МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

ФАРАФОНОВ ГЛІБ РУСЛАНОВИЧ

УДК 004.056:004.75

КВАЛІФІКАЦІЙНА РОБОТА

Дослідження механізмів забезпечення безпеки в децентралізованих системах

Спеціальність – 125 «Кібербезпека»

Галузь знань – 12 «Інформаційні технології»

Подається на здобуття освітнього ступеня магістр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____Г.Р.Фарафонов

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Євсеєєв Сергій Петрович

доктор технічних наук, професор

Житомир – 2023

Анотація

Фарафонов Г. Р. Дослідження механізмів забезпечення безпеки в децентралізованих системах. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 125 «Кібербезпека» галузі знань 12 «Інформаційні технології». – Поліський національний університет, Житомир, 2023.

Метою роботи є розробка та впровадження ефективних механізмів забезпечення безпеки в децентралізованих системах. Задачі включають аналіз існуючих ризиків та вразливостей децентралізованих систем, розробку стратегій захисту та вдосконалення методів виявлення та реагування на кіберзагрози.

Об'єктом дослідження є процес забезпечення кібербезпеки в децентралізованих системах. Предметом є конкретні аспекти цього процесу, такі як розподілена архітектура, криптографічні принципи, мережеві протоколи та методи управління доступом.

Предметом дослідження є конкретні аспекти цього процесу, такі як розподілена архітектура, криптографічні принципи, мережеві протоколи та методи управління доступом.

У дослідженні розглядаються аспекти безпеки децентралізованих систем, механізми, спрямовані на забезпечення конфіденційності та автентичності. Вивчено структури криптобірж, розробку технології Blockchain та смарт-контрактів, їх функціональність та тенденції розвитку, а також можливі ризики.

Зокрема, виконано створення програмного продукту для генерації криптовалют.

Дослідження базується на використанні публікацій, наукових видань та навчальних посібників.

У процесі роботи використовувалися такі програмні засоби, як Visual Studio 2019, Postman та Visio 2016.

Ключові слова: блокчейн, децентралізовані системи, інформаційна безпека, конфіденційність, криптовалюта.

SUMURRY

Farafonov G. R. Research of Security Mechanisms in Decentralized Systems. – Qualification work as a manuscript.

Qualification work for the acquisition of the second (master's) level of higher education in the specialty 125 "Cybersecurity" in the field of knowledge 12 "Information Technologies." – Polissia National University, Zhytomyr, 2023.

The aim of the work is the development and implementation of effective security mechanisms in decentralized systems. Tasks include analyzing existing risks and vulnerabilities of decentralized systems, developing protection strategies, and improving methods for detecting and responding to cyber threats.

The object of the research is the process of ensuring cybersecurity in decentralized systems. The subject includes specific aspects of this process, such as distributed architecture, cryptographic principles, network protocols, and access control methods.

The study examines security aspects of decentralized systems, mechanisms aimed at ensuring confidentiality and authenticity. It explores the structures of crypto exchanges, the development of Blockchain technology and smart contracts, their functionality, trends in development, and potential risks.

In particular, the creation of software for cryptocurrency generation was implemented.

The research is based on the use of publications, scientific publications, and educational materials.

During the work, software tools such as Visual Studio 2019, Postman, and Visio 2016 were used.

Keywords: blockchain, decentralized systems, information security, confidentiality, cryptocurrency.

ЗМІСТ

ВСТУП

Актуальність даного дослідження базується на швидкому розвитку децентралізованих технологій та їхньому значущому впливі на різні сфери життя, включаючи економіку, фінанси, управління та інформаційні системи. Використання блокчейну та інших децентралізованих підходів стає все більш поширеним, особливо з урахуванням їхніх переваг, таких як підвищена безпека, відсутність посередників і можливість побудови довіреності в умовах, коли це важко досягти.

Проте, разом із зростанням популярності децентралізованих систем, збільшується ймовірність кіберзагроз і порушень безпеки в цьому контексті. Відсутність централізованого управління і потенційні вразливості нових технологій створюють виклики для забезпечення конфіденційності, цілісності та доступності даних. Аналіз ідентифікованих загроз та розробка ефективних стратегій забезпечення безпеки стають важливими завданнями в контексті стрімкого розвитку цих технологій.

Наша країна активно розвиває інформаційні технології та є учасником глобальних технологічних процесів. Особливо важливо вивчати та розуміти вплив децентралізованих систем на кібербезпеку. Результати дослідження можуть допомогти удосконалити національні стратегії забезпечення безпеки в інформаційному просторі та зробити їх більш адаптованими до викликів, що виникають у зв'язку із зростанням застосування децентралізованих технологій.

Дане дослідження вирішує актуальне завдання розробки та оптимізації механізмів забезпечення безпеки в умовах швидкого розвитку децентралізованих систем, а його результати можуть мати значення для подальшого вдосконалення кібербезпеки як в глобальному, так і в національному контекстах.

Метою даної роботи є розробка та впровадження ефективних механізмів забезпечення безпеки в децентралізованих системах. Задачі включають аналіз існуючих ризиків та вразливостей децентралізованих систем, розробку стратегій захисту та вдосконалення методів виявлення та реагування на кіберзагрози.

Об'єктом дослідження є процес забезпечення кібербезпеки в децентралізованих системах.

Предметом дослідження є конкретні аспекти цього процесу, такі як розподілена архітектура, криптографічні принципи, мережеві протоколи та методи управління доступом.

Для досягнення поставленої мети використовуються такі методи дослідження:

1. Визначення потенційних загроз та їхній вплив на децентралізовані системи.
2. Вивчення різних сценаріїв кібератак та їхніх наслідків.
3. Створення ефективних планів для запобігання та відповіді на кіберзагрози.
4. Використання практичних випробувань для перевірки ефективності запропонованих механізмів.

Наукова новизна полягає в розробці нових підходів до забезпечення кібербезпеки в децентралізованих системах, що може відобразитися на подальшому розвитку галузі. Практичне значення роботи полягає в можливості застосування розроблених механізмів у реальних умовах, що сприятиме забезпеченню стійкості та надійності децентралізованих систем.

1 **АНАЛІЗ БЕЗПЕКИ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ**

1.1. **Основні принципи побудови децентралізованих систем**

Традиційна система готівкових платежів поступово втрачає свою популярність на користь електронних технологій, які включають в себе використання кредитних та дебетових карток, інтернет-банкінгу, електронної торгівлі та інших аспектів. Усі ці електронні засоби оплати функціонують у централізованому режимі, де існує один центральний орган - інстанція або установа, така як банки, уряди чи компанії, що випускають кредитні картки.

Під час обробки електронних транзакцій, користувачі взаємодіють з цими централізованими системами та довіряють їм як надійним третім сторонам. Незважаючи на те, що ці платіжні системи надають ряд переваг, таких як автентифікація та цифровий підпис, вони також мають свої недоліки.

Однією з головних проблем цієї моделі є вразливість перед атаками хакерів. Крім того, міжнародні грошові перекази в таких системах можуть бути повільними, інколи супроводжуються високими комісіями та не можуть уникнути незворотних транзакцій. Все це пов'язано із централізованою структурою, де існує єдиний центр, до якого звертаються всі учасники. Схема централізованої мережі наведена на рис. 1.1.

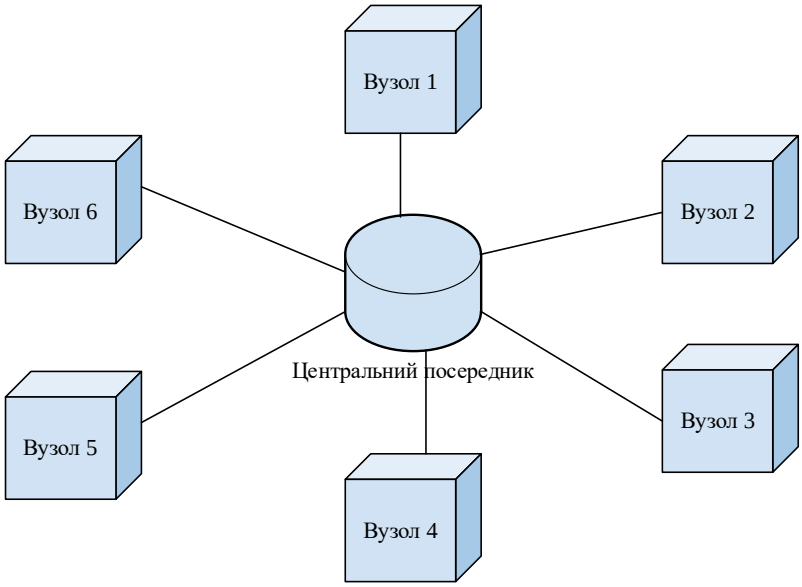


Рисунок 1.1 – Схема централізованої системи

Централізована система має низку важливих переваг. Управління такою системою є зручнішим, і зазвичай відомо, хто несе відповідальність за її функціонування. Рішення приймаються швидко, і можна легко розробити конкретну бізнес-модель і дотримуватися її, що стає перевагою при монетизації.

