовують майнінг для перевірки і прийняття транзакцій.

Означення 1.6 [2] Майнінг — діяльність з підтримки розподіленої платформи і створення нових блоків з можливістю отримати винагороду в формі емітованої валюти і комісійних зборів у різних платформах.

Дані ланцюга блоків зберігаються на вузлах мережі, які узгоджуються щодо оновлення стану системи. Оновлення стану системи відбувається в результаті досягнення консенсусу між вузлами.

Означення 1.7 Стан системи - сукупність станів облікових записів у певний момент часу.

Означення 1.8 Консенсус - угода множини вузлів мережі щодо стану системи у певний момент часу.

Означення 1.9 Вузол мережі - комп'ютер, що перевіряє усі транзакції у системі, і у разі обчислення блоку передає цю інформацію іншим вузлам мережі. Працює під керівництвом спеціалізованого програмного забезпечення.

Для забезпечення автентифікації дій користувачів та забезпечення цілісності транзакцій в платформах, побудованих на основі технології ланцюгів блоків, використовується електронний цифровий підпис.

Означення 1.10 Електронний цифровий підпис (ЕЦП) — вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього повідомлення або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписанта [3]. Електронний цифровий підпис

накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Будь-які дії користувача в платформах, побудованих на основі технології ланцюгів блоків, виконується за допомогою виконання транзакцій.

Означення 1.11 Операція - це певна одинична дія з обмеженої множини усіх можливих дій у системі, що визначає певні зміни облікових записів.

Означення 1.12 Транзакція - група послідовних операцій зі зміни станів облікових записів, що може атомарно прийнятися ядром системи або відхилитися.

Транзакції приймаються або відхиляються за певним протоколом консенсусу. Для опису протоколу консенсусу необхідно ввести наступі означення:

Означення 1.13 Федеративна візантійська система узгодження, або FBAS, є пара $\langle V, Q \rangle$, що складається з набору вузлів V та кворумної функції $Q : V \to 2^{2^V} \backslash \{\emptyset\}$, що визначає один або декілька кворумних підмножин для кожного вузла, де вузол входить до всіх власних кворумних підмножин, тобто $\forall v \in V, \forall q \in Q(v), v \in q$ [4]. (2^X позначає потужність иножини X)

Означення 1.14 Кворум - множина вузлів $U \subseteq V$ в FBAS $\langle V, Q \rangle$ представляє собою кворум якщо $U \neq \emptyset$ і U містить кворумні підмножини для кожного члена, тобто $\forall v \in U, \exists q \in Q(v), v \in q$ такий, що $q \subseteq U$ [4].

При гешуванні даних для забезпечення більш високого рівня безпеки використовується сіль.

Означення 1.15 Сіль - рядок даних, який передається геш-функції разом з паролем. Значенням солі є рядок непов'язаних із собою даних, що мають довільний розмір [5].

1.2 Механізми електронного голосування

При розгляді електронних рішень для голосування, повинна бути проведена чітка різниця між тим, що зазвичай називають І-голосуванням, або інтернет-голосуванням, та більш традиційною формою електронного голосування, яка використовує контрольовані кабіни для голосування, де фактично голосування проводиться на електронному пристрої.

Також слід зазначити, що існує категорія електронних рішень для голосування, які залишаються дуже близькі до поняття фізичного, паперового бюлетеня. Ці системи, в основному, є електронним інтерфейсом для створення паперових голосувань, які потім можуть бути більше ефективно підраховані. Для цього існують різні підходи, наприклад перфокарти, електронне розпізнавання маркерних міток або друкованих голосів.

Схема голосування повинна задовольняти певним *загальним вимогам до безпеки* та *додатковим вимогам* (які показують захищеність від атак зловмисника), а також деяким *специфічним вимогам по впровадженню системи*.

Загальні вимоги [6]:

a) *Правомочність (Eligibility)*. У будь-якій схемі голосування мають право голосувати, тільки правомочні виборці, які відповідають певним, заздалегідь визначеним, критеріям. Можливість перевіряти правомочність виборця і механізм забезпечення контролю кількості голосів, які може віддати виборець, є обов'язковим для схеми голосування.

b) *Конфіденційність (Privacy)*. Таємне голосування. Голосування не повинно ідентифікувати виборця. Будь-який зв'язок між виборцем та його голосом повинен бути знищений. Максимальна конфіденційність схеми голосування досягається, якщо конфіденційність виборця порушується тільки при змові всіх інших суб'єктів (виборців та влади).

c) *Перевірка (Verifiability)*. Виборець повинен мати можливість перевірити, чи був його голос записаний правильно та враховано в остаточному підрахунку. Є два різновиди цієї вимоги. Одним з них є *індивідуальна перевірка*, де тільки виборець може перевірити свій голос. Другий - *універсальна перевірка*, де після того, як підрахунок публікується, кожен може переконатися, що всі правомочні голоси були включені, і процес підрахунку був точний. Універсальна перевірка є більш практичною, оскільки є нереальним дозволити виборцям перевіряти свої голоси окремо. Вимога *перевірки* потребує щоб виборець був зв'язаний зі своїм голосом, що є протиріччям до вимоги конфіденційності. Однак, цей пункт має вирішальне значення в отриманні довіри виборця до системи голосування.

d) *Врегулювання (Dispute-freeness)*. Схема голосування повинна надати механізм для врегулювання всіх суперечок на будь-якій стадії. Поняття універсальної перевірки аналогічне, але обмежується голосуванням і підрахунком етапів. *Dispute-freeness* може зробити схему складною.

e) *Точність (Accuracy)*. Схеми голосування повинні бути вільні від помилок. Голоси повинні бути правильно записані і підраховані. Голоси неправомочних виборців не повинні враховуватися. Універсальна перевірка - властивість прямо пов'язана з точністю.

f) *Довгострокова конфіденційність (Long-term privacy)*. В деяких випадках, виборцю повинна бути надана довгострокова конфіденційність.

g) *Чесність (Fairness)*. Для того, щоб провести неупереджені вибори, ніхто не повинен бути в змозі обчислити частковий результат (як вибори прогресують).

Крім загальних функцій безпеки, схема голосування повинна бути стійка до атак зловмисника. Для забезпечення стійкості, повинні бути дотримані такі вимоги:

a) *Надійність (Robustness)*. Схема повинна бути стійкою до активних або пасивних атак (корумпованої влади/виборців), а також проблеми та перешкоди (влади/виборців, які не беруть участі у голосуванні). Схема

голосування досягає *максимальної надійності*, якщо вона надійна при змові всіх виборчих органів влади зірвати вибори. Але це також вимагає, щоб усі виборчі органи влади брали участь у проведенні виборів. Будь-який виборчий орган, який не бере участі у голосуванні, може також зірвати вибори, що призводить до нульової стійкості та помилок. В схемі голосування існує, внутрішній конфлікт між стійкістю до атак зловмисника і стійкістю до несправностей.

b) *Не підтвердження голосу (Receipt-freeness)*. Виборець не повинен бути забезпечений «квитанцією», з якої він може підтвердити свій голос будь-якій іншій особі. *Підтвердження голосу* має те ж саме поняття, що і непростежуваність або конфіденційність.

c) *Маніпуляція (Incoercibility)*. Забезпечення стійкості від маніпуляції. Зловмисник може спробувати примусити виборця і маніпулювати голосуванням. Зловмисник може, також, змусити виборця утримуватися від голосування, або навіть може представляти правомочного виборця на будь-якій стадії схеми голосування, шляхом отримання секретного ключа виборця.

3) Вимоги до реалізації системи:

a) *Масштабованість (Scalability)*. Складність протоколів, використовуваних в схемі голосування, є основним фактором його практичної реалізації. Ефективна схема голосування повинна бути масштабованою для зберігання, обчислення та комунікаційних потреб.

a) *Практичність (Practicality)*. Практична схема голосування не повинна мати вимоги, які можуть бути важко реалізувати у великих масштабах.

Розподіл секрету. Обмін секретом - це методика, що дозволяє n сторонам ділитися секретом, з властивістю, що жодна сторона не може відновити секрет без участі інших. Існують також схеми, де потрібне співробітництво, принаймні, $n - a$ сторін, так звана *порогова* схема. (k, n) порогова схема ділить секрет на $k + n$ частин і вимагає співпраці, принаймні, k сторін для того, щоб відновити секрет.

Добре відомим алгоритмом, який реалізує дану стратегію, є поділ секрету Шаміра[7]. Алгоритм заснований на властивості, що многочлен ступеня m однозначно визначається $M + 1$ точками на кривій. Поліном вважається секретом, і одна точка на кривій полінома буде дана кожній зі сторін. Якщо ми маємо k сторін, ми повинні використовувати поліном ступеня $k + 1$ для того, щоб мати можливість відновити многочлен. Можна легко розширити цю схему до порогової схеми, просто генеруючи ще кілька точок на кривій і розподіливши їх до "додаткових" сторін. Таким чином, можна побудувати (k, n) порогову схему, вибираючи поліном ступеня $k - 1$ і генеруючи $k + n$ частин, з яких кожна зі сторін отримує одну.

У той час, як поліном може бути відновлений тільки принаймні k частинами, то зловмисник, який має меншу кількість частин, має ще деяку інформацію про таємниці. Проте, поділ секрету Шаміра повинен використовуватися в скінченному полі, що підвищить безпеку, так як знання безлічі точок розміром менше k більше не буде надавати атакуючому знання про таємницю.

Гомоморфне шифрування. Інший метод, який має дуже цікаві застосування в електронних схемах голосування, це гомоморфне шифрування. Схема з гомоморфними властивостями дозволяє проводити обчислення, основані на шифротексту, без необхідності дешифрування вхідних значень, а потім повторно шифрувати результат. Схема шифрування гомоморфна, якщо з даних шифртекстів $\varepsilon(a)$ та $\varepsilon(b)$ можна отримати $\varepsilon(a \times b)$ для деякої операції $\times$.

У випадку електронного голосування, ми зацікавлені в операції додавання, що дозволяє нам обчислити $\varepsilon(a + b)$ без дешифрування $\varepsilon(a)$ або $\varepsilon(b)$. Це цікава властивість, так як вона дозволяє додавати голоси разом без необхідності розшифрування будь-якого з окремих голосів. Таким чином, підрахунок може бути зроблено в безпечний та конфіденційний для користувача спосіб, так як тільки результат складання повинен бути розшифрований.

Прикладом криптосистеми з гомоморфною властивістю є модифікований алгоритм ElGamal [8]. У звичайному ElGamal ми маємо відкритий ключ $h = g^x$ з секретним ключем x. Функція шифрування

$$E(m) = (g^r, m \cdot h^r)$$

з випадковим r. ElGamal набуває гомоморфності при вдосконаленні, коли замість шифрування m використовується g^m. Цей варіант називається експоненціальною схемою ElGamal [9][10].

