Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *«Дослідження механізмів забезпечення безпеки децентралізованих системах»*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Столярик Ю. П.
підпис (прізвище та ініціали)

Керівник *Карпінський М.П.*
підпис (прізвище та ініціали)

Нормоконтроль *Лобур Т.Б.*
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«19» __червня__ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Столярику Юрію Павловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження механізмів забезпечення безпеки в децентралізованих системах

Керівник роботи Карпінський Микола Петрович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «03» 04 2023 року № 4/7-349

2. Термін подання студентом завершеної роботи 17.06.2023

3. Вихідні дані до роботи Вимоги до програмного забезпечення

4. Зміст роботи (перелік питань, які потрібно розробити)
Основні принципи побудови та аналіз безпеки децентралізованих систем
Механізми забезпечення безпеки в децентралізованих системах
Використання децентралізованих систем
Дослідження механізмів забезпечення конфіденційності, цілісності та автентичності децентралізованих систем
Реалізація створення та використання криптовалют на основі децентралізованих систем
Оцінка поточного стану безпеки децентралізованих систем
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Пилипець.М.І., проф. кафедри МТ		

7. Дата видачі завдання 16.01.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	16.01 – 19.01	*Виконано*
2.	Підбір джерел про принципи побудови та методи забезпечення безпеки в децентралізованих системах	20.01 – 05.02	*Виконано*
3.	Опрацювання джерел в галузі дослідження	06.02 – 22.02	*Виконано*
4.	Розроблення програмного коду	23.02 – 20.03	*Виконано*
5.	Тестування роботи програми та верифікація результатів	21.03-05.04	*Виконано*
6.	Оформлення розділу «Аналіз безпеки децентралізованих систем»	06.03 – 17.04	*Виконано*
7.	Оформлення розділу «Аналіз механізмів забезпечення конфіденційності та автентичності в децентралізованих системах»	18.04 – 29.04	*Виконано*
8.	Оформлення розділу «Реалізація створення та використання криптовалют на основі децентралізованих систем»	30.04 – 13.05	*Виконано*
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	14.05 – 21.05	*Виконано*
10.	Оформлення кваліфікаційної роботи	22.05 – 05.06	*Виконано*
11.	Нормоконтроль	06.06 – 11.06	*Виконано*
12.	Перевірка на плагіат	12.06 – 15.06	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.06 – 19.06	*Виконано*
14.	Захист кваліфікаційної роботи	23.06.2023	

Студент _____ *Столярик Ю.П.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Карпінський М.П.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження механізмів забезпечення безпеки в децентралізованих системах // Кваліфікаційна робота ОР «Бакалавр» // Столярик Юрій Павлович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-41 // Тернопіль, 2023 // С. __ , рис. – 50, табл. – 1, кресл. – , додат. –

Ключові слова: АВТЕНТИЧНІСТЬ, КОНФІДЕНЦІЙНІСТЬ, БЕЗПЕКА, БІТКОЇН, БЛОКЧЕЙН, ДЕЦЕНТРАЛІЗОВАНІ СИСТЕМИ, ІНТЕРНЕТ-БАНКІНГ, КРИПТОБІРЖА, КРИПТОВАЛЮТА, СМАРТ-КОНТРАКТ.

Кваліфікаційна робота присвячена аналіз систем методики оцінки поточного стану безпеки децентралізованих систем на основі аналізу сучасних загроз та механізмів протидії та програмна реалізація створення криптовалют на основі децентралізованих систем.

Об'єкт дослідження – процес забезпечення безпеки в децентралізованих системах.

Предмет дослідження – створення криптовалют із використанням технології Blockchain, та аналіз механізмів забезпечення безпеки в децентралізованих системах.

У роботі розглядаються питання аналізу безпеки децентралізованих систем, механізми забезпечення конфіденційності та автентичності. Розглянуто побудови криптобірж, формування технології Blockchain та смарт-контрактів, механізми їх функціонування, а також тенденції розвитку та можливі ризики. Зокрема, здійснено реалізацію програмного продукту для створення криптовалют.

В якості інформаційної бази дослідження були використані публікації, наукові видання, навчальні посібники.

Для реалізації даної роботи були використані програмні продукти: Visual Studio 2019, Postman, Visio 2016.

ABSTRACT

Study of mechanisms for ensuring security in decentralized systems // Thesis of educational level "Bachelor" // Stoliaryk Yurii Pavlovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБс-41 group // Ternopil, 2023 // P. ___ , fig. -____, table. - __ , chair. - ___ , added. -___.

Keywords: AUTHENTICITY, CONFIDENTIALITY, SECURITY, BITCOIN, BLOCKCHAIN, DECENTRALIZED SYSTEMS, INTERNET BANKING, CRYPT EXCHANGE, CRYPT CURRENCY, SMART CONTRACT.

Qualification thesis is devoted to analysis of the systems of methods for assessing the current security of decentralized systems based on the analysis of modern threats and countermeasures and software implementation of cryptocurrencies based on decentralized systems.

The object of study – the process of security in decentralized systems.

The subject of research is the creation of cryptocurrencies using Blockchain technology, and the analysis of security mechanisms in decentralized systems.

The thesis considers the issues of security analysis of decentralized systems, mechanisms for ensuring confidentiality and authenticity. The construction of cryptocurrencies, the formation of Blockchain technology and smart contracts, the mechanisms of their operation, as well as development trends and possible risks are considered. In particular, a software product for creating cryptocurrencies was implemented.

Publications, scientific publications, textbooks were used as the information base of the research.

For the implementation of this master’s thesis were used software products: Visual Studio 2019, Postman, Visio 2016. The work contains: 86 pages, 50 figures, 1 tables, 21 literary sources.

ЗМІСТ

ВСТУП 8
1 АНАЛІЗ БЕЗПЕКИ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ 10
1.1. Основні принципи побудови децентралізованих систем 10
1.2 Загрози на централізовані та децентралізовані системи 15
1.3 Механізми забезпечення безпеки в децентралізованих системах 23
1.3.1 Алгоритм SHA-2 (256 bit) 25
1.3.2 Алгоритм RIPEMD (160 bit) 28
1.3.3 Алгоритм SHA-3 (Keccak-512 bit) 29
1.3.4 Цифровий підпис 32
1.4 Використання децентралізованих систем 33
1.4.1 Принципи побудови криптобірж 33
1.4.2 Інтернет-банкінг на основі децентралізованих систем 37
1.4.3 Принципи формування смарт контрактів 39
1.5 Висновки до розділу 42
2 АНАЛІЗ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ та АВТЕНТИЧНОСТІ В ДЕЦЕНТРаЛІЗОВАНИХ СИСТЕМАХ 43
2.1 Дослідження механізмів забезпечення конфіденційності 43
2.2 Дослідження механізмів забезпечення цілісності 60
2.3 Дослідження механізмів забезпечення автентичності 61
2.3.1 Безпека відкритого ключа 62
2.3.2 Безпека закритого ключа 63
2.4 Висновки до розділу 2 63
3 РЕАЛІЗАЦІЯ СТВОРЕННЯ ТА ВИКОРИСТАННЯ КРИПТОВАЛЮТ НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ 65
3.1 Вибір програмних засобів для реалізації майнінгу 65
3.2 Програмна реалізація імітація майнінгу криптовалют 66
3.3 Оцінка поточного стану безпеки децентралізованих систем 77
3.4 Висновки до розділу 3 78

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ 79
4.1 Управління та нагляд за безпекою життєдіяльності в Україні..... 79
4.2 Соціальне значення охорони праці 82
ВИСНОВКИ..... 84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 86

ВСТУП

У сучасному світі ми стоїмо на порозі цифрової революції, яка торкнеться всіх сфер життя. Час від часу відбуваються технологічні прориви, які відкривають новий світ можливостей. Наприклад, винахід Інтернету був таким проривом, який змінив світ майже з усіх боків. Зі швидким розвитком інформаційних технологій громадськість стає більш чутливою до безпеки своїх особистих даних. Люди хочуть, щоб їх дані оброблялися швидко, але безпечно.

Децентралізовані системи та технологія Blockchain є одним із нових технологічних проривів, який, як очікується, революціонізує спосіб виконання транзакцій, вплинувши тим самим на широкий спектр потенційних сфер застосування. Децентралізація в інформаційних системах стала не просто черговим витком технологічної еволюції, вона пропонує кардинально новий підхід, який здатний змінити принципи взаємодії людей. В теперішній час усі користувачі хочуть не тільки довіряти, але і мати можливість провірити. Раніше фінансові системи вважалися закритими і були захищені стандартними методами. Але з появою Bitcoin виявилось, що будь-яка фінансова система може бути повністю прозорою для усіх та існувати без жодного центру прийняття рішень, при цьому зберігати приватність платежів користувачів. У Bitcoin набір транзакцій зберігається у незмінній структурі, званої блоками, а ще через одну рангову мережу блоки дублюються і зберігаються всім учасникам. Протоколи консенсусу гарантують, що більшість учасників мають послідовне уявлення про порядок блоків, які генеруються учасниками, відомими як майнери. Оскільки транзакції в блоці перевіряються всіма учасниками, не існує вимоги щодо залучення довіреної третьої сторони до системи. Функція бездовірної безпеки викликала інтерес дослідників, і схема дизайну Bitcoin була узагальнена і реалізована в системі, відомій як Blockchain. Таким чином, технологію Blockchain можна використовувати для усунення потреби в централізованих установах і базах даних, де кожен, хто бере участь у транзакції, може безпосередньо переглядати та підтверджувати транзакцію, що призводить до створення справжньої прозорості та

бездовірних систем. Концепція наявності комп’ютерного протоколу, призначеного для цифрового полегшення, перевірки або забезпечення виконання переговорів або виконання контракту без потреби третіх сторін, була досягнута за допомогою Blockchain.

Хоч технологія Blockchain приносить велику зручність галузі, але громадськість все ще стурбована безпекою та ефективністю систем Blockchain.

Для того, щоб виконати поставлені цілі, необхідно спочатку вирішити ряд завдань:

- виконати аналіз безпеки децентралізованих систем;
- ознайомитись з основними принципами побудови та механізмами безпеки децентралізованих систем;
- ознайомитись з технологією Blockchain та процесу створення криптовалют;
- виконати аналіз механізмів забезпечення конфіденційності та автентичності в децентралізованих системах;
- виконати програмну реалізацію створення криптовалюти з використанням технології Blockchain.

Тому питання пов’язані з децентралізованими системами, технологією Blockchain та програмна реалізація пов’язані з розробкою нових криптовалют із використанням технології Blockchain є актуальними в цей час.

1 АНАЛІЗ БЕЗПЕКИ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ

1.1 Основні принципи побудови децентралізованих систем

Класична система оплати готівкою все частіше витісняється абразивною електронікою, яка включає кредитні та дебетові картки, інтернет-банкінг, електронну торгівлю та багато іншого. Усі ці об’єкти централізовані. Тобто одна центральна влада; установи, що належать таким установам, як банки, уряди, компанії, що випускають кредитні картки та інші. При обробці електронних лабораторій люди повинні покладатися на свої електронні продукти як довіреній третій стороні. Всі ці платіжні системи працюють добре і пропонують, наприклад, автентифікацію чи цифровий підпис. Однак слабкість усієї моделі – проблема, пов’язана з централізованими платіжними системами. Фінансові установи можуть зазнавати атак хакерів, грошові перекази між країнами здійснюються повільно, вимагають високих комісій і не можуть повністю уникнути незворотних транзакцій. В централізованій системі є один великий центр, до якого всі звертаються. Схема централізованої мережі наведена на рис. 1.1.

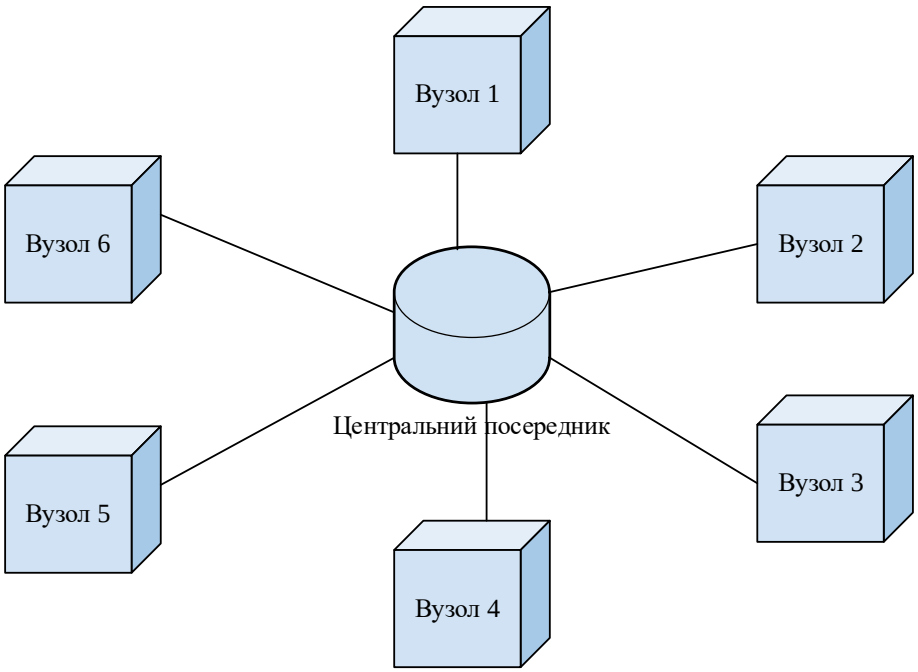

Рисунок 1.1 – Схема централізованої системи

Централізована система має низку корисних властивостей. Така система легше керується і зазвичай відомо, хто відповідає за її роботу. Рішення приймаються швидко. Можна побудувати цілком конкретну бізнес-модель і дотримуватися її, що є перевагою, коли йдеться про монетизацію.

Для децентралізованої системи є багато таких центральних вузлів, вони всі рівноправні, та всі користувачі звертаються до них. І якщо якийсь із вузлів відмовить, то користувачі можуть звернутися до іншого вузла. Така система більш стійка.

Основна мета децентралізації – забезпечити для системи властивості, що дозволяють ефективно та надійно взаємодіяти її користувачам один з одним у ситуації, коли вони не довіряють якомусь центральному органу чи посереднику.

Децентралізація є протилежним до централізації процесом і передбачає розподіл функцій системи між її учасниками, причому без єдиного органу, що керує.

Сьогодні протоколи для децентралізованих платіжних систем (таких як Bitcoin, Ethereum та інші) вже існують. Це дозволяє людям зміцнювати довіру через однорангові системи електронних грошей та проводити транзакції без необхідності участі третьої сторони. Вони не контролюються централізовано ані банком, ані компанією, ані урядом. Чим більше мережа, тим більш децентралізована і безпечна. Blockchain - це базова технологія, яка завдяки співпраці безлічі лічильників і криптографії дозволяє встановлювати довірчі відносини не тільки у великих організаціях. Кожна окрема транзакція (у разі з криптовалютами) планується та аутентифікується випадковим користувачем. Це створює зростаючий перелік довірених записів, блоків. Окремі блоки з’єднуються за допомогою криптографічних інструментів для створення глобальної розподіленої “бухгалтерської книги”, яку кожен може перевірити у мережі у будь-який час. Таким чином, мережа уникає центральної влади, яка може бути корумпована або атакувати.

Окремі криптовалюти багато в чому різняться. Однак технологія Blockchain – це не лише криптовалюта. Ця технологія має набагато більше застосувань, ніж

цифрові гроші. Діаграма децентралізованої мережі, що є підмножиною розподіленої системи зображена на рис. 1.2.

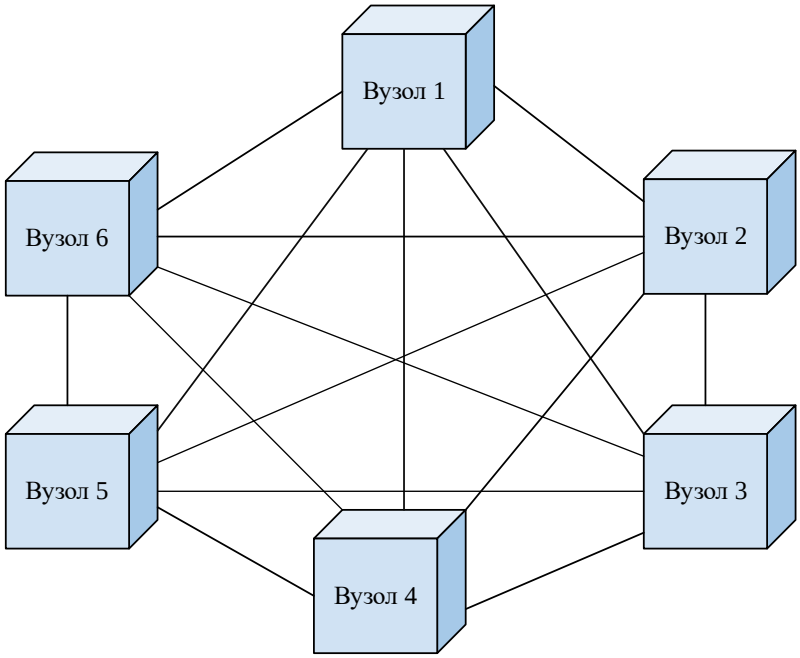

Рисунок 1.2 – Схема децентралізованої системи

Приклад децентралізованої системи – робота мережі BitTorrent або Torrent. У цьому прикладі немає центрального вузла, тобто всі події відбуваються між усіма комп’ютерами. Суть BitTorrent або Torrent у тому, що відбувається поширення дрібними частинами інформації та збереження її в усіх користувачів. Коли користувач включає BitTorrent або Torrent, то він починає передавати інформацію.

У централізованої та децентралізованої системи є свої плюси та мінуси.

До плюсів децентралізованої системи відноситься:

1. Якщо з ладу вилетить частина мережі, то мережа продовжить працювати.

2. Немає одноосібної цензури, цензура може бути в якомусь вигляді, але немає такого, що хтось там єдиний вирішив вас забанити.

3. Ваші дані знаходяться у вас, тобто дані належать вам, і вони не зберігаються на сервері і ви не залежите від інших учасників мережі.

До мінусів децентралізованої системи належить:

1. Немає відповідальних суб'єктів, тобто всі рівні один з одним і відповідального важко призначити.

2. У зв'язку з першим пунктом організувати службу підтримки складно та практично неможливо.

3. Складність прийняття рішень (якщо потрібно щось змінити у правилах, у протоколі, необхідно провести голосування).

4. Повільні процеси.

Далі перелічені основні аспекти децентралізації, які пов'язані з новими технологічними та організаційними ризиками: максимальне підвищення рівня незалежності кожного компонента системи; балансування між застосованими засобами та ефективністю роботи; децентралізація зберігається цілісності всієї системи.

Далеко не завжди порівняння децентралізації та централізації є гарною ідеєю, оскільки все залежить від вимог та умов експлуатації.

Першою успішною реалізацією технології Blockchain стала мережа Bitcoin. З цієї причини переважна більшість людей також асоціюють термін Blockchain виключно з криптовалютами. Однак потенціал Blockchain не обмежується Bitcoin. Технологія як така привертає велику увагу у різних галузях, включаючи фінансові послуги, благодійні та некомерційні організації, мистецтво та електронну торгівлю.

Децентралізація – одна з найбільших переваг Bitcoin. Для виконання Bitcoin-транзакцій сторонні особи не потрібні. Уряди, банки та фінансові посередники не мають можливості втручатися у транзакції користувачів. Bitcoin є одноранговою системою і тому дає користувачам більше свободи. Тому що валюта децентралізована і контролюється лише з боку користувачів, не стягуються збори третіх осіб або податки на транзакції.

У децентралізованих системах обліку фінансів, таких як криптовалюти, передбачається, що всі учасники зберігають копії однієї і тієї ж бази даних та оновлюють їх спільно, використовуючи алгоритми досягнення консенсусу.

Зокрема, Blockchain – розподілений цифровий реєстр, який записує транзакції, в яких відбувається обмін цінностями. Розподілене означає, що існує

кілька копій книги. У випадку децентралізована система передбачає, що учасники обмінюються інформацією безпосередньо, використовуючи Peer-to-peer (P2P) протоколи (рис. 1.3).

Рисунок 1.3 – Схема протоколу Peer-to-peer

Архітектура P2P – це розподілена архітектура без використання централізованої системи. У цій архітектурі кілька користувачів, які завантажують один і той же файл, наприклад, підключені один до одного. Це дає змогу обмінюватися даними між користувачами, що значною мірою зменшує відповідальність за завантаження, яка покладається на сервер в архітектурі клієнт-сервер. Фактично кожен підключений користувач одночасно є сервером і клієнтом.

Експерти можуть стверджувати, що P2P не слід характеризувати суворо ступенем централізації та децентралізації. Це вірно, оскільки централізація в P2P може існувати в різних ступенях, які можна класифікувати як:

1. Повністю централізований: наприклад, клієнт-сервер, де послуга доступна на одному хості, а клієнт, якому потрібна ця служба, повинен зв’язатися з цим конкретним хостом або сервером.

2. Посередництво: централізовано лише деякі, але не всі функції. Централізованими функціями часто є операції з бухгалтерського обліку, такі як реєстраційна інформація, моніторинг послуг, активний пошук серверів.

3. Повністю децентралізована: у якій немає двох однорангових пристроїв (відрізняються за своєю функціональністю), та ці пристрої різняться лише даними або вмістом, які вони передають. Наприклад, Gnutella – це розподілена мережа

обміну файлами, де ніхто не відрізняється від іншого, крім вмісту, яким він ділиться.

Системи P2P були розроблені та розгорнуті для широкого спектра додатків, таких як розповсюдження вмісту, розподілене сховище та розподілені обчислення, і сьогодні складають значну частину інтернет-трафіку. Їх децентралізована природа дає багато переваг, головним чином здатність до самомасштабування: нові учасники мотивовані вносити ресурси, які компенсують додаткове робоче навантаження, яке вони створюють.

Peer-to-peer (P2P) протоколи самостійно зберігають дані, які їм потрібні. Для цього вони запускають спеціальне програмне забезпечення, яке підтримує необхідний P2P-протокол конкретної децентралізованої системи. Децентралізація також пропонує кращу відмовостійкість і стійкість до атак (оскільки не існує єдиної точки збою, яка могла б зруйнувати всю систему), низький бар'єр для розгортання (оскільки для розгортання P2P-сервісу потрібно менше виділеної інфраструктури) і покращену конфіденційність користувачів.

1.2 Загрози на централізовані та децентралізовані системи

Як для централізованих, так і для децентралізованих систем є певні загрози, які так чи інакше діють на ці системи. Для того, щоб краще ознайомитися з загрозами на децентралізовані системи, розглянемо для початку приклади загроз на централізовані системи. Загалом це загрози на банківський сектор. Реалізація таких загроз призводять до збитків банкам, втрати прибутків, соціальною чи психологічною напруженістю навколо установи банків або в їх колективах. До числа загроз на банківський сектор належать як внутрішні, так і зовнішні загрози. Існує три складові загроз:

- безпека інформації: знищення інформаційних даних на магнітних носіях, відмова клієнта від прийому/передачі даних, знаходження необхідних паролів при спостереженнях, копіювання даних з терміналів / обладнання / магнітних носіїв, скрімінг, фармінг, фішинг / телефонний фішинг;

- інформаційна безпека: внесення змін до даних та програми для підробки і фальсифікації фінансових документів витяг інформації зі статичних баз даних, нелегальне використовування секретної інформації та ведення даних, створення помилкових тверджень про отримання платіжних документів;
- кібернетична безпека: віртуальне викрадення, виявлення паролів користувачів, знищення / модифікація / блокування інформації, несанкціоноване перевищення повноважень на доступ, DoS / U2R / R2L – атаки.

Отже, безпека є ключовим компонентом для кожної системи. Основні завдання захисту даних:

- забезпечення неможливості доступу сторонніх осіб до змісту документа;
- забезпечення впевненості одержувача у тому, що документ цілісний і справжній, тобто при передачі не було підмінено або відредаговано інформацію.

Відповідно модель загроз, так само як і модель порушника в таких системах, можуть сильно відрізнятися від своїх традиційних аналогів. Модель загроз є структурованим описом можливих загроз. А модель порушника є структурною характеристикою порушника.