У децентралізованій системі існує безліч центральних вузлів, які є рівноправними, і всі користувачі мають доступ до них. В разі відмови одного вузла користувачі можуть звернутися до іншого, що робить систему більш стійкою.

Основна мета децентралізації полягає в створенні системи, яка забезпечує ефективну та надійну взаємодію користувачів один з одним у ситуаціях, коли вони не довіряють центральному органу чи посереднику. Цей підхід передбачає розподіл функцій системи між її учасниками, у відсутності єдиного органу управління.

На сьогоднішній день вже існують протоколи для децентралізованих платіжних систем, таких як Bitcoin, Ethereum та інші. Це дозволяє людям зміцнювати довіру через використання однорангових систем електронних грошей та здійснювати транзакції без участі третьої сторони, не контрольованої централізованими структурами. Чим більше користувачів у мережі, тим вона стає більш децентралізованою та безпечною.

Використання технології блокчейн у цих системах дозволяє встановлювати відносини довіри в розподіленому середовищі, уникаючи централізованого контролю. Хоча різні криптовалюти мають свої особливості, технологія блокчейн виявляється не лише в контексті цифрових грошей, але й має значно більше застосувань. Діаграма децентралізованої мережі, що є підмножиною розподіленої системи зображена на рис. 1.2.

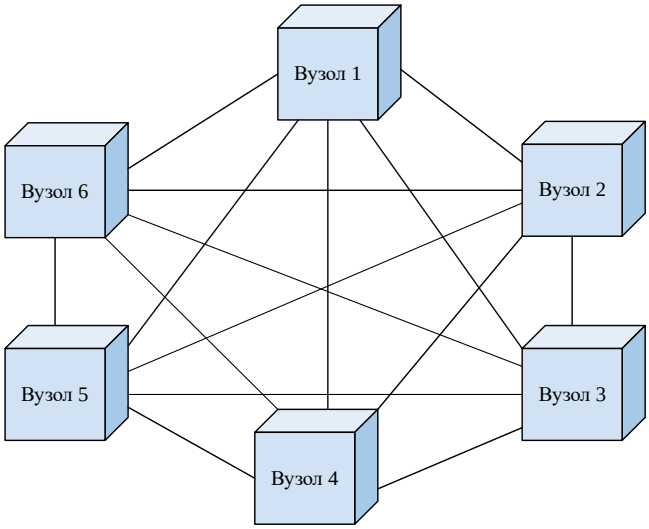


Рисунок 1.2 – Схема децентралізованої системи

Приклад децентралізованої системи - робота мережі BitTorrent або Torrent. У цьому випадку відсутній центральний вузол, і всі події розгортаються між усіма комп'ютерами. Суть BitTorrent або Torrent полягає в тому, що інформація поширюється частинами і зберігається в усіх користувачів. При включенні BitTorrent або Torrent користувач починає передавати інформацію.

Як у централізованій, так і у децентралізованій системі є свої переваги та недоліки.

До переваг децентралізованої системи відносяться:

1. Відсутність великої вразливості при виході з ладу частини мережі, що дозволяє системі продовжувати працювати.
2. Відсутність одноосібної цензури; можливість наявності цензури, але відсутність однієї центральної влади, що має право забороняти.
3. Ваші дані перебувають під вашим контролем, не зберігаються на сервері, і ви не залежите від інших учасників мережі.

До недоліків децентралізованої системи відносяться:

1. Відсутність визначених відповідальних суб'єктів, що робить складним призначення відповідальності.
2. Складність організації служби підтримки через відсутність визначених структур.

3. Складність в ухваленні рішень та проведенні змін у правилах чи протоколах, оскільки вимагається голосування.
4. Повільні процеси.

Нижче перераховані ключові аспекти децентралізації, пов'язані з новими технологічними та організаційними ризиками:

- Максимальне підвищення рівня незалежності кожного компонента системи.
- Збереження балансу між ефективністю та застосованими засобами.
- Децентралізація можлива лише за умови збереження цілісності всієї системи.

Важливо розуміти, що порівняння децентралізації та централізації не завжди є простим рішенням, оскільки це залежить від вимог та умов конкретного використання.

Першою успішною реалізацією технології Blockchain стала мережа Bitcoin, що використовується в основному для криптовалют. Проте потенціал технології Blockchain не обмежується лише цифровими грошима, і вона привертає увагу в різних галузях, таких як фінансові послуги, благодійні та некомерційні організації, мистецтво та електронна торгівля.

Децентралізація - це одна з найбільших переваг Bitcoin, яка дозволяє здійснювати транзакції без участі сторонніх осіб. Уряди, банки та фінансові посередники не мають можливості втручатися в транзакції користувачів. Bitcoin пропонує однорангову систему, що забезпечує більше свободи користувачам, оскільки валюта децентралізована і контролюється виключно користувачами, уникаючи стягнення зборів чи податків на транзакції.

У децентралізованих системах фінансового обліку, таких як криптовалюти, учасники зберігають копії однієї бази даних та оновлюють їх спільно за допомогою алгоритмів досягнення консенсусу. Blockchain, як розподілений цифровий реєстр, фіксує транзакції обміну цінностями. 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 представляє собою розподілену систему без централізованого контролю. У такій архітектурі користувачі, які завантажують один і той же файл, підключені один до одного, що дозволяє обмінюватися даними і розподіляти відповідальність за завантаження між учасниками, зменшуючи необхідність в центральному сервері, як у клієнт-серверній архітектурі. Фактично, кожен користувач виступає як сервер і клієнт одночасно.

Експерти можуть стверджувати, що P2P не слід характеризувати суворо в термінах централізації та децентралізації. Це вірно, оскільки централізація в P2P може існувати в різних формах, які можна класифікувати наступним чином:

Повністю централізований: наприклад, клієнт-сервер, де послуга доступна на одному хості, і клієнт, який потребує цю службу, повинен звертатися до цього конкретного хоста або сервера.

Посередництво: централізовані функції обмежені, але не всі. Типові централізовані функції включають операції з обліку, такі як реєстраційна інформація, моніторинг послуг і активний пошук серверів.

Повністю децентралізований: відсутність чіткої ієрархії, де обидва пристрої на одному рівні і відрізняються лише переданими даними або вмістом. Наприклад, Gnutella - розподілена мережа обміну файлами, де всі користувачі рівнозначні і відрізняються лише змістом, яким вони діляться.

Системи P2P були створені для різних застосувань, таких як розповсюдження вмісту, розподілені сховища та розподілені обчислення, і зараз вони є значною частиною інтернет-трафіку. Вони відзначаються здатністю до саморегулювання: нові учасники стимулюються надавати ресурси для компенсації додаткового робочого навантаження, яке вони створюють.

Протоколи Peer-to-peer (P2P) самостійно зберігають потрібні дані. Для цього вони використовують спеціальне програмне забезпечення, яке підтримує конкретний P2P-протокол для децентралізованої системи. Децентралізація також забезпечує високу відмовостійкість і стійкість до атак, оскільки відсутність єдиного пункту відмови уникає руйнування всієї системи. Також вона дозволяє швидке розгортання (бо для P2P-сервісу потрібно менше виділених ресурсів) і покращує конфіденційність користувачів.

1.2. Загрози на централізовані та децентралізовані системи

Існують певні ризики як для централізованих, так і для децентралізованих систем, які можуть впливати на їхню надійність. Для кращого розуміння загроз децентралізованим системам розглянемо спочатку приклади загроз централізованим системам, зокрема у банківському секторі. Реалізація таких загроз може призвести до фінансових втрат для банків, збитків у сфері прибутку, а також викликати соціальну чи психологічну напругу серед клієнтів або співробітників установи банку. Внутрішні та зовнішні загрози включаються до переліку загроз для банківського сектору. «Існує три складові загроз:

- безпека інформації: відмова абонента від факту прийому (передачі), виявлення паролів при викраденні або візуальному спостереженні, зміна або знищення даних на магнітних носіях, 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 на основі зв’язків між секретною та несекретною інформацією, несанкціоноване введення даних, несанкціоноване

використання інформації підвищеного рівня секретності, 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»[5]. «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»[5].

Модель загроз та модель порушника у таких системах можуть значно відрізнятися від своїх традиційних аналогів. Модель загроз є систематизованим описом можливих небезпек, тоді як модель порушника представляє собою структурний опис потенційного порушника.

Питання захисту персональних даних є ключовим у контексті технології Blockchain. Оскільки розподілений реєстр надає розподіленість, де ми не можемо контролювати місцезнаходження особистої інформації, вона може раптово з'явитися у всьому світі. У випадку технології Blockchain ми втрачаємо контроль над тим, де зберігаються особисті дані. З одного боку, це становить виклик для обробки персональних даних; з іншого боку, аналіз ризиків вказує на мінімальний ризик неправомірного використання цих даних через вбудовані в зашифровані блоки або хеші. Таким чином, хоча обробка персональних даних може бути незаконною, ризик неправомірного використання даних є мінімальним, що зменшує потребу в їхньому захисті.