Одна з проблем полягає в тому, що при розшифруванні кінцевого результату, отримуємо g^{a+b}. Для того, щоб отримати фактичне значення $a + b$ потрібно було б вирішити проблему дискретного логарифму. У цьому випадку, це можливо. Припустимо, що ми маємо n голосів $(v_0, v_1, \dots, v_{n-1})$, де кожен голос або 0 («ні») або 1 («так»).

Оскільки простір рішень невеликий, ми можемо просто перебирати різні можливі значення сум для того, щоб знайти розв'язок.

Є також підходи до електронного голосування, які використовують регулярне шифрування ElGamal, як гомоморфне при множенні. Для того, щоб розробити надійне рішення для електронного голосування, яке використовує гомоморфне шифрування, повинно бути розглянуто багато ускладнень, щоб зменшити витік інформації і гарантувати цілісність [11].

Розглянемо опис криптографічних примітивів та модулів, які складають протоколи електронного голосування.

Анонімний канал зв'язку (mixnet). Під час подачі голосів, недоторканість приватного життя виборців може бути забезпечена шляхом приховування особи виборця. Це досягається за допомогою використання анонімного каналу. Анонімний канал - це канал зв'язку, де виборець (sender) є анонімним до влади (receiver) і до будь-якого спостерігача повідомлення.

Рисунок 1.1 представляє собою схематичне зображення загальної системи mixnet. На етапі i (Mix_i), вхід $E_K(m_j, r_j)$, $j = \overline{1, n}$ приймаються і перетворюються з використанням або розшифрування або шифрування,

переставляються і передаються далі на $i+1$. На основі криптографічного перетворення, що використовується, mixnet називається розшифрувальний mixnet (decryption mixnet) або перешифрувальний mixnet (re-encryption mixnet).

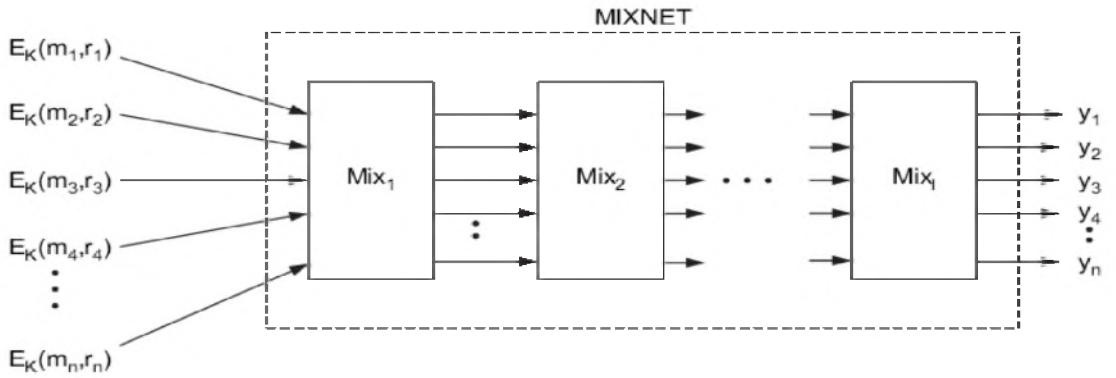

Рисунок 1.1 - Загальна схема mixnet.

Розшифрувальний mixnet. Нехай K_i відкритий ключ і-го етапу. Нехай, у схемі голосування, відправник mixnet буде виборець, а отримувач - орган влади. Відправник V_j зв'язує повідомлення m_j з випадковою рядком r_j як $(m_j \| r_j)$, далі шифрує $E_{K_1}\left(E_{K_2}\left(\ldots\left(E_{K_l}(m_j\|r_j)\right)\ldots\right)\right)$ та передає. Mix_i з особистим ключем K_i^{-1}, приймає вхідні сигнали, як $E_{K_i}\left(E_{K_{i+1}}\left(\ldots\left(E_{K_l}(m_j\|r_j)\right)\ldots\right)\right)$ від Mix_{i-1}.

Алгоритм розшифрувального mixnet:

Вхід. $E_{K_1}\left(E_{K_2}\left(\ldots\left(E_{K_l}(m_j\|r_j)\right)\ldots\right)\right); j=\overline{1,n};\ i=\overline{1,l}.$

1) Розшифровуємо $D_{K_i}\left(E_{K_i}\left(E_{K_{i+1}}\left(\ldots\left(E_{K_l}(m_j\|r_j)\right)\ldots\right)\right)\right)=$

$=E_{K_{i+1}}\left(\ldots\left(E_{K_l}(m_j\|r_j)\right)\ldots\right)$

2) У лексикографічному порядку розшифровуємо величини, які отримані на попередньому кроці.

Вихід. $\{m_j\}_R$ - набір змішаних повідомлень, які не можуть бути простежені назад до відправників.

Перешифровальний mixnet. Відправник V_j , використовуючи випадковий рядок r_j, шифрує повідомлення m_j *як* $E_K(m_j, r_j) = (g^{r_j}, K^{r_j} m_j)$, де $K = g^S$ є відкритим ключем отримувача. Тоді алгоритм mixnet виглядає наступним чином.

Алгоритм *перешифрувального* mixnet:

Вхід. $E_K(m_j, r_j) = (g^{r_j}, K^{r_j} m_j)$; $j = \overline{1,n}$; $i = \overline{1,l}$.

1) Перешифровуємо з випадковим рядком $r_{i,j}$:

$$E_K\left(m_j, r_j + \sum_{a=1}^{i-1} r_{a,j} + r_{i,j}\right) = \left(g^{r_j + \sum_{a=1}^{i} r_{a,j}}, K^{r_j + \sum_{a=1}^{i} r_{a,j}} m_j\right).$$

2) Випадковим чином переставляємо всі повторно зашифровані величини, отримані на кроці 1.

Вихід. $\left\{E_K\left(m_j, r_j + \sum_{a=1}^{l} r_{a,j}\right) = \left(g^{r_j + \sum_{a=1}^{l} r_{a,j}}, K^{r_j + \sum_{a=1}^{l} r_{a,j}} m_j\right)\right\}_R$, набір ймовірнісних шифрувань, які не можуть бути простежені назад до відправників.

У mixnet, порушення можуть відбуватись, коли «корумпований» Міх може спробувати змінити деякі з його входів або дефектний Міх просто не в змозі виконати свою роботу. Шляхом введення перевірки змішування, mixnets були зроблені стійкими до кількох «корумпованих» Міх, але стійкість до дефектної Міх все ще залишається проблемою в модифікованих mixnet.

Підходи до покращення надійності та ефективності mixnets можна широко класифікувати як оптимістичний та песимістичний підхід. Оптимізм дозволяє досягти максимальної продуктивності при відсутності порушень в будь-якому Міх. Проте, міцність буде принесена в жертву, так як порушення вимагають повторення всього перемішування, що призводить до погіршення ефективності. З іншого боку, песимізм забезпечує перевірку на кожному етапі і не має механізму для підвищення ефективності, коли немає збою.

Дошка оголошень (bulletin board)

Вимога перевірки голосування і виборців може бути досягнута за допомогою загальнодоступної дошки оголошень [12]. Дошка оголошень є публічним, широкомовним каналом зв'язку, який має пам'ять. Будь-яка інформація, яка транслюється зберігається в пам'яті і доступна для читання.

Дошка оголошень може містити розділи для виборців, які мають право голосу. Правомочний виборець має доступ до призначеної йому секції. Така дошка може бути реалізована з використанням декількох надійних серверів. У схемі голосування кожен виборець має право опублікувати свій голос в розділі. Це дозволяє перевіряти, що голос зарахований правильно. Орган також використовує дошку оголошень для розміщення інформації. Спеціальна форма дошки не містить ніяких спеціальних розділів для визначення зв'язку між виборцем і його голосом.

Сліпий підпис представляє собою криптографічний протокол, який можна використаний для того, щоб зробити голос анонімним. Протокол може бути описаний таким чином:

Крок 1. Виборець V приховує свій голос v використовуючи випадковий рядок r і відкритий ключ K_A вповноваженого A, $BV = blind(v, r, K_A)$. Потім підписує BV, використовуючи свій закритий ключ K_V^{-1}, *як* $sign_V(BV, K_V^{-1})$ та відсилає його до A.

Крок 2. A перевіряє правильність V (шляхом перевірки підпису з відкритим ключем K_V), потім підписує BV закритим ключем K_A^{-1}, як $sign_A(BV, K_A^{-1})$ та відсилає його до V.

Крок 3. V перевіряє підпис A а потім видаляє r , щоб отримати $sign_A(v, K_A^{-1})$, що є сліпим підписом голосу v.

Такий протокол був запропонований з використанням криптосистеми RSA, де $blind(v, r, K_A) = r^{K_A} v$.

Гомоморфне шифрування.

Алгоритм шифрування E_K, називається гомоморфним, якщо дано $E_K(m_1)$ та $E_K(m_2)$, можна отримати $E_K(m_1 \odot m_2)$, не розшифровуючи

m_1 та m_2 окремо, для деяких операцій $\odot$. Операція $\odot$ може бути модульним додаванням ($\oplus$, адитивний гомоморфізм) або множення ($\otimes$, мультиплікативний гомоморфізм).

RSA з відкритим ключем криптосистема, яка володіє мультиплікативним гомоморфізмом, тоді як ElGamal криптосистем адитивного гомоморфізму.

Секретний обмін. Єдиний довірений орган влади може призвести до того, що вибори можуть бути корумповані або дефектні. Надійність, в цьому випадку, може бути вирішена шляхом розподілу довіри (секрету; наприклад, ключа розшифрування) між декількома органами. $(\boldsymbol{t}, \boldsymbol{k})$-порогова схема розподілу секрету, де $\boldsymbol{t} \leq \boldsymbol{k}$ може бути використана для спільного поділу секрету $\boldsymbol{S}$ між $\boldsymbol{k}$ довіреними органами. Схема потребує, щоб існувала певна довірена сторона $\boldsymbol{T}$, яка, за допомогою протоколу генерації спільного ключа будує секретний ключ $\boldsymbol{S} = \boldsymbol{K}^{-1}$, публікує відкритий ключ $\boldsymbol{K}$ і генерує $\boldsymbol{k}$ частин секретного ключа. Довірений орган $\boldsymbol{A_i}$, отримує частку $\boldsymbol{s_i}$ від $\boldsymbol{T}$. Для відновлення секретного ключа, $\boldsymbol{t}$ або більше органів влади повинні представити свої частини секрету, які потім об'єднуються. Секретний ключ захищений від змови $\boldsymbol{t-1}$ корумпованих і $\boldsymbol{k-t}$ недійсних органів влади. У схемах голосування, Т може бути третя сторона або виборець.