Дуже цікаве питання щодо технології Blockchain, це питання захисту персональних даних. Це спільне питання у Blockchain. Оскільки розподілений реєстр є розподіленим, тобто ми не можемо контролювати, де знаходиться особиста інформація, тому що вона раптово виявляється у всьому світі. А у випадку з технологією Blockchain у нас немає жодного контролю над тим, де всюди з’являються особисті дані. З одного боку, це справді питання обробки персональних даних, але, з іншого боку, коли справа доходить до цього, якщо ми проводимо аналіз ризиків, ми виявляємо, що ризик неправомірного використання цих персональних даних є абсолютно мінімальним, тому що особисті дані вбудовуються в зашифровані блоки або у вигляді геша. Звідси випливає, що така обробка персональних даних є незаконною, але насправді тому, що існує мінімальний ризик, тому мінімальна потреба у захисті даних.

Для побудови будь-якої надійної облікової системи, перш за все, необхідно визначити з якими проблемами система може зіткнутися. Для кожної

інформаційної системи необхідна розробка політики безпеки, яка включає розробку моделі загроз, порушника і також модель інформаційної безпеки системи.

Безпеку Blockchain можна розглядати з різних точок зору. Безпека мережі Blockchain як інфраструктури та безпека Blockchain-додатків, тобто контрактів.

Якщо розглядати перший варіант, то можна зазначити, що історія блоків захищена; тобто блоки є незмінними після того, як вони додані до книги Blockchain. Техніка ланцюжка зв'язує кожен блок з попереднім за допомогою геш-показівників. Таким чином, вміст блоку n + 1 містить геш блоку n. Тому будь-яке загартовування в блоці n миттєво ставить під загрозу дійсність усіх наступних блоків. Комбінація дерева Merkle (рис. 1.4) і геш-показівників забезпечує безпечну та ефективну модель даних, яка відстежує несанкціоновані зміни або зловмисне втручання в реєстр Blockchain.

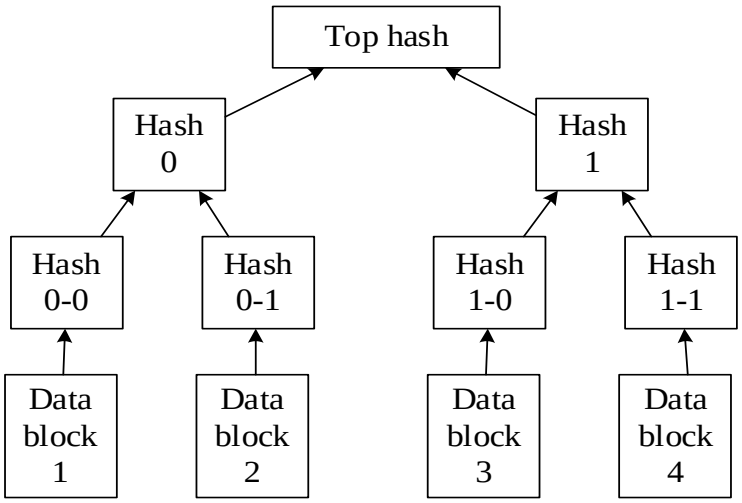

Рисунок 1.4 – Комбінація дерева Merkle

Як приклад децентралізованої системи буде облікова система Bitcoin. Опис можливих загроз наведено на рис. 1.5.

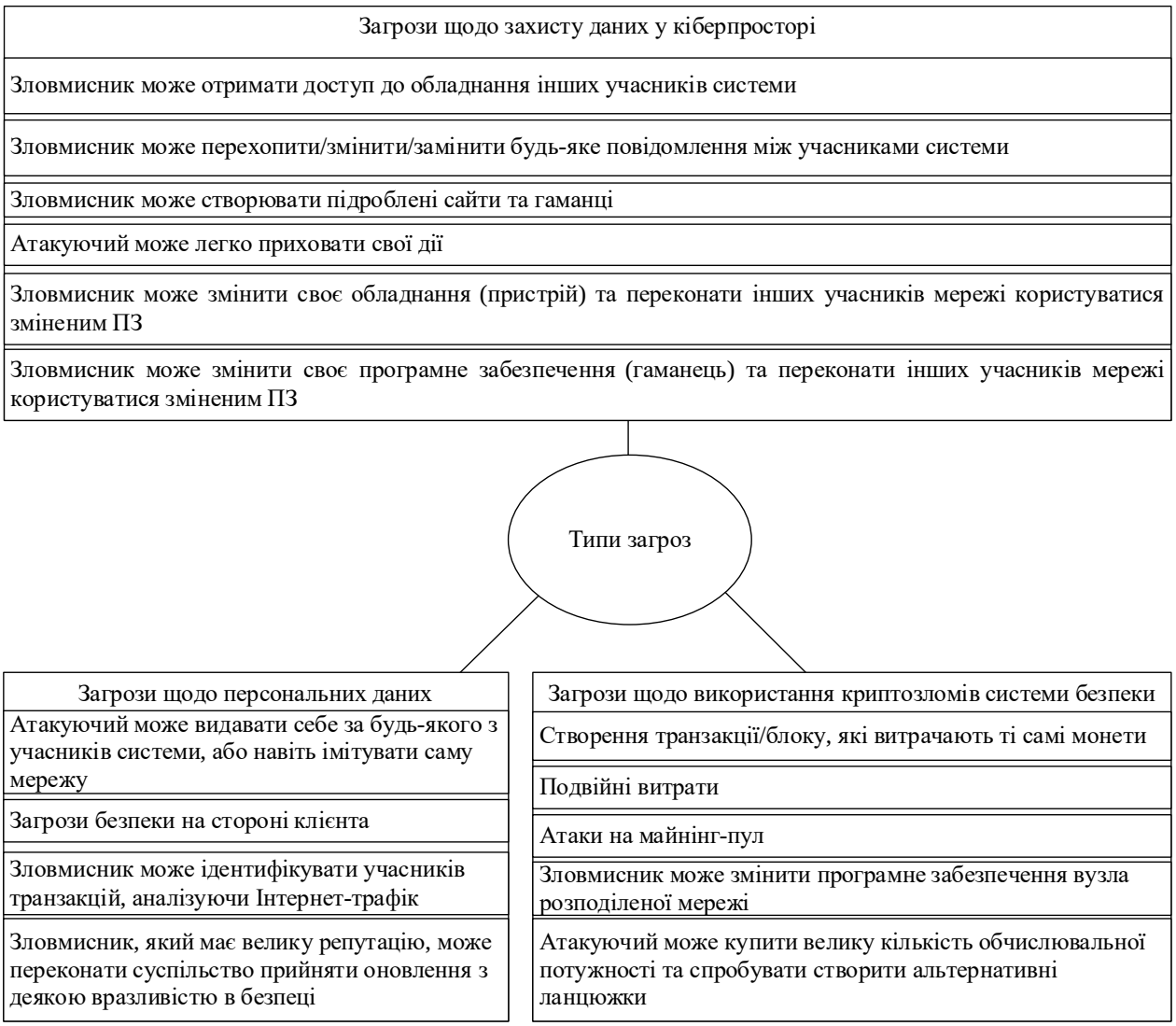

Рисунок 1.5 – Типи загроз

Загрози, що стосуються створенню транзакції/блоку, які витрачають ті самі монети, або якщо зловмисник, який має велику репутацію, може переконати суспільство прийняти оновлення з деякою вразливістю в безпеці, вважаються найскладніші з погляду реалізації, оскільки для їхнього успішного проведення необхідна велика підтримка із боку учасників системи. А оскільки від запропонованих оновлень безпосередньо залежить безпека монет користувачів, такі пропозиції ретельно розглядатимуться і впровадитимуть у них backdoors досить проблематично. В цю категорію також підходять загрози стосовно того, що зловмисник може створити форк та додати деяку вразливість безпеки, та може побудувати небезпечний протокол поверх системи Bitcoin.

Метою загроз стосовно атакуючого, який може купити велику кількість обчислювальної потужності та спробувати створити альтернативні ланцюжки блоків та зловмисник, який може отримати доступ до обладнання інших учасників системи (тим самим заволодіти переважно обчислювальною потужністю) є володіння великою кількістю обчислювальної потужності. Реалізація цих загроз дуже дорога і рідко вигода від таких атак перевищує витрати на її проведення. З іншого боку, для користувачів такі атаки дуже небезпечні, оскільки в цих випадках атакуючий обманює користувачів, при цьому уникаючи порушення самого протоколу.

Загрози, в яких зловмисник може створювати підроблені сайти та гаманці, змінити програмне забезпечення вузла розподіленої мережі, змінити своє обладнання (пристрій) та переконати інших учасників мережі користуватися зміненим ПЗ та змінити своє програмне забезпечення (гаманець) та переконати інших учасників мережі користуватися зміненим ПЗ, їх реалізувати набагато простіше попередніх. Найчастіше вони можуть бути реалізовані постачальниками програмного та апаратного забезпечення. І враховуючи те, що далеко не кожен користувач перевіряє вихідний код гаманця на предмет уразливостей, такі атаки більш ніж ймовірні і можуть призвести до втрати користувачами монет. Як захист від подібних атак користувачам рекомендується перевіряти ПЗ гаманців на предмет вразливостей і використовувати ПЗ тільки від довірених джерел.

Загрози, в яких зловмисник може перехопити/змінити/замінити будь-яке повідомлення між учасниками системи, видає себе за будь-якого з учасників системи, або навіть імітує саму мережу, легко приховує свої дії та може ідентифікувати учасників транзакцій, аналізуючи Інтернет-трафік провести найпростіше, оскільки вони реалізуються на мережевому рівні. Наслідки таких атак дуже рідко можуть призвести до втрати монет користувачами, проте обмежити доступ користувача до актуальної інформації та обмежити його дії в мережі можуть. Захистити децентралізовану систему від подібних атак проблематично, але можливо за рахунок збільшення кількості зв’язків кожного користувача з іншими вузлами мережі.

Стосовно подвійних витрат, то якщо споживач використовує ту саму криптовалюту для кількох транзакцій, то це вважається подвійними витратами. Зловмисник може проводити гоночні атаки, щоб ініціювати подвійні витрати. У Blockchain на основі POW ці типи атак порівняно легко реалізувати, тому що зловмисник може легко використовувати час між ініціюванням двох транзакцій, а також підтвердженням двох транзакцій. Перш ніж друга транзакція зловмисника стала недійсною, він отримав результат першої транзакції, що може призвести до подвійних витрат.

Атаки на майнінг-пул використовуються для збільшення обчислювальної потужності або геш-потужності блоку. Та створюються пули для майнінгу. Ці пули впливають на час, необхідний для перевірки блоку. Ці пули для майнінгу також збільшують шанси на нагороду за майнінг (рис. 1.6).

Рисунок 1.6 – Атака на майнінг-пул

Пули для майнінгу розвиваються і вразливість для використання цих пулів також збільшується. Атаки на майнінг-пул бувають двох типів:

1. Внутрішні атаки - це атаки, у яких майнер зловмисно збирає більше, ніж потрібно, і порушує нормальну функціональність, у результаті пул ігнорує успішні спроби майнінгу.

2. Зовнішні атаки – це атаки, які виникають, коли майнер використовує вищу геш-потужність для атаки пула, що призводить до подвійних витрат. Атаки на пул

майнінгу включають егоїстичний майнінг, стрибкоподібні атаки, утримання блоків, хабарництво тощо.

Загрози безпеки на стороні клієнта також вважається значною загрозою, бо кожен користувач у мережі Blockchain має набір закритих відкритих ключів для доступу до своїх гаманців із криптовалютою. Тому необхідно безпечно керувати цими ключами. Важливим аспектом безпеки на стороні клієнта є те, що якщо клієнт втратить або скомпрометує ключі, він не зможе отримати доступ до свого гаманця, що призведе до безповоротних грошових втрат. Безпека клієнта скомпрометована за допомогою різних механізмів, таких як зламування, використання програмного забезпечення з помилками або неправильне використання гаманця.

Беручи до уваги загрози, які були перелічені вище, можна зробити висновок, що злочинна діяльність тільки зростає. До одного користувача можна прив'язати кілька Bitcoin-адрес, і ця адреса не має відношення до її справжньої особи в житті. Тому Bitcoin використовувався у незаконній діяльності. Bitcoin підтримується різними сторонніми платформами, і користувачі можуть купувати або продавати продукти за допомогою цих платформ. Відстежити поведінку користувачів досить складно, оскільки процес продажу та купівлі з використанням сторонніх платформ є анонімним. Злочинні дії, які часто здійснюються з використанням Bitcoin, полягають у наступному.

Програми-вимагачі. Він часто використовується злочинцями для вимагання грошей з використанням Bitcoin як торговельну валюту. У липні 2014 року програма-вимагач CTB-Locker поширилася по всьому світу у вигляді поштового вкладення. Якщо користувач клацає вкладення, програма-вимагач запускається у фоновому режимі системи та шифрує приблизно 114 типів кожного з них. Жертва має заплатити зловмиснику певну суму в біткойнах Wi-Fi протягом 96 годин. В іншому випадку encrypted_les не буде відновлено.

Підземний ринок. Bitcoin часто використовується на підпільному ринку як валюта. Silk Road, наприклад, є анонімним міжнародним онлайн-ринком, який працює як прихована служба Tor і використовує Bitcoin як валюту. Більшість

продуктів, що продаються на Шовковому шляху - це наркотики або інші контрольовані продукти в реальному світі.

Оскільки на Шовковий шлях припадає значна частина міжнародних транзакцій, Bitcoin робить транзакцію зручнішою на підпільному ринку, що завдає шкоди соціальному забезпеченню.

Відмивання грошей. Оскільки Bitcoin має такі функції, як анонімність та оплата через віртуальну мережу, і був прийнятий у багатьох країнах, Bitcoin несе найнижчий ризик відмивання грошей у порівнянні з іншими валютами. О, Коді та ін. Запропонувати Темний гаманець, Bitcoin додаток, який може повністю приховати та приховати транзакцію Bitcoin. Dark Wallet може шифрувати та змішувати інформацію про транзакції користувача з дійсними монетами, що значно спрощує відмивання грошей.

Активно розвиваються технології пов’язані з криптовалютою та подібними їй інструментами. А так як капітал привертав і продовжує привертати увагу зловмисників, загальнодоступні Blockchain-мережі регулярно піддаються різним атакам. Багато атак на самі Blockchain-мережі як такого впливу немають, тому що основна суть – це викрадення активів. Достут до них забезпечується тільки за допомогою приваних ключів. Перед тим як розбирати цей вид загроз, необхідно описати принцип володіння цифровим активом, зокрема криптовалютою.

Приватний ключ може бути згенерований особисто користувачем. Він забезпечить доступ до адреси в мережі Blockchain, де в свою чергу і зберігається цифровий актив. Розпоряджатися цим активом може тільки той, хто знає значення приватного ключа.

Ще один вид захисту цифрових активів – це технологія мультипідпису. Суть технології заключається в тому, що для підтвердження транзакції необхідно декілька підписів. Учасники таких транзакцій, зберігаються в таємниці, і це збільшує захист самого гелектроннго гаманця.

Як правило, ця конфігурація у гаманці реалізується за допомогою смарт-контрактів.

Окрім цього, існує кілька загроз безпеки, пов’язаних із хмарою, які перешкоджають її широкомасштабному застосуванню для цілей обміну даними. Однією з основних загроз, пов’язаних із хмарою, є інсайдерські атаки, коли хмарні постачальники використовують свої привілеї для витоку або маніпулювання даними користувачів. Хмарне середовище також уразливе до кількох шкідливих атак. Зловмисники можуть використовувати вразливості хмарної інфраструктури за допомогою зловмисного програмного забезпечення, включаючи віруси та руткіти, для крадіжки даних користувачів, скомпрометувати здатність механізму контролю доступу захищати дані та застосовувати та оцінювати свої політики або навіть маніпулювати цими політиками, які можуть залишатися непоміченими значну кількість часу.

1.3 Механізми забезпечення безпеки в децентралізованих системах

Для забезпечення безпеки використовуються геш-функції та цифровий підпис (рис. 1.7).

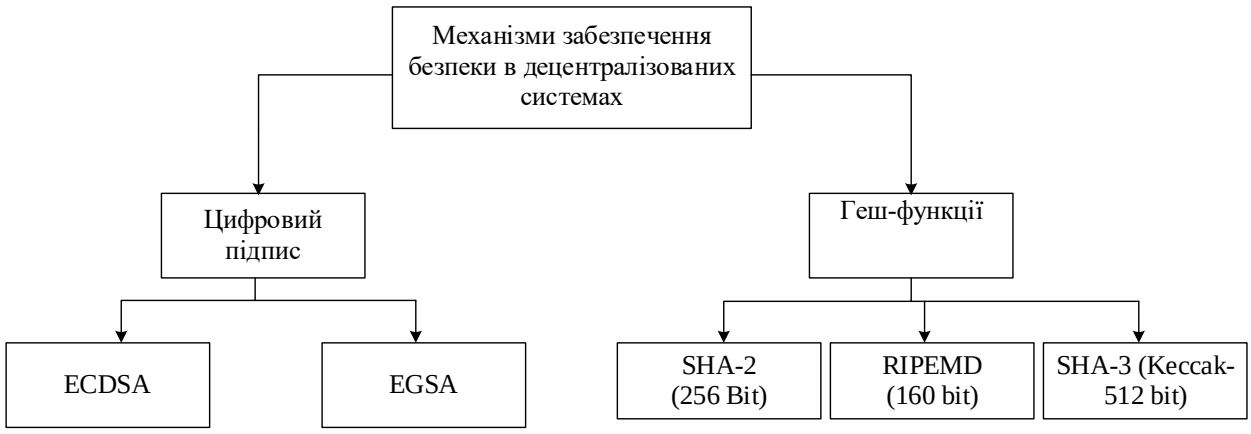

Рисунок 1.7 – Механізми забезпечення безпеки в децентралізованих системах

Геш-функції – являє собою функції, які перетворюють вхідних даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Геш-функція

гарантує, що якщо інформація буде змінена будь-яким чином – навіть на один біт, – в результаті вийде зовсім інше геш-значення.

Геш виглядає як набір випадкових чисел та літер. Це буквено-цифрове значення, яке однозначно ідентифікує дані або цифровий відбиток даних.

Функція гешування приймає дані як вхідні дані і повертає унікальний геш. Оскільки геш є “цифровим відбитком” всього блоку, дані є комбінацією індексу, позначки часу, попереднього геша, даних блоку та одноразового номера (nonce).

Основним алгоритмом шифрування, що використовується в Blockchain, є SHA-256 для обчислення результату гешування під час майнінгу блоку, який є ключовим моментом під час процесу майнінгу. Одноразове значення вимагає обчислення відповідно до складності, встановленої Blockchain.

SHA-256 є найпоширенішим і ефективним криптографічним геш-алгоритмом, який широко використовується при шифруванні даних у техніці Blockchain. Якщо йдеться про Bitcoin, то використовується алгоритм RIPEMD-160, а якщо говорити про Ethereum, то використовується алгоритм SHA-3, а саме Keccak-256 й Keccak-512.

Криптографічний геш-алгоритм – це метод, який перетворює вихідні дані за допомогою спеціального алгоритму в інший тип нових даних із фіксованою довжиною для пошуку. За допомогою алгоритму SHA-256 дані будуть передані у 256-бітний геш, і він підходить для захисту з двох основних причин. По-перше, це свого роду одностороння функція, яка не може бути розшифрована назад. Насправді ви не можете знайти жодної підказки щодо вихідних даних через зашифрований результат. Іншою причиною, яку не можна ігнорувати, є те, що будь-яка незначна зміна даних призведе до помітної різниці в результатах шифрування. Таким чином, практично неможливо, щоб різні входи виробляли однаковий вихід.

Алгоритм гешування SHA-256, який використовує Bitcoin-Blockchain, є широко використовуваною і дуже сильною криптографічною геш-функцією. Сильна геш-функція створює вихідне повідомлення, яке не можна використовувати для точного вгадування вхідного повідомлення. Геш-функції, такі як SHA-256,

часто використовуються для шифрування паролів. Таким чином, якщо сервер, що містить геш-виходи, буде зламаний, він не може бути використаний для отримання вхідного повідомлення

Відкритий ключ піддається гешуванню за допомогою SHA-256, а для результату знову розраховується геш-значення, але вже за допомозі RIPEMD160. На виході виходить число завдовжки 160 біт (20 байт).

Геш-функції є компонентами для багатьох важливих програм інформаційної безпеки, включаючи

1) генерацію та перевірку цифрових підписів;

2) отримання ключів;

3) генерацію псевдовипадкових бітів.

1.3.1 Алгоритм SHA-2 (256 bit)

SHA-2 (256 bit) – геш-функція призначена для створення так званих “відбитків” (fingerprint) або “дайджестів” (digest) фіксованої довжини для повідомлень довільної довжини. Вхідне повідомлення після доповнення до ннеобхідної довжини останнього блоку розбивається на блоки, кожен блок – на 16 слів. Алгоритм ітераційно опрацьовує кожен блок повідомлення через цикл із 64 або 80 ітераціями (раундами). Результати обробки кожного блоку залежать від результатів обробки попереднього і є значенням суми за модулем попередньої геш-функції та результатом обробки поточного блоку. Однак, ініціалізація внутрішнього стану здійснюється результатом обробки попереднього блоку. Тому незалежно обробляти блоки та складати результати не можна.

Як ми бачимо на рис. 1.8 сині блоки нелінійно перемішують біти для ускладнення криптографічного аналізу. Причому для ще більшої надійності використовуються різні функції перемішування (якщо ви зможете знайти математичну лазівку для швидкого генерування валідних геш, то візьмете під контроль весь процес Майнінг Bitcoin).

Функція більшості (Ma блок) побітово працює зі словами A, B і C. Для кожної бітової позиції вона повертає 0, якщо більшість вхідних бітів в цій позиції – нулі, інакше поверне 1.

Блок Σ0 циклічно зсуває A на 2 біти, потім вихідне слово A циклічно зсувається на 13 біт, і, аналогічно, на 22 біти. Утворені три зсунуті версії A побітово складаються по модулю 2 (звичайний xor, (A ror 2) xor (A ror 13) xor (A ror 22)).

Ch реалізує функцію вибору. На кожній бітової позиції перевіряється біт з E, якщо він дорівнює одиниці, то на вихід йде біт з F з цієї позиції, інакше біт з G. Таким чином, біти з F і G перемішуються, виходячи зі значення E.

Σ1 за структурою аналогічний Σ0, але працює зі словом E, а відповідні зсувні константи – 6, 11 і 25.

Червоні блоки виконують 32-бітове складання, формуючи нові значення для вихідних слів A і E. Значення Wt генерується на основі вхідних даних (це відбувається в тій ділянці алгоритму, який отримує і обробляє гешовані дані). Kt – своя константа для кожного раунду.