Побудова надійної облікової системи передбачає визначення можливих проблем, з якими система може зіткнутися. Для цього розробляється політика

безпеки, яка включає в себе модель загроз, модель порушника та модель інформаційної безпеки системи.

Безпеку Blockchain можна розглядати з різних точок зору, включаючи безпеку мережі як інфраструктури та безпеку додатків Blockchain, зокрема смарт-контрактів. Історія блоків у Blockchain є захищеною, оскільки блоки залишаються незмінними після додавання до ланцюжка. Механізм ланцюжка пов'язує кожен блок з попереднім за допомогою хеш-показівників. Таким чином, будь-яке втручання в блок n миттєво ставить під загрозу дійсність всіх наступних блоків. Комбінація дерева Merkle та хеш-показівників забезпечує безпечну та ефективну модель даних, яка виявляє несанкціоновані зміни чи зловмисне втручання в реєстр Blockchain. 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.

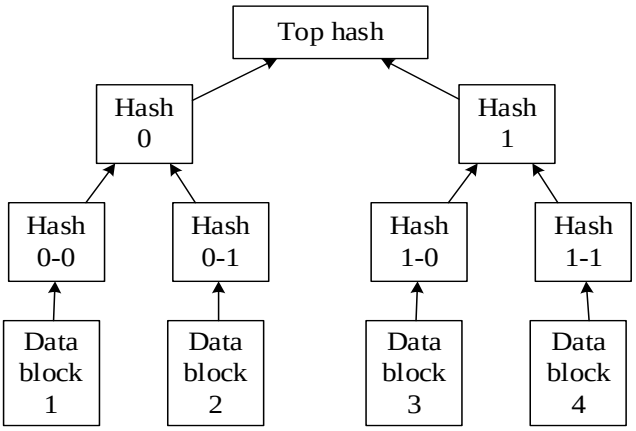


Рисунок 1.4 – Комбінація дерева Merkle

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

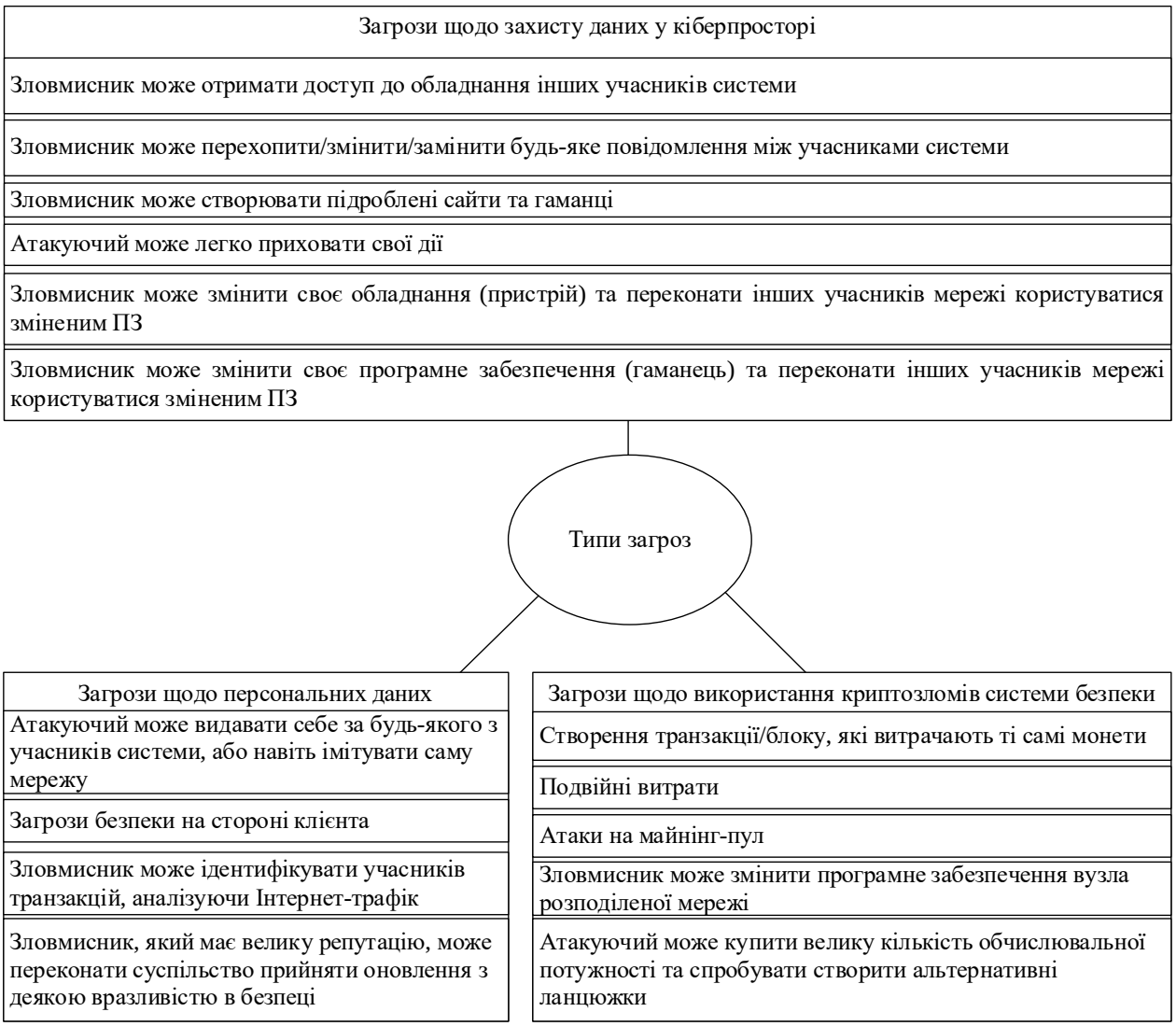


Рисунок 1.5 – Типи загроз

Атаки на майнінг-пули використовуються з метою збільшення обчислювальної потужності або геш-потужності блоку. Майнінгові пули створюються для майнінгу, впливаючи на час, необхідний для перевірки блоку. Ці майнінгові пулі також збільшують шанси на отримання нагороди за майнінг. (рис. 1.6).

Рисунок 1.6 – Атака на майнінг-пул

Пули для майнінгу постійно еволюціонують, і разом із цим зростає вразливість їх використання. Атаки на майнінг-пули можна класифікувати на два типи:

Внутрішні атаки:

Це ситуації, в яких майнер зловмисно збирає більше ресурсів, ніж потрібно, порушуючи нормальну функціональність. У результаті пул ігнорує успішні спроби майнінгу.

Зовнішні атаки:

Це випадки, коли майнер використовує вищу геш-потужність для атаки пула, що може призводити до подвійних витрат. Серед атак на майнінг-пул варто відзначити егоїстичний майнінг, стрибкоподібні атаки, утримання блоків, хабарництво та інші.

Загрози безпеки на стороні клієнта також є значущими, оскільки кожен користувач мережі Blockchain має набір ключів для доступу до свого гаманця. Необхідно ефективно управляти цими ключами, оскільки їх втрата чи компрометація може призвести до безповоротних грошових втрат.

Однією з основних загроз є зростання кількості злочинної діяльності, особливо в контексті використання Bitcoin. Програми-вимагачі, які вимагають виплату у Bitcoin, андеграундні ринки та відмивання грошей - усе це становить реальні загрози для безпеки мережі.

Програми-вимагачі, такі як CTB-Locker, використовують Bitcoin для анонімних виплат в обмін на відновлення зашифрованих даних. Підпільні ринки, такі як Silk Road, використовують Bitcoin як анонімну валюту для торгівлі контрольованими та незаконними продуктами.

З метою забезпечення відмивання грошей злочинці можуть використовувати Dark Wallet та інші інструменти, що ускладнюють відстеження транзакцій.

Криптовалюти продовжують розвиватися, але разом з цим з'являються нові виклики безпеки. Захист від злочинної діяльності вимагає постійного удосконалення технічних та організаційних заходів безпеки.

Основні атаки впливають на мережу Blockchain не прямолінійно. Вони спрямовані на викрадення активів, доступ до яких забезпечується за допомогою приватного ключа. Перш ніж вивчати ці загрози, слід розглянути принцип володіння цифровим активом, зокрема криптовалютою.

Кожна транзакція всередині блоку, така як переказ криптовалюти з одного гаманця в інший, підписується асиметричним електронним підписом. У громадській мережі, де учасники анонімні та не довіряють один одному, цей додатковий криптографічний шар є необхідним.

Використання асиметричного механізму цифрового підпису відрізняється від традиційного асиметричного шифрування. Підпис здійснюється закритим ключем, а перевірка – відкритим ключем (яку може виконати будь-який учасник). Обчислення значення відкритого ключа базується на закритому ключі, а обернене перетворення вимагає практично неможливих обчислень.