У перевірочній схемі розподілу секрету, тільки всі k органів можуть перевірити протокол і, отже, будь-яка суперечка повинна бути вирішена третьою довіреною стороною. У публічній перевірочній схемі розподілу секрету, будь-яка зовнішня сторона може перевірити правильність протоколів на кожному етапі схеми.

Система Helios. Helios - це система, яка реалізує “голосування з криптографічним аудитом”. *Helios* є програмним забезпечення з відкритим вихідним кодом. Через особливості побудови системи Helios, довіра серверу не потрібна: процес голосування залишається повністю перевірним, навіть якщо адміністратори системи є зловмисниками. Helios забезпечує гарантію

голосування, яка означає, що користувач може переконатися, що його голос був фактично взятий до уваги при остаточному підрахунку. Вона також забезпечує універсальну здатність підтвердження, маючи на увазі, що будь-який спостерігач (або користувач) може перевірити, що всі голоси були підраховані і підраховані правильно.

У статті, в якій Helios був вперше представлений на Usenix в 2008 році, автори стверджують, що якщо система повністю скомпрометована, конструкція системи може підтримувати лише або цілісність або конфіденційність [13]. Цей перший варіант Helios був розроблений, щоб забезпечити здатність перевірки, при цьому зменшивши рівень секретності. У більш пізній версії, на основі пропозиції Д. Деміреля та ін., гарантії конфіденційності були посилені за допомогою гомоморфності шифрування та багатопартійну підрахунку [14]. Це було реалізовано у третій версії Helios, яка буде розглянута в цьому розділі.

Важливий вибір дизайну Helios є те, що вони повністю ігнорують захист від примусу. У той час як інші системи обмежують можливість примусу, автори Helios прийняли його і навіть реалізували кнопку “виконай”, яка буде надсилати голос у вигляді звичайного тексту, в супроводі деталей, необхідних для перевірки факту здійснення голосу, з використанням адреси електронної пошти користувача.

Наслідком такої конструкції є те, що вибір Helios не є підходящим вибором для виборів з високою явкою, де інтереси такі, що великомасштабний примус може стати ефективним варіантом для атакуючого.

Використання. Коротко розглянемо, як працює процес голосування Helios. Ми припускаємо, що вибори вже створені, і що якийсь користувач Аліса хоче проголосувати. Зверніть увагу, що порожній виборчий бюлетень можна розглядати і заповнювати будь-ким, в будь-який час, без автентифікації. Автентифікація (в даному контексті, верифікація ідентичності і права на виборах) проводиться тільки в момент видання бюлетеня системою Helios.

1) Аліса починає процес голосування, вказавши в яких виборах вона хоче прийняти участь.

2) Виборча Система Підготовки (BPS) проводить Алісу через всі питання, що були винесені на голосування, і записує її вибір.

3) Після підтвердження BPS шифрує відповіді разом з деякими випадковими даними. Геш шифр тексту служить підтвердженням завершення процесу шифрування.

4) Тепер Аліса може вибрати: перевірити бюлетень, або спакувати та відправити.

a. Якщо вона здійснює перевірку: шифр текст і ключ віддається Алісі, яка тепер може перевірити, чи збігаються ці дані із підтвердженням, і якщо він розшифровується до голосу, який вона хотіла відправити. Якщо все в порядку, BPS поновлюється на кроці 3 та повторно шифрує вибір з новою випадковістю, ще раз дозволяє Алісі вибрати перевірку або відправку.

b. Якщо вона бажає відправити бюлетень, усі випадкові дані та відкриті тексти будуть остаточно видалені. Залишаться тільки шифртексти. BPS продовжується з наступного кроку.

5) Алісі пропонується здійснити перевірку на справжність, і якщо перевірка завершується успішно, зашифрований голос реєструється, як голос Аліси.

6) Helios робить можливість примусу явною, показуючи кнопку “виконай”. Ця кнопка дозволяє Алісі посилати зашифрований текст, випадкові дані та відкритий текст на будь-яку адресу електронної пошти. Як згадувалося раніше, це робиться, щоб явно показати, що протидія примусу не є властивістю Helios.

7) Зашифрований голос Аліси тепер показується на дошці оголошень. Тут містяться всі голоси, які були зроблені до даного моменту. Кожен голос або пов'язаний з ім'ям виборця або до деякої кількості ідентифікаторів. Той, хто голосував, може знайти його зашифрований голос на

дошці. Аліса може перевірити, чи її голос дійсно вірно зарахований, і чи це дійсно її голос.

8) Після закриття виборів, вибори офіційно працюють разом для обчислення суми зашифрованих голосів. Це робиться за допомогою безпечного обчислення роздільної системи і гомоморфності шифрування.

9) Публікуються результати виборів. Будь-який бажаючий може перевірити, використовуючи дошку голосів, що його голос був прийнятий до уваги і що додавання голосів було зроблено правильно.

1.3 Опис технології ланцюгів блоків

Ланцюг блоків представляє собою побудований за певними правилами ланцюг блоків, що містять інформацію певну інформацію [2]. Найчастіше копії ланцюгів блоків зберігаються і незалежно один від одного обробляються множиною вузлів мережі.

Блок транзакцій - спеціальна структура для запису групи транзакцій в ланцюг блоків. Транзакція вважається завершеною і достовірною, коли вона пройшла перевірку на правильність формату і підпису, і коли сама транзакція об'єднана записана в блок транзакцій. Всі блоки записуються в один ланцюг, який містить інформацію про всі проведені транзакції. Найперший блок в ланцюжку – первинний блок. Первинний блок - розглядається як окремий випадок, так як у нього відсутній батьківський блок.

Блок складається з заголовку і списку транзакцій. Заголовок блоку включає в себе геш блоку, геш попереднього блоку, геші транзакцій і додаткову службову інформацію. Список транзакцій, формується з черги транзакцій, не записаних в попередні блоки. Критерій відбору з черги задається вузлами мережі. Це не обов'язково повинна бути хронологія за часом. Наприклад, можуть включатися тільки операції з високою комісією

(наприклад в системі Bitcoin) або за участю заданого списку адрес [15]. Для транзакцій в блоці використовується деревоподібна схема гешування – дерево Меркла (зображене на рисунку 1.2), аналогічне формування геш-суми для файлу використовується в протоколі BitTorrent [16].

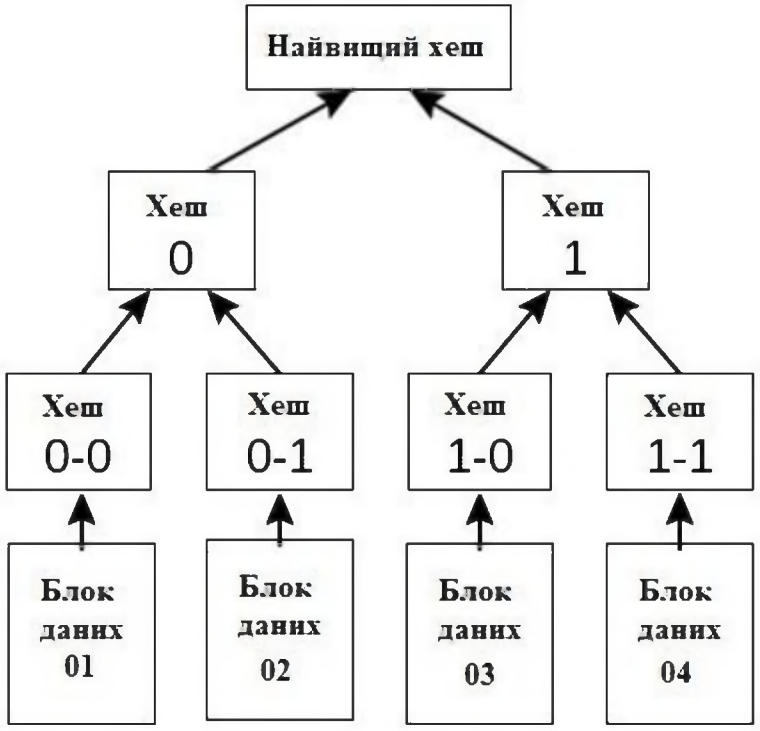

Рисунок 1.2 - Дерево Меркла

Створений певним вузлом мережі блок буде прийнятим іншими вузлами у тому випадку, якщо він був сформований відповідно до протоколу досягнення консенсусу. Наприклад, в системі Bitcoin блок буде прийнятий вузлами мережі, якщо числове значення гешу заголовка рівне або менше від певного параметру складності, величина якого змінюється в визначений період часу. Так як результат гешування функції SHA-256 [17] вважається незворотнім, на даний момент немає алгоритму отримання бажаного результату, крім випадкового перебору. Якщо геш не задовольняє умові, то в заголовку блоку змінюється параметр *nonce* і геш перераховується [15]. Зазвичай потрібна велика кількість перерахунків. Коли бажане значення гешу знайдене, вузол надсилає отриманий блок іншим вузлам мережі, які

перевіряють блок. Якщо помилок немає, то блок вважається доданим в ланцюг і наступний блок повинен додавати в свій заголовок його геш.

Значення параметру складності, з яким порівнюється геш, в системі Bitcoin коригується через кожні 2016 блоків. Мережа Bitcoin повинна витрачати на генерацію одного блоку приблизно 10 хвилин, а на 2016 блоків - близько двох тижнів. Якщо 2016 блоків сформовані швидше, то значення складності зменшується і знайти геш блоку стає складніше. Зміна параметру складності обчислень не впливає на надійність мережі Bitcoin і потрібна лише для того, щоб система генерувала блоки з визначеним інтервалом.

Робота по формуванню блоків ведеться одночасно всіма вузлами мережі. Ті блоки, що задовольняють критеріям системи, додаються в усі реплікації розподіленої бази блоків. Регулярно виникають ситуації, коли кілька нових блоків в різних частинах розподіленої мережі називають попереднім один і той же блок, тобто ланцюг блоків може розділитися. У цьому випадку можливе паралельне нарощування різних ланцюгів. У кожному з блоків, що знаходяться на однаковій висоті можуть зустрічатися як однакові транзакції, так і різні, що увійшли тільки в один з них. В такому випадку вузли мережі починають вважати головним той ланцюг, який отримує своє продовження. Таким чином у разі рівного розподілу складності і довжини перевага віддається тому ланцюгу, кінцевий блок якого з'явиться раніше. Транзакції, що увійшли в відхилений ланцюг (в тому числі по виплаті винагороди), втрачають статус підтверджених. Якщо це транзакція з передачі Bitcoin, то вона буде поставлена в чергу і потім включена в черговий блок. Транзакції з отримання винагороди за створення відкинутих блоків не дублюються в іншому ланцюгу, тобто винагорода виплачена за формування відсічених блоків втрачається.