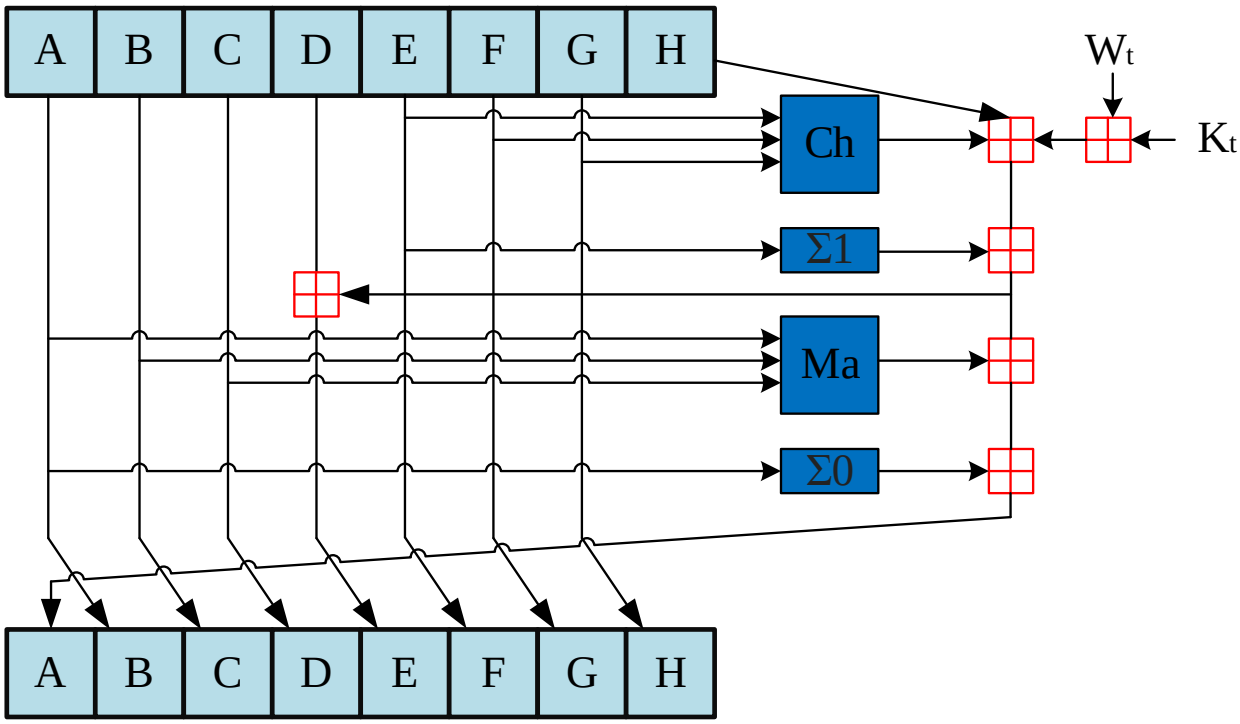

Рисунок 1.8 – Схема однієї ітерації алгоритмів SHA-256

Що стосується Bitcoin, алгоритм SHA-256 використовується тричі. Для перших двох раундів вага розраховується з використанням blockhead, тоді як вага третього раунду використовує результат другого раунду. У першому раунді повідомлення - це номер версії та попередній геш, а також більша частина кореня Меркла. Другий тур містить частину blockhead. Це включає решту кореня Меркла, всю мітку часу, мету складності і одноразовий номер. Початкові значення геш-функції для першого та третього раундів зумовлені, тоді як початкові геш-значення других раундів є результатом першого раунду (рис. 1.9).

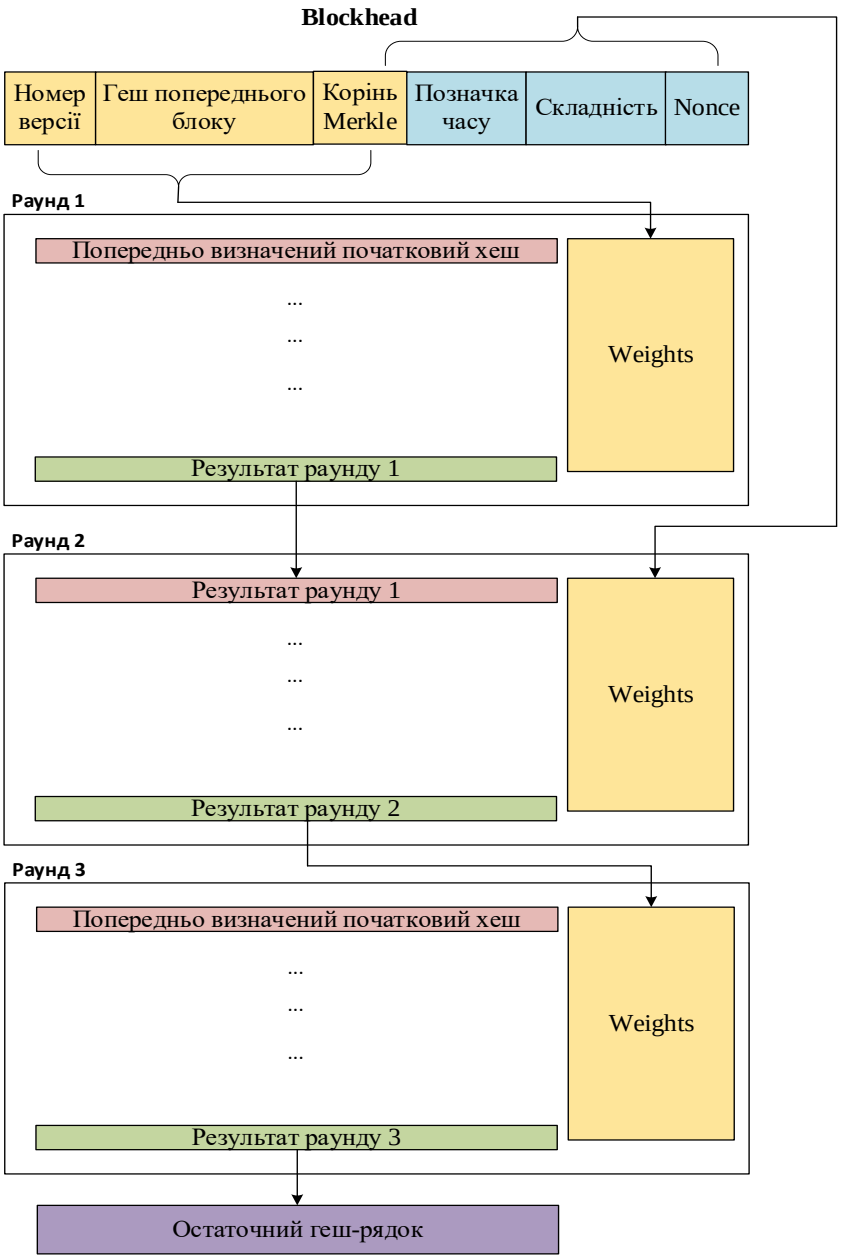

Рисунок 1.9 – Три раунди SHA-256 щодо Blockchain Bitcoin

1.3.2 Алгоритм RIPEMD (160 bit)

Також в криптовалютних системах на основі технології Blockchain Bitcoin використовується RIPEMD (160 bit) – RIPEMD-160 – це криптографічна геш-функція, заснована на конструкції Меркла-Дамгарда. Він використовується в стандарті Bitcoin. Це розширена версія алгоритму RIPEMD, яка створює 128-бітний геш-дайджест, тоді як алгоритм RIPEMD-160 видає 160-бітний вихід. Функція стиснення складається з 80 етапів, що складаються з 5 блоків, кожен з яких виконується 16 разів. Цей шаблон виконується двічі, а результати об’єднуються внизу за допомогою додавання за модулем 32. Функція стиснення складається із підблока змінної, який блок повідомлення передається 16 разів. Є 5 різних варіантів, всього 80 прогонів. Цей процес відбувається двічі, коли дані, що зустрічаються внизу, переміщаються в наступний блок (якщо він є) або додаються до геш-реєстру, якщо його немає. Підблок може бути змінений конструкцією нелінійної функції, порядку, в якому блок повідомлення читається за раунд, величиною повороту вліво і константою Загальна схема функції стиснення показані на рис. 1.10.

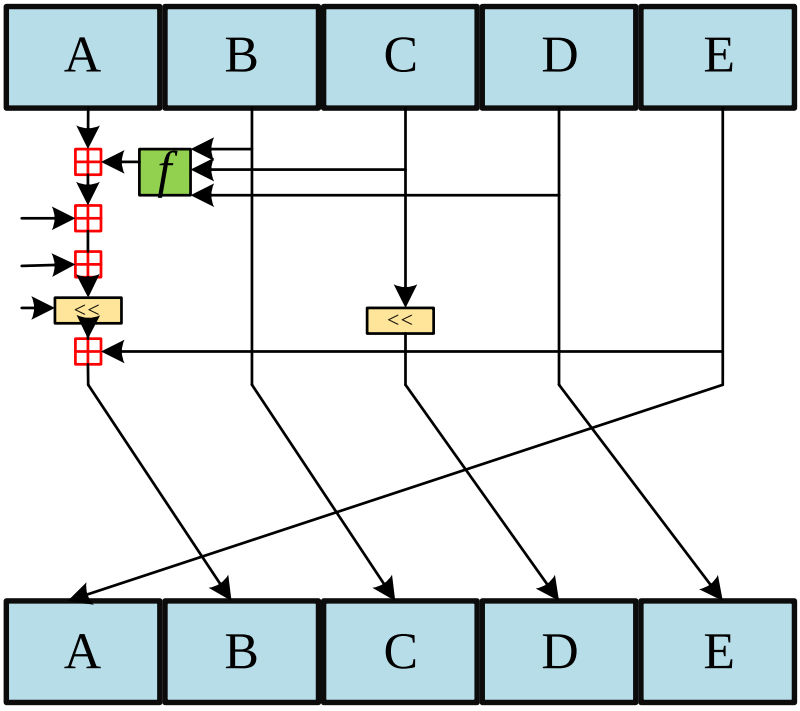

Рисунок 1.10 – Реалізація алгоритму RIPEMD-160

1.3.3 Алгоритм SHA-3 (Keccak-512 bit)

Алгоритм SHA-3 (Keccak-512) використовується в Ethereum. Це децентралізована програмна платформа, яка працює на смарт-контрактах. Це децентралізовані програми, які запрограмовані на самостійну роботу без ризику збою або без втручання третіх осіб. Ці програми базуються на базі даних розподіленого ланцюжка блоків. Платформа Ethereum підтримується криптовалютою Ether, яка потрібна на її роботи. Ethash – це алгоритм інтелектуального аналізу, реалізований мережею Ethereum та криптовалютами на основі Ethereum. Ethash є наступником попереднього алгоритму Ethereum, званого Dagger-Hashimoto, і насправді, є його оновленням. Проте, поточні етапи розробки обох алгоритмів зробили їх занадто відмінними від того, щоб вважатися одним і тим самим алгоритмом. Ethash використовує алгоритми гешування “Keccak-256” і “Keccak-512”, що призводить до деякої плутанини через одночасну розробку криптографічних стандартів SHA-3 (Secure Hash Algorithm 3) поряд з розробкою Ethash.

Стандарт SHA-3 (Secure Hash Algorithm-3) визначає нове сімейство функцій, які доповнюють SHA-1 і SHA-2 сімейство геш-функцій. Сімейство SHA-3 складається з чотирьох криптографічних геш-функцій і двох функцій розширюваного виводу. Ці шість функцій мають губчату конструкцію, функції з такою структурою називаються функціями губки. SHA-3 (Keccak) – алгоритм гешування змінної розрядності. Кожна з функцій SHA-3 ґрунтується на екземплярі алгоритму KECCAK.

Конструкція губки - це структура завдання функцій над двійковими даними з довільною вихідною довжиною. У конструкції використовуються такі три компоненти:

1. Базова функція на рядках фіксованої довжини, що позначається f.

2. Параметр, званий швидкістю, позначається r.

3. Правило наповнення, що позначається pad.

Ця функція приймає два входи: бітовий рядок, позначений N, і довжину в бітах, позначену d. Аналогія з губкою полягає в тому, що довільна кількість вхідних

бітів поглинається станом функції, після чого довільна кількість вихідних бітів видавлюється з її стану. Функція f показує рядки однієї фіксованої довжини, позначені b, на рядки такої ж довжини. b називається шириною f. Функції SHA-3 є прикладами конструкції губки, в якій базова функція f є оборотною, хоча конструкція губки не вимагає, щоб f була зворотною. Швидкість r – це додатне число, яке строго менше за ширину b. Місткість, позначена c, є цілим натуральним числом b-r. Таким чином, r + c = b. Правило заповнення, pad, є функцією, яка створює заповнення, тобто рядок відповідної довжини для додавання до іншого рядка. Загалом, якщо врахувати ціле додатне число x і ціле невід'ємне число m, вихідний pad(x, m) є рядком із властивістю, що m + len(pad(x, m)) є додатним кратним x. У конструкції губки x = r і m = len(N), щоб доповнений вхідний рядок можна було розділити на послідовність r-бітових рядків. Отже, схема SHA-3 (Keccak) проілюстрована на рис. 1.11.

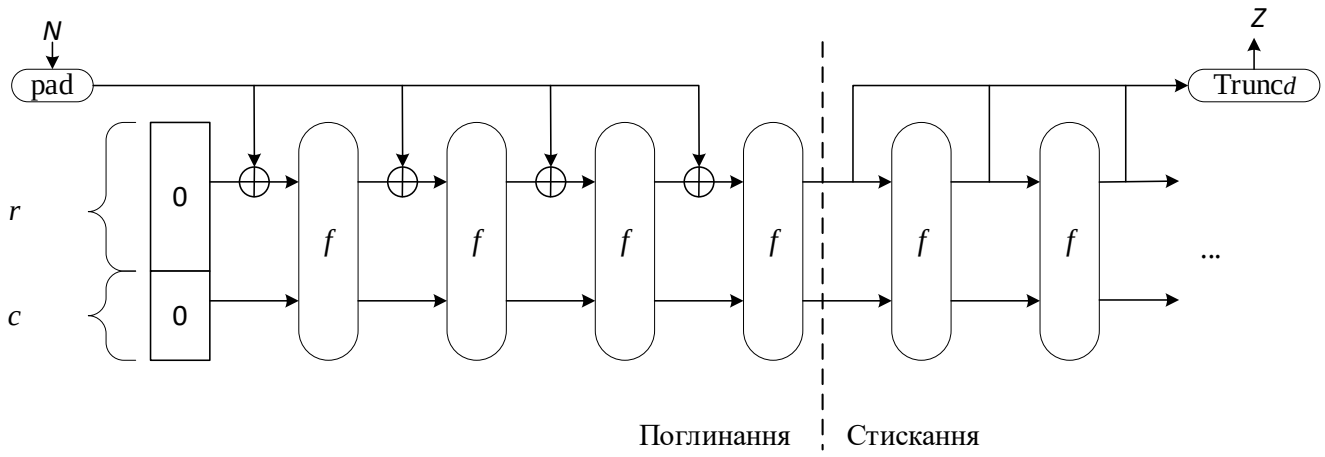

Рисунок 1.11 – Схема SHA-3 (Keccak) складається із двох етапів:

1) Поглинання. Початкове повідомлення M піддається багатораундовим перестановкам f; 2) Стискання. Висновок значення в результаті перестановок Z

Чотири геш-функції SHA-3 є альтернативами функціям SHA-2, і вони призначені для забезпечення стійкості проти зіткнень, атак прообразу та другого прообразу, що дорівнює або перевищує опір, який надають відповідні функції SHA-2. Функції SHA-3 також призначені для протидії іншим атакам, таким як атаки розширення довжини, яким протистояла б випадкова функція тієї ж вихідної

довжини, загалом забезпечуючи ту саму силу безпеки, що й випадкова функція, аж до вихідних даних довжина.

Деякі з ключових особливостей вибору губчастої конструкції для Keccak є

1. Конструкція Sponge, яка дозволяє використовувати режими, захищені від загальних атак.

2. Це свого роду блоковий шифр без розкладу ключа.

3. Вибір операцій, з яких він складається, – це прості зміщення XOR, AND, NOT та циклічні зсуви. Немає таблиць пошуку, арифметичних операцій чи операцій, що залежать від даних.

4. Це функція, здатна генерувати вихідні дані змінної довжини.

5. Він гнучкий з точки зору рівня безпеки, торгуючи бітрейтом за місткість, не змінюючи перестановки.

Завдяки всім вищезгаданим властивостям його можна використовувати для кількох функцій, таких як потоковий шифр, функція генерування маски, генератор псевдовипадкових бітів з можливістю повторного використання, а також ефективне автентифіковане шифрування.

При взаємодії конкретного користувача з мережею Bitcoin потрібно враховувати низку аспектів. Їх не можна ігнорувати, оскільки у цих випадках користувач змушений комусь довіряти. Наприклад, якщо ви вірите, що баланс на вашому bitcoin-гаманці правильний, це означає, що ви довіряєте:

- тим, хто радив встановити цей конкретний гаманець;
- що гаманець, який ви використовуєте, розроблений та реалізований коректно;
- що ваша операційна система не зламана;
- що апаратне забезпечення вашого комп’ютера не було зламано;
- що ваш гаманець підключений до вірогідного Bitcoin вузла (а не до зловмисного);
- що імплементація криптографічної бібліотеки не має прихованих уразливостей;
- що конкретний криптографічний алгоритм не містить помилок.

Децентралізований підхід дозволяє зменшити можливість компрометації системи. У контексті системи Bitcoin це означає, що кожен користувач мережі може самостійно та незалежно перевірити коректність будь-якої транзакції чи блоку, йому не доводиться довіряти даним, наданим третьою стороною.

1.3.4 Цифровий підпис

Цифровий підпис закладається в тому, що людина, яка хоче підписати повідомлення, спочатку генерує ключову пару (рис. 1.12).

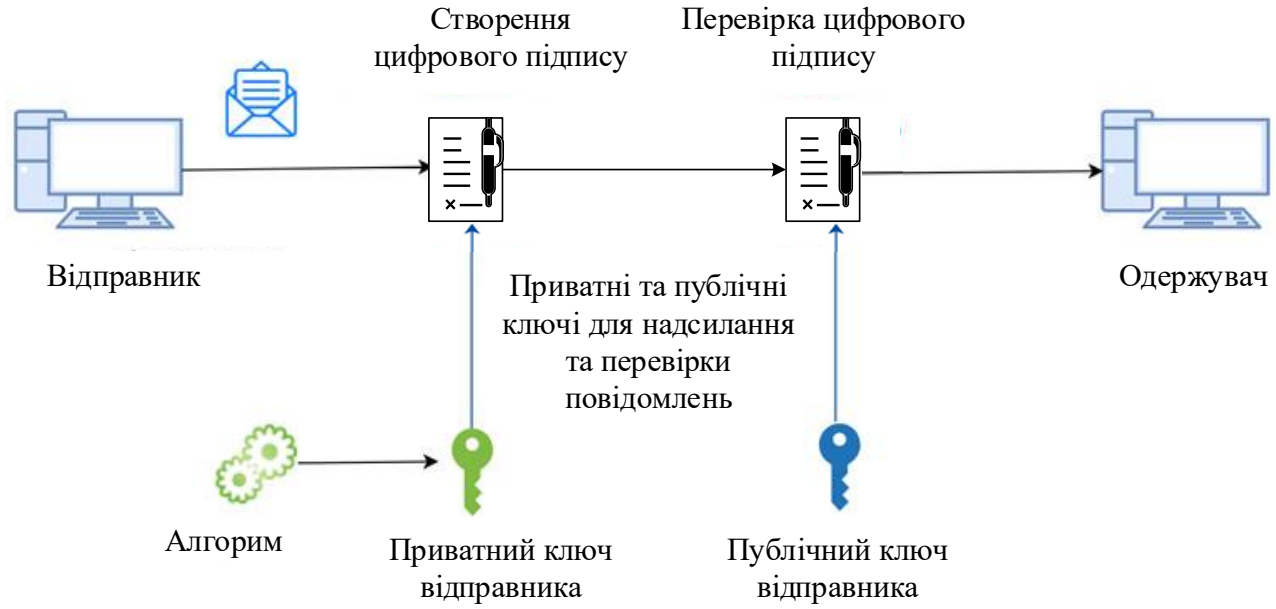

Рисунок 1.12 – Цифровий підпис

Особистий ключ ця людина зберігає у себе, щоб сформувати цифровий підпис. А відкритий публікує, щоб одержувач підписаного повідомлення міг перевірити автентичність цього повідомлення. Тепер, щоб підписати повідомлення, він використовує свій особистий ключ. Після цього він розповсюджує два файли: підписане повідомлення та прикріплений підпис. Інша людина, яка хоче переконатися, що документ було правильно сформовано та правильно підписано, може взяти відкритий ключ, повідомлення, після чого перевірити відповідність підпису відкритому ключу та повідомлення. Функція

формування підпису приймає на вхід повідомлення та особистий ключ автора, а на виході формує дані підпису, що прикріплюються до повідомлення (рис. 1.13).

Рисунок 1.13 – Функції створення цифрового підпису

Далі повідомлення разом із підписом надсилається одержувачу. Одержувач, щоб переконатися, що підпис є коректним, викликає функцію перевірки. Ця функція приймає три параметри: повідомлення, значення підпису та відкритий ключ. Вона повертає значення логічної змінної, чи вірний підпис. Виявитися неправильним підпис може у випадках, коли або повідомлення було підмінено, або дані підписи було порушено, або для перевірки використовувався неправильний (пошкоджений або підмінений) відкритий ключ.

1.4 Використання децентралізованих систем

1.4.1 Принципи побудови криптобірж

Існує ряд сервісів, які обмінюють криптовалюти на фіатні гроші і навпаки. BTCX є прикладом цього. Однак цей конкретний сервіс продає лише bitcoin та ефір. Це обмеження, коли користувач може обмінювати лише фіатні валюти на обмежений набір криптовалют, є загальним для багатьох чинних бірж.

Криптовалютні гаманці потрібні користувачам, щоб вони могли керувати своїми грошима. Такий додаток дуже схожий на традиційний гаманець. Ви зберігаєте в ньому свої гроші до тих пір, поки не захочете щось купити або отримати гроші, які хочете зберегти в ньому, тобто здійснити операції. Першим

кроком для користувача є рішення, який тип гаманця використовувати. Наступним кроком для користувача є створення облікового запису. Для цього існує кілька рівнів анонімності. Деякі гаманці вимагають ім’я та адреси електронної пошти, а інші взагалі не вимагають особистої інформації. Оскільки криптовалюта є віртуальною валютою, користувачі повинні зберігати її на носії, який називається криптовалютним гаманцем. Ці гаманці можуть існувати в різних форматах (рис. 1.14).

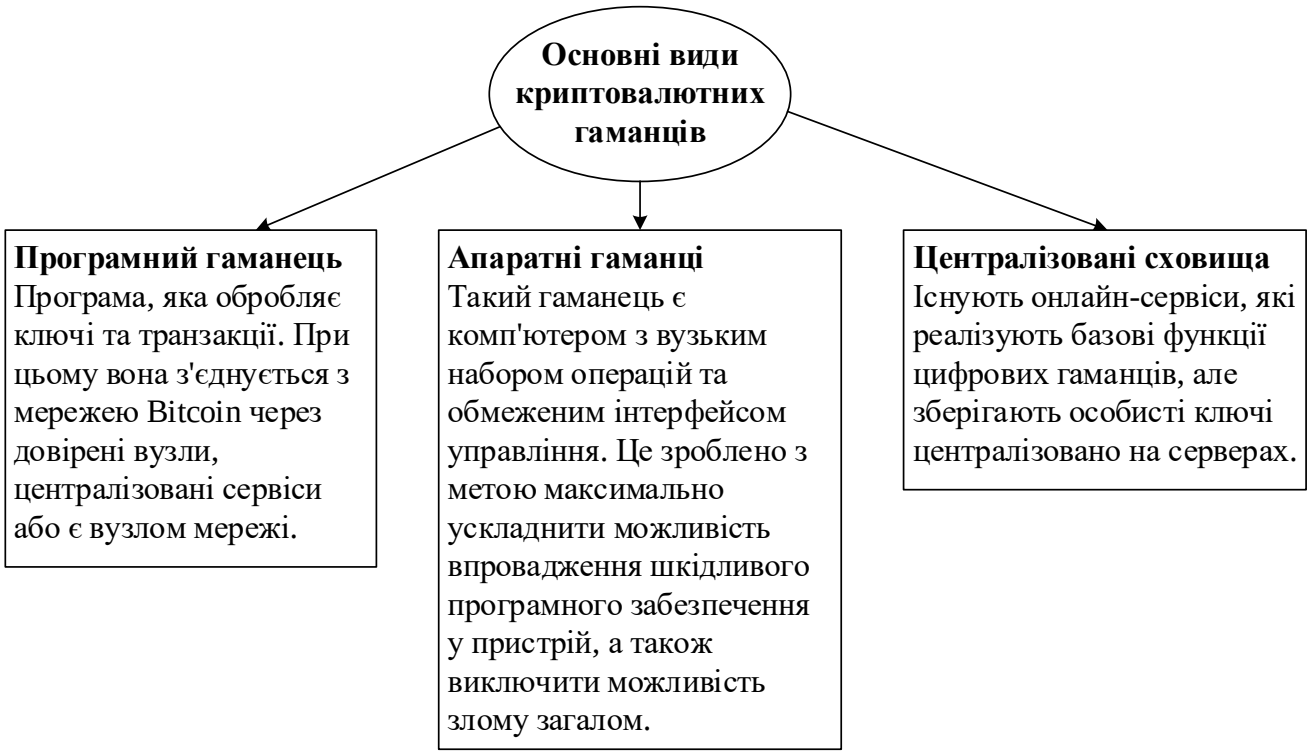

Рисунок 1.14 – Криптовалютні гаманці

Криптовалютні гаманці поділяються на категорії залежно від того, як вони отримують доступ до мережі, на якому носії зберігається гаманець, чи зберігається весь ланцюжок або лише його частини, а також ключі зберігаються локально чи на (хмарному) сервері. На рис. 1.15 наведено короткий як саме відповідальна особа може організувати процеси зберігання особистих ключів та підписання транзакцій.