Приватний ключ генерується користувачем і дає доступ до адреси в Blockchain, де зберігається цифровий актив. Знаючи приватний ключ, користувач фактично володіє цим активом. Публічний ключ виступає як адреса гаманця та автентифікація підпису в інших блоках.

Атаки, спрямовані на отримання доступу до приватного ключа, не є прямими загрозами для Blockchain. Однак захист від цього ризику покращується через децентралізовані послуги, що відрізняються від традиційних бірж. Вони не

зберігають приватні ключі та особисті дані на своїх серверах і діють як посередники для зіставлення заявок на купівлю та продаж активів.

Технологія мультипідпису, що використовує кілька підписів для підтвердження транзакції, також додає додатковий рівень захисту гаманця. Це реалізується за допомогою смарт-контрактів.

Також важливо враховувати загрози безпеки, пов'язані з хмаровими сервісами, такими як інсайдерські атаки та вразливості інфраструктури. Ці аспекти вимагають постійного вдосконалення заходів безпеки.

1.3. Механізми забезпечення безпеки в децентралізованих системах

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

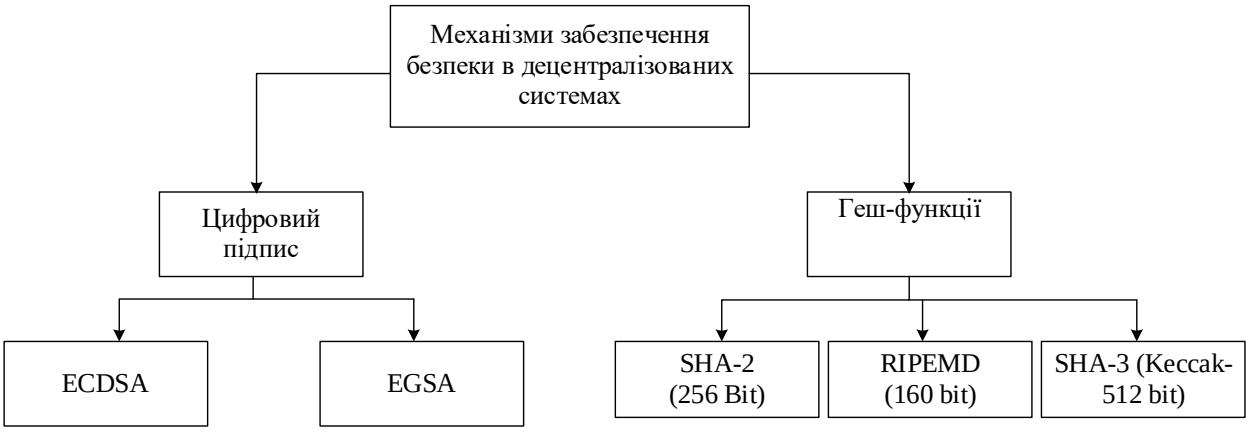


Рисунок 1.7 – Механізми забезпечення безпеки в децентралізованих системах

Геш-функції представляють собою механізм перетворення масиву вхідних даних будь-якої довжини в вихідний бітовий ряд фіксованої довжини за допомогою конкретного алгоритму. Основна характеристика геш-функції полягає в тому, що будь-яка зміна вхідних даних, навіть на один біт, призводить до повного змінення геш-значення.

Геш-значення виглядає як набір випадкових чисел і літер, що створює буквено-цифровий ідентифікатор або цифровий відбиток даних. Функція

гешування приймає вхідні дані і генерує унікальний геш. Оскільки геш служить "цифровим відбитком" всього блоку, дані включають індекс, мітку часу, попередній геш, дані блоку та одноразовий номер (nonce).

Основним криптографічним алгоритмом, який застосовується в технології Blockchain, є SHA-256. Цей алгоритм використовується для обчислення геш-значення під час майнінгу блоку, що є ключовим етапом в процесі майнінгу. Одноразове значення обчислюється відповідно до складності, встановленої в Blockchain.

SHA-256 є ефективним та поширеним криптографічним геш-алгоритмом, який широко використовується у шифруванні даних в Blockchain. Наприклад, Bitcoin використовує алгоритм RIPEMD-160, а Ethereum застосовує алгоритм SHA-3, зокрема Keccak-256 та Keccak-512.

Криптографічний геш-алгоритм є методом, який перетворює вихідні дані за допомогою спеціального алгоритму в інший тип нових даних фіксованої довжини. Алгоритм SHA-256, використовуваний в Bitcoin-Blockchain, є сильним і надійним, забезпечуючи велику стійкість шифрування. Його одностороння функція та чутливість до змін даних роблять його неперевершеним для захисту від різноманітних атак.

Важливим етапом у Blockchain є гешування відкритого ключа за допомогою SHA-256, а потім застосування RIPEMD160 для отримання унікального геш-значення довжиною 160 біт. Геш-функції використовуються в різних сферах інформаційної безпеки, включаючи генерацію та перевірку цифрових підписів, отримання ключів та генерацію псевдовипадкових бітів.

1.4. Використання децентралізованих систем

Існують різні сервіси, які дозволяють обмінювати криптовалюти на фіатні гроші та навпаки, і BTCX є прикладом такого сервісу. Проте цей конкретний сервіс обмінює лише bitcoin, що становить типове обмеження для багатьох інших бірж.

Гаманці для криптовалют необхідні користувачам для ефективного управління своїми фінансами. Це додаток, аналогічний традиційному гаманцю, де

гроші можна зберігати, доки не виникне потреба у їх витрачанні чи здійсненні інших фінансових операцій. Перший крок для користувача - визначити тип гаманця, який він хоче використовувати. Після цього слід створити обліковий запис, зазначаючи, що різні гаманці можуть вимагати різного рівня особистої інформації. Деякі вимагають ім'я та адресу електронної пошти, в той час як інші надають можливість анонімного користування без введення особистих даних.

Оскільки криптовалюта існує в електронній формі, користувачам потрібно використовувати спеціальний пристрій для її зберігання, що називається криптовалютним гаманцем. Ці гаманці можуть існувати в різних форматах (рис. 1.14).

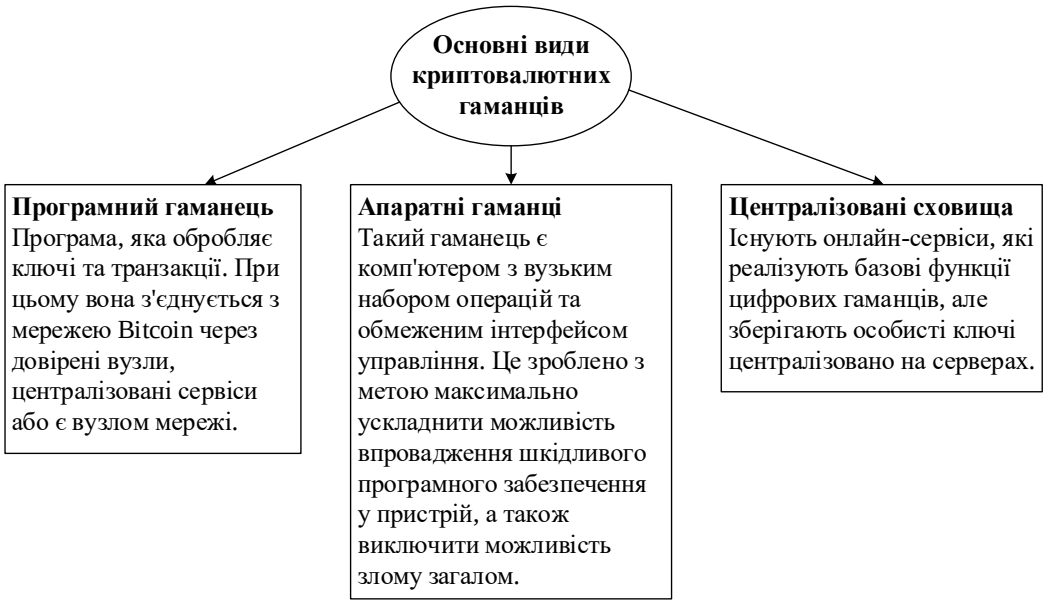


Рисунок 1.14 – Криптовалютні гаманці

Криптовалютні гаманці розділяються за категоріями в залежності від способу доступу до мережі, місця зберігання гаманця на носії, а також того, чи зберігається весь ланцюжок блоків або лише його фрагменти, а також чи ключі знаходяться локально чи на сервері (в хмарі). На рисунку 1.16 представлена спрощена ілюстрація основних компонентів, необхідних для проведення транзакції з використанням криптовалютного гаманця.