Копії ланцюга блоків або її частини одночасно зберігаються на всіх вузлах мережі і синхронізуються відповідно до формальних правил побудови ланцюга блоків. Інформація в блоці не зашифрована і доступна у відкритому

вигляді, але відсутність змін засвідчується криптографічними засобами за допомогою геш-функції, яка підтверджує цілісність та незмінність ланцюга блоків.

Відомі протоколи досягнення консенсусу:

Proof of Work (PoW - дослівно: доказ виконання роботи) - алгоритм захисту розподілених систем від зловмисних дій [18]. Суть протоколу зводиться до двох основних пунктів:

1. Необхідності виконання певної досить складного і тривалого завдання;
2. Можливості швидко і легко перевірити результат.

Завдання PoW завжди можна виконати за скінченну кількість часу, однак це вимагає великих обчислювальних потужностей. При цьому перевірка отриманого рішення займає набагато менше ресурсів і часу.

В мережі Bitcoin механізм PoW використовується як засіб досягнення консенсусу. За основу був взятий проект Hashcash, до якого був доданий механізм змінюваної складності - зменшення або збільшення N (потрібного числа нулів) залежно від загальної потужності учасників мережі. Обчислювальною функцією стала криптографічна геш функція SHA-256.

Механізм PoW забезпечує здатність учасників системи перевірити правильність розрахунків виконаних вузлом мережі. Даний процес включає в себе спробу знайти геш блоку, який відповідатиме поточному рівню складності.

Ці розрахунки можуть проводитися тільки в інтерактивному режимі і складність встановлюється на такому рівні, щоб ланцюг доповнювався новими блоками в певні часові інтервали. У той же час перевірка результатів обчислень залишається простою. Учасники системи завжди можуть підтвердити, що вузол мережі знайшов коректне значення, однак, оскільки

процес обчислення гешу блоку випадковий, неможливо точно визначити який саме вузол вирішить завдання і додасть новий блок до ланцюгу.

Для того, щоб система визнала блок вузол, необхідно, щоб значення його гешу було меншим в порівнянні з поточним значенням складності. Таким чином, геш значення кожного блоку доводить, що була виконана певна робота по його виявленню.

Кожен блок містить попередній блок, що утворює ланцюг. Змінити блок заднім числом неможливо. Для проведення такого процесу необхідно виконати роботу по обчисленню геш значень всіх попередніх блоків. Висока складність цього процесу захищає ланцюг блоків від несанкціонованого доступу та подвійних витрат.

Proof of Stake (PoS) - альтернативний механізм консенсусу, вперше реалізований в 2012 році в криптовалюті PPCoin (зараз відома під назвою PeerCoin) [18]. Ідея полягає у використанні «частки» (stake) в якості ресурсу, який визначає, який саме вузол мережі отримує право додавання наступного блоку в ланцюг.

У підході Proof-of-Stake вузли також виконують гешування даних в пошуках результату меншого за певне значення складності, але складність в даному випадку розподіляється пропорційно, відповідно до балансу даного вузла. Іншими словами - відповідно до кількості монет на рахунку користувача.

Таким чином, більше шансів згенерувати наступний блок має вузол з великим балансом. Схема виглядає досить привабливо насамперед через невеликі вимоги до обчислювальних ресурсів.

Також існує певна множина протоколів консенсусу, в основі яких лежить «Byzantyne Fault Tolerance» [20]. Це підхід для досягнення консенсусу в мережі, яка включає в себе вузли, що виконують зловмисні дії або вузли з технічними помилками. Протокол продовжує безвідмовно працювати в

умовах, коли певна кількість вузлів (менше третини від загальної кількості) ненавмисно або навмисно намагається порушити коректне функціонування системи шляхом відправки хибних або недійсних повідомлень.

1.4. Порівняння публічних і приватних платформ, побудованих на основі технології ланцюгів блоків

Масштабованість

Публічний Ланцюг блоків Bitcoin може обробляти близько 7 транзакцій в секунду (TPS - transactions per second), в той час як в Ethereum це число є приблизно вдвічі більшим (в залежності від виду транзакцій), але все ж воно знаходиться в межах одного порядку [21][22]. До того ж, в Ethereum виконання транзакцій з викликом розумного-контракту вимагає на 1-5 хвилин більше часу. Такий час може викликати незручності для систем голосування. Запланований перехід на Proof of Stake в Ethereum, швидше за все підвищить показник TPS, але масштабованість мережі не збільшиться. Іншими словами, кількість транзакцій в секунду не збільшиться пропорційно зростанню числа вузлів.

У той же час приватний ланцюг блоків можна налаштувати таким чином, що показник транзакцій в секунду буде значно більшим, ніж можуть запропонувати в найближчому майбутньому публічні мережі.

Вартість транзакцій

Поточні витрати на обслуговування додатків на публічному ланцюзі блоків, в основному, складаються із загальної суми комісій за здійснення транзакцій, які необхідні для його експлуатації. У мережі Bitcoin вони зараз складають приблизно $0.4 за транзакцію. У мережі Ethereum вартість транзакції залежить від її типу. Розмір комісії за звичайну транзакцію

становить кілька центів, в той час, як комісія за виконання смарт контракту може становити більше $0.3. Для звичайного користувача такий розмір комісії не завжди може бути прийнятним. У той же час розмір комісії в приватному ланцюзі блоків є регульованим, що при необхідності дозволяє відмовитися від будь-яких комісій на виконання платежів.

Контроль

Непідконтрольність публічного ланцюгу блоків є з одного боку перевагою, а з іншого недоліком. Публічна мережа керується спільнотою її учасників - розробників, користувачів, постачальників послуг (бірж), майнерів і інших. Спільнота забезпечує цілісність мережі та зручність роботи в ній. Ефективність роботи публічної мережі досягається за допомогою оновлень протоколу, реалізація яких відбувається за допомогою так званих форків. Форк представляє собою внесення несумісних з попередньою версією програмного забезпечення змін до вузлів. Крім того, що для проведення форків потрібна згода з боку майнерів, сам факт проведення форку говорить про те, що база даних публічного ланцюгу блоків фактично перестала бути постійною. З іншого боку, багато розробників додатків хочуть мати можливість довільно і легко змінювати дані або програмний код своїх додатків. Приватний ланцюг блоків володіє достатньою гнучкістю і дозволяє безболісно змінювати програмний код. Ця властивість дає абсолютну гнучкість і гарантує незмінність головної бази даних системи.

Висновки до розділу 1

В даному розділі були розглянуті механізми електронного голосування та технологія ланцюгів блоків. Також були розглянуті найбільш поширені протоколи досягнення консенсусу. Проведений порівняльний аналіз публічного і приватного ланцюгів блоків. Визначено, що публічний ланцюг

блоків має певні переваги, але, тим не менш, має значні недоліки, серед яких незначна конфіденційність, труднощі введення оновлень, мала масштабованість, відносно висока комісія за виконання транзакцій. Таким чином використання приватного ланцюгу блоків дозволить знизити або зовсім скасувати комісію на виконання транзакцій, забезпечити максимальну швидкість підтвердження транзакцій, простоту проведення оновлень, а також незмінність головної бази даних приватного ланцюга блоків.

2 СИСТЕМА ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ ПЛАТФОРМИ WAVES

В даному розділі будуть розглянуті основні особливості платформи Waves, а також побудованої на її основі системи електронного голосування. Також в кінці розділу дана система буде перевірена на відповідність основним вимогам до систем електронного голосування.

2.1 Особливості платформи Waves

Waves - це платформа побудована на основі технології ланцюга блоків, яка має відкритий код і виконує функції передачі, випуску і обміну криптовалютами, наприклад, таких як Bitcoin і Ethereum та інших криптовалют[23]. Система є децентралізованою і прозорою.

Платформа функціонує на основі консенсусу LPoS, модифікації консенсусу Proof-of-stake. Особливістю LPoS є можливість для користувачів передавати власні баланси повним вузлам на правах оренди в обмін на частину прибутку від майнінгу.

У орендодавців залишається право повернути передані кошти в будь-який момент. Відповідно до протоколу досягнення консенсусу найвищу ймовірність знаходження блоку мають вузли з найбільшою кількість криптовалюти Waves на своєму балансі. Право на майнінг в рамках платформи є тільки у повних вузлів. Рядові користувачі можуть лише передавати їм свої баланси в оренду і в процесі майнінгу участі не приймають.

Для забезпечення незмінності ланцюга блоків, що є основою операцій з криптовалютами, використовуються криптографічні алгоритми аналогічні до тих, що використовуються в системі Bitcoin. Максимальний розмір блоку становить 100 транзакцій і кожен новий блок генерується в кожну хвилину. Таким чином система може пропускати 1,6 транзакцій в секунду.

2.2 Властивості системи голосування, побудованої на основі платформи Waves

Дана система побудована на основі платформи Waves і не потребує внесення змін до протоколу Waves [24].

Основні етапи системи голосування:

1) Реєстрація

Кожен кандидаті кожен виборець повинен мати свій особистий гаманець в системі Waves. Таким чином, виборці повинні створити власний гаманець, після чого система повинна прив'язати публічні адреси надані виборцем до його паспортних даних.

2) Створення питання, яке буде виноситись на голосування

Деталі виборчого процесу та визначення основної сфери голосування. Адреси кожного опитування та кожного конкретного голосу будуть доступні публічно.

3) Голосування

Голосування представляє собою відправку виборцем транзакції платежу на адресу кандидата. Сума платежу дорівнює 1. Платіж може здійснюватися в спеціально випущеній на основі платформи Waves криптовалюті або в самій криптовалюті Waves. Виборець не може голосувати, не витративши певну

кількість криптовалют Waves, активів або валют (що вирішується залежно від вимог щодо створення опитування).

Таким чином, загальна кількість криптовалют Waves або іншої валюти на балансах кандидатів і будуть відображати остаточні результати.

4) Перевірка

Верифікацію можна виконати, перевіривши ідентифікаційний номер виборця, пароль і список виборців.

Кожен користувач може перевірити факт надходження голосу на баланс кандидата. Крім того, всі інші транзакції можна перевірити таким чином, щоб підрахувати результати виборів.

Кожен кандидат вказує власну адресу Waves. Після чого виборці відправляють транзакцію платежу обраному кандидату. Будь-яка спроба порушення правил голосування (наприклад, надання двох голосів одному виборцю) може бути вистежена шляхом перевірки ланцюгу блоків. Результати виборів визначаються кількістю монет, що використовувались при голосуванні, на балансі кандидатів.