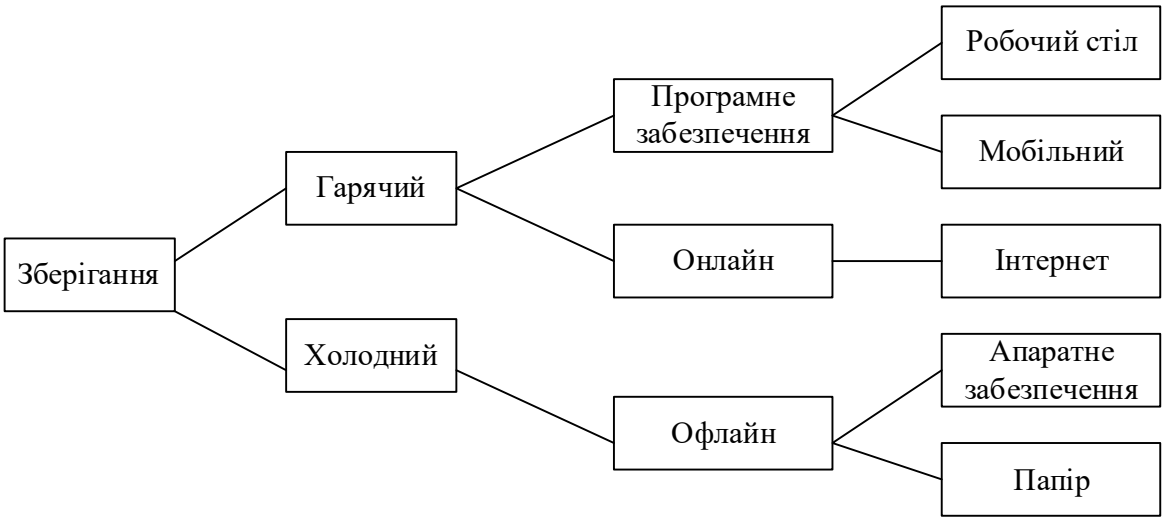

Рисунок 1.15 – Зберігання особистих ключів та підписання транзакцій

Перша відмінність, яка є між гаманцями, — це гарячий чи холодний гаманець. До гарячих гаманців належать усі типи гаманців, які мають підключення до Інтернету. Зазвичай вони мають форму програми або програми, встановленої або доступної на смартфоні чи комп'ютері. Люди можуть використовувати їх як онлайн або веб-гаманці, які є додатками для браузера, які не потрібно завантажувати та встановлювати. Або у вигляді програмних гаманців, які додатково розрізняють настільні та мобільні додатки. Перші – це програми, які встановлюються та використовуються на комп'ютері, а другі – мобільні додатки. Холодні або автономні гаманці не мають доступу до мережі чи Інтернету. Тому вони реалізовані як спеціальний пристрій, який використовується тільки для цієї мети. Цей пристрій ще називають апаратним гаманцем. Іншим типом холодного гаманця буде паперовий гаманець, який просто означає запис необхідної інформації як закритий ключ. Однак у цій дипломній роботі ми не будемо акцентувати на цьому типі гаманця. Загалом, передбачається, що холодні гаманці більш безпечні.

Настільні гаманці – це програмне забезпечення, яке завантажується та встановлюється на комп'ютер. Зазвичай вони надають користувачеві повний контроль над ключами та коштами. Крім того, конфіденційні дані, такі як ключі, зберігаються в локальних файлах на пристрої і зазвичай шифруються. Тому дуже важливо створити резервну копію гаманців на випадок втрати або пошкодження.

Більшість програм пропонують можливість відновлення за допомогою початкової фрази.

Мобільні гаманці схожі на настільні гаманці. Вони встановлюються у вигляді додатків на мобільний пристрій і дають користувачеві керування ключами. Оскільки зламання мобільних пристроїв, крадіжка або втрата, а також порушення безпеки є звичайними, важливо регулярно створювати резервні копії програми. Залежно від використовуваної програми резервна копія зберігається на пристрої або передається на хмарний сервер.

Веб-гаманці часто вважаються найменш безпечним типом гаманця, оскільки вони належать стороннім постачальникам, які відповідають за критичне сховище. Однак перевага у зручності використання полягає в тому, що вони не вимагають встановлення, а отже, доступ до них можна отримати з будь-якого місця. Більш того, їх таким чином не можна втратити і швидко відновити.

Апаратні гаманці, як відомо, є найбільш безпечними, оскільки вони не мають підключення до Інтернету. Однак неможливо відновити гаманець, якщо не матиме додаткового резервного гаманця на випадок втрати. Керування ключами повністю в руках користувача. Цей тип гаманця часто поєднується з програмним забезпеченням або онлайн-гаманцями для здійснення транзакцій.

На рис. 1.16 наведено спрощену ілюстрацію основних компонентів, необхідних для транзакції з використанням криптовалютного гаманця. Сам гаманець не зберігає жодних монет. Однак він зберігає та захищає приватний ключ, який необхідний для здійснення транзакцій і, таким чином, для використання монет. Інша інформація, як геш відкритого ключа, що дорівнює адресі, історії транзакцій та кількості криптовалют, зберігається в Blockchain. Одним зі способів захисту приватного ключа та інших конфіденційних даних, що зберігаються в гаманці, є шифрування. Це може бути поєднане з механізмом блокування, для доступу до програми може знадобитися PIN-код або пароль.

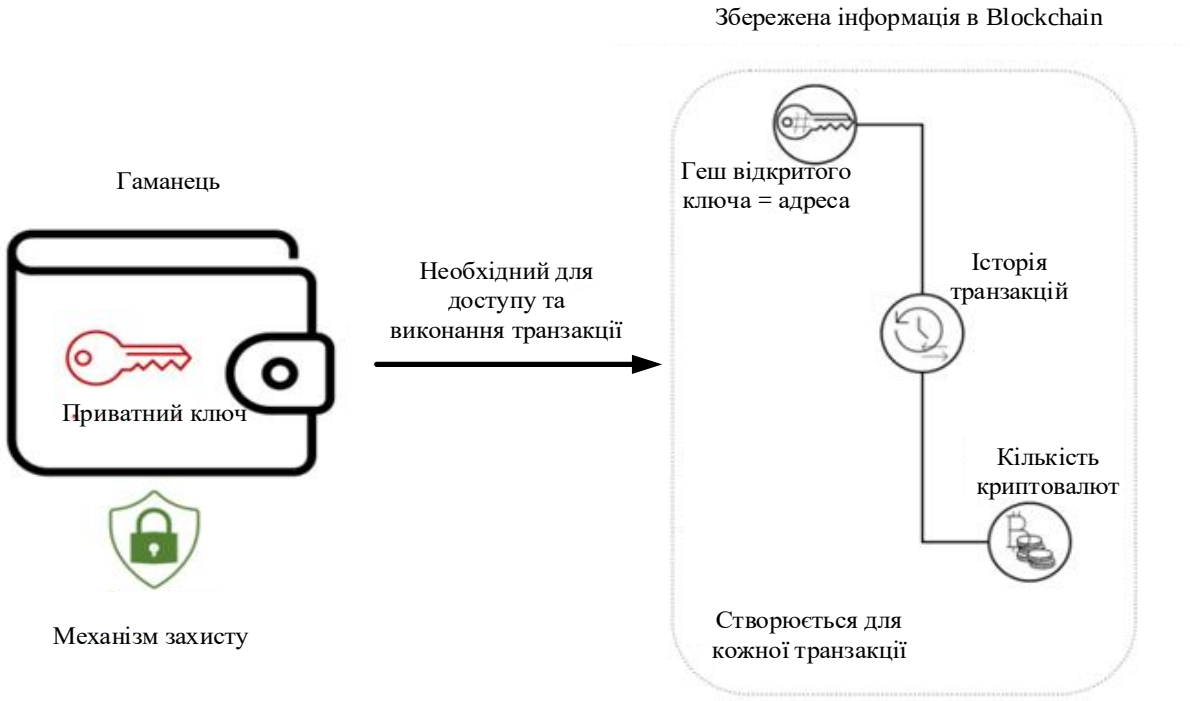

Рисунок 1.16 – Спрощене пояснення ролі гаманців у криптовалютних транзакціях

1.4.2 Інтернет-банкінг на основі децентралізованих систем

Розвиток інтернет-технологій та їх застосування у банківській справі призвело до виникнення нових видів фінансових послуг та зміни структури банківських ринків. В цей час банки розвинених країн просувають послуги інтернет-банкінгу як одного з найважливіших напрямів дистанційного банківського обслуговування клієнтів.

Інтернет-банкінг належить до систем, які дозволяють клієнтам отримувати доступ до рахунків та загальної інформації про банківські послуги, видах обслуговування за допомогою персонального комп’ютера або іншого пристрою.

Поточна банківська система вважається централізованою. Нижче приведені її характеристики:

1. Емісія грошей не задана на програмному рівні, а є централізованим рішенням кількох груп людей;

2. Гроші у банку не належать вам, вони належать банку, а ви просто маєте до них доступ;

3. Банк може просто заборонити переміщати кошти (наприклад, блокування рахунку);

4. Система не прозора, ми не знаємо, скільки грошей буде надруковано завтра, і навіть зараз багато людей не знають, скільки вже надруковано, іноді є відкрита інформація про це, але перевірка її справжності часто проблематична. Ми можемо перевірити інформацію на сайті, але фактично перевірити не вдасться, тому що в нас немає доступу до баз даних банку безпосередньо.

Якщо взяти вище перечисленні пункти, то можна описати проблеми чинних платіжних систем, а саме те, що користувач не контролює свій баланс, кошти можуть бути зараховані на рахунок одержувача через кілька днів, а іноді кілька десятків днів після їх відправлення, платежі можуть бути скасовані у процесі виконання, кошти можуть бути заблоковані.

Як наслідок виникла потреба реалізувати систему, в якій люди зможуть незалежно і повністю самостійно керувати своїми грошима. Рішенням став Bitcoin.

Bitcoin – децентралізований протокол, який описує, як відбувається комунікація між учасниками децентралізованої мережі та які властивості мають різні сутності мережі (наприклад, задається кількість монет та правила їх емісії).

Альтернатива банківської системи – це криптовалюта.

Основні пункти пов’язані з криптовалютою:

1. Емісія грошей задана на рівні коду, всі знають, скільки буде цифрових одиниць обліку та за якими правилами вони створюються.

2. Криптовалюта належить тільки вам, якщо у вас є від неї приватний ключ або seed-фраза;

3. Ніхто неспроможний заборонити переміщати кошти;

4. Правила мережі задані в протоколі, поміняти їх можна лише голосуванням, і воно буде прозорим;

5. Помилки окремих осіб не призводить до краху системи, тому що кожна особа має таку ж владу, як і інша особа, тобто криптовалюту.

6. Ви маєте право витрачати цифрову валюту в мережі Інтернет скрізь, де вона застосовується;

7. Система повністю прозора, тобто ви знаєте, скільки монет створено на поточний момент, ви можете переглянути всі транзакції та переглянути всі інші параметри, які доступні та необхідні вам.

1.4.3 Принципи формування смарт контрактів

Смарт-контракт – це угода про перерозподіл цінностей між контрагентами, яка має на увазі суворе та однозначне завдання умов, автоматизацію процесів виконання та мінімізацію залучення довірених сторін. Найчастіше це досягається за допомогою об’єднання механізмів завдання умов і механізмів суворого виконання, для чого зазвичай застосовують комп’ютерні протоколи, що описують порядок створення, обробки та реалізації смарт-контрактів. Система під управлінням такого протоколу називається платформою смарт-контрактів, де контракти міститься у спеціальні транзакції (або запити), які мають бути підписані всіма залученими до угоди сторонами. Після того, як таку транзакцію було передано на платформу смарт-контрактів, змінити умови контракту, порушити або скасувати їх виконання вже неможливо. Смарт контракти формуються за способом завдання і виконання умов (рис. 1.17)

Рисунок 1.17 – Формування смарт-контрактів за способом завдання і виконання умов

Що стосується повних за Тьюрингом контрактів, перевага такого підходу полягає у більшій гнучкості при програмуванні смарт-контракту.

Недоліком є неможливість гарантувати безпеку платформи, оскільки існують ситуації, за яких виконання смарт-контракту, написаного повною за Тьюрингом мовою програмування, може бути непередбачуваним.

До довільних повних за Тьюрингом контрактів відносять платформу Ethereum, RootStock, EOS, Cardano.

Оракули транслюють дані із зовнішнього середовища до децентралізованої облікової системи. Найчастіше оракул має identity та вимагає деякої довіри користувачів до даних, які він надає. Усі дані, які передає оракул, мають бути підписані цифровим підписом, щоб забезпечити цілісність даних та неможливість відмови (кожен користувач системи може перевірити, чи конкретні дані передав певний оракул).

Можна мінімізувати необхідність довіри користувачів шляхом збільшення кількості оракулів (припустимо, щоб запобігти корупції). Тоді смарт-контракт отримуватиме відомості від безлічі незалежних оракулів і погоджуватиме їх, після чого прийматиме деяке рішення.

Смарт-контракти сприймаються як перспективні та багатообіцяючі технології для впровадження в існуючу регульовану інфраструктуру.

Почнемо з того, що ми говоримо про регульоване середовище, в якому відповідальність за ризики несуть конкретні сторони. Перехід в епоху цифрової взаємодії вимагає застосування максимально надійних технологій, які забезпечать учасникам системи впевненість у тому, що дані, що вони бачать, є справжніми. Інформація, яка зберігатиметься в електронних реєстрах, має бути захищена від несанкціонованих змін.

За рівнем приватності смарт-контракти можуть бути повністю відкритими, або частково конфіденційними, або повністю конфіденційними (сторонні спостерігачі не можуть бачити умови смарт-контрактів). За способом ініціації смарт-контракти також можна розділити мінімум на дві групи: автоматизовані та ручні (неавтоматизовані). Для автоматизованих характерно, що при всіх відомих

параметрах і умовах, смарт-контракт повністю виконується автоматично, тобто не вимагає відправки якихось додаткових транзакцій і витрати додаткової комісії при кожному наступному виконанні.

Найбільш часто використовувані шаблони смарт-контрактів включають наступне:

1. Тимчасове блокування (підтвердження результату відкладено на певний термін).

2. Блокування по геш (активи можна розблокувати, тільки якщо надати прообраз конкретного геш-значення).

3. Тимчасове блокування по геш (активи можуть бути розблоковані, якщо прообраз геш-значення буде надано протягом заданого часового інтервалу).

4. Мультипідпис та її модифікації.

5. Атомарний обмін (обмін цифровими активами, що враховуються в різних системах, між двома користувачами, з гарантією виконання угоди повністю або повної її скасування).

Впровадження смарт-контрактів пов’язане з великою кількістю очікувань із боку бізнесу. Створення умов для функціонування смарт-контракту є необхідним кроком у напрямку автоматизації бізнес-процесів. А все-таки, ми повинні підкреслити поняття соціального консенсусу. Це поняття може вплинути на очікувані результати смарт-контракту до зміни стану бази даних.

Смарт-контракт може бути адаптований у будь-яких середовищах не лише за допомогою Blockchain. Оскільки концепція Blockchain спрямована на видалення стороннього посередника для транзакцій. Традиційно ця третя сторона несе відповідальність за підтримання та виконання контрактів та побудову довіри між усіма залученими сторонами. Наприклад, якщо ви купуєте продукт в Інтернеті, сторонній банк у цьому випадку переконається, що платіж надіслано на правильний рахунок, платіж виконаний успішно, та запит на повернення може бути узгоджений між зацікавленими сторонами. Зважаючи на те, що ризик мережі Blockchain полягає в тому, що якщо ви купуєте товар, наприклад, з використанням Bitcoin, після здійснення платежу немає гарантії, що ви отримаєте куплений товар.

Оскільки інша залучена сторона може ухвалити рішення не доставляти придбаний продукт або претензія не отримана. Крім того, Bitcoin-гаманці анонімні, тому ви можете не знати, куди саме було відправлено транзакцію. Таким чином, інтелектуальний контакт є важливою частиною концепції Blockchain для побудови транзакцій, не пов'язаних з довірою, і для вирішення безлічі ризиків, пов'язаних із перетином мережі Blockchain.

1.5 Висновки до розділу

Якщо підбивати та робити висновки, то технологія blockchain може змінити спосіб обміну цінністю. Попри те, що технологія знаходиться в зародковому стані, все більше людей починають усвідомлювати її потенціал змінити світ. Більшості людей досі важко зрозуміти, чому blockchain настільки значущий, заявляючи, що нинішні системи проведення фінансових транзакцій, здається, працюють досить добре. Blockchain здатний усунути потребу в багатьох роботах у секторі фінансових послуг, які зараз виконують люди. Послуги з фінансового посередництва та страхування можуть виконуватися на blockchain без необхідності посередника. Посередництво в спорах та врегулювання кредитних вимог стали б нескінченно ефективнішими, якби необхідна інфраструктура була побудована з використанням технології blockchain. Короткострокові наслідки можуть означати, що цілий ряд людських ресурсів і спеціалізацій більше не будуть потрібні, і люди побачать, що цінність своїх навичок швидко зменшується.

Децентралізовані системи та технологія Blockchain може забезпечити надійне зберігання даних та наділити облікову систему рядом цінних властивостей. Однак складність проектування, налаштування та підтримки децентралізованої облікової системи, як правило, набагато вища, ніж для централізованих альтернатив, тому необхідно усвідомлено підходити до цього питання. Варто враховувати той факт, що ці переваги не завжди гарантовані на 100%.

2 АНАЛІЗ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ТА АВТЕНТИЧНОСТІ В ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМАХ

2.1 Дослідження механізмів забезпечення конфіденційності

Останнім часом Blockchain привертає багато уваги. Попри те, що різні функції Blockchain-технологій надали нам зручні та надійні послуги, проблеми безпеки та конфіденційності, як і раніше, викликають занепокоєння. Різні автори проводили дослідження з питань безпеки та конфіденційності технології Blockchain, але необхідне докладне та систематичне дослідження, яке намагатиметься охопити всі важливі аспекти.

Технологія Blockchain дає формування електронної книги і для забезпечення безпеки, а саме конфіденційності та цілісності використовується саме її розподіл на всіх повних вузлах.

Вузли Біткойна взаємодіють один з одним через мережевий протокол Bitcoin P2P, тим самим вони гарантують цілісність системи. Вузол, який неправильно використовує або намагається розповсюджувати неправильну інформацію.

Попри те, що запуск повного перевіряючого вузла не дає фінансових винагород, він настійно рекомендується, оскільки він забезпечує довіру, та конфіденційність та захист для користувачів. Крім того, повний вузол не повинен довіряти іншим, і він дозволяє користувачеві повністю контролювати свої гроші.

Вузли виконують три основні типи функцій; виявлення та ретрансляція транзакцій, оновлення Blockchain новими блоками транзакцій (консенсус) і ретрансляція блоків транзакцій.

Блокчейни можна розділити на різні підкатегорії залежно від того, чи потрібна авторизація мережевим вузлам, щоб вони діяли як верифікатори, і від того, чи є доступ до даних блокчейна публічним чи приватним. Перша категорія полягає в тому, чи дозволено процес перевірки та консенсусу:

- блокчейни без дозволу, будь-хто може налаштувати вузол, підключитися до мережі та взяти участь у процесі перевірки;

- дозволені блокчейни, де привілеї майнінгу делегуються центральним органом або консорціумом.

Друга категорія полягає в тому, чи є книга публічною чи приватною:

- публічні блокчейни — це блокчейни, де кожен може отримати копію книги та ініціювати транзакції;
- приватні блокчейни — це блокчейни, дозвіл на які обмежено користувачами в організації або організації.

У загальнодоступних блокчейнах будь-хто може приєднатися, підключившись до одного або кількох вузлів і транслюючи трансакцію. Коли користувач здійснює транзакцію, кожен вузол-одержувач передає транзакцію своїм з'єднанням, доки всі вузли не отримають копію транзакції. Нові блоки створюються, коли деякі або всі вузли збирають транзакції в блоки транзакцій з мітками часу, які потім транслюються через мережу. Консенсус встановлюється, коли всі вузли або більшість вузлів отримають дійсний блок транзакцій, який додається до попередніх блоків у блокчейні. Кожен новий блок має цифровий підпис і містить підпис попереднього блоку. Пов'язані цифрові підписи гарантують цілісність транзакцій, зареєстрованих у блокчейні, і нема потреби підтримувати центральну копію.

Як зазначалося раніше, всі розподілені реєстри вимагають підтвердження транзакцій, після чого має бути досягнутий консенсус. Однак Blockchain об'єднують перевірку та консенсус у процесі, відомому як майнінг (майнінг = перевірка транзакцій + консенсус). Процес підтвердження транзакцій називається майнінгом, а люди, які виконували верифікацію, називаються майнерами. Майнінг використовується для захисту та перевірки транзакцій (незалежно від того, пов'язані вони з криптовалютами, такими як Bitcoin, або з будь-якими іншими записами). У майнінгу беруть участь майнери, які додають дані про транзакції до глобального журналу завершених транзакцій. У цих книгах блоки захищені майнерами та з'єднані між собою, утворюючи ланцюжок. Приклад ланцюжка блоків показано на рис. 2.1.

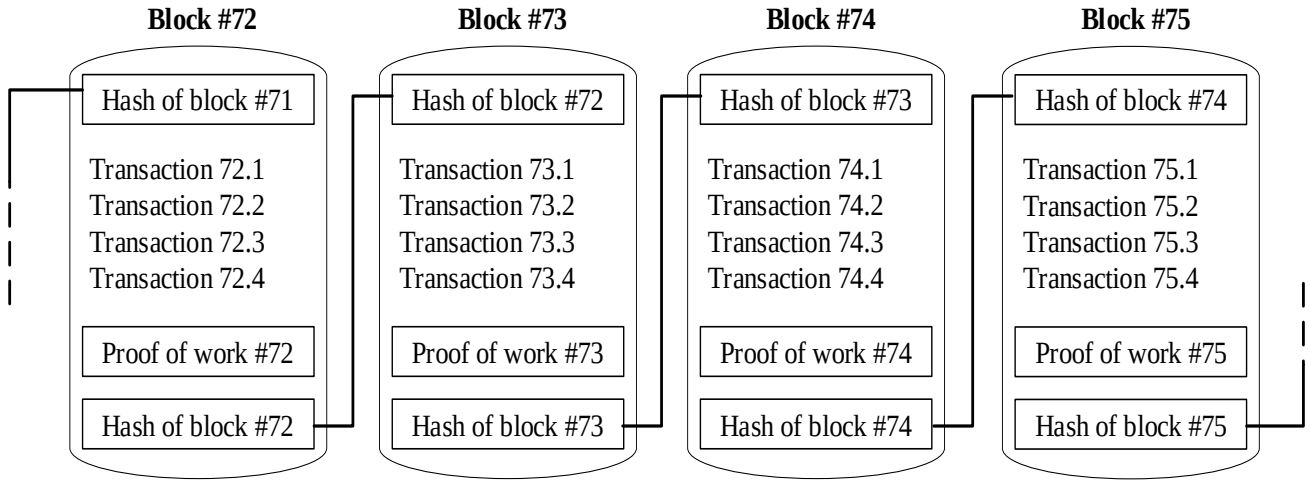

Рисунок 2.1 – Схема ланцюжка блоків

У різних Blockchain використовуються різні механізми консенсусу. Деякі Blockchain, такі як Bitcoin, загальнодоступні; це означає, що будь-хто може придбати обладнання, під'єднатися до мережі і почати “майнінг”. Інші вимагають, щоб учасники процесу консенсусу виконували деякі заздалегідь визначені вимоги, сформульовані розробниками-засновниками. Майнери обчислюють криптографічний підпис чи водяний знак останнього блоку транзакцій. Bitcoin використовує стандартні геш-функції як водяні знаки, де найдовший ланцюжок блоків транзакцій із дійсними гешами забезпечує консенсус. Майнери запитують нові Bitcoin у спеціальній транзакції, яка називається транзакцією coinbase. Кожен існуючий Bitcoin можна простежити до транзакції з базою монет.