Сам гаманець не зберігає жодних монет, але зберігає та захищає приватний ключ, який необхідний для здійснення транзакцій і, отже, для використання монет. Інша інформація, така як геш відкритого ключа (який дорівнює адресі), історія транзакцій та кількість криптовалют, зберігається в Blockchain. Для захисту приватного ключа та інших конфіденційних даних, що зберігаються в гаманці, може використовуватися шифрування, яке може бути поєднане з механізмом блокування, щоб отримати доступ до програми потрібно ввести PIN-код або пароль.

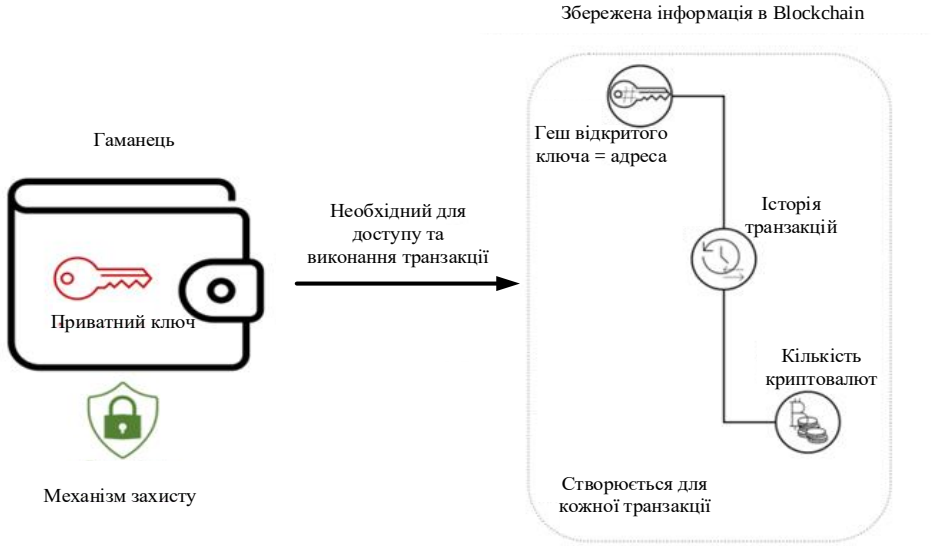


Рисунок 1.16 – Пояснення ролі гаманців у криптовалютних транзакціях

Висновки до розділу 1

Отже, в даному розділі розглянуто різноманітні аспекти криптовалютних гаманців. Визначено їхні категорії відповідно до способу доступу до мережі, умов зберігання та місця зберігання ключів. Головний акцент приділено компонентам, необхідним для здійснення транзакцій з використанням гаманців, зокрема захисту приватних ключів, їхнього зберігання та взаємодії з Blockchain. Розглянуто заходи безпеки, такі як шифрування та механізми блокування, які використовуються для забезпечення конфіденційності та недоступності ключів, важливих для виконання криптовалютних операцій.

2 АНАЛІЗ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ТА АВТЕНТИЧНОСТІ В ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМАХ

2.1. Дослідження механізмів забезпечення конфіденційності

Технологія Blockchain використовується для створення електронної книги з метою забезпечення безпеки, конфіденційності та цілісності. Її розподілення на всі повні вузли мережі гарантує найвищий рівень безпеки. Взаємодія мережі Bitcoin P2P допомагає забезпечити цілісність системи, реагуючи на неправильні дії вузлів. Використання повних перевіряючих вузлів, хоча не призводить до фінансових вигід, рекомендується заради довіри, безпеки та конфіденційності користувачів, адже вони гарантують дотримання правил, захист від атак та повний контроль користувача над своїми активами. Вузли виконують ключові функції, такі як виявлення та ретрансляція транзакцій, оновлення Blockchain і ретрансляція блоків транзакцій.

Блокчейни поділяються на категорії в залежності від авторизації мережевих вузлів і доступу до даних. Дозволені блокчейни делегують привілеї майнінгу центральному органу або консорціуму, тоді як публічні блокчейни відкриті для всіх. Приватні блокчейни обмежені доступом користувачів в організації або конкретній групі.

У публічних блокчейнах будь-хто може приєднатися, ініціювати транзакції та отримувати копії книги. Консенсус досягається шляхом об'єднання вузлів, які перевіряють блоки транзакцій і додають їх до блокчейну, забезпечуючи цифрові підписи та цілісність.

У всіх розподілених реєстрах необхідно забезпечити підтвердження транзакцій та досягти консенсусу. Проте Blockchain об'єднує ці кроки в процесі, відомому як майнінг (майнінг = перевірка транзакцій + консенсус). Майнінг, або підтвердження транзакцій, виконується майнерами, які виконують верифікацію. Цей процес використовується для захисту та підтвердження різноманітних транзакцій, чи то пов'язаних із криптовалютами, такими як Bitcoin, чи іншими видами записів. Майнери додають дані про транзакції до глобального журналу

завершених транзакцій під час майнінгу, створюючи блоки, які захищені та пов'язані між собою, утворюючи ланцюжок. Приклад ланцюжка блоків показано на рис. 2.1.

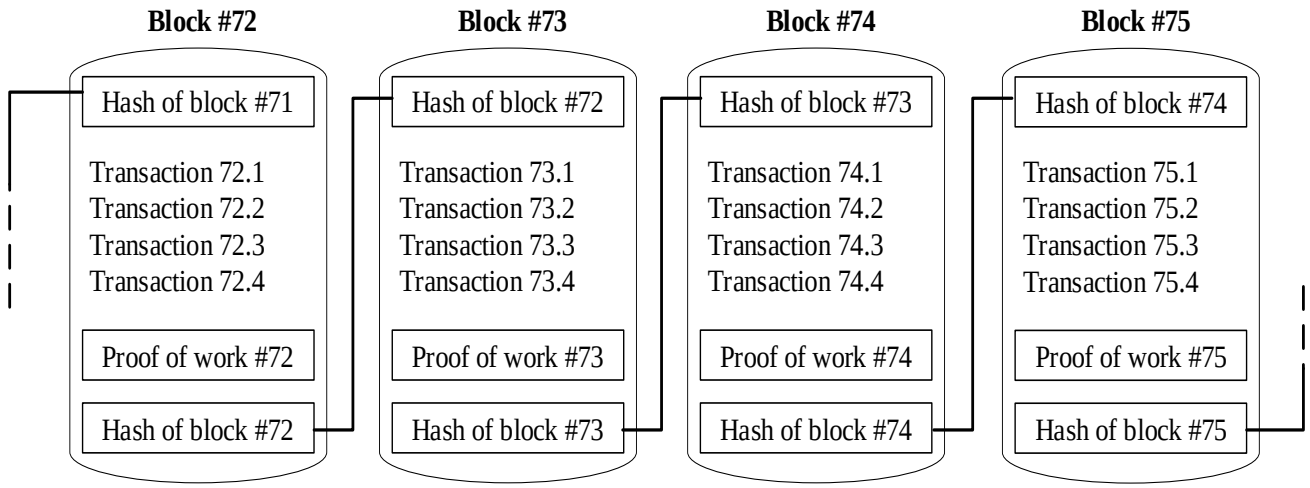


Рисунок 2.1 – Схема ланцюжка блоків

Blockchain використовує різноманітні механізми консенсусу, які різняться залежно від конкретної реалізації. Наприклад, у випадку Bitcoin він є загальнодоступним, що означає можливість будь-якого користувача придбати обладнання, приєднатися до мережі і розпочати майнінг. Інші Blockchain встановлюють певні вимоги для учасників процесу консенсусу, які заздалегідь визначені розробниками-засновниками.

Майнери, які беруть участь у майнінгу, обчислюють криптографічний підпис або водяний знак для останнього блоку транзакцій. У випадку Bitcoin використовуються стандартні геш-функції як водяні знаки, і консенсус визначається найдовшим ланцюжком блоків транзакцій із дійсними гешами. Майнери отримують нові Bitcoin у спеціальній транзакції, відомій як транзакція coinbase. Кожен Bitcoin можна простежити до відповідної транзакції з базою монет.

Роль криптографічних гешів важлива для забезпечення безпеки реєстру Bitcoin. Кожен новий блок транзакцій 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).