Система голосування на основі платформи Waves дозволить ставити питання щодо яких буде виконуватись голосування "Опитування" для будь-якого дійсного облікового з можливістю вибору одної зі 100 відповідей на одне й те саме питання. У системі можна додати умову, щоб надавати право на голосування тільки тим користувачам, що мають на балансі мінімальну кількість криптовалют Waves, валют чи активів. Кожній відповіді n відповідає певне ціле число в визначеному діапазоні. Відповідь може мати певну вагу w в залежності від різних умов голосування.

Все вищеописане визначається під час створення "Опитування". Після цього результат r визначається наступним чином $r = \sum_{i=0}^{n} w * n$

Результати голосування будуть збережені в ланцюгу блоків і не можуть бути змінені у майбутньому.

Транзакції "створення опитування" або "голосування" у разі проходження перевірки на правильність зберігаються в децентралізованій мережі.

Комісії, що стягуються з кожної транзакції "створення опитування" або "голосування" є статичними і складають 1 і 0,001 Waves відповідно.

Транзакція не є підтверджено до того моменту поки вона не буде додана до блоку, який був доданий до ланцюгу блоків і має своє продовження.

Також кожна транзакція що була відправлена в мережу після певної, визначеної правилами голосування, мітки часу буде відкинута вузлами мережі.

Транзакції "створення опитування" або "голосування" мають різні параметри:

- Адреса виборця.
- Конкретна вартість транзакції, яка є "комісією за транзакцію".
- Часова мітка, яка вказує момент коли було здійснено голосування.
- Додаткові параметри, що визначаються правилами голосування і додаються в опис транзакції.

Таким чином, якщо на балансі користувача достатньо коштів для "створення опитування" або "голосування" то процес голосування поділяється на наступні етапи:

- Під час створення нової транзакції генерується її ідентифікатор та додаються параметри, що визначені правилами голосування.
- Транзакція підписується особистим ключем виборця.
- Відправка і розповсюдження транзакції по всім вузлам мережі.

- Сервер надсилає дає відповідь з результатами голосування. Якщо якийсь з голосів був не зарахований сервер надсилає помилку з описом, що вказує на неправильність параметрів транзакції.

2.3 Перевірка на відповідність вимогам до систем електронного голосування

В системі не передбачена централізована реєстрація користувачів контролюючими органами. Тому властивість правомочності не може бути досягнута.

В публічній історії лежить виключно відкритий ключ учасника голосування і відсутні будь-які інші дані система є конфіденційною.

За рахунок публічності ланцюга блоків Waves перевірка результатів займе не більше декількох секунд адже кількість отриманих голосів визначається перевіркою балансу кандидата. Таким чином досягається точність результатів і можуть бути врегульовані будь-які конфлікти, так як всі вхідні транзакції на адреси кандидатів, а також всі вихідні транзакції можуть бути відстеженні.

За рахунок того, що дані в ланцюгу блоків не можуть бути змінені з часом система є надійною.

Тим не менш масштабованість не може бути досягнена, адже швидкість обробки та прийняття транзакцій на платформі Waves складає 1,6 транзакцій в секунду. Таким чином дана система не може бути ефективною для широкомасштабних голосувань.

Реалізація даної системи не потребує затрат на сервери, але кожен голос потребує наявність криптовалюти Waves на балансі виборця, що може призводити до певних незручностей.

Система дозволяє користувачам доводити третій стороні факт голосування за певного кандидата. Тим не менш вона є стійкою до маніпуляцій

адже для отримання доступу до облікового запису зловмиснику необхідно здійснювати підбір особистого ключа. Складність такого перебору складає 2^{128}. Така складність робить підбір малоймовірним.

Висновки до розділу 2

Система електронного голосування побудована на основі ланцюга блоків Waves не відповідає основним вимогам до систем електронного голосування. Основними недоліками є відсутність правомочності, можливість доведення третій стороні і низький показник обробки та прийняття транзакцій, а саме – 1.6 транзакцій в секунду. Таким чином дана система не може забезпечити голосування навіть на рівні міста, адже підрахунок голосів на голосуванні з 1 мільйоном виборців займе, як мінімум, 1 тиждень.

3 СИСТЕМА ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ ПЛАТФОРМИ STELLAR

В цьому розділі буде розглянута платформа Stellar, а також буде побудована система електронного голосування на основі цієї платформи. Побудована система буде перевірена на відповідність вимогам до систем електронного голосування.

3.1 Побудова системи електронного голосування

Платформа Stellar є стабільною, швидкою та відмовостійкою, але її використання для проведення електронного голосування є достатньою проблемним, адже для участі в голосуванні виборці повинні будуть купувати внутрішню криптовалюту – люмени [25]. Тому пропонується побудова платформи з модифікованою під потреби електронного голосування логікою. Зміни до протоколу Stellar не повинні привести до появи нових вразливостей.

В модифікованому під потреби голосування протоколі не повинна існувати головна внутрішня криптовалюта адже для кожного голосування повинна створюватися нова криптовалюта.

Варто зазначити, що зберігається незмінною властивість протоколу Stellar, що дозволяє емітенту криптовалюти повністю контролювати її розповсюдження та встановлювати ліміти для облікових записів. Саме ця властивість дозволить зробити процес голосування більш прозорим.

В системі голосування визначені наступні учасники:

- **Майстер** - головна відповідальна особа, може бути тільки одна в

системі. Призначенням майстра є керування процесом емісії голосів. Він представляє собою верхівку ієрархії розподілення прав учасників системи. Майстер володіє рядом повноважень, які притаманні лише йому. Відповідає за призначення та відкликання учасників: адміністраторів, емітентів.

- **Емітент -** довірена особа майстра. Може бути декілька в системі. Емітентом виступає учасник, якому в установленому порядку майстер надає право здійснювати емісію голосів. Повноваженням емітента є підписання операції емісії.
- **Адміністратор** - довірена особа майстра. Може бути декілька осіб в системі. Адміністратор керує такими учасниками, як кандидати та виборці і відповідає за їх реєстрацію. Призначенням адміністратора є розподілення прав учасників системи нижчого рівня.
- **Кандидат –** особа, яка отримує голоси в ході голосування.
- **Виборець -** особа, що є кінцевим користувачем в системі. В ході ініціювання голосування отримує не більше одного голосу на свій обліковий запис. Його взаємодія з системою відбувається за допомогою програмного застосунку за допомогою якого він може приймати участь в голосуванні.

Усіх учасників системи можна розділити на дві групи: системні - учасники, що керують обліком голосів у системі, несистемні - учасники, що користуються системою обліку голосів. До **системних** учасників відносяться: майстер, емітенти, адміністратори, кандидати. До **несистемних** учасників відносяться виборці.

Кожен учасник системи має власний обліковий запис. Обліковий запис характеризується набором значень, які містять сукупність даних про зареєстрованого користувача в системі. Обліковим записом в системі є наступний набір значень:

- головний відкритий ключ, він же є унікальним ідентифікатором облікового запису;

- тип облікового запису;
- баланс облікового запису одиницях;
- кількість підписаних та прийнятих транзакцій;
- кількість підписантів;
- відкриті ключі підписантів;
- лічильник;
- права підписантів.

Із кожним обліковим записом пов'язана щонайменше одна пара ключів: відкритий ключ і особистий ключ відповідно до алгоритму електронного цифрового підпису $ed25519$. Для побудови цифрового підпису використовується скручена крива Едвардса $ed25519$, запропоновану Бернштейном [26] Ця крива задається рівнянням

$-x^2 + y^2 = 1 + \frac{121665}{121666} * x^2 * y^2$ над полем F_q,

де $q = 2^{255} - 19$.

Параметри цієї кривої вибрані таким чином, щоб оптимізувати швидкодію операції додавання точок кривої та мінімізувати вплив вхідних даних на процес генерації ключів для цифрового підпису.

Крива Едвардса $ed25519$ застосується в ряді протоколів, бібліотек і програмних продуктів. Зокрема, $ed25519$ застосовується для автентифікації та обміну ключів в таких системах як: OpenSSH, I2p, Tor, Tox.

Закон додавання точок для цієї кривої задається наступним чином: сумою двох точок $P(x_1, y_1)$ та $Q(x_2, y_2)$ кривої E буде точка $P + Q$ з координатами:

$\frac{(x_1y_2+x_2y_1}{1+m}, \frac{x_1x_2+y_2y_1}{1-m})$ (1)

де $m = dx_1x_2y_1y_2$, $d = -\frac{121665}{121666}$, а всі операції (додавання, множення, знаходження оберненого) виконуються у полі Fq.

З рівняння кривої випливає, що для будь-якого $y \in Fq$ існує не більше двох відповідних значень $x \in Fq$ таких, що пара (x, y) є розв'язком рівняння (1). При цьому, якщо пара (x, y) є розв'язком (1), то другим розв'язком цього

рівняння буде пара $(q-x, y)$. Зауважимо, що, оскільки число q є непарним, то одне з чисел x та $(q-x)$ обов'язково буде парним, а друге – непарним. Тому у двійковому записі чисел x та $(q-x)$ останні (наймолодші) біти завжди будуть різними. Цей факт можна використати для так званого "стискання" точки кривої. Замість того, щоб задавати точку кривої двома координатами (x, y), достатньо задавати її у вигляді $(b_0(x), y)$, де $b_0(x)$ є останнім (наймолодшим) бітом числа x. За заданими y та $b_0(x)$ координату x можна однозначно відновити як розв'язок відповідного квадратного рівняння у полі Fq з заданним молодшим бітом:

$$x^2 = \sqrt{\frac{y^2 - 1}{1 + \frac{121665}{121666} * y^2}}$$

Особистим ключем користувача є b-бітовий рядок k. Відкритий ключ генерується за наступним алгоритмом [26]:

1) До особистого ключу k застосовується криптографічна геш функція H, $H(k) = (h_0, h_1, \ldots, h_{2b-1})$, де h_i –біти отриманого гешу.

2) З використанням бітів h_i обчислюється число $s = 2^n + \sum_{c \le i < n} 2^i h_i$, де $c = 3$, а $n = b - 2$.

3) За отриманою s та базовою точкою G, обчислюється відкритий ключ користувача $A = sG$.

Біти $h_b, \ldots, h_{2b-1}$ використовуються безпосередньо при накладанні цифрового підпису.

Для виконання будь-яких дій для учасників визначений певний набір операцій: емісія, голосування, створення облікового запису. Будь-яка операція підписується її ініціатором і, у разі прийняття на рівні ядра, змінює поточний стан системи. Алгоритм цифрового підпису виглядає наступним чином:

1) До особистого ключу k застосовується криптографічна геш

функція H. З отриманих $2b$ бітів беруться останні b бітів $h_b, \ldots, h_{2b-1}$.

2) Обчислюється значення $r = (h_b, \ldots, h_{2b-1}, M)$, де M повідомлення на яке накладається підпис.