Криптографічні геші, відіграють ключову роль у забезпеченні безпеки реєстру Bitcoin, Blockchain. Кожен дійсний новий блок транзакції містить геш попереднього блоку в Blockchain. Новий блок n пов’язаний з попереднім блоком n-1. Наступний дійсний блок n+1, своєю чергою, включатиме посилання на n (рис. 2.2).

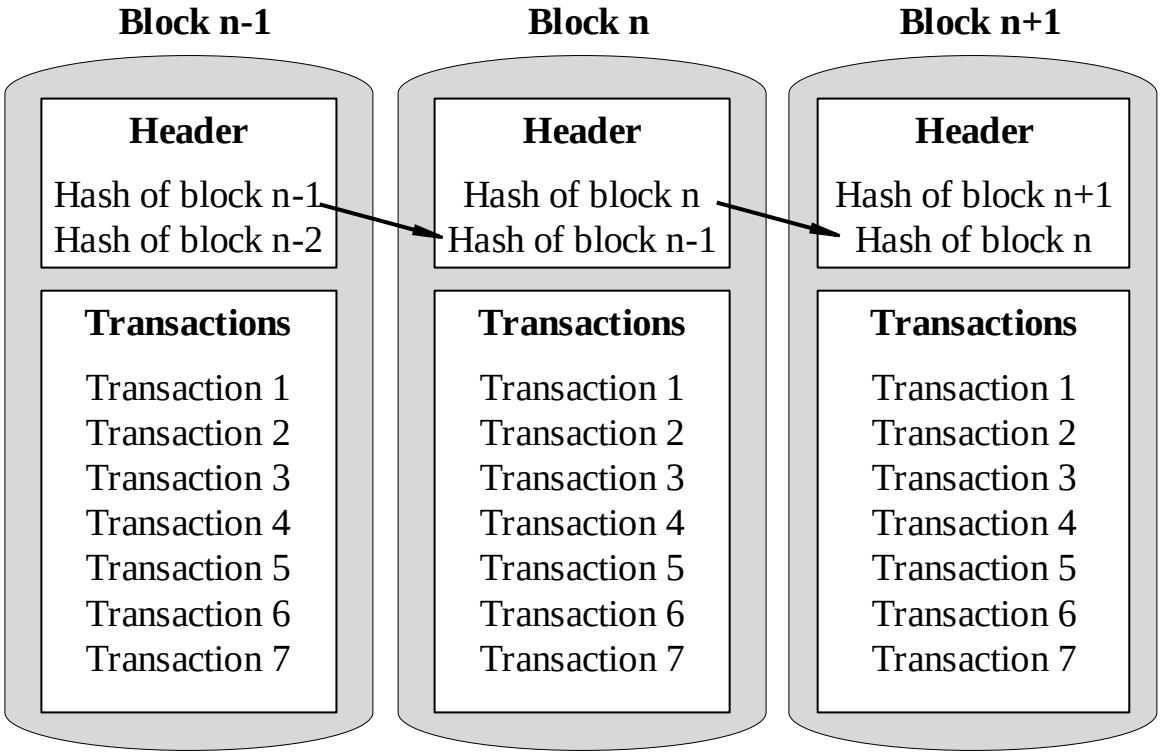

Рисунок 2.2 – Реєстрація транзакцій Bitcoin із взаємопов’язаними блоками транзакцій

Тому будь-яка спроба змінити транзакцію в ланцюжку блоків вимагає перерахунку не тільки гешу для цільового блоку, але і всіх наступних гешей блоків. Якщо зловмисник не управляє 51% або більше майнерів, йому практично неможливо перерахувати альтернативний Blockchain зі зміненою транзакцією. Пов’язані геші гарантують цілісність ланцюжка блоків та усувають необхідність у довіреному центрі для ведення центрального реєстру транзакцій.

Усі розподілені книги вимагають підтвердження транзакцій, після чого має бути встановлений консенсус.

Різні блокчейни використовують різні механізми консенсусу. Деякі блокчейни, як-от біткойн, є загальнодоступними; це означає, що будь-хто може придбати обладнання, під'єднатися до мережі та почати “майнінг”. Інші, однак, вимагають від учасників процесу консенсусу виконати деякі заздалегідь визначені вимоги, встановлені основними розробниками-засновниками.

Протокол “докази роботи”:

Proof-of-Work: блокчейн Bitcoin, використовує процедуру під назвою Proof-of-Work (PoW) для перевірки транзакцій і створення нових блоків.

Спочатку PoW призначався як економічний захід для запобігання розподілених атак відмови в обслуговуванні (DDoS) та інших зловживань послугами, таких як спам, у мережі. Він досягає цього, вимагаючи виконання певної “роботи” від запитувача послуг, що зазвичай вимагає часу на обробку комп’ютером. Ланцюжок блоку наведений на рис. 2.3.

Рисунок 2.3 – Proof-of-Work

“Робота”, яку зобов’язаний виконати запитувач послуг, вимагає витрат капіталомістких ресурсів, таких як обчислювальна потужність, що підвищує альтернативні витрати на участь у шахрайській або неетичній поведінці. Однак це являє собою компроміс між безпекою даних і марнуванням капітальних ресурсів. Тому робота не повинна бути ні занадто важкою, ні занадто легкою. Якщо

необхідна робота занадто важка, це створює неефективність, оскільки робота може бути простішою і все одно давати ті ж результати. Однак, якщо необхідна робота надто проста, система не виконує своєї мети, що призводить до неефективного використання ресурсів. Наразі всі публічні блокчейни покладаються на певну форму перевірки Proof-of-Work і певний тип процесу консенсусу.

Однак головна проблема PoW полягає в тому, що це неймовірно енергомісткий процес. Було викладено багато потенційних рішень, які спрямовані на зниження потреби в енергії для підтримки блокчейну. Найбільш перспективною заміною PoW є концепція Proof-of-Stake.

Протоколи "доказу частки"

Proof-of-Stake (PoS): це алгоритм для досягнення розподіленого консенсусу в мережах блокчейн. Proof-of-Stake був запропонований як потенційна заміна Proof-of-Work і призначений для розв'язання проблеми неефективного використання капітальних ресурсів, таких як обчислювальна потужність та енергія. Основна ідея Proof-of-Stake полягає в тому, щоб розподілити привілеї майнінгу залежно від того, скільки "ставок" має учасник у мережі (рис. 2.4).

Рисунок 2.4 – Proof-of-Stake

Було запропоновано багато різних версій системи Proof-of-Stake, де привілеї майнінгу залежать не тільки від власності на валюту, але й від інших факторів. До таких факторів можна віднести частоту транзакцій і те, як довго учасник був частиною мережі. На сьогодні ніхто успішно не створив системи, яка повністю базується на Proof-of-Stake. Однак більшість експертів галузі погоджуються, що це лише питання часу, коли перший успішний блокчейн PoS побачить світ. У разі успіху Proof-of-Stake може істотно зменшити кількість енергії, необхідної для підтримки блокчейн-мереж, і, за інших рівних умов, створити тиск на зниження цін на електроенергію.

Механізм перевірки та консенсусу Proof-of-Work гарантує, що для маніпулювання транзакцією, що зберігається в блокчейні, потрібно контролювати мінімум 51% або більше загальної обчислювальної потужності (геш-потужності) в мережі.

Для розвитку взаємної довіри Blockchain працює з допомогою інтеграції механізму розподіленого консенсусу. У цьому механізмі обчислювальна потужність розподіляється між усіма доступними майнерами даних. Робота цих майнерів даних полягає у перевірці гешей, згенерованих циклами процесора. Якщо ці майнери об’єднаються, вони можуть стати великим майнінговим пулом із максимальною обчислювальною потужністю. Якщо майнінговий пул має 51% або більше обчислювальних потужностей, вони можуть взяти під контроль Blockchain та створити серйозні ризики для безпеки. Наприклад, у Blockchain на основі доказу роботи (POW), якщо потужність гешування одного майнера на 50% більше, ніж загальна потужність гешування всієї системи Blockchain, то майнер може легко запустити атаку 51% і викликати вразливість, як:

- атаки зі зворотною транзакцією;
- двойні витрати;
- виключити транзакції;
- змінювати транзакції;
- тривожні операції інших майнерів;
- завершення процесу перевірки.

В інших прикладах, один майнер, який працює над Blockchain на основі proof of stake, може мати 50% від загальної кількості монет, може почати атаку на 51% і може змінювати та використовувати інформацію системи Blockchain.

Delegated Proof-of-Stake (DPoS): це механізм, який дозволяє ефективно створювати блоки та обробляти більше транзакцій за секунду, порівняно з іншими алгоритмами консенсусу, шляхом скорочення кількості валідаторів При голосуванні власники монет мають можливість вибирати валідаторів транзакцій, які формують блоки. Власники монет можуть голосувати за кандидатів у будь-який момент часу. Це забезпечує високу стійкість мережі: якщо більшість виконавців зазнають невдачі, спільнота негайно проголосує за їхню заміну (рис. 2.5).

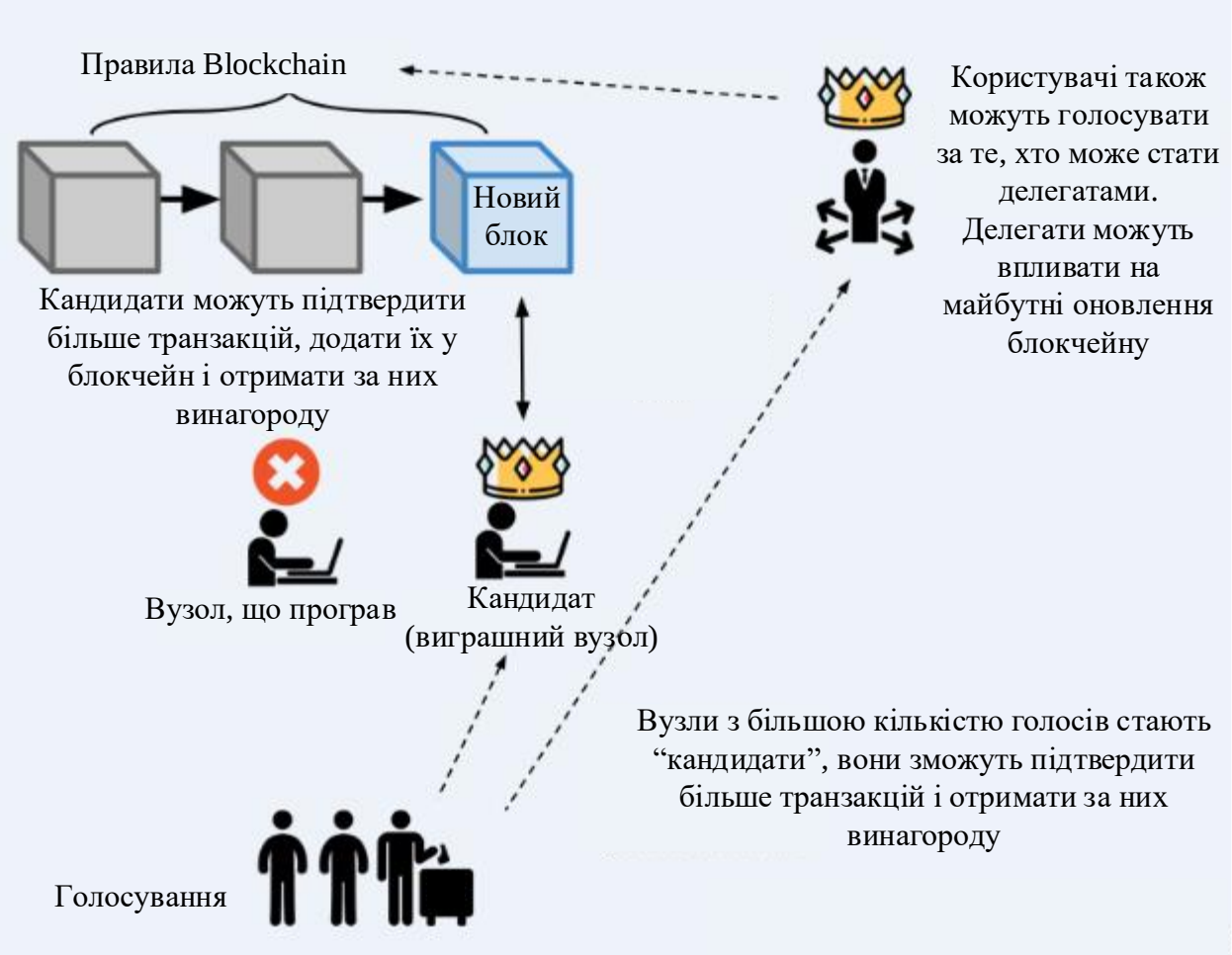

Рисунок 2.5 – Delegated Proof-of-Stake

У цьому протоколі нові блоки генеруються кожні 1-2 секунди. Цей протокол не лише швидший, але й справедливіший, оскільки "делегований" валідатор поділяє свої токени з власниками, які його обрали. Однак, підтвердження готових блоків все ще залежить від інших учасників мережі.

Byzantine Fault Tolerance (BFT) протоколи:

Delegated Byzantine Fault Tolerance (DBFT): Система продовжуватиме працювати навіть у випадку, коли вузол переходить у режим офлайн. Тим самим, протокол консенсусу BFT виглядає як рятувальний засіб від недоліків протоколів PoW та PoS. Однак, навіть з великою кількістю валідаторів, він все ще змагається з проблемою швидкості. Саме тому розробники запропонували делеговану модель BFT - DBFT.

Визнані валідатори цього протоколу консенсусу значно випереджають інші протоколи, такі як Ethereum з 15-20 транзакціями в секунду або NEO з майже 10 000 т/с. Зручно мати кілька довірених сторін, які перевіряють транзакції перед випуском інших вузлів. У випадку, якщо валідатор "відмовляє" або недоступний, учасники можуть делегувати новий вузол. Незважаючи на те, що цей протокол розрахований на громадське середовище, він все ж є дещо централізованим (рис. 2.6).

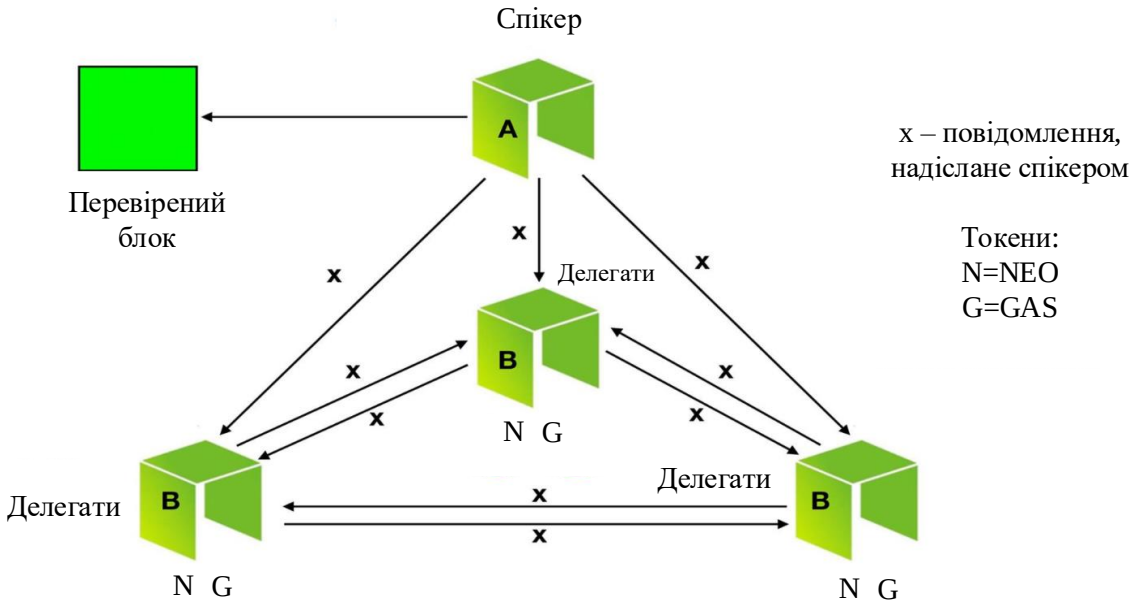

Рисунок 2.6 – Delegated Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance: Протокол практичного виправлення вад у випадку відмов (PBFT) дуже подібний до DBFT, особливо за їх централізованим характером. Однак, єдине відмінність полягає в тому, що PBFT має простішу реалізацію і часто використовується в приватних середовищах з відомими учасниками.

Коли валідатор отримує повідомлення, він зобов'язаний прийняти рішення - вважати його достовірним або недостовірним. Для цього валідатор перевіряє повідомлення самостійно, а потім запитує всі інші вузли послідовно, що вони думають про цю транзакцію. Якщо 2/3 учасників підтримують цю транзакцію, валідатор приймає її і передає своє рішення до мережі для інших валідаторів. Таким чином, досягається консенсус на основі підтвердження, яке представлене всіма валідаторами. Точна структура мережі залежить від механізму консенсусу, наприклад у PBFT є лідери, які підтверджують і не підтверджують peer (рис. 2.7).

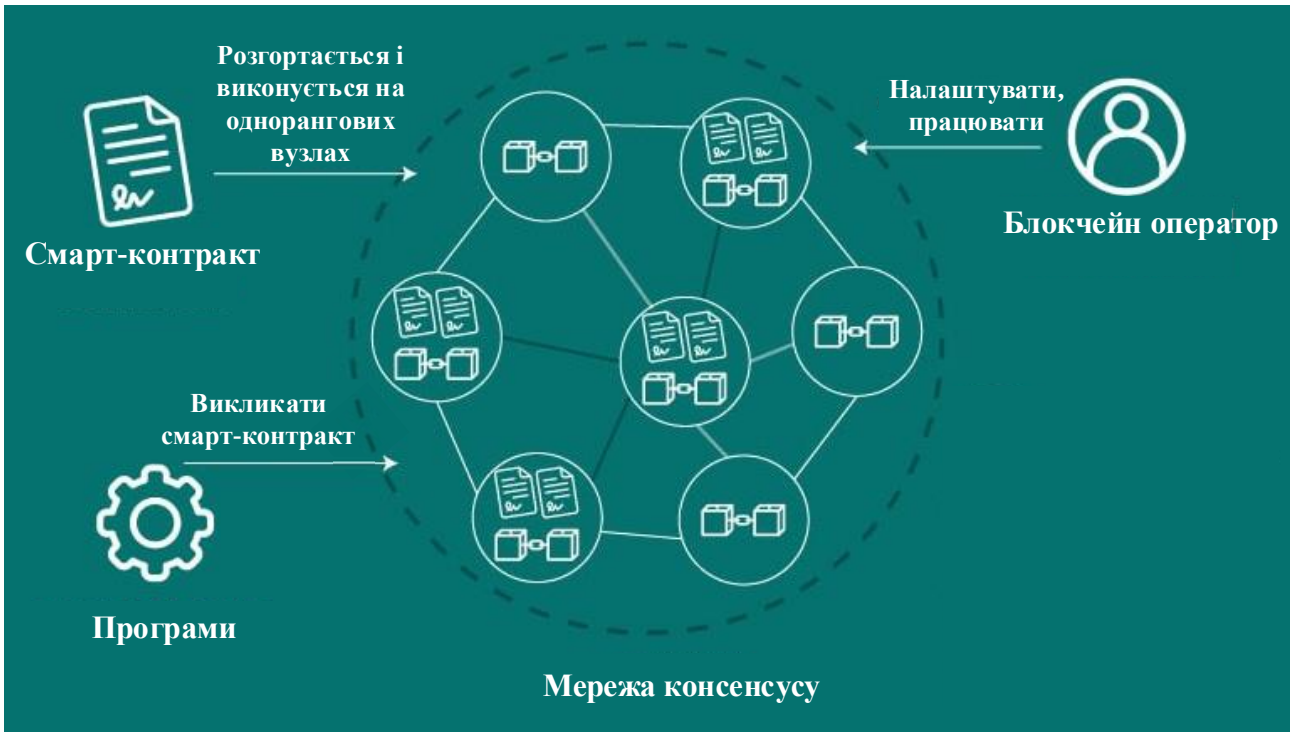

Рисунок 2.7 – Practical Byzantine Fault Tolerance

Потік повідомлень консенсусу між відповідними одноранговими партнерами, щоб забезпечити належне підтримання транзакції смарт-контракту блокчейну; світовий стан зберігається послідовним протягом відтворення локальної транзакції.

PBFT є ефективним у системах з низькою затримкою, але вразливим до кількості валідаторів та пропускної здатності, оскільки кожне повідомлення генерує багато запитів та перевірок. Він ідеально підходить для приватних середовищ, де немає потреби великого навантаження, але є потреба у великій кількості транзакцій. PBFT гарантує остаточність рішень щодо транзакцій у мережі, оскільки вони приймаються абсолютною більшістю в будь-який момент часу.

Federated Byzantine Agreement (FBA) не потребує заздалегідь відомого набору учасників, на відміну від PBFT та інших варіацій BFT. FBA дозволяє будь-кому приєднатися до мережі. У цьому протоколі транзакції валідуються фіксованою кількістю учасників, які обираються з поточних учасників мережі. Використовуються два набору даних (рис. 2.8): глобальний (G), однаковий для всіх партнерів, використовуваний для контролю цілісності моделей, і локальний (L), окремий для всіх партнерів, який використовується для оцінки ефективності. При більшості голосів модель кандидата приймається або немає.

За правилами FBA існують Gateways (шлюзи) та Market-Makers (мейкери), які забезпечують чесність та ліквідність мережі. Шлюзи виступають як банки, що управляють фінансовими коштами та створюють віртуальні токени. Мейкери підтримують облікові записи з різними шлюзами та валютами.

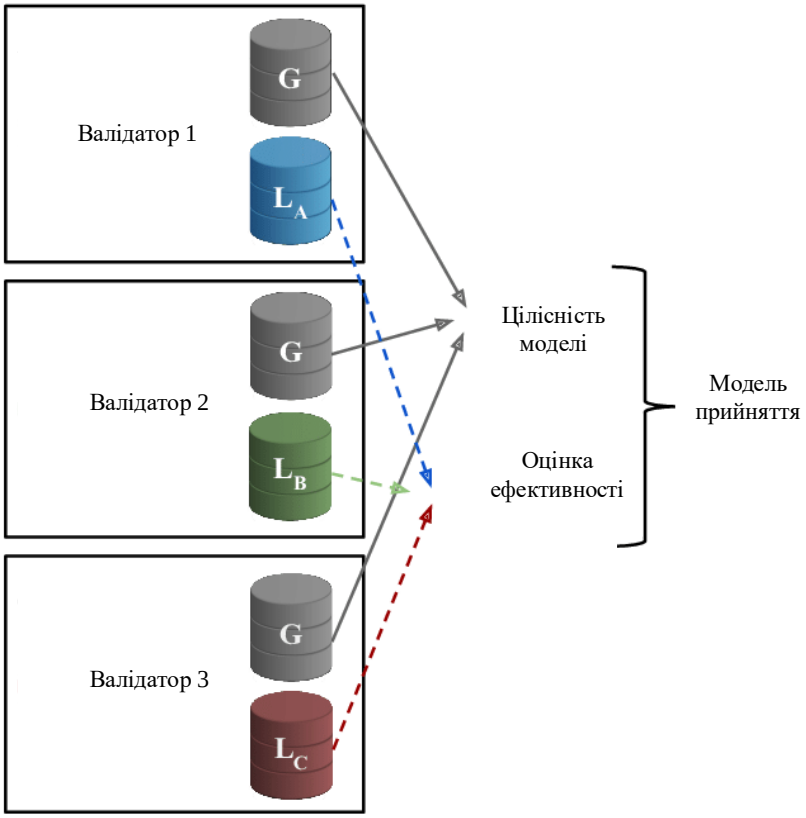

Рисунок 2.8 – Federated Byzantine Agreement

“Неблокчейни”:

Directed Acyclic Graph (DAG): Блокчейни не можуть бути паралельними. Існує можливість змінювати розмір або частоту блоків, а також учасників, які їх валідують, проте це призведе до строго лінійного характеру всієї історії подій. Натомість, технологія Directed Acyclic Graph (DAG) є асинхронною і надає конкурентну перевагу для одночасних подій. Результивний граф має бути ациклічним, що означає, що будь-який заданий кортеж обробляється оператором лише один раз. Винятком є ітеративна обробка, також підтримувана Apex, коли вихідні дані оператора стають вхідними даними попередника (або висхідного оператора), створюючи цикл у графі щодо потоків (рис. 2.9).