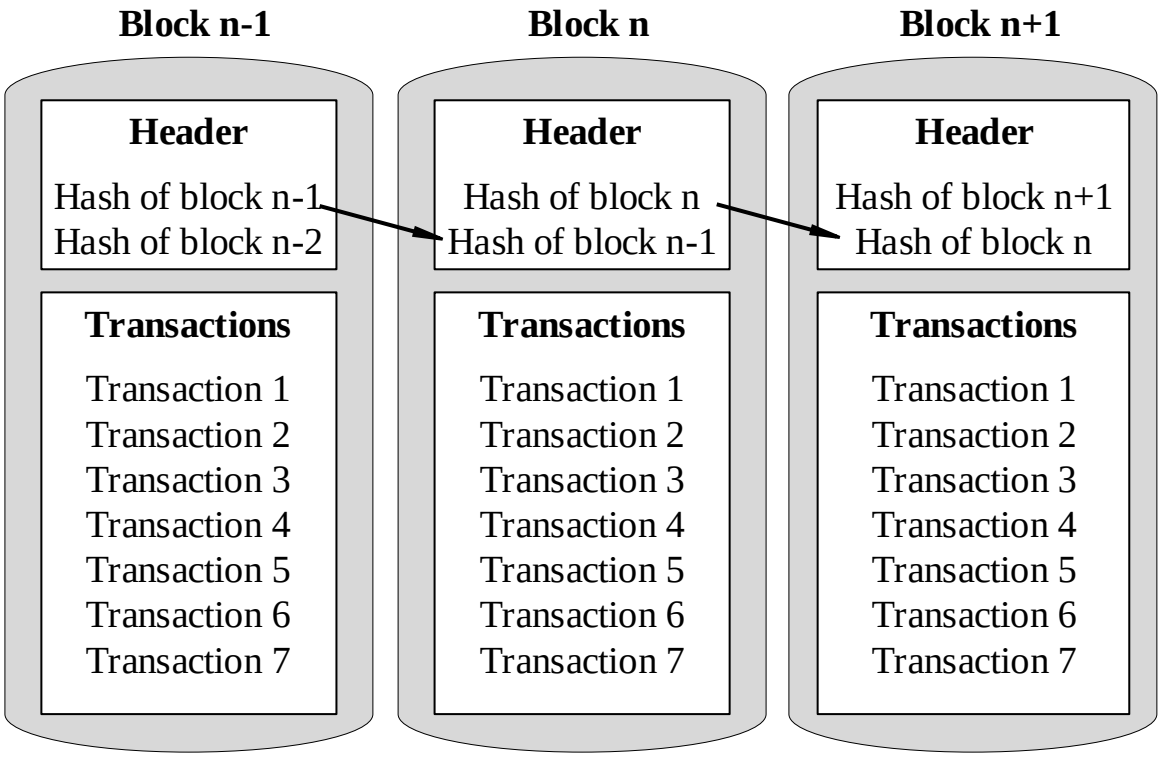


Рисунок 2.2 – Реєстрація транзакцій Bitcoin із взаємопов’язаними блоками транзакцій

Отже, будь-яка спроба змінити транзакцію в ланцюжку блоків вимагає перерахунку не лише гешу для конкретного блоку, але й усіх наступних гешів блоків. Якщо зловмисник не контролює 51% або більше майнерів, практично неможливо перерахувати альтернативний блокчейн зі зміненою транзакцією. Пов'язані геші гарантують цілісність ланцюжка блоків та усувають необхідність у довіреному центрі для ведення централізованого реєстру транзакцій.

Усі розподілені реєстри потребують підтвердження транзакцій, після чого має бути досягнутий консенсус.

Різні блокчейни використовують різні механізми консенсусу. Деякі, наприклад, біткойн, є загальнодоступними, де будь-хто може придбати обладнання, приєднатися до мережі і почати майнінг. Інші ж встановлюють певні

вимоги для учасників процесу консенсусу, які визначаються розробниками-засновниками.

Одним з протоколів консенсусу є "Proof-of-Work" (Доказ роботи), який використовується у блокчейні Bitcoin. В цьому процесі майнери вирішують складні завдання, щоб додати нові блоки до ланцюжка. Цей процес вимагає значних обчислювальних ресурсів і енергії.

Важливою проблемою Proof-of-Work є його велике споживання енергії. Існують різні пропозиції для зменшення цього енергетичного споживання, і однією з перспективних альтернатив є концепція "Proof-of-Stake".

Протоколи “доказу частки”:

Proof-of-Stake (PoS) є алгоритмом для досягнення розподіленого консенсусу в блокчейн-мережах. Запропонований як альтернатива Proof-of-Work (PoW), він призначений для подолання неефективного використання капітальних ресурсів, таких як обчислювальна потужність та енергія. Основна концепція PoS полягає в розподілі привілеїв майнінгу залежно від обсягу "ставок" учасника у мережі.

Існує кілька варіантів PoS, де привілеї майнінгу залежать від різних факторів, таких як власність монет, частота транзакцій і тривалість участі в мережі. Хоча на сьогодні немає повноцінної системи PoS, більшість експертів вважають, що це питання часу до появи першого успішного блокчейну, що використовує PoS. У разі успіху цей механізм може істотно знизити витрати енергії для утримання блокчейн-мереж і тиснути на зниження цін на електроенергію.

Delegated Proof-of-Stake (DPoS) дозволяє швидше створювати блоки та обробляти більше транзакцій за секунду, зменшуючи кількість валідаторів. Учасники мережі голосують за валідаторів, враховуючи вагу кожного голосу, яка визначається сумою активів голосуючого. Збереження вартості монет надає можливість проголосувати в будь-який момент, забезпечуючи високу стійкість мережі.

Byzantine Fault Tolerance (BFT) протокол Delegated Byzantine Fault Tolerance (DBFT) дозволяє системі продовжувати роботу, навіть коли вузол виходить з ладу. Цей протокол є вдосконаленою моделлю BFT, забезпечуючи швидкість та стійкість

мережі. Хоча це централізованіша модель, вона ефективно вирішує проблеми PoW та PoS, пропонуючи делеговані валідатори із спільнотою як механізмом контролю.

Practical Byzantine Fault Tolerance (PBFT) та Federated Byzantine Agreement (FBA) є двома різними протоколами консенсусу у блокчейні, які дозволяють досягнути узгодженості серед різних вузлів мережі. Давайте перефразуємо вихідний текст, щоб зрозуміти їхні ключові властивості та відмінності.

PBFT подібний до DBFT і відрізняється своєю більшою централізованістю. Основна відмінність полягає у простоті реалізації PBFT та його використанні у приватних мережах з відомими учасниками. У PBFT, коли валідатор отримує повідомлення, він приймає рішення, проводячи перевірки та опитуючи інші вузли. Якщо ⅔ учасників підтримують транзакцію, вона приймається та передається мережі. PBFT ефективний при низькій затримці, але чутливий до кількості валідаторів та пропускної спроможності.

З іншого боку, FBA не вимагає заздалегідь відомого набору учасників і дозволяє приєднатися до мережі будь-кому. Транзакції в FBA валідуються фіксованою кількістю вибраних учасників у мережі. Протокол використовує два набори даних - глобальний для контролю цілісності та локальний для оцінки ефективності. При більшості голосів модель кандидата приймається.

Важливо зауважити, що в FBA існують шлюзи та мейкери, які забезпечують чесність та ліквідність мережі. Шлюзи володіють фінансовими коштами та створюють їх віртуальні еквіваленти, тоді як мейкери ведуть облікові записи в різних валютах.

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)»[7].

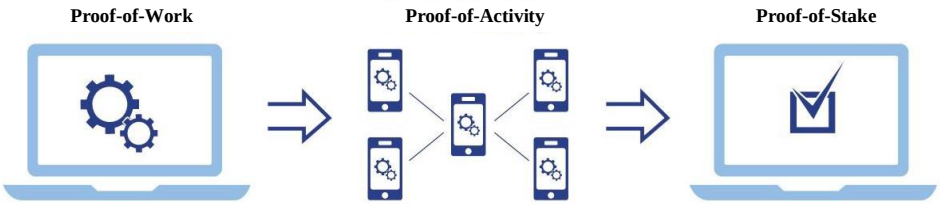


Рисунок 2.11 – Proof-of-Activity

Proof-of-Location (PoL): цей механізм дозволяє користувачам закріпити за собою конкретну GPS-локацію та, таким чином, підтверджувати свою автентичність у мережі. Цікаво, що протокол використовує BFT-маячки (beacons), які реєструють геолокацію та часові маркери у блокчейні, що забезпечує надійність та запобігає можливим збоям та шахрайству в системі.

Протоколи PoL включають в себе процедуру, яка вставляє проміжних посередників між користувачами. Ці посередники - це пристрої, які виконують протокол. Важливо зауважити, що навіть якщо ці пристрої належать фізичним користувачам, вони повинні бути присутніми поруч із користувачем у момент створення підтвердження. В результаті, щоб забезпечити повний PoL для користувача пристрою, необхідно встановити три кореляції (рис. 2.12).