3) За отриманим r та базовою точкою G, обчислюється значення $R = rG$.

4) Підпис S обчислюються наступним чином: $S = (r + H(R, A, M) * s) mod(l)$, де $l -$ непарне просте число таке, що $l * G = 0$ і $2^c * L = |E|$, а $|E|$ – потужність множини точок на еліптичній кривій.

5) Отримана пара чисел (R, S) є підписом повідомлення M.

Перевірка підпису здійснюється наступним чином:

1) До отриманих відкритих даних R, A, M застосовується криптографічна геш функція $H(R, A, M)$.

2) Перевіряється рівність $2^c SB = 2^c R + 2^c A * H(R, A, M)$ над E. Якщо рівність виконується то підпис вірний. В іншому випадку – ні.

Стійкість алгоритму цифрового підпису базується на складності задачі дискретного логарифмування у групі точок еліптичної кривої. Стійкість до злому для $ed25519$ становить приблизно 2^{128} операцій (у середньому для атаки на $ed25519$ потрібно здійснити 2^{140} бітових операцій), що відповідає стійкості таких алгоритмів, як NIST P-256 і RSA з розміром ключа у 3000 бітів або 128-бітного блокового шифру. Алгоритм цифрового підпису на кривій $ed25519$ не залежить від алгоритму гешування та не є чутливим до часових та ємнісних атак[27].

Зняття голосу з балансу користувача може виконуватись виключно операцією голосування. Ця операція знімає голос з облікового запису відправника платежу та збільшує кількість голосів на обліковому записі кандидата. Ця зміна є атомарною і її неможливо відмінити, що забезпечується особливостями ланцюгів блоків. Кожен голос збільшує лічильник облікового запису на 1. Це забезпечує неможливість проведення атаки подвійної трати [28].

Дії, які змінюють стан облікового запису користувача, такі як голосування, додавання підписантів або відкликання підписантів називаються операціями. Для того, щоб виконати операцію, користувач має створити транзакцію, яка буде включати визначену кількість операцій. За структурою транзакція складається з наступних значень:

- відкритий ключ ініціатора транзакції;
- значення комісії;
- порядковий номер транзакції від даного ініціатора;
- мітка часу;
- додаткові дані (повідомлення);
- список операцій;
- список підписів.

Перевірка і прийняття транзакцій здійснюється децентралізованою мережею вузлів. Кожен вузол перевіряє усі транзакції незалежно від інших вузлів та розповсюджує їх в мережу вузлів. Узгодження щодо прийняття певного блоку транзакцій відбувається за протоколом досягнення консенсусу [4].

Протокол консенсусу складається з двох підпротоколів: протокол висування кандидатів і протокол голосування. Протокол висування кандидатів визначає певні значення кандидатів на наступний блок. За певний час протокол висування кандидатів створює однаковий набір значень кандидатів на кожному неушкодженому вузлі. Це означає, що вузли можуть комбінувати значення кандидатів певним визначеним шляхом для отримання одного спільного значення для наступного блоку. Після того, як протокол висування кандидатів виконав свою роботу, вузли починають виконувати протокол голосування. Протокол голосування використовує механізм федеративного голосування, щоб прийняти або не прийняти наступний блок. Коли неушкоджені вузли приймають кандидата на наступний блок до ланцюгу, блок вважається закритим. Якщо ж більшість вузлів підкидує блок, що був

отриманий після виконання протоколу висування кандидатів, то вузли переходять до наступного кандидата. Якщо вузли не зійдуться на певному кандидаті то в якості кандидата на блок вибирається пустий блок, який після виконання протоколу голосування буде доданий до ланцюгу.

Під час роботи протоколу консенсусу вузли обмінюються повідомленнями для отримання спільного кандидата на наступний блок. Всі повідомлення приводяться до спільного формату та підписуються особистими ключами вузлів.

Якщо вузол отримує певне значення кандидата на наступний блок від вузлів, яким довіряє, він не буде суперечити додаванню цього значення до списку кандидатів. Множина вузлів, яким довіряє вузол називається кворумними підмножинами. Кожен вузол може мати декілька кворумних підмножин. Таким чином система складається з множини вузлів, кожен з яких вибрав одну або декілька кворумних підмножин.

Кворум - це сукупність вузлів, достатніх для узгодження між вузлами. Кворумна підмножина є підмножиною кворуму, якій довіряє певний вузол. Множина вузлів кворумної підмножини може бути меншою, ніж множина кворуму. Розглянемо систему з чотирьох вузлів на рисунку 3.1. Кворумної підмножини $\{v_1; v_2; v_3\}$ вузла v_1 достатньо, щоб переконати v_1 в твердженні. Але підмножини v_2 і v_3 включають v_4, що означає, що ні v_2, ні v_3 не можуть приймати рішення без згоди v_4. Отже, узгодження не можливе без участі v_4.

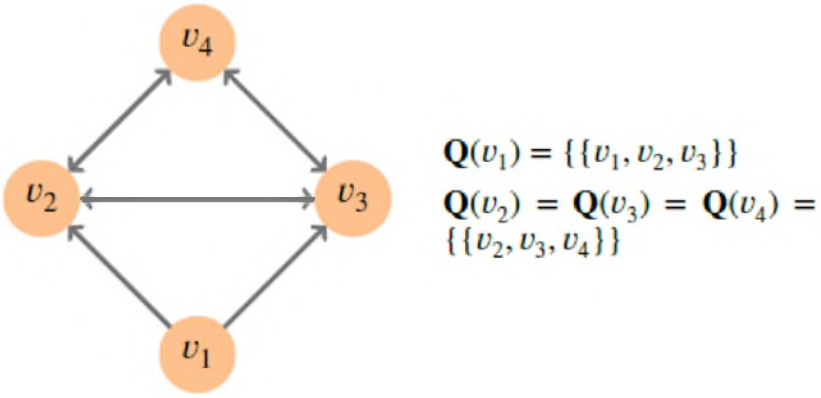

Рисунок 3.1 - Кворумна підмножина v_1 не є кворумом без вузла v_4.

Традиційна, не федеративна візантійська система узгодження вимагає від усіх вузлів прийняття одних і тих самих кворумних підмножин, тобто $\forall v_1; v_2, \quad Q(v_1) = Q(v_2)$. Оскільки кожен учасник приймає кожну підмножину, традиційні системи не розрізняють кворумні підмножини та кворуми. Недоліком є те, що членство та кворуми повинні якимось чином бути попередньо визначеними, що виключає можливість відкритого членства та децентралізованого контролю. Традиційна система, що працює за протоколом візантійської угоди, як правило, має складатись з $3f + 1$ вузлів, будь-які $2f + 1$ з яких складають кворум. Тут f - максимальна кількість ушкоджених вузлів.

Федеративна візантійська система узгодження, узагальнює протокол візантійської угоди, щоб забезпечити більший діапазон налаштувань. Ключове нововведення FBA дає змогу кожному вузлу v вибрати свою власну підмножину вузлів $Q(v)$. Таким чином, кожен вузол може вибрати свою підмножину вузлів, яким він довіряє.

Першим кроком, протокол висування досягає сходжень на певному наборі значень кандидатів на наступний блок. Способи отримання значень кандидатів можуть бути різними. Значення кандидатів може отримуватись шляхом об'єднання наборів транзакцій. Також отримувати значення кандидатів можна шляхом перетину або вибору кандидата на наступний блок з найбільшим геш-значенням.

Вузли отримують значення кандидата x на наступний блок шляхом федеративного голосування.

До тих пір поки вузол v не отримав значення кандидатів від інших вузлів, v може сприяти висуванню кандидата з будь-яким значенням x, яке проходить перевірку достовірності на рівні додатків. Вузол, як правило повинен висувати будь-які значення кандидатів, які висувають інші вузли. Кількість кандидатів на наступний блок повинна бути обмежена, щоб уникнути переповнення кандидатами.

Протокол висування має певні властивості тоді, коли система має необхідну кількість неушкоджених вузлів. Зокрема для кожного наступного блоку:

1) Неушкоджені вузли виробляють щонайменше, одне значення кандидата.

2) В певний момент, кількість можливих кандидатів перестає збільшуватись.

3) Якщо один неушкоджений вузол розглянув x на місце кандидата, тоді кожен неушкоджений вузол розгляне x на місце кандидата.

Чим менше комбінацій значень, тим ефективнішим буде процес висування кандидатів. Для вузлів визначається тимчасовий пріоритет і кожен вузол, якщо це можливо, висуває ті ж самі значення кандидатів, що і вузол з більш високим пріоритетом. Нехай H геш-функція зі значенням виходу, що лежить в межах $\{0,\dots,h_{max}-1\}$. У разі використання геш-функції SHA-256 $h_{max}=2^{256}$. $G_i(m)=H(i,x_{i-1},m)$ – значення конкретної геш-функції отримане для блоку i, де x_{i-1} значення гешу попереднього блоку. Беручи до уваги наступний блок i і раундовий номер n, кожен вузол v обчислює набір сусідів і пріоритет для кожного сусіда наступним чином:

- $weight(v,v')=|\{q|q\in Q(v)\wedge v'\in q||Q(v)|$
- $neighbours(v,n)=\{v'|Gi(N,n,v')<h_{max}*weight(v,v')\}$
- $priority(n,v')=Gi(P,n,v')$

N і P – константи для отримання двох різних значень гешу. Функція $weight(v,v')$ повертає ті значення з $Q(v)$, які містить v'. Використання ваги, як ймовірності над n, що v' з'явиться в $neighbours(v,n)$, зменшується ймовірність того, що вузли, які можуть виявитись зловмисними не будуть переважати в раунді.

Кожен вузол v, повинен спочатку знайти вузол $v_0\in neighbours(v,0)$, який дозволяє отримати максимальне значення пріоритету $priority(0,v_0)$ між вузлами, з якими він може контактувати і потім голосувати за висування тих

самих значень, що і v_0. Вузол v використовує мітки часу для того, щоб вирішити чи голосувати за нові нових кандидатів на наступний блок. Після проходження n −тої мітки часу, вузол v знаходить вузол $v_n \in neighbours(v, n)$, який дозволяє отримати максимальне значення $priority(0, v_n)$. Після того, як вузол v знайшов v_n він голосує за висування тих самих кандидатів на наступний блок, що і v_n.

Після того, як вузли отримали певні значення кандидатів, вони беруть участь в протоколі голосування. Бюлетень b є парою, що має вигляд $b = \langle n, x \rangle$, де x- значення кандидата, а $n \geq 1$-це лічильник. Лічильник і значення кандидата записуються, як $b.n$ і $b.x$, таким чином $b = \langle b.n, b.x \rangle$. Бюлетені повністю впорядковуються за значеннями $b.n$.