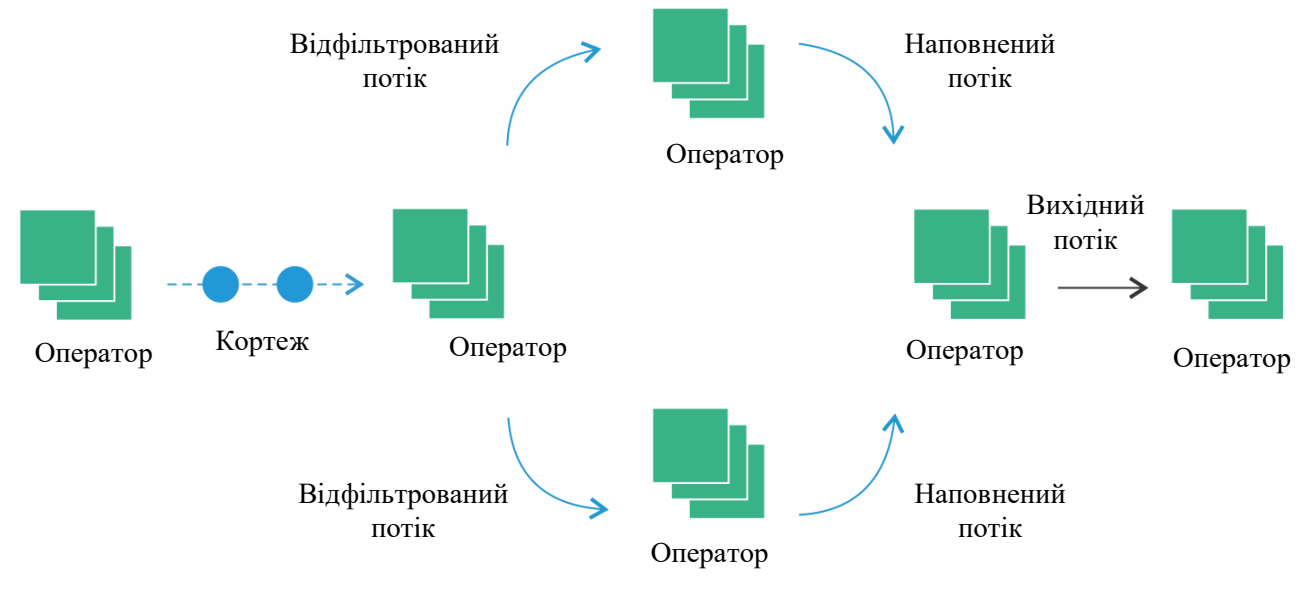

Рисунок 2.9 – Directed Acyclic Graph

У таких системах протокол дозволяє учасникам додати один блок транзакцій і підтвердити кілька попередніх блоків. Це означає, що "швидкість підтвердження старих блоків залежить від кількості нових транзакцій". Хоча це призводить до високої швидкості для мережі, технологія DAG виявляється повільнішою на менших масштабах.

HashGraph: розробники цього протоколу стверджують, що блокчейн є застарілою системою, і пропонують DAG як його альтернативу. Однак основною відмінністю HashGraph є використання протоколу "gossip about gossip", де вузол отримує набір транзакцій з відомим часовим позначенням, про які він вже знає від інших вузлів. Для працездатності цього алгоритму всі учасники мережі повинні бути взаємно відомі. Завдяки синхронізації кожен вузол зберігає повну інформацію та історію отримання цієї інформації всіма вузлами мережі. Як тільки вузол бачить у своїй історії, що певне повідомлення вже було отримано та підтверджено більшістю вузлів, він не сумнівається у його правдивості.

Прорив гешграфу полягає в його консенсусному протоколі. Математично доведено, що він допомагає реплікувати дані набагато швидше, ніж блокчейн. На абстрактному рівні гешграф складається зі стовпців і вершин (рис. 2.10).

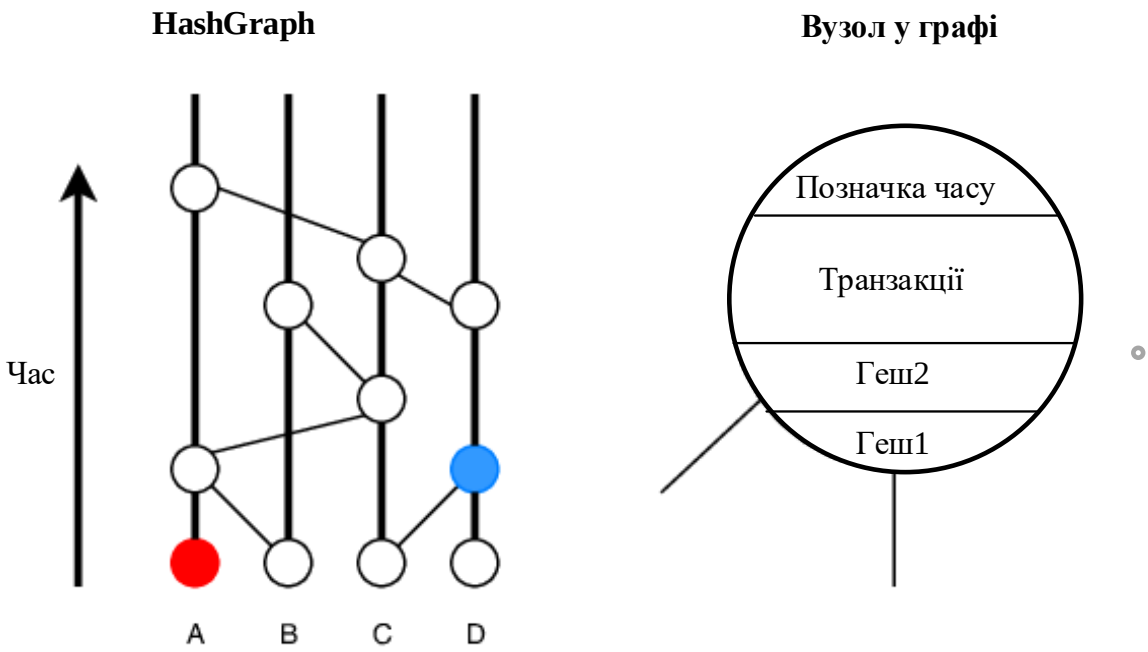

Рисунок 2.10 – HashGraph

Кожен стовпець представляє користувача в мережі, а вершини – події. Користувачі в основному можуть виконувати дві дії в гешграфі:

- подати транзакцію;
- інформація про транзакцію.

Кожна подія містить чотири частини інформації. Геш 1 – це геш попередньої події, створеної користувачем, який отримує інформацію, а Геш 2 – це геш попередньої події користувача, який надсилає інформацію. Транзакція містить будь-які транзакції, надіслані користувачем, який надсилає інформацію, а поле міток часу використовується для відстеження того, коли подія була подана. Зауважимо, що навіть якщо користувач спробує підробити часові позначки, алгоритм голосування, який використовується для досягнення консенсусу, виявить це.

Інші протоколи консенсусу для конкретних завдань:

Proof-of-Activity (PoA): поєднує протоколи PoW та PoS, що означає, що учасники можуть як майнути, так і закладати частку для валідації блоків. Отже, протокол PoA забезпечує баланс між майнерами та звичайними учасниками мережі (рис. 2.11).

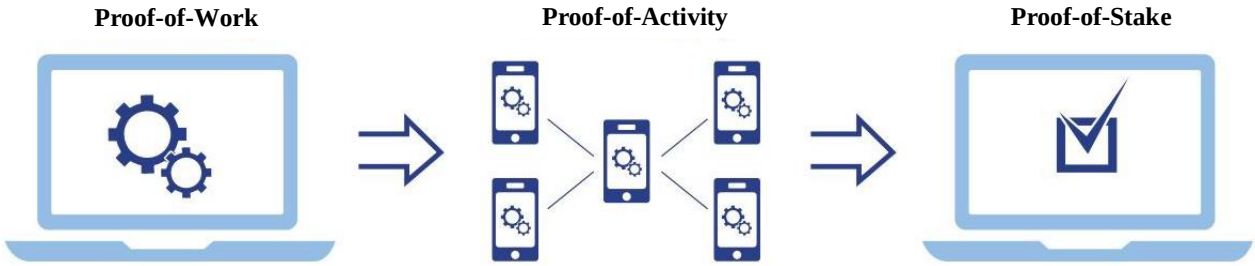

Рисунок 2.11 – Proof-of-Activity

Proof-of-Location (PoL): дозволяє користувачам закріпити за собою конкретну GPS-локацію і таким чином автентифікувати себе в мережі. Цікаво те, що протокол спирається на BFT маячки (beacons), які записують геолокацію та маркери часу в блокчейні, що запобігає збоям та шахрайству в системі.

Протоколи PoL покладаються на процедуру, яка вставляє проміжних посередників між користувачами. Проміжні посередники - це пристрої, які виконують протокол. Однак, хоча пристрої належать фізичним користувачам, вони повинні бути з ними на момент створення підтвердження. В результаті, щоб забезпечити повний PoL для користувача пристрою, необхідно встановити три кореляції (рис. 2.12).

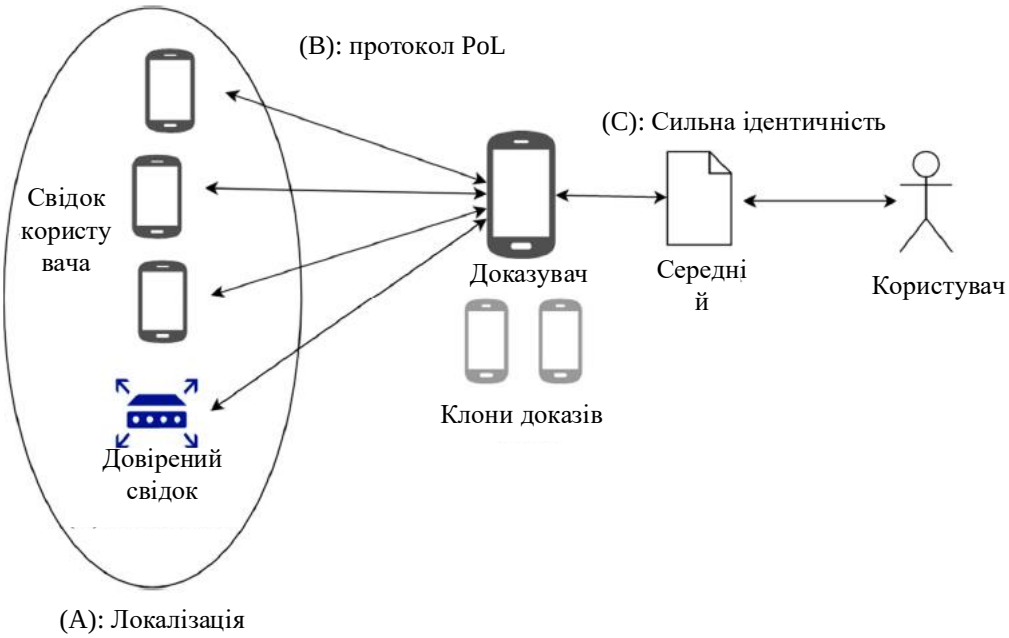

Рисунок 2.12 – Proof-of-Location

1. Кореляція A пов’язує свідків з певним місцем. Його задовольняють або надійні пристрої з відомим місцеперебуванням, або пристрої інших користувачів, які визначають своє положення за допомогою методу локалізації, наприклад GPS. Сам протокол PoL може забезпечити інфраструктуру та механізм локалізації.

2. Співвідношення B пов’язує пристрій користувача (доказувач) зі свідками. Це задовольняється протоколом PoL, укладеним між доказівником та свідками. Мета полягає в тому, щоб засвідчити, що доказувач знаходиться поруч зі свідками в конкретний час. Більшість протоколів PoL у бібліографії мають лише цю кореляцію. Вони засвідчують, що пристрій з обліковими даними користувача знаходиться поблизу свідків. Однак може бути багато пристроїв перевірок, які мають однакові облікові дані користувача. Таким чином, підключення B не прив’язує користувача до розташування.

3. Кореляція C пов'язує користувача з пристроєм у певний час, коли виконується протокол PoL. Щоб задовольнити цей зв’язок, під час сеансу протоколу розвідник створює носій. Цей носій підтверджує, що під час створення PoL перебуває у розпорядженні користувача. Його називають «сильною ідентифікацією», оскільки він засвідчує не лише місцезнаходження пристрою, а й місцезнаходження користувача як такого. Протокол PoL може задовольняти всім перерахованим вище співвідношенням. Принаймні, він повинен задовольняти B.

Proof-of-Importance (PoI): алгоритм консенсус PoI працює аналогічно як PoS, складається три компоненти (рис. 2.13).

Рисунок 2.13 – Proof-of-Importance

Перший параметр впливає на рейтинг транзакцій, тоді як другий і третій параметри мають меншу вагу, але все ж враховуються для визначення "важливості" облікового запису. Чим менша сума токенів, тим більший вплив мають інші параметри.

Тому обліковий запис з великою кількістю токенів може підвищити коефіцієнт значущості майже в 3 рази завдяки активності та постійній присутності у мережі. Однак це має незначний вплив на тих, у кого є сотні мільйонів токенів на їх облікових записах.

Proof-of-Elapsed-Time (PoET): Intel, як провідна компанія в галузі технологій, розробила свій власний блокчейн під назвою IntelLedger. Для алгоритму консенсусу IntelLedger використовується Proof-of-Elapsed-Time (PoET). Цей алгоритм подібний до Proof-of-Work, але він вимагає значно менше електроенергії (див. рис. 2.14). Замість того, щоб учасники розв'язували криптографічні головоломки, алгоритм працює в середовищі надійного виконання (Trusted Execution Environment, TEE), такому як Intel Software Guard Extensions (SGX).

Рисунок 2.14 – Proof-of-Importance

Також даний протокол PoET гарантує, що блоки будуть створюватися випадково.

Згідно статистичних даних компанії Intel алгоритм PoET можна масштабувати до тисяч нод, при цьому він коректно працюватиме на будь-якому процесорі Intel, що підтримує SGX.

2.2 Дослідження механізмів забезпечення цілісності

Використання технології blockchain у Bitcoin орієнтоване саме на забезпечення цієї послуги безпеки (відповідно, її забезпечення вимагає максимальної уваги до всіх процесів, пов’язаних з обробкою даних: зберіганням, передачею тощо).

Цілісність даних є однією з основних характеристик, які сприяють і зміцнюють довіру до технології Blockchain. Ця цілісність даних досягається завдяки розумній ідеї криптографії, яка складає механізм консенсусу. Для досягнення консенсусу мережа Blockchain використовує так званий механізм доказу роботи. Ці механізми гарантують, що зміна будь-якої одиниці інформації в Blockchain означатиме використання величезної обчислювальної потужності для перевизначення всієї мережі. Крім того, Blockchain пропонує рішення, яке може вирішити одну зі складних відкритих проблем, відомих у літературі як проблема подвійних витрат. Це розглядається як помітна вразливість, яка порушує цілісність системи. Мережа Blockchain відмічає час першої транзакції, на яку власник витрачає певну монету, і відхиляє наступні, таким чином виключаючи подвійні витрати. Довірі також можна підвищити її прозорий характер, коли кожен користувач може перевіряти трансляцію трансакції на основі заздалегідь визначених правил.

Впевненість у цілісності даних у Blockchain залежить від трудомістких завдань PoW у процесі майнінгу. Але майнінг викликає головну нестачу неефективності зберігання даних: високу затримку підтвердження та низьку повсюдну затримку. Затримка - це часовий інтервал між надсиланням даних та підтвердженням збереження. У біткойнах це 10 хвилин, що приблизно дорівнює часу обчислення блоку. Пропускна здатність становить близько семи транзакцій на

секунду. У порівнянні з класичним сховищем даних Blockchain досить неефективний. Це створює ще одну проблему забезпечення такої ж цілісності.

Зокрема, підвищення цілісності вводиться трирівнева Blockchain-мережа. На першому рівні перевіряється політика доступу. Пропонований реєстр безпеки на основі Blockchain, що базується на загальному вимірі віртуальних машин, покращує політику безпеки. Це робиться у другому шарі. Зрештою, на третьому рівні виконується перевірка безпеки для відповідних пакетів даних віртуальної машини.

На третьому рівні основна робота - поділ конфіденційних та нечутливих даних. Три типи Blockchain - це загальнодоступний Blockchain, Blockchain консорціуму та приватний Blockchain. Кожен може перевірити транзакцію та підтвердити її, а також може брати участь у процесі досягнення консенсусу, який називається публічним ланцюжком блоків.

Blockchain консорціуму означає, що вузол, який мав повноваження, можна вибрати заздалегідь, зазвичай має партнерські відносини, такі як бізнес-бізнес, дані в ланцюжку блоків можуть бути відкритими або приватними, розглядатися як частково децентралізовані. Як Hyperledger, і R3CEV є Blockchain консорціуму. У приватному ланцюжку блоків вузол буде обмежений, не кожен вузол може брати участь у цьому ланцюжку блоків, має строго управління повноваженнями при доступі до даних.

2.3 Дослідження механізмів забезпечення автентичності

Завдання із забезпечення перевірки автентичності авторства (автентичності) найчастіше вирішується за допомогою цифрового підпису.

Цифровий підпис - це аналог рукописного підпису, який забезпечує дві властивості: можливість перевірки справжності та цілісності документа, що захищає його від модифікації та підміни.

Сучасний криптографічний ключ – це цифрова послідовність певної довжини, створена за певними правилами, з використанням генераторів

випадкових чисел та/або розрахована відповідно до спеціального алгоритму з інших значень. Криптографічний ключ є важливою складовою під час здійснення криптографічних операцій.

2.3.1 Безпека відкритого ключа

Відкритий ключ піддається гешування за допомогою SHA256, а для результату знову розраховується геш-значення, але вже за допомогою RIPEMD160. На виході виходить число завдовжки 160 біт (20 байт). Біткоїн-адреси майже завжди представлені користувачам у кодуванні, що називається Base58Check, в якому використовується алфавіт з 58 символів та контрольна сума, що дозволяє позбутися двозначностей у написанні та запобігти помилкам. Приховування відкритого ключа захищає користувачів від крадіжки монет, якщо буде зламано алгоритм цифрового підпису. По-друге, різні функції були застосовані з метою зниження ризику атаки на геш-функції – були взяті незалежні стандарти та ймовірність, що обидва містять backdoor, гранично мала (рис. 2.15).

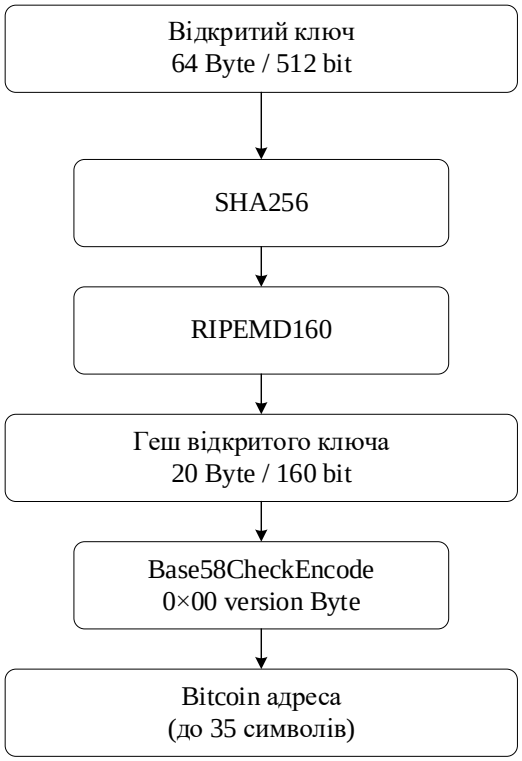

Рисунок 2.15 – Адреса виходить з відкритого ключа користувача внаслідок застосування алгоритмів гешування

2.3.2 Безпека закритого ключа

У системах Blockchain закритий ключ користувача розпізнається як облікові дані безпеки та автентифікації, створені користувачем, і третя сторона не бере участі в цьому процесі. Щоразу, коли користувач створює гаманець для криптовалюти, він / вона також повинен імпортувати закритий ключ у гаманець. Цей закритий ключ імпортується в гаманець, щоб гарантувати безпеку та автентифікацію криптовалюту. Якщо закритий ключ втрачено або вкрадено, його неможливо відновити, що означає, що користувач не може отримати доступ до гаманця іншими альтернативними способами і що його криптовалюти в гаманці недоступні.

Системи Blockchain не контролюються сторонніми організаціями, тому сценарії втрати або крадіжки закритого ключа призводять до ризику зміни даних зловмисниками, що не відстежуються.

2.4 Висновки до розділу 2

Підбиваючи підсумки, головна перевага blockchain полягає в тому, що він підтримує ідею створення відкритої, загальнодоступної та надійної книги, яка успішно народилася з Bitcoin. Таким чином, blockchain надав перше розв'язання проблеми встановлення довіри в незахищеному середовищі, не покладаючись на третю сторону. Що є добре відомим завданням у розподілених обчисленнях. Коротко кажучи, проблема полягає в спробі домовитися про курс дій або стан системи шляхом обміну інформацією через ненадійну та потенційно скомпрометовану мережу. Рішення Саточі Yакамото, яке використовує концепцію proof-of-work для досягнення консенсусу без центрального надійного органу. Взагалі перелічують близько 18 переваг (гарантій), які пропонує технологія blockchain, які випливають із її неявних принципів проектування. Ці гарантії розглядаються як будівельні блоки, які можна використовувати для створення різних додатків з різними гарантіями на основі їх природи та використання.

Протоколи консенсусу є дуже важливою складовою розподілених систем. Завдяки їх можна встановити справедливість, уникаючи збоїв системи після того

один з учасників виходить із ладу. По-друге, децентралізоване середовище потребує прийняття рішень навіть в такому середовищі де існує недовіра, але є правила, які допоможуть досягнути консенсусу.

В даному розділі було розглянуто найпоширеніші протоколи,які в свою чергу використовуються у десятках проектів. Хоча з питань безпеки та конфіденційності Blockchain було проведено безліч досліджень, систематичного вивчення безпеки систем Blockchain все ще немає.

3 РЕАЛІЗАЦІЯ СТВОРЕННЯ ТА ВИКОРИСТАННЯ КРИПТОВАЛЮТ НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ

3.1 Вибір програмних засобів для реалізації майнінгу

За допомогою функції гешування SHA-256, яка використовується як для майнінгу, так і для процедури додавання монет у ланцюжок Bitcoin, а також для створення Bitcoin-адрес, я реалізую програму, яка покаже, як додається ланцюжок блоків, і що саме міститься в кожному блоку. Сам майнінг включає налаштування числа так, щоб результат алгоритму гешування був нижче певного значення. Натомість майнеру передаються Bitcoin. Потім знову здобутий блок міститься в ланцюжок блоків.

Реалізація програмного продукту було зроблено мовою програмування C#.

Переваги C#:

Незважаючи на простоту функціонування, C# є потужною мовою програмування, завдяки якій розробники мають можливість створювати багатофункціональні програми. Ця мова ставитися до мов компільованого типу, тому він має всі переваги таких мов, як Java, C++, Visual Basic і т.д.

Через велику різноманітність синтаксичних конструкцій та можливості працювати з платформою .Net, C# дозволяє швидше, ніж будь-яку іншу мову, розробляти програмні рішення.

C# відрізняється є надійною.

Postman – це HTTP-клієнт використовується для тестування API.

API (Application Programming Interface) – це інтерфейс для обміну даними із сервера між двома програмами або компонентами ПЗ. Postman відіграє важливу роль в допомозі тестувальникам у проєктуванні дизайну API та створенні імітаторів роботи програми (mock-серверів).

Тестувальник за допомогою HTTP-клієнта (Postman) може складати та відправляти HTTP-запити до API; створювати папки запитів, щоб скоротити час тестування, створювати набір послідовних запитів та змінювати їхні параметри,

оточення та фіксувати моментну передачу, а також здійснювати тестування да допомогою Collection Runner.

Щоб виявити чи є помилки в роботі API, тестувальний відправляє тестові запити від клієнта до сервера. Цей запит обробляється програмою, яка працює на протоколі HTTP, і відправляє тестувальнику відповідь.