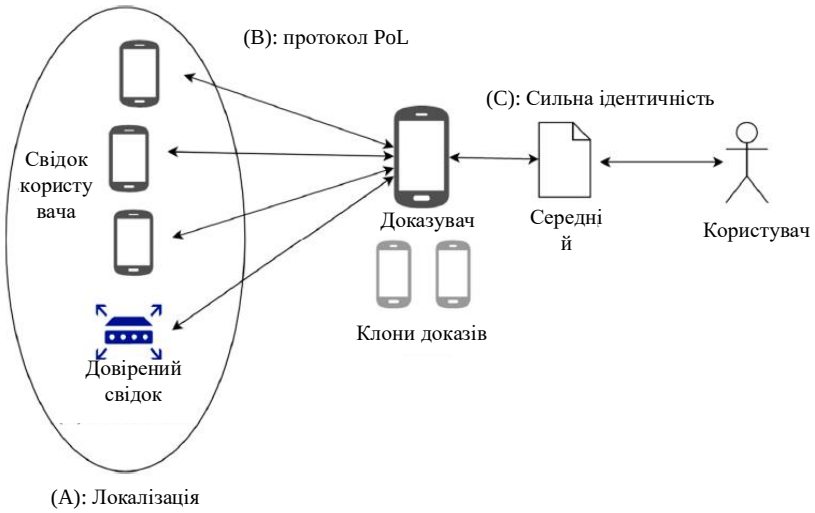


Рисунок 2.12 – Proof-of-Location

Кореляція A пов'язує свідків із конкретним місцем, що може бути досягнуто за допомогою надійних пристроїв із відомим місцезнаходженням або пристроїв

інших користувачів, які визначають свою локацію, наприклад, за допомогою GPS. Сам протокол PoL може надати необхідну інфраструктуру та механізм локалізації.

Співвідношення B встановлює зв'язок між пристроєм користувача (доказувачем) та свідками через протокол PoL. Його мета полягає в тому, щоб підтвердити, що доказувач перебуває поруч із свідками в конкретний час. Більшість протоколів PoL в бібліографії обмежені лише цією кореляцією, підтверджуючи, що пристрій із обліковими даними користувача знаходиться біля свідків. Однак може бути багато пристроїв перевірок із однаковими обліковими даними користувача, тому підключення B не прив'язує користувача до конкретного місцезнаходження.

Кореляція C пов'язує користувача із пристроєм у певний час, коли виконується протокол PoL. Для задоволення цього зв'язку розвідник створює носій під час сесії протоколу. Цей носій підтверджує, що під час створення PoL він перебуває у розпорядженні користувача. Це відомо як "сильна ідентифікація", оскільки вона підтверджує не лише місцезнаходження пристрою, але й самого користувача. Протокол PoL може відповідати всім вищезазначеним зв'язкам, принаймні, він повинен задовольнити B.**Proof-of-Importance (PoI):** алгоритм консенсус PoI діє майже як PoS, але включає три компоненти (рис. 2.13).

Рисунок 2.13 – Proof-of-Importance

Хоча значення першого параметра важливе для рейтингу при перевірці транзакцій, другий і третій параметри виявляються досить слабкими, проте вони сприяють визначенню "важливості" облікового запису. Чим менше сума токенів, тим суттєвіший вплив інших параметрів. Отже, обліковий запис з сотнями тисяч токенів може збільшити коефіцієнт значущості майже втричі завдяки своїй активності та постійній присутності в мережі. З іншого боку, це не має значення для тих, хто володіє сотнями мільйонів токенів у своєму обліковому записі.

Proof-of-Elapsed-Time (PoET): Виробник чіпів Intel, не відстаючи, розробив свій власний блокчейн, який отримав назву IntelLedger. Алгоритм консенсусу IntelLedger, відомий як Proof-of-Elapsed-Time (PoET) або "доказ минулого часу", схожий на Proof-of-Work, але ефективно використовує енергію(рис. 2.14). Замість вирішення криптографічної головоломки, алгоритм працює в середовищі Trusted Execution Environment (TEE), такому як Intel Software Guard Extensions (SGX).

Рисунок 2.14 – Proof-of-Importance

Протокол PoET також забезпечує те, що блоки створюються випадковим чином, але без необхідності виконання будь-якої додаткової роботи.

Як альтернативу, Intel пропонує рішення вигляду гарантованого часу очікування, що базується на Trusted Execution Environment (TEE). За інформацією компанії, алгоритм PoET може масштабуватися до тисяч вузлів і ефективно працювати на будь-якому процесорі Intel, що підтримує SGX.

2.2. Дослідження механізмів забезпечення цілісності

Використання технології блокчейн у Bitcoin спрямоване на забезпечення високого рівня безпеки, вимагаючи уваги до всіх аспектів обробки даних, таких як зберігання та передача.

Цілісність даних виступає ключовою характеристикою, яка підвищує довіру до технології блокчейн. Це досягається завдяки розумній ідеї криптографії, в основі якої лежить механізм консенсусу. Для досягнення консенсусу мережа блокчейн використовує механізм доказу роботи, який гарантує, що зміна будь-якої одиниці інформації вимагає великої обчислювальної потужності для переоцінки всієї

мережі. Крім того, блокчейн пропонує рішення для проблеми подвійних витрат, що визнається як важлива уразливість, що порушує цілісність системи.

Впевненість у цілісності даних в блокчейн залежить від трудомістких завдань Proof-of-Work у процесі майнінгу. Проте майнінг призводить до неефективності зберігання даних, включаючи високу затримку підтвердження та низьку загальну пропускну здатність. Це створює проблеми забезпечення цілісності.

Для вирішення цього виклику введено трирівневу мережу блокчейн. На першому рівні перевіряється політика доступу, а другий шар представляє собою пропонований реєстр безпеки на основі блокчейн, який покращує політику безпеки. У третьому рівні виконується перевірка безпеки для відповідних пакетів даних віртуальної машини. Ця робота включає поділ конфіденційних та нечутливих даних, дозволяючи використовувати різні типи блокчейнів, такі як загальнодоступний, консорціум та приватний, залежно від потреб користувачів.

Висновки до розділу 2

Отже, у розділі проаналізовано ключові аспекти технології блокчейн та її застосування в різноманітних сферах. З'ясувано, що головна перевага блокчейну полягає в створенні відкритої, надійної та загальнодоступної системи, що виникла з концепції Bitcoin. Технологія блокчейн вирішує проблему довіри в ненадійних умовах, уникнувши потреби у посередниках та спираючись на концепцію proof-of-work для досягнення консенсусу.

Крім того, розглянуто гарантії, які пропонує технологія блокчейн, і які можуть служити будівельними блоками для різноманітних застосувань залежно від їхнього призначення та вимог. Найпопулярніші протоколи, які застосовуються в більшості проєктів блокчейн, розглянуті для кращого розуміння їхніх особливостей та можливостей.

Розглягуто сутність технології блокчейн та її потенціал у різних галузях, відкриваючи широкі можливості для подальших досліджень та впроваджень в інноваційних проєктах та застосунках.

3 РЕАЛІЗАЦІЯ СТВОРЕННЯ ТА ВИКОРИСТАННЯ КРИПТОВАЛЮТ НА ОСНОВІ ДЕЦЕНТРАЛІЗОВАНИХ СИСТЕМ

3.1. Вибір програмних засобів для реалізації майнінгу

З використанням алгоритму гешування SHA-256, який використовується для майнінгу та додавання монет у ланцюжок Bitcoin, а також для створення Bitcoin-адрес, я розробляю програму, яка демонструє процес додавання блоків у ланцюжок та вміст кожного блоку. Майнінг включає в себе установку числа так, щоб результат гешування був менше певного значення. Винагорода майнера складається з Bitcoin. Додатково, отриманий блок додається до ланцюжка блоків.

Для реалізації цього програмного продукту використано середовище розробки Microsoft Visual Studio 2019. Visual Studio - це інтегроване середовище, яке служить для написання, відлагодження та збірки коду, а також для публікації програм. Зокрема, воно має редактор, відладчик, компілятори, автозавершення коду та інші зручні функції для поліпшення розробки.

Мова програмування C# використана для реалізації програмного продукту. C# - це об'єктно-орієнтована мова програмування, що дозволяє створювати потужні та багатофункціональні програми. Завдяки об'єднанню ідей з Java, C++, Visual Basic та інших мов, C# дозволяє розробляти програмні рішення швидше.

Postman використовується як HTTP-клієнт для тестування API, де можна відправляти запити на сервер та отримувати відповіді. Це корисний інструмент для тестування API та взаємодії з сервером, а також для створення mock-серверів для імітації роботи програм.

«За допомогою Postman тестувальник може:

- складати та відправляти HTTP-запити до API;
- створювати колекції (набір послідовних запитів) та папки запитів для скорочення часу тестування;
- змінювати параметри запитів (наприклад, ключі авторизації та URL);
- змінювати оточення для запитів (наприклад, на тестовому стенді, локально або на сервері);

- додавати під час виклику API контрольні точки (фіксацію моменту передачі);
- проводити автоматизоване тестування API з колекції запитів за допомогою Collection Runner.

Для роботи із серверами програма використовує протокол HTTP. Тестувальник відправляє тестові запити від клієнта на сервер та отримує відповідь, чи є помилка в роботі API»[8].

3.2. Програмна реалізація імітація майнінгу криптовалют

Для створення симуляції майнінгу в Blockchain необхідно заздалегідь налаштувати середовище розробки та додаткове програмне забезпечення. Покрім встановлення Microsoft Visual Studio 2019 (див. рис. 3.1), також потрібно встановити HTTP-клієнт, який я обрав у вигляді Postman (див. рис. 3.2).