Бюлетень b приймається або відкидується шляхом виконання федеративного голосування. Прийняття або відмова у прийнятті бюлетеню є взаємно несумісними тому неушкоджені вузли можуть голосувати тільки за один з цих варіантів. Оскільки тільки одне значення кандидата може бути вибране в якості наступного блоку, всі прийняті бюлетені повинні бути сумісними.

Два бюлетеня b_1 і b_1 називаються сумісними, якщо $b_1.x = b_2.x$.

В разі успішного завершення фази протоколу голосування до ланцюгу додається новий блок транзакцій. Якщо ж вузли не змогли досягти консенсусу то до ланцюгу додається пустий блок.

Історія усіх прийнятих транзакцій та стан усієї системи зберігається кожним вузлом незалежно. Будь-які прийняті системою транзакції є незворотними. Неможливо змінити або скасувати прийняті системою транзакції, навіть якщо буде доведено, що власник про них не знав або не хотів їх проводити.

Взаємодія користувача з системою здійснюється за допомогою програмного застосунку. У своєму застосунку користувач може отримати поточний стан свого облікового запису. Також за допомогою застосунку користувач може створювати, підписувати і надсилати в мережу транзакції

при виконанні голосування. Користувачі можуть зберігати особисті ключі на пристроях або у вигляді мнемонічної фрази.

Генерація мнемонічної фрази відбувається за алгоритмом описаним в bip-0039 [29]. Алгоритм отримання мнемонічної фрази складається з наступних кроків:

1) Обчислюється контрольна сума шляхом ділення довжини особистого ключа на число 32, $CS = length(PK)/32$.

2) Контрольна сума додається до довжини ключа і ділиться на 11 $MS = (CS + length(PK))/11$.

3) Кожному значенню в діапазоні $0, \dots, 2^{11} - 1$ відповідає певне слово. Таким чином мнемонічна фраза складається з MS взаємно не пов'язаних слів, де кожне слово вибирається з 2048 варіантів визначеним шляхом.

Для кожного нового голосування створюється і випускається в обіг унікальна монета, будь-які операції з цією монетою будуть заборонені після закінчення процесу голосування.

Випуск голосів в обіг здійснюється за допомогою процесу емісії. Емісія в системі є централізованою і повністю контролюється емітентами. Процес емісії голосів представляє собою платіж з облікового запису майстра на облікові записи виборців. Розмір емісії дорівнює кількості виборців, що повинні приймати участь у голосуванні. Обліковий запис майстра має від'ємний баланс, адже він відповідає за емісію в системі.

Кожен з учасників системи повинен мати власний програмний застосунок, який дасть йому можливість виконувати своє призначення. Серед необхідних програмних застосунків:

- Застосунок користувача - програмне забезпечення та інтерфейс користувача для роботи з його обліковим записом. За допомого якого користувач може створювати та відправляти в мережу транзакції з операцією голосування, переглядати історію голосування
- Панель управління адміністратора - інтерфейс адміністратора для управління системою. За допомогою панелі управління адміністратор може

реєструвати кандидатів та виборців, переглядати статистику голосувань, блокувати операції з монетами після закінчення етапу голосування.

- Панель управління емітента - інтерфейс емітента для створення і відправки в мережу транзакцій емісії голосів.
- Панель управління кандидата - інтерфейс кандидата для перегляду історії отриманих голосів.

Життєвий цикл голосування

1) Емісія голосів. Емітент створює нові монети та поповнює баланси виборців у розмірі 1 монета на кожного.

2) Процес голосування. Виборці надсилають свої монети на облікові записи кандидатів, за яких вони бажають проголосувати.

3) Процес підрахунку голосів. Після закінчення процесу голосування фіксуються баланси виборців, а адміністратор встановлює обмеження на виконання будь-які операції з монетою, яка використовувалась при голосування. Кандидат на балансі, якого буде найбільша кількість голосів буде переможцем.

В системі визначені наступні операції:

- Створення облікового запису. Здійснюється адміністратором.
- Голосування. Може здійснюватися виключно виборцями.
- Додавання або відкликання підписанта. Здійснюється майстром.
- Емісія голосів. Здійснюється емітентом.

Будь-які операції в системі відбуваються тільки після прийняття транзакцій, що містять ці операції. Транзакція має наступну структуру:

- Відкритий ключ користувача
- Порядковий номер транзакції
- Часова мітка
- Список операцій

- Множина $Ed25519$ підписів

У разі втрати особистого ключа адміністратором або емітентом, майстер здійснює транзакцію відкликання емітента або адміністратора в якості підписанта. Таким чином будь-які транзакції, що містять операції підписані відкликаним ключем будуть відхилені.

Для зручності користувачів особисті ключі можуть зберігатися на віддаленому сховищі ключів. Ця опція дає можливість отримувати доступ до облікового запису з різних пристроїв, уникаючи ручного перенесення особистих ключів з пристрою на пристрій. Метою такої процедури є забезпечення надійного та безпечного зберігання особистих даних. В порівнянні з іншими способами збереження особистих даних такий сервіс має ряд переваг пов'язаних зі зручністю.

Збереження даних на цьому сервісі передбачає, що учасник зберігатиме та оброблятиме особисті дані для ініціювання запитів на захищеному пристрої.

Відправка даних на віддалене сховище ключів відбувається у наступному порядку:

1. В застосунку користувача генерується випадкове значення - сіль $salt$.

2. До відкритих даних користувача $data$, паролю $pass$ та солі приміняться адаптивна криптографічна геш-функція $scrypt$ для отримання унікального ідентифікатора користувача $userID = scrypt(data, pass, salt, kdf)$, де kdf – параметри, що задають складність обчислення гешу [30].

3. В користувацькому застосунку відбувається шифрування особистого ключа $priv$ алгоритмом AES-256 [31]. В якості ключа шифрування виступає геш від паролю користувача $h = H(pass)$. В якості геш функції H виступає SHA-256. В результаті шифрування буде отриманий зашифрований особистий ключ $EPK = E_{AES256}(priv, h)$.

4. Користувацький застосунок формує запит, який містить дані користувача $data$, сіль $salt$, унікальний ідентифікатор користувача $userID$, зашифрований особистий ключ EPK та ініціює з'єднання зі сховищем ключів.

5. Сховище ключів перевіряє чи не співпадають дані користувача з іншими. Якщо такого користувача не існує - створює запис у своїй базі даних.

Отримання даних з віддаленого сховища ключів відбувається в наступному порядку:

1. Користувацький застосунок відправляє запит, що містить відкриті дані користувача $data$ на віддалене сховище ключів.

2. У відповідь сервер надсилає сіль $salt$, що відповідає даним користувача.

3. До даних користувача $data$, паролю $pass$ та солі приміняться адаптивна криптографічна геш-функція $scrypt$ для отримання унікального ідентифікатора користувача $userID = scrypt(data, pass, salt, kdf)$, де kdf – параметри, що задають складність обчислення гешу.

4. Застосунок користувача формує запит, який містить відкриті дані $data$, сіль $salt$ та унікальний ідентифікатор користувача $userID$ та ініціює з'єднання, надіславши запит в сховище ключів.

5. У разі співпадіння даних надісланих користувацьким застосунком з записом в базі даних, сховище ключів у відповідь надсилає зашифрований особистий ключ EPK.

6. Використовуючи геш від паролю, користувацький застосунок дешифрує особистий ключ користувача $priv = D_{AES256}(EPK, SHA256(pass))$

Отримана система має значні переваги над іншими системами, що також побудовані на основі технології ланцюгів блоків по показникам правомочності, масштабованості та практичності. Тим не менше ця система не може забезпечити чесність виборів, а також дає користувача можливість підтвердити факт голосування за певного кандидата. Це робить можливим підкуп виборців

Для забезпечення властивостей анонімності, не можливості доведення третій стороні, чесності та вирішення проблеми доведення третій стороні може бути використана наступна схема генерації одноразових адрес, що використовується в протоколі CryptoNote [32]:

Вхідні дані: G – базова точка на еліптичній кривій $Ed25519$, a - особистий ключ кандидата, тоді $A = aG$ - відкритий ключ кандидата.

З перспективи програмного застосунку виборця буде виглядати наступним чином:

1) На стороні програмного застосунку виборця генерується випадкове число r, яке повинно бути видалено відразу після проведення транзакції.

2) В програмному застосунку виборця обчислюється $D = rA$.

3) Число D використовується для отримання значення $f = H(D)$, де H криптографічна геш функція.

4) На виході отримуємо $F = fG$, яке і є одноразовою адресую кандидата.

5) Виборець формує та відправляє в мережу транзакцію з операцією голосування на отриману одноразову адресу і прикладає число R, як додаткові дані.

Перспектива кандидата:

1) Програмний застосунок кандидата перевіряє всі транзакції пройшли перевірку на правильність.

2) Програмний застосунок кандидата витягує R, з кожної прийнятої системою транзакції.

3) В програмному застосунку кандидата обчислюється $D' = aR$.

4) З використанням отриманого значення D' обчислюється $f' = H(D')$, де H криптографічна геш функція.

5) За допомогою отриманого f' обчислюється значення $F' = f'G$. Після чого програмний застосунок кандидата звіряє F з отриманим F'. Якщо $F = F'$ то голос належить кандидату, а якщо ні – то голос належить комусь із опонентів.

Обчислення D і D' вимагає наявності секретних даних: або r (Виборця) або a (Кандидата). Так як r - вибирається випадковим чином і видаляється зразу після отримання одноразової адреси, зовнішній спостерігач не зможе обчислити кому призначався голос. Після завершення голосування кандидати мають опублікувати свої особисті ключі для того, щоб кожен міг перевірити результати голосування.

Запропонована схема забезпечить більш високий рівень конфіденційності та забезпечить чесність виборів. Але водночас реалізація цієї схеми знизить практичність системи адже збільшиться рівень складності реалізації.

3.2 Перевірка на відповідність вимогам до систем електронного голосування

Одною з найбільш важливих вимог до систем електронного голосування є чесність. Адже наявність цієї властивості зменшує ймовірність маніпуляцій.

Твердження 3.1 Побудована система відповідає вимогам чесності.

Доведення. Під час голосування використовуються одноразові адреси. Генерація одноразової адреси відбувається наступним чином $F = H(rA) * G$, де A −відкритий ключ кандидата, r − випадкове число, H − криптографічна геш функція, а G − базова точка. Чесність досягається за рахунок складності обчислення особистого ключа за відомим значенням відкритого ключа. Щоб визначити кому призначався голос необхідно знати r або a, де a − особистий

ключ кандидата. Складність такого перебору при довжині ключа 256 бітів складає 2^{128}. *Твердження доведено.*

Чесність дозволить зробити результати виборів максимально об'єктивними. Окрім цього необхідно зберігати конфіденційність виборців та точно порахувати результати проведеного голосування.