3.2 Програмна реалізація імітація майнінгу криптовалют

Для створення імітації майнінгу Blockchain необхідно попередньо виконати налаштування середовища розробки та додаткового програмного забезпечення. Крім встановлення Microsoft Visual Studio 2019 (рис. 3.1), є потреба у встановленні HTTP-клієнта, для цього було обрано Postman (рис. 3.2).

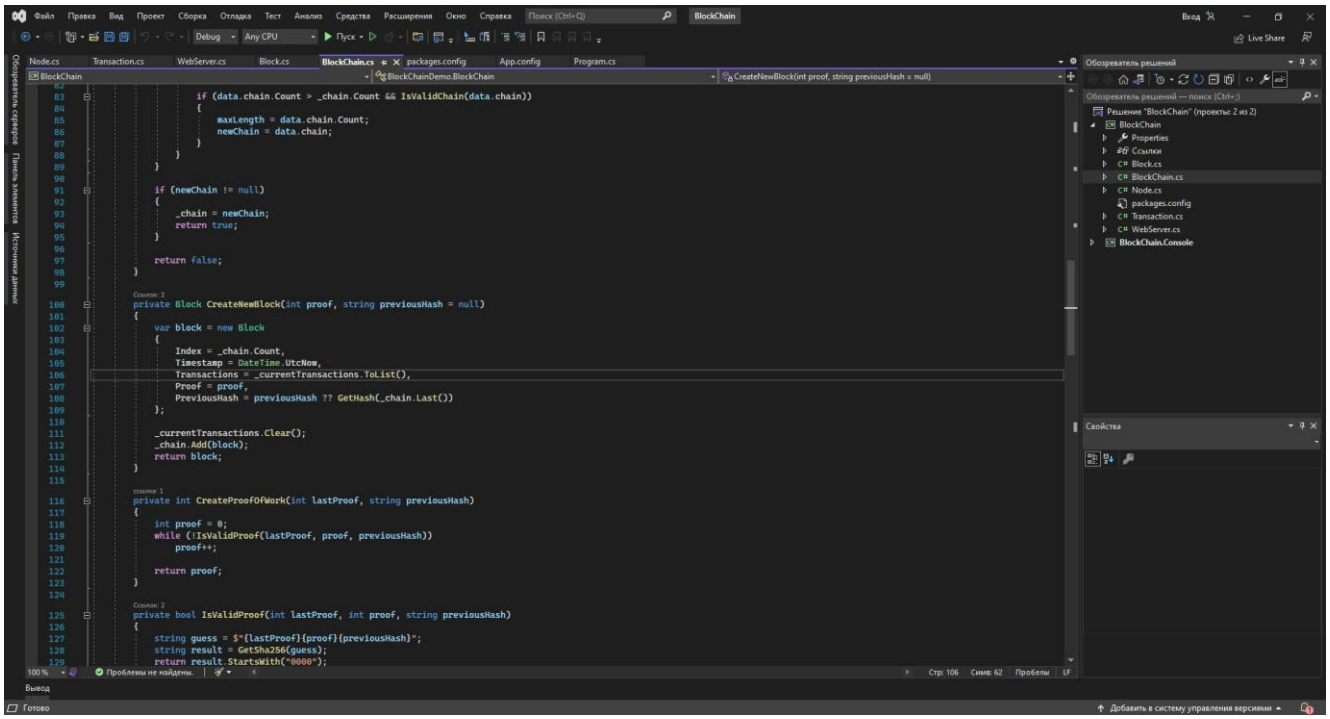

Рисунок 3.1 – Інтерфейс Microsoft Visual Studio

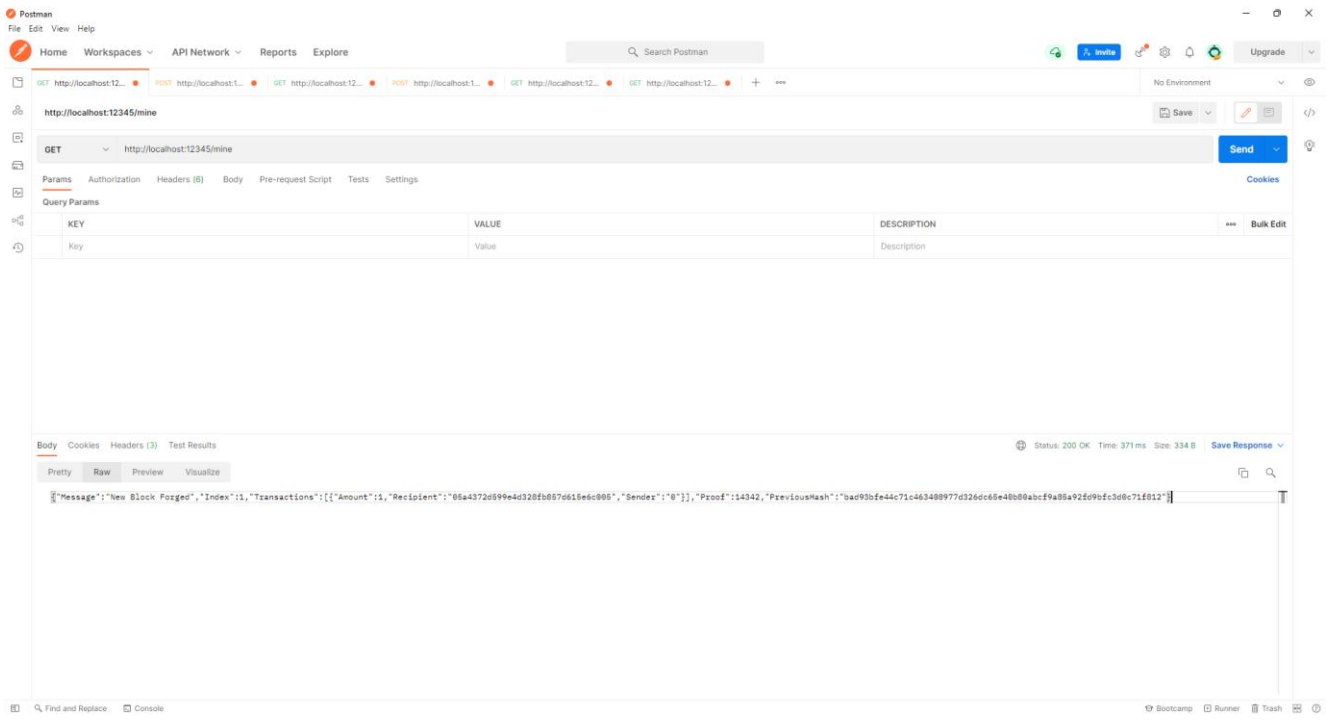

Рисунок 3.2 – Інтерфейс Postman

На початку був створений клас для реалізації Blockchain. Даний клас забезпечує роботу з нашим Blockchain, несе відповідальність за збереження транзакцій та містить методи для внесення нових блоків у ланцюжок блоків. Приклад кода зображено на рис. 3.3 – 3.6

```
ссылка: 1
public BlockChain()
{
    NodeId = Guid.NewGuid().ToString().Replace("-", "");
    CreateNewBlock(proof: 100, previousHash: "1"); //genesis block
}
```

Рисунок 3.3 – Приклад коду

```
Ссылок: 2
internal int CreateTransaction(string sender, string recipient, int amount)
{
    var transaction = new Transaction
    {
        Sender = sender,
        Recipient = recipient,
        Amount = amount
    };

    _currentTransactions.Add(transaction);

    return _lastBlock != null ? _lastBlock.Index + 1 : 0;
}
```

Рисунок 3.4 – Приклад коду

```
Ссылок: 2
private string GetSha256(string data)
{
    var sha256 = new SHA256Managed();
    var hashBuilder = new StringBuilder();

    byte[] bytes = Encoding.Unicode.GetBytes(data);
    byte[] hash = sha256.ComputeHash(bytes);

    foreach (byte x in hash)
        hashBuilder.Append($"{x:x2}");

    return hashBuilder.ToString();
}
```

Рисунок 3.5 – Приклад коду

```
ссылка: 1
private bool IsValidChain(List<Block> chain)
{
    Block block = null;
    Block lastBlock = chain.First();
    int currentIndex = 1;
    while (currentIndex < chain.Count)
    {
        block = chain.ElementAt(currentIndex);
        Debug.WriteLine($"{lastBlock}");
        Debug.WriteLine($"{block}");
        Debug.WriteLine("-------------------------");

        //Check that the hash of the block is correct
        if (block.PreviousHash != GetHash(lastBlock))
            return false;

        //Check that the Proof of Work is correct
        if (!IsValidProof(lastBlock.Proof, block.Proof, lastBlock.PreviousHash))
            return false;

        lastBlock = block;
        currentIndex++;
    }

    return true;
}
```

Рисунок 3.6 – Приклад коду

Розглянемо більш детальніше програмний код та його реалізацію. Згідно з описом Blockchain, кожний окремий блок якого зберігає такі дані, як: тимчасова мітка, індекс блоку, доказ консенсусу та геш попереднього блоку (рис. 3.7).

```
Ссылок: 2
private Block CreateNewBlock(int proof, string previousHash = null)
{
    var block = new Block
    {
        Index = _chain.Count,
        Timestamp = DateTime.UtcNow,
        Transactions = _currentTransactions.ToList(),
        Proof = proof,
        PreviousHash = previousHash ?? GetHash(_chain.Last())
    };

    _currentTransactions.Clear();
    _chain.Add(block);
    return block;
}
```

Рисунок 3.7 – Приклад коду

CreateTransaction (рис. 3.8, рис. 3.9) відповідає за нові транзакції у блок. Заносить транзакцію до списку, повертаючи індекс блоку.

```
Ссылок: 2
internal int CreateTransaction(string sender, string recipient, int amount)
{
    var transaction = new Transaction
    {
        Sender = sender,
        Recipient = recipient,
        Amount = amount
    };

    _currentTransactions.Add(transaction);

    return _lastBlock != null ? _lastBlock.Index + 1 : 0;
}
```

Рисунок 3.8 – CreateTransaction()

```
Ссылок: 2
private Block CreateNewBlock(int proof, string previousHash = null)
{
    var block = new Block
    {
        Index = _chain.Count,
        Timestamp = DateTime.UtcNow,
        Transactions = _currentTransactions.ToList(),
        Proof = proof,
        PreviousHash = previousHash ?? GetHash(_chain.Last())
    };

    _currentTransactions.Clear();
    _chain.Add(block);
    return block;
}
```

Рисунок 3.9 – Створення нового блоку CreateNewBlock()

Для створення першого блоку генези – першого блоку без попередника використовується фрагмент коду:

Створення блоку генези

self.new_block (previous_hash=1, proof=100)

Також блок має мати справедливий консенсус, який дає результат майнінгу. CreateProofOfwork дуже схожий з алгоритмом Hashcash в Bitcoin по реалізації алгоритму доказу роботи (рис. 3.10).

```
ссылка: 1
private int CreateProofOfWork(int lastProof, string previousHash)
{
    int proof = 0;
    while (!IsValidProof(lastProof, proof, previousHash))
        proof++;

    return proof;
}
```

Рисунок 3.10 – Метод proof_of_work

Взаємодія з нашим Blockchain виконуються за допомогою HTTP-запитів, і для цього створюють такі три методи:

- /transactions/new – створює у блоці нову транзакцію;
- /mine - дає завдання серверу початку майнінгу нового блока;

- /chain - для повернення всього Blockchain

Розглянемо шаблонний код сервера вузла мережі Blockchain.

```
public WebServer(BlockChain chain)
{
    var settings = ConfigurationManager.AppSettings;
    string host = settings["host"]?.Length > 1 ? settings["host"] : "localhost";
    string port = settings["port"]?.Length > 1 ? settings["port"] : "12345";

    var server = new TinyWebServer.WebServer(request =>
        {
            string path = request.Url.PathAndQuery.ToLower();
            string query = "";
            string json = "";
            if (path.Contains("?"))
            {
                string[] parts = path.Split('?');
                path = parts[0];
                query = parts[1];
            }

            switch (path)
            {
                //GET: http://localhost:12345/mine
                case "/mine":
                    return chain.Mine();

                //POST: http://localhost:12345/transactions/new
                //{ "Amount":123, "Recipient":"ebeabf5cc1d54abdbca5a8fe9493b479", "Sender":"31de2e0ef1cb4937830fcfd5d2b3b24f" }
                case "/transactions/new":
                    if (request.HttpMethod != HttpMethod.Post.Method)
                        return $"{new HttpResponseMessage(HttpStatusCode.MethodNotAllowed)}";

                    json = new StreamReader(request.InputStream).ReadToEnd();
                    Transaction trx = JsonConvert.DeserializeObject<Transaction>(json);
                    int blockId = chain.CreateTransaction(trx.Sender, trx.Recipient, trx.Amount);
                    return $"Your transaction will be included in block {blockId}";

                //GET: http://localhost:12345/chain
                case "/chain":
                    return chain.GetFullChain();

                //POST: http://localhost:12345/nodes/register
                //{ "Urls": ["localhost:54321", "localhost:54345", "localhost:12321"] }
                case "/nodes/register":
                    if (request.HttpMethod != HttpMethod.Post.Method)
                        return $"{new HttpResponseMessage(HttpStatusCode.MethodNotAllowed)}";

                    json = new StreamReader(request.InputStream).ReadToEnd();
                    var urlList = new { Urls = new string[0] };
                    var obj = JsonConvert.DeserializeAnonymousType(json, urlList);
                    return chain.RegisterNodes(obj.Urls);

                //GET: http://localhost:12345/nodes/resolve
                case "/nodes/resolve":
                    return chain.Consensus();
            }

            return "";
        },
        $"http://{host}:{port}/mine/",
        $"http://{host}:{port}/transactions/new/",
        $"http://{host}:{port}/chain/",
        $"http://{host}:{port}/nodes/register/",
        $"http://{host}:{port}/nodes/resolve/"
    );
```

Рисунок 3.11 – Програмний код для реалізації майнінгу на вузлі мережі

Для того, щоб на вузлі виконувався майнінг нового блоку, потрібно вирішити задачу для алгоритму доказу роботи (PoW). Після чого необхідно створити новий блок, який буду містити в собі ланцюжки блоків.

Перед тим як використовувати клієнт HTTP-запитів Postman, потрібно запустити процес майнінгу блоку, а для цього потрібно надіслати GET-запит серверу вузла. Результат виконання запиту на майнінг блоку подано на (рис. 3.12).

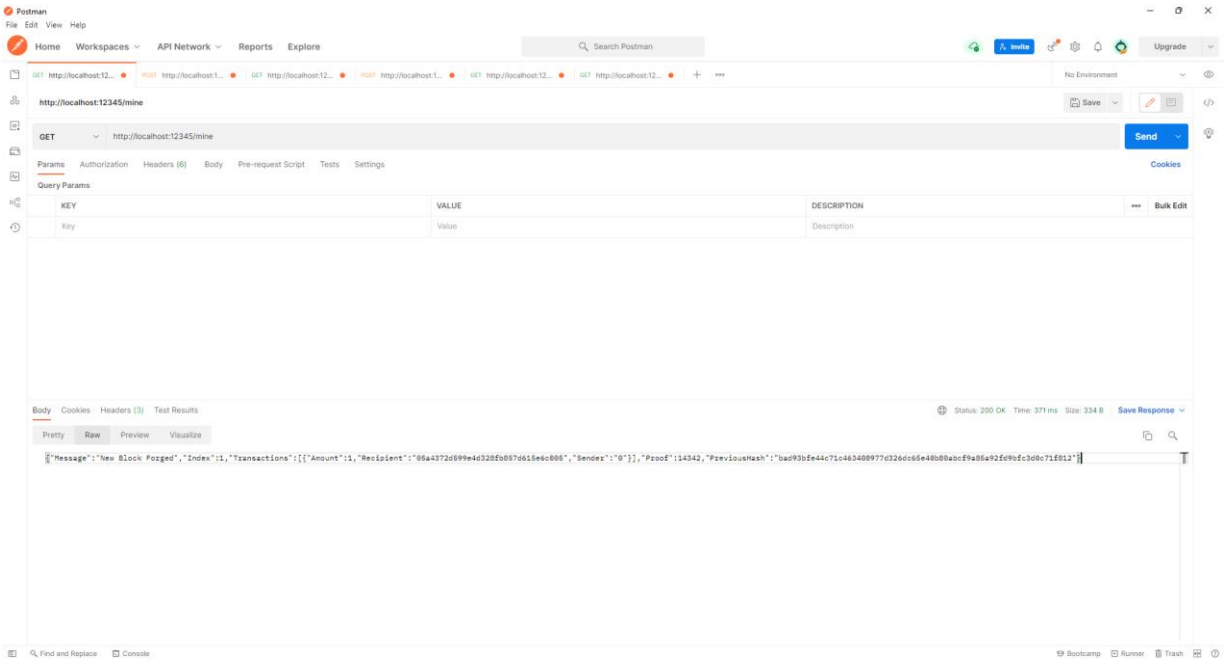

Рисунок 3.12 – Запит на майнінг блоку та результат виконання запиту

Щоб була можливість сторення нової транзакції треба здійснити POST-запит до вузла мережі з тілом (рис. 3.13).

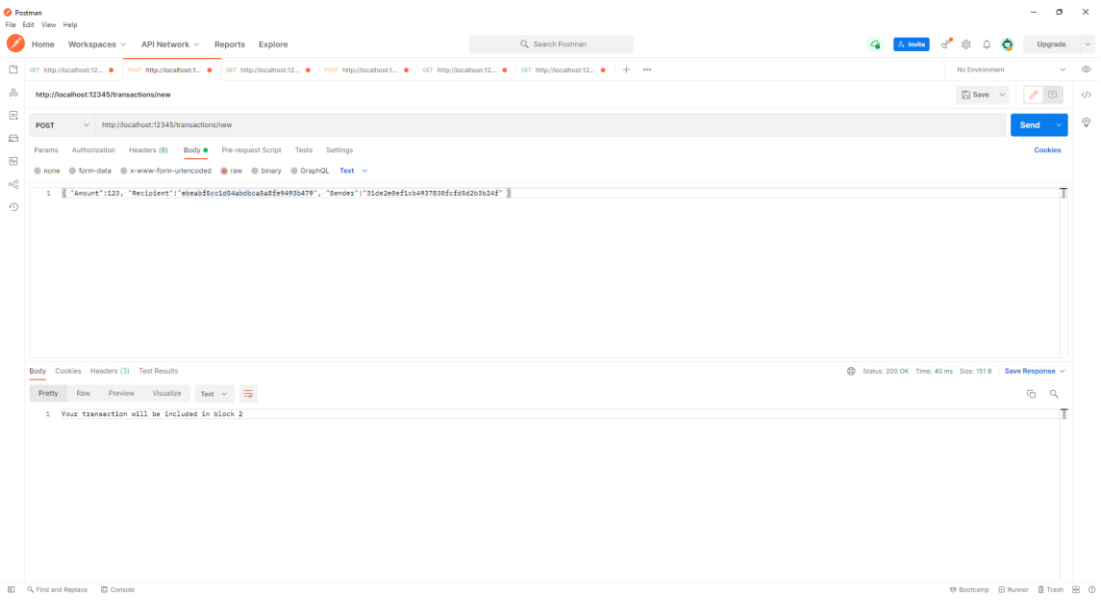

Рисунок 3.13 – Запит на виконання транзакції

Реалізація децентралізованої мережі вузлів здійснюється шляхом реалізації програмного коду для реєстрації нових вузлів мережі та реалізації алгоритмів консенсусу. Програмний код реалізації даних кінцевих точок представлений на малюнках (рис. 3.14).

```
ссылка: 1
internal string RegisterNodes(string[] nodes)
{
    var builder = new StringBuilder();
    foreach (string node in nodes)
    {
        string url = $"http://{node}";
        RegisterNode(url);
        builder.Append($"{url}, ");
    }

    builder.Insert(0, $"{nodes.Count()} new nodes have been added: ");
    string result = builder.ToString();
    return result.Substring(0, result.Length - 2);
}
```

Рисунок 3.14 – Реалізація кінцевої точки для реєстрації нових вузлів мережі Blockchain

Згідно алгоритму консенсусу PoW дійсним ланцюгом являється найдовший і є авторитетним на цьому вузлі. Метод ResolveConflicts дозволяє конфлікти та робить звміну на найдовший ланцюжок (рис. 3.15).

```
ссылка: 1
private bool ResolveConflicts()
{
    List<Block> newChain = null;
    int maxLength = _chain.Count;

    foreach (Node node in _nodes)
    {
        var url = new Uri(node.Address, "/chain");
        var request = (HttpWebRequest)WebRequest.Create(url);
        var response = (HttpWebResponse)request.GetResponse();

        if (response.StatusCode == HttpStatusCode.OK)
        {
            var model = new
            {
                chain = new List<Block>(),
                length = 0
            };
            string json = new StreamReader(response.GetResponseStream()).ReadToEnd();
            var data = JsonConvert.DeserializeAnonymousType(json, model);

            if (data.chain.Count > _chain.Count && IsValidChain(data.chain))
            {
                maxLength = data.chain.Count;
                newChain = data.chain;
            }
        }
    }

    if (newChain != null)
    {
        _chain = newChain;
        return true;
    }

    return false;
}
```

Рисунок 3.15 – ResolveConflicts

Надалі відбувається реєстрація концевих точок для API додавання нових вузлів і вирішення конфліктів (рис. 3.16).

```
public WebServer(BlockChain chain)
{
    var settings = ConfigurationManager.AppSettings;
    string host = settings["host"]?.Length > 1 ? settings["host"] : "localhost";
    string port = settings["port"]?.Length > 1 ? settings["port"] : "12345";

    var server = new TinyWebServer.WebServer(request =>
        {
            string path = request.Url.PathAndQuery.ToLower();
            string query = "";
            string json = "";
            if (path.Contains("?"))
            {
                string[] parts = path.Split('?');
                path = parts[0];
                query = parts[1];
            }

            switch (path)
            {
                //GET: http://localhost:12345/mine
                case "/mine":
                    return chain.Mine();

                //POST: http://localhost:12345/transactions/new
                //{ "Amount":123, "Recipient":"ebeabf5cc1d54abdbca5a8fe9493b479", "Sender":"31de2e0ef1cb4937830fcfd5d2b3b24f" }
                case "/transactions/new":
                    if (request.HttpMethod != HttpMethod.Post.Method)
                        return ${new HttpResponseMessage(HttpStatusCode.MethodNotAllowed)}";

                    json = new StreamReader(request.InputStream).ReadToEnd();
                    Transaction trx = JsonConvert.DeserializeObject<Transaction>(json);
                    int blockId = chain.CreateTransaction(trx.Sender, trx.Recipient, trx.Amount);
                    return $"Your transaction will be included in block {blockId}";

                //GET: http://localhost:12345/chain
                case "/chain":
                    return chain.GetFullChain();

                //POST: http://localhost:12345/nodes/register
                //{ "Urls": ["localhost:54321", "localhost:54345", "localhost:12321"] }
                case "/nodes/register":
                    if (request.HttpMethod != HttpMethod.Post.Method)
                        return ${new HttpResponseMessage(HttpStatusCode.MethodNotAllowed)}";

                    json = new StreamReader(request.InputStream).ReadToEnd();
                    var urlList = new { Urls = new string[0] };
                    var obj = JsonConvert.DeserializeAnonymousType(json, urlList);
                    return chain.RegisterNodes(obj.Urls);

                //GET: http://localhost:12345/nodes/resolve
                case "/nodes/resolve":
                    return chain.Consensus();
            }

            return "";
        },
        $"http://{host}:{port}/mine/",
        $"http://{host}:{port}/transactions/new/",
        $"http://{host}:{port}/chain/",
        $"http://{host}:{port}/nodes/register/",
        $"http://{host}:{port}/nodes/resolve/"
    );
```

Рисунок 3.16 – Кінцеві точки для API

Провірка роботи алгоритмів консенсусу та транзакцій, відбувається при запуску нових вузлів (рис. 3.17 – 3.18).

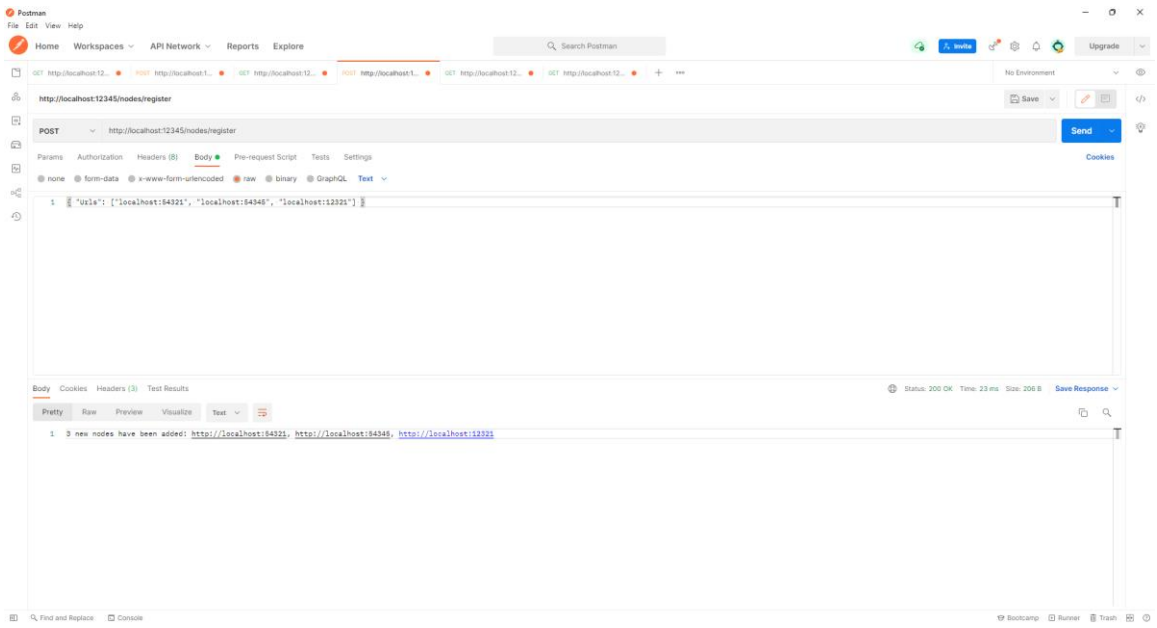

Рисунок 3.17 – Реєстрація нового вузла мережі

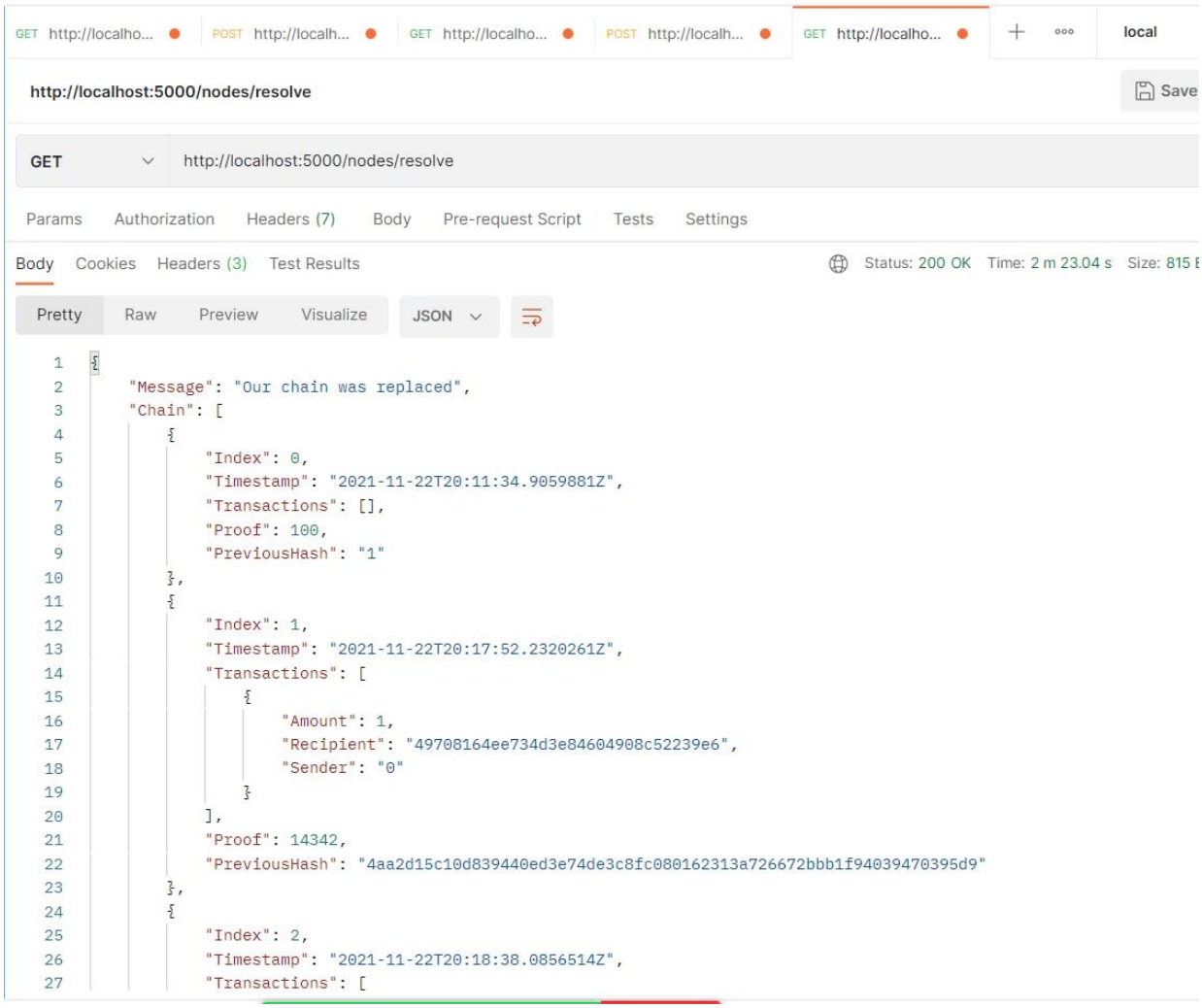

Рисунок 3.18 – Виконання транзакції та перевірка алгоритму консенсусу

3.3 Оцінка поточного стану безпеки децентралізованих систем

Оскільки blockchain розподіляються між безліччю однорангових, забезпечення безпеки ланцюга має велике значення. Особливо великою проблемою, яку потрібно було вирішити, була проблема подвійних витрат, тобто коли актор може витратити одні й ті ж монети кілька разів. Blockchain розв'язує цю проблему за допомогою методу криптографічного гешування, який називається proof-of-work, який вимагає розв’язування криптографічних головоломок. На практиці ці головоломки об’єднують усі транзакції, які потрібно включити до блоку, додають до нього криптографічний одноразовий номер і обчислюють геш, який починається з певної кількості нулів, зазвичай називають складністю блокувати. Складність можна підвищити або зменшити, щоб переконатися, що лише невелика підмножина всіх можливих значень, створених геш-функцією, відповідає вимогам. Оскільки ця підмножина може бути досить маленькою, у порівнянні з усіма можливими значеннями гешування, багато одноразових значень потрібно гешувати, перш ніж знайти той, який вдасться, що вимагає великої обчислювальної потужності. Однак спробу відтворити геш під час перевірки нового блоку можна зробити, знаючи одноразовий номер, тому перевірка набагато швидша. Це призводить до того, що створення нових блоків вимагає значної обчислювальної потужності в порівнянні з верифікацією, і це теоретично має забезпечити мережу без подвійних витрат.

Однак безпека blockchain не може бути детерміновано гарантована. Якщо один майнер або група майнерів контролює 51% обчислювальної потужності в мережі, вони теоретично можуть подвоїти витрати в мережі і призвести до втрати грошей іншими користувачами. Це стало відомо як атака 51%. Після відправки коштів на інший рахунок, щоб здійснити подвійну витрату, майнери могли відокремити blockchain у попередній точці та відновити один або кілька блоків, замінивши транзакцію на адресу, що належить комусь іншому, трансакція, що надсилає ту саму суму на адресу, що належить зловмиснику. Коли в кінцевому підсумку довжина blockchain в розгалуженій мережі перевищує довжину основної

мережі, зловмисники можуть об’єднати свою мережу з основною мережею, а їхній blockchain стане консенсусним blockchain. Теоретично також можна переписати всю історію blockchain за допомогою атаки, яка називається Genesis Attack. Варто також зазначити, що зловмисник, який контролює частку мережі, меншу за 51%, може здійснити атаку 51%. Ризик цього стає меншим, чим менша частина мережі, яку контролює зловмисник. Розділ blockchain, а потім спроба наздогнати довжину основного blockchain мають дуже високі вимоги до обчислювальної потужності. І для кожного додаткового блоку цей попит зростає. Таким чином, після того, як блок, що містить транзакцію, видобуто, зазвичай потрібно чекати певну кількість блоків, поки ця транзакція не буде визнана безпечною. Наприклад, на blockchain Bitcoin.

3.4 Висновки до розділу 3

У цьому розділі було представлено реалізацію створення та використання криптовалют на основі децентралізованих систем. Основна мета цього розділу полягала в тому, щоб створити імітацію майнингу криптовалют та спосіб створення розподіленої криптовалюти, який уникає вузьких місць і проблем централізації, які супроводжуються реалізацією blockchain. Розділ продемонстрував, що цей процес як і раніше гарантує, що всі нові зусилля з перевірки забезпечують захист кожної попередньої транзакції після короткого періоду зближення для кожної транзакції. Створюючи таким чином структуру наочний приклад, з чого складається blockchain, та як проходить процес майнингу.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Управління та нагляд за безпекою життєдіяльності в Україні

Основою управління безпекою є система організаційно-розпорядчих заходів з профілактики і протидії негативним факторам - недоліки, проблеми, кризові ситуації, які порушують стійке функціонування та розвиток держави, регіона, погрожуючи небезпекою окремій людині.

Управління - це завжди система взаємопов'язаних елементів. Вона складається з чотирьох основних структур. Перша - суб'єкти управління - органи, що відповідають за управління безпекою. Друга - об'єктивна система принципів, правил, певних обмежень, які формують структуру управління. Третя - сукупність інформаційних процесів, необхідних для регулювання стану об'єкта, його контролювання. Четверта - кваліфікований, структурований персонал, здатний контролювати ситуацію, приймати рішення. Ця структура багатопланова, складна. Її потрібно розглядати на загальнодержавному, регіональному, місцевому та локальному рівнях.

Вся державна система - Кабінет Міністрів України, Національна Рада з питань безпеки життєдіяльності населення, Комітет з нагляду за охороною праці, структури Міністерства безпечної життєдіяльності, Служба безпеки України, Міністерство внутрішніх справ, органи державного пожежного нагляду, місцеві державні адміністрації та Ради депутатів, їх Виконавчі комітети.

Важливим державним органом є Національна рада з питань безпечної життєдіяльності населення, яка створена відповідно до Закону України "Про охорону праці". Основне її призначення розробка та реалізація державної політики в галузі охорони життя людей на виробництві та профілактики побутового травматизму, створення системи державного управління цією галуззю.

Національна рада у своїй діяльності керується Конституцією і законами України, постановами Верховної Ради України, указами і розпорядженнями Президента України, декретами, постановами і розпорядженнями Кабінету

Міністрів України, а також Положенням про Національну раду з питань безпеки. Вона розробляє та здійснює заходи щодо створення цілісної системи державного управління охороною життя людей на виробництві та профілактики побутового травматизму, вносить на розгляд Кабінету Міністрів України пропозиції про вдосконалення цієї системи; організує і забезпечує контроль за виконанням законодавчих актів і рішень Уряду України.

Рада сприяє впровадженню в життя Національної програми з безпечної життєдіяльності та законів, пов'язаних з реалізацією державної політики з питань безпечної життєдіяльності населення. Вона подає Кабінету Міністрів України пропозиції щодо вдосконалення законодавства з цих питань та координує діяльність центральних і місцевих органів державної виконавчої влади в галузі охорони життя людей на виробництві та профілактики побутового травматизму.

Цей орган організує перевірки діяльності центральних і місцевих органів державної виконавчої влади і заслуховує на своїх засіданнях або засіданнях бюро Національної ради звіти керівників з питань, що входять до її компетенції. Її представники беруть участь у міжнародному співробітництві, сприяючи вивченню, узагальненню і поширенню досвіду у галузі охорони життя людей на виробництві та профілактики побутового травматизму, вирішує питання контролю за виконанням укладених договорів і угод у цій галузі.

Рішення Національної ради та її бюро, прийняті в межах їх компетенції, є обов'язковими для виконання центральними та місцевими органами державної виконавчої влади, підприємствами, установами, організаціями та громадянами.

Крім центральних державних органів управління та нагляду за станом безпеки важливе місце займають місцеві органи - обласні, міські, на виробничих об'єктах, які повинні контролювати стан безпеки, охорони праці на кожному підприємстві.

Важливу роботу виконують державні інспектори, які проводять обстеження стану підприємств, фіксують порушення нормативних актів з охорони праці, призупиняють роботу виробництв та об'єктів, на яких виникала пряма загроза

здоров'ю, життю працюючих. Велику роль відіграють експертно-технічні центри, які займаються технічною експертизою стану обладнання, промислових об'єктів.

Важливе значення має робота з перегляду нормативно правових актів щодо державної експертизи проектної документації з питань охорони праці, здоров'я робітників, експертизи екологічного стану підприємств, місць їх розташування.

Враховуючи багаторівневість та багатоаспектність системи управління безпекою життєдіяльності, необхідно використовувати методи програмно-цільового та програмно-орієнтовного управління, тобто враховувати специфічні особливості кожної конкретної ситуації, місцевості, об'єкта.

Програмно-цільовий метод вимагає участі структур, які відносяться до різних відомств у рішенні системних задач, орієнтуючи всю систему управління безпекою життєдіяльності на кінцеву ціль - безпечна життєдіяльність всього суспільства.

Проблемно-орієнтовний метод акцентує увагу на прийняття профілактичних заходів, які б попереджували виникнення кризових ситуацій. Разом узяті ці два методи обумовлюють необхідність проведення робіт за двома напрямками. Перший - зумовлює необхідність загальних розробок концептуально-методологічних основ забезпечення безпеки життєдіяльності на державному рівні. Другий - потребує конкретних розробок, які б забезпечили управління безпекою на інших, нижчих рівнях - місцевому, локальному, об'єктному.

Отже, фахівець у будь-якій сфері діяльності повинен знати законодавство як державне, так і міжнародне. Тоді він грамотно, свідомо може впливати на події, передбачати їх результати, щоб завчасно організувати належні заходи, забезпечити безперервну роботу підприємства, його окремих виробництв. Фахівець-менеджер повинен так організувати діяльність, щоб природні, технічні та інші джерела загальної небезпеки, на які впливають люди, суспільство не створювали б небезпеки для життя та здоров'я працівників, для збереження матеріальних цінностей.

Сучасний стан світу, проблеми його збереження та питання безпеки у побутовій, виробничій, природній сферах ставлять перед фахівцем, керівником, будь-якою людиною вимоги збереження життя - людського, всієї природи.

4.2 Соціальне значення охорони праці

Соціальне значення охорони праці полягає в сприянні зростанню ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення його безпеки, зниження виробничого травматизму і захворюваності.

У зв'язку з цим соціальне значення охорони праці проявляється, перш за все, у впливі на зміну наступних трьох основних показників, що характеризують рівень розвитку суспільного виробництва.

1. Зростання продуктивності праці в результаті збільшення фонду робочого часу за рахунок:

• скорочення внутрішньо змінних простоїв шляхом попередження передчасного стомлення, а також зниження числа або ліквідації мікротравм, обумовлених несприятливими умовами праці. Попередження передчасного втоми з допомогою раціоналізації умов праці, введення оптимальних режимів праці та відпочинку та інших заходів на харчових підприємствах сприяє збільшенню ефективного використання робочого часу. Цей же результат дає ліквідація мікротравм, так як кожна з них супроводжується втратою до 2-х годин робочого часу;

• скорочення цілоденних втрат робочого часу в результаті зниження рівня або ліквідації тимчасової непрацездатності через виробничого травматизму, професійної і загальної захворюваності. Цей показник має важливе значення для харчових виробництв, на яких кожна травма в даний час супроводжується втратою працездатності в середньому більш ніж на 26 днів.

2. Збереження трудових ресурсів і підвищення професійної активності працюючих за рахунок:

• поліпшення стану здоров'я працюючих і збільшення середньої тривалості їх життя шляхом поліпшення умов праці, що також супроводжується збільшенням виробничого стажу працюють при їх високої трудової активності;

• підвищення професійного рівня внаслідок зростання кваліфікації і майстерності у зв'язку зі збільшенням виробничого стажу;

• можливості використання залишкової трудової активності, великого практичного досвіду та професійних знань пенсіонерів по старості та інвалідів на доступних для них роботах і забезпеченні відповідних їх фізичним можливостям умов праці.

3. Збільшення сукупного національного продукту за рахунок поліпшення зазначених вище показників і складових їх компонентів.

ВИСНОВКИ

Метою магістерської роботи є аналіз систем методики оцінки поточного стану безпеки децентралізованих систем на основі аналізу сучасних загроз та механізмів протидії та програмна реалізація створення криптовалют на основі децентралізованих систем. Створення криптовалют із використанням технології Blockchain, та аналіз механізмів забезпечення безпеки в децентралізованих системах.

У роботі розглядаються питання аналізу безпеки децентралізованих систем, механізми забезпечення конфіденційності та автентичності. Розглянуто побудови криптобірж, формування технології Blockchain та смарт-контрактів, механізми їх функціонування, а також тенденції розвитку та можливі ризики. Зокрема, здійснено реалізацію програмного продукту для створення криптовалют.

Перша частина роботи присвячена аналізу безпеки децентралізованих систем. Цей розділ охоплював такі теми як основні принципи побудови децентралізованих систем, загрози на ці системи, а також механізми забезпечення безпеки в децентралізованих системах за допомогою різних алгоритмів гешування.

Та в значній мірі зазначалися теми про використання децентралізованих систем у світі.

Другий розділ був присвячений аналізу механізмів забезпечення конфіденційності, автентичності та цілісності в децентралізованих системах.

Третій розділ включав в себе реалізацію створення та використання криптовалют на основі децентралізованих систем та оцінку поточного стану безпеки децентралізованих систем.

Технологія blockchain все ще знаходиться на стадії становлення. Передбачуваною метою blockchain було вирішити проблему подвійних витрат цифрових валют. Вирішивши проблему подвійних витрат, можна було б створити цифрову валюту, яка не вимагала б центрального органу для її контролю. По суті, bitcoin був доказом концепції для прямих цифрових транзакцій. Після виходу bitcoin швидко стало очевидним, що потенціал технології blockchain виходить

далеко за межі сфери цифрової валюти. В основі його лежить здатність blockchain усувати ризики контрагентів і необхідність фінансового посередництва в цифрових транзакціях.

Як і у випадку з усіма новими технологіями, лише час покаже, якими будуть довгострокові наслідки. Успіх будь-якої технологічної інновації залежить від її здатності створювати вартість. Майбутнє blockchain залежить від того, що він зможе успішно реалізувати обіцянку більшої ефективності та меншої вартості для користувачів. Якщо blockchain можуть успішно створювати цінність і підвищувати соціальний добробут, неминуче почнеться активне впровадження користувачів і подальші інвестиції. Було б несправедливо припускати, що будь-який blockchain був би бездоганним на такій ранній стадії. Існує ще багато проблем, які потрібно вирішити, перш ніж технологія blockchain зможе повністю розкрити свій потенціал. Щоб знайти вирішення даних проблем знадобиться співпраця всіх зацікавлених сторін, і процес, ймовірно, займе деякий час.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Наскрізна програма практики для студентів спеціальності 125 “Кібербезпека” другого (магістерського) рівня [Електронний ресурс] / уклад. С. П. Євсеєв, О. В. Мілов, О. Г. Король. – Харків : ХНЕУ ім. С. Кузнеця, 2021. – 32 с.

2. Методичні рекомендації до виконання дипломних проєктів (робіт) для студентів спеціальності 125 “Кібербезпека” другого (магістерського) рівня [Електронний ресурс] / уклад. С. П. Євсеєв, О. Г. Король, А. А. Гаврилова, О. В. Мілов. – Харків : ХНЕУ ім. С. Кузнеця, 2021. – 47 с.

3. Вимоги до оформлення курсових і дипломних проектів : методичні рекомендації для студентів галузі знань 12 “Інформаційні технології” / уклад. А. А. Гаврилова, С. П. Євсеєв, Г. П. Коц, О. Г. Руденко. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 50 с.

4. Децентрализованные приложения. Технология Blockchain в действии. – СПб.: Питер, 2017. – 240 с.: ил. – (Серия “Бестселлеры O’Reilly”).

5. Осваиваем биткоин / пер. С анг. А. В. Снастина. – М.: ДМК Пресс, 2018. – 428 с.: ил.

6. Машина правды. Блокчейн и будущее человечества / Пол Винья, Майкл Кейси ; пер. с англ. М. Сухотиной ; [науч. ред. К. Щеглова]. – М .: Манн, Иванов и Фербер, 2018. – 320 с.

7. Эпоха криптовалют. Как биткоин и блокчейн меняют мировой экономический порядок / Пол Винья, Майкл К ейси ; пер. с англ. Э. Кондуковой; [науч. ред. А. Форк]. – 2–е изд. – М .: Манн, Иванов и Фербер, 2018. – 432 с.

8. Блокчейн и децентрализованная денежная система: принципы построения и пути развития [Электронный ресурс] Режим доступу: https://cyberleninka.ru/article/n/blokcheyn–i–detsentralizovannaya–denezhnaya–sistema–printsipy–postroeniya–i–puti–razvitiya/viewer

9. Блокчейн и децентрализованные системы : учеб. пособие для студ. заведений высш. образования : в 3 частях. Ч. 1 / П. Кравченко, Б. Скрябин, О. Дубинина. – Харьков, 2019. – 488 с. : ил. 191; табл. 13; библиогр.: 124 назв.

10. Лелу, Л. Блокчейн от А до Я. Все о технологии десятилетия / Л. Лелу. – М.: Эксмо, 2018, – 256 с.

11. Integration DEFinition for function modeling (IDEF0). Draft Federal Information Processing Standards Publication 183, 1993 December 21. – URL: http://idef.com/wp–content/uploads/2016/02/idef0.pdf. (дата обращения 19.02.2019 г.)

12. Карпычев, В.Ю. Функциональное моделирование (IDEF0) как метод исследования блокчейнтехнологии / В.Ю. Карпычев // Труды НГТУ им. Р.Е. Алексеева. – 2018. – № 4 (123). – С. 22–32.

13. Иванов, О. Все об атаке “Человек посередине” (Man in the Middle, MitM). – URL: https://www.antimalware.ru/analytics/Threats_Analysis/man–in–the–middle–attack (дата обращения 29.03.2019 г.).

14. DOS и DDoS–атаки: понятие, разновидности, методы выявления и защиты. – URL: https://compconfig.ru/net/dos–i–ddos–ataki.html (дата обращения 29.03.2019 г.).

15. Дрешер, Д. Основы блокчейна: вводный курс для начинающих в 25 небольших главах / Д. Дрешер. – М.: ДМК Пресс, 2018, – 320 с

16. Bitcoin Developer Guide [Електронний ресурс] / – Режим доступу до ресурсу: https://developer.bitcoin.org/devguide/block_chain.html

17. Bitcoin [Електронний ресурс] / – Режим доступу до ресурсу: https://en.bitcoin.it/wiki/Main_Page.

18. Egill Már Hreinsson. The future of blockchain technology and cryptocurrencies. [Електронний ресурс] / Egill Már Hreinsson – Режим доступу до ресурсу: https://skemman.is/bitstream/1946/30832/1/The%20future%20of%20blockchain%20technology%20and%20cryptocurrencies..pdf.

19. A Decentralised Secure and Privacy-Preserving E-Government System [Електронний ресурс] – Режим доступу до ресурсу: https://nrl.northumbria.ac.uk/id/eprint/47353/1/nnko.noe_phd_16042130.pdf.

20. Achieving trust-oriented data protection in the cloud environment [Електронний ресурс] – Режим доступу до ресурсу: https://opus.lib.uts.edu.au/handle/10453/29219.

21. Authentication, Authorization and Accounting with Ethereum Blockchain [Електронний ресурс] – Режим доступу до ресурсу: https://helda.helsinki.fi/bitstream/handle/10138/228842/aaa-ethereum-blockchain.pdf?sequence=2&isAllowed=y.