Твердження 3.2 Побудована система відповідає вимогам конфіденційності, довгострокової конфіденційності, перевірки, точності і врегулювання.

Доведення. Конфіденційність та довгострокова конфіденційність досягається за рахунок того, шо в ланцюгу блоків лежить виключно відкритий ключ виборця $V = v * G$, де $v -$ особистий ключ виборця, а $G -$ базова точка еліптичної кривої.

Властивості точності, врегулювання і перевірки голосів досягаються за рахунок публічності ланцюга блоків. Після опублікування кандидатами особистих ключів a, що дозволить перебрати всі проведені транзакції діставши з опису транзакцій числа R та обчислити $F = H(aR) * G$ і визначити, які з одноразових адрес належать певному кандидату. Сумарний баланс одноразових адрес дозволить порахувати кількість отриманих кандидатом голосів. Фальсифікація голосів не є можливою адже для цього необхідно мати доступ до особистих ключів виборців v або кандидатів a. Складність підбору ключів довжини 256 бітів складає 2^{128}. Ймовірність такого підбору дуже низька. *Твердження доведено.*

Забезпечення конфіденційності та точності збільшить довіру до системи, а врегулювання спорів щодо результатів виборів займатиме мінімум часу, адже результати голосування можна буде переглядати зразу після опублікування особистих ключів кандидатів. При цьому будь-які маніпуляції з голосами після закінчення виборів буде легко відслідкувати за рахунок того, що історія голосувань є публічною. Окрім цього адміністратори системи будуть блокувати будь-які операції в криптовалюті, що була випущена для проведення голосування після його закінчення.

Твердження 3.3 Побудована система є надійною.

Доведення. Для підробки результатів голосування зловмиснику необхідно ефективно підробляти цифровий підпис або видаляти транзакції за ланцюга блоків. В першому випадку необхідно підбирати особисті ключі виборців, що малоймовірно з урахуванням факту складності підбору - 2^{128}. У другому випадку необхідно замінити існуючий блок, блоком з таким самим значенням гешу, що не містить небажаних для зловмисника транзакцій. Складність цієї задачі також складає 2^{128}. Тому ймовірність такої подій дуже низька. *Твердження доведено.*

Забезпечення надійності системи є важливим фактором. Якщо система не надійна, то зловмисник зможе підробити результати виборів.

Реєстрація виборців в системі є централізованою і здійснюється контролюючим органом, який назначає адміністраторів системи. Тому дана система має властивість правомочності.

Система електронного голосування має обробляти велику кількість транзакцій за невеликі інтервали часу. Масштабованість досягається за рахунок того що система була побудована на основі протоколу Stellar без внесення змін до протоколу консенсусу. Тому швидкість обробки та прийняття транзакцій буде аналогічною, а саме до 1000 транзакцій в секунду. За рахунок цього система буде ефективною. При цьому вага ланцюга буде незначною адже вага однієї транзакції складає не більше 1 кілобайта.

Практичність досягається за рахунок того, що для розгортання і запуску системи необхідно лише 5 недорогих серверів. При цьому система буде обробляти до 3600000 голосів за годину, що задовольняє умовам голосування на національному рівні.

Стійкість системи від маніпуляцій забезпечується за рахунок того, що зловмисник не може отримати доступ до особистого ключа користувача не знаючи паролю від його облікового запису в програмному застосунку.

Як наслідок використання одноразових адрес і видалення випадкового значення r одразу після генерації одноразової адреси кандидата, користувач не зможе довести третій стороні факт голосування за певного кандидата.

Висновки до розділу 3

В даному розділі була проаналізована платформа Stellar та виділені її основні особливості. На основі цих особливостей було зазначено, що дана платформа є оптимальною для побудови систем голосування на її основі. На основі платформи Stellar була побудована система електронного голосування, а також були запропоновані методи покращення системи для забезпечення більшої ефективності. Також було доведено, що дана система відповідає вимогам до систем електронного голосування: чесність, конфіденційність, довгострокова конфіденційність, правомочність, точність, врегулювання, масштабованість, надійність, перевірка результатів, не підтвердження голосу, стійкість до маніпуляцій.

ВИСНОВКИ

В ході роботи було проведено огляд класичних систем електронного голосування, а також визначені основні властивості, яким повинна відповідати система електронного голосування. В результаті виявлено, що частина з визначених властивостей присутня в протоколах побудованих на основі технології ланцюгів блоків. Також розглянуті основні алгоритми досягнення консенсусу: Proof of Work, Proof of Stake, BFT.

Проведено порівняльний аналіз публічного і приватного ланцюгів блоків в контексті їх використання для проведення електронного голосування. В результаті визначено, що публічні ланцюги блоків не можуть забезпечити необхідну швидкість обробки транзакцій та практичність системи голосування.

Проаналізовано систему електронного голосування, побудовану на основі платформи Waves та її властивості. В результаті виявлені основні недоліки цієї системи, а саме: низький показник обробки транзакцій в секунду, висока вартість транзакцій і відсутність конфіденційності для виборців.

Проведено аналіз протоколу Stellar та виділені його основні особливості. Як результат була запропонована система електронного голосування, побудована на основі протоколу Stellar. Досліджено властивості побудованої системи голосування, в результаті якого було визначено, що вона відповідає вимогам до сучасних систем електронного голосування, а саме: чесність, конфіденційність, довгострокова конфіденційність, правомочність, точність, врегулювання, масштабованість, надійність, перевірка результатів, не підтвердження голосу, стійкість до маніпуляцій, практичність. Окрім цього система є дешевою в використанні і при цьому є достатньо швидкою так як може обробляти до 1000 транзакцій в секунду. Таким чином побудована

система може використовуватись для проведення голосувань на національному рівні.

ПЕРЕЛІК ПОСИЛАНЬ

1. Menezes A. Handbook of Applied Cryptography / A. Menezes, P. Oorschot, S. Vanstone. // CRC Press. – 1996.

2. Gervais A. Do you need a Blockchain? / A. Gervais, G. Karame. // International Association for Cryptologic Research. – 2017. – С. 1–2.

3. Закон України про електронний цифровий підпис [Електронний ресурс]. – 2003. – Режим доступу до ресурсу: http://zakon4.rada.gov.ua/laws/show/852-15.

4. MAZIERES D. The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus / D. MAZIERES, 2015. – С. 4–24.

5. Patel P. A Cryptography Application using Salt Hash Technique / P. Patel, J. Patel, P. Virparia. // International Journal of Application or Innovation in Engineering & Management. – 2013. – С. 1–4.

6. Lambrinoudakis C. The current landscape / C. Lambrinoudakis, D. Gritzalis // Secure Electronic Voting / C. Lambrinoudakis, D. Gritzalis., 2003. – С. 101–122.

7. Shamir A. How to share a secret / A. Shamir. // Communications of the ACM. – 1979. – С. 612–613.

8. Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms / T. Elgamal. // IEEE Transactions on Information Theory. – 1985.

9. Cramer R. A secure and optimally e_cient multiauthority election scheme / R. Cramer, R. Gennaro, B. Schoenmakers. // European transactions on Telecommunications. – 1997. – С. 481–490.

10. Peng K. Multiplicative homomorphic e-voting / K. Peng, R. Aditya, C. Boyd. // Progress in Cryptology – INDOCRYPT 2004. – 2005. – С. 61–72.

11. Fontaine C. A survey of homomorphic encryption for nonspecialists / C. Fontaine, F. Galand. // EURASIP Journal on Information Security. – 2007.

12. Jonker H. Bulletin Boards in Voting Systems: Modelling and Measuring Privacy / H. Jonker, G. Pang. // Availability, Reliability and Security (ARES). – 2011. – С. 1–8.

13. Adida B. Helios: Web-based open-audit voting / B. Adida. // USENIX Security Symposium. – 2008. – №17. – С. 335–348.

14. Demirel D. Improving helios with everlasting privacy towards the public / D. Demirel, J. Van De Graaf, R. Araujo. // Electronic Voting Technology Workshop/Workshop on Trustworthy Elections. – 2012.

15. Andreas M. Aggregating Transactions into Blocks / M. Andreas // Mastering Bitcoin / M. Andreas., 2010. – (O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472). – С. 184–189.

16. Erman D. BitTorrent Traffic Measurements and Models : дис. канд. техн. наук / Erman D., 2005. – 166 с.

17. Dobraunig C. Analysis of SHA-512/224 and SHA-512/256 / C. Dobraunig, M. Eichlseder, F. Mendel. // ASIACRYPT 2015. – 2015. – С. 1–5.

18. Gervais A. On the Security and Performance of Proof of Work Blockchains / A. Gervais, G. Karame, K. Wüst. // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. – 2016. – С. 2–6.

19. King S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake / S. King, S. Nadal., 2012.

20. Castro M. Practical Byzantine Fault Tolerance / M. Castro, B. Liskov. // Massachusetts Institute of Technology. – 2001. – С. 1–5.

21. Nakamoto S. A Peer-to-Peer Electronic Cash System / S. Nakamoto, 2009. – С. 1–5.

22. Wood G. Ethereum: A secure decentralised generalised transaction ledger eip-150 revision / G. Wood, 2015. – С. 1–3.

23. Ivanov S. Waves Whitepaper / S. Ivanov, 2015. – С. 1–13.

24. Faour N. Transparent Voting Platform Based on Permissioned Blockchain / N. Faour. // ArXiv. – (2018). – С. 26–31.

25. Stellar [Електронний ресурс]. – 13.05.2018. – Режим доступу до ресурсу: https://www.stellar.org/developers/guides/get-started/.

26. Bernstein D. High-speed high-security signatures / D. Bernstein, N. Duif, P. Schwabe. // Springer-Verlag. – 2011.

27. Bernstein D. EdDSA for more curves [Електронний ресурс] / D. Bernstein. – 2015. – Режим доступу до ресурсу: https://eprint.iacr.org/2015/677.pdf.

28. Podolanko J. Countering Double-Spend Attacks on Bitcoin Fast-Pay Transactions / J. Podolanko, J. Ming, M. Wright. // In Workshop on Technology and Consumer Protection. – 2017. – С. 1–3.

29. Mnemonic code for generating deterministic keys [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki.

30. Percival C. The scrypt Password-Based Key Derivation Function [Електронний ресурс] / C. Percival. – 2016. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc7914.

31. Daemen J. AES Proposal: Rijndael / J. Daemen, V. Rijmen., 199. – 47 с.

32. Saberhagen N. CryptoNote v 2.0 [Електронний ресурс] / N. Saberhagen. – 2013. – Режим доступу до ресурсу: https://cryptonote.org/whitepaper.pdf.