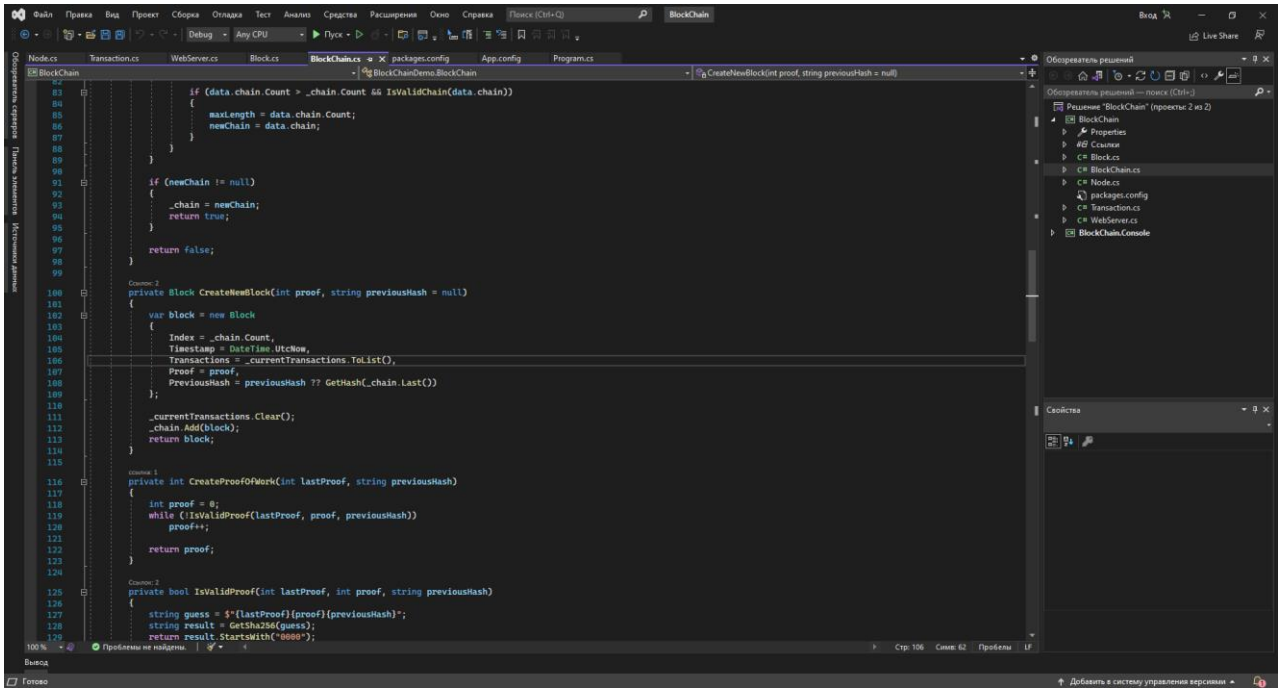


Рисунок 3.1 – Інтерфейс Microsoft Visual Studio

На початку був створений клас для реалізації Blockchain, який містить початковий порожній лист для зберігання Blockchain, який зберігає транзакції. Даний клас забезпечує роботу з нашим Blockchain, відповідає за зберігання 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
}
```

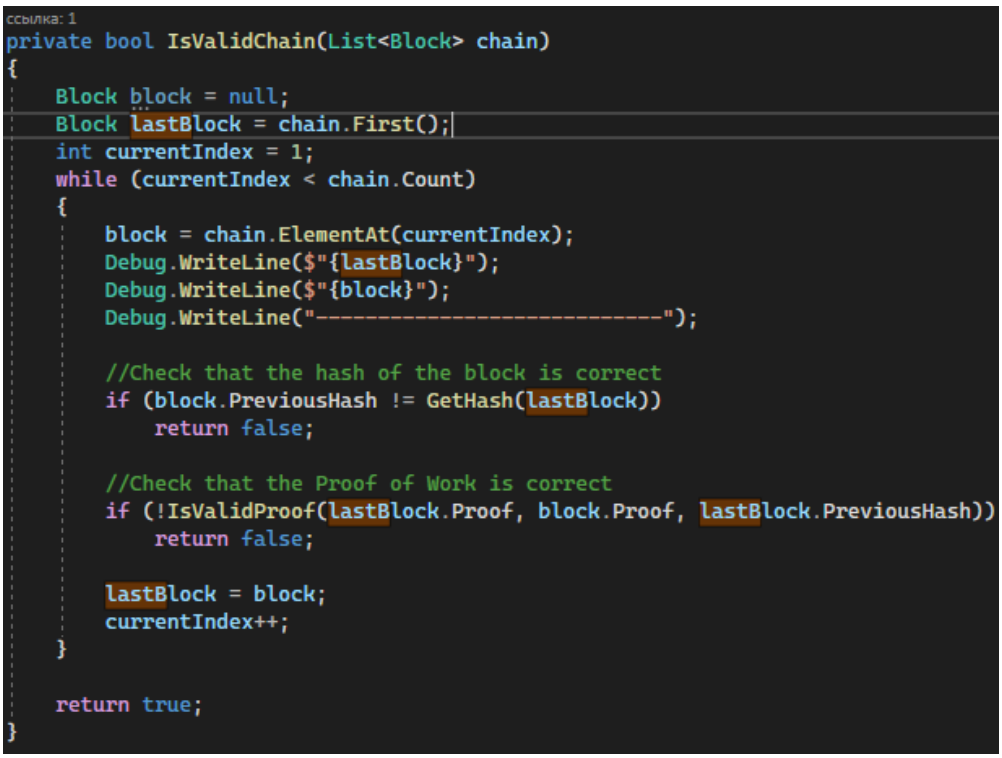
Рисунок 3.3 – Приклад коду

```
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();
}
```

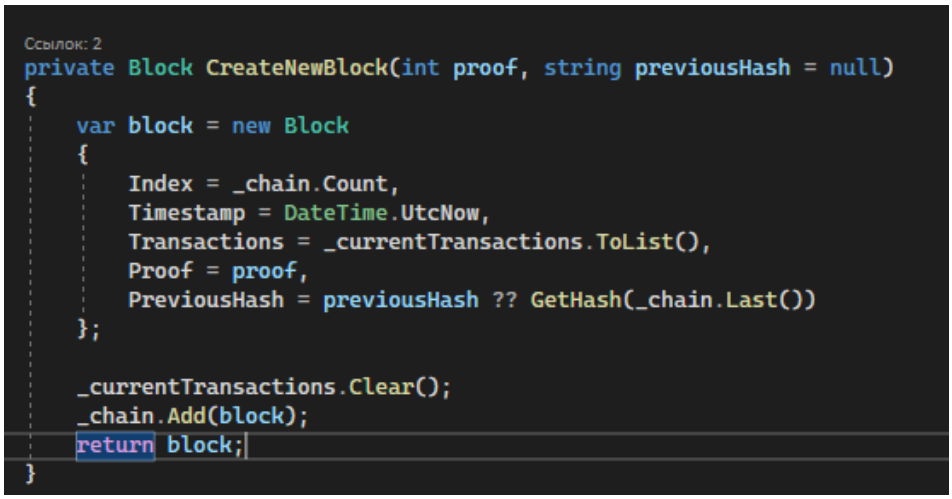
Рисунок 3.5 – Приклад коду



```
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

Давайте детальніше розглянемо реалізацію програмного коду. Згідно з визначенням Blockchain, у кожному блоку потрібно зберігати наступну інформацію: індекс блоку, часову мітку, список транзакцій, підтвердження консенсусу та геш попереднього блоку (див. 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) важливий для внесення нових транзакцій у блок. Заносить транзакцію до списку і повертає індекс блоку, до якого буде заноситись транзакція.

```
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 у блоці має бути доказ консенсусу, який надає результат майнінгу.

CreateProofOfwork – реалізує алгоритм доказу роботи (рис. 3.10), який схожий на алгоритм Hashcash в Bitcoin.

```
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 Blockchain відбувається шляхом використання HTTP-запитів. Для цього необхідно створити три додаткові методи:

- /transactions/new - для створення нової транзакції в блоку;
- /mine - для ініціювання сервером початку майнінгу нового блоку;
- /chain - для отримання повної інформації про Blockchain.

Також необхідно розробити функцію, яка відповідає за обробку транзакцій.

```
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

На вузлі потрібно здійснити майнінг нового блоку. Для цього необхідно вирішити завдання за допомогою алгоритму доказу роботи (PoW), видачі винагороди майнеру за вирішення криптографічного завдання (1 монета) і створення нового блоку з включенням його до ланцюжка блоків.Для взаємодії з Blockchain можна використовувати клієнт HTTP-запитів Postman. Спочатку необхідно запустити сервер вузла мережі Blockchain. Для запуску процесу майнінгу блоку необхідно надіслати GET-запит серверу вузла. Результат виконання запиту на майнінг блоку подано на (рис. 3.12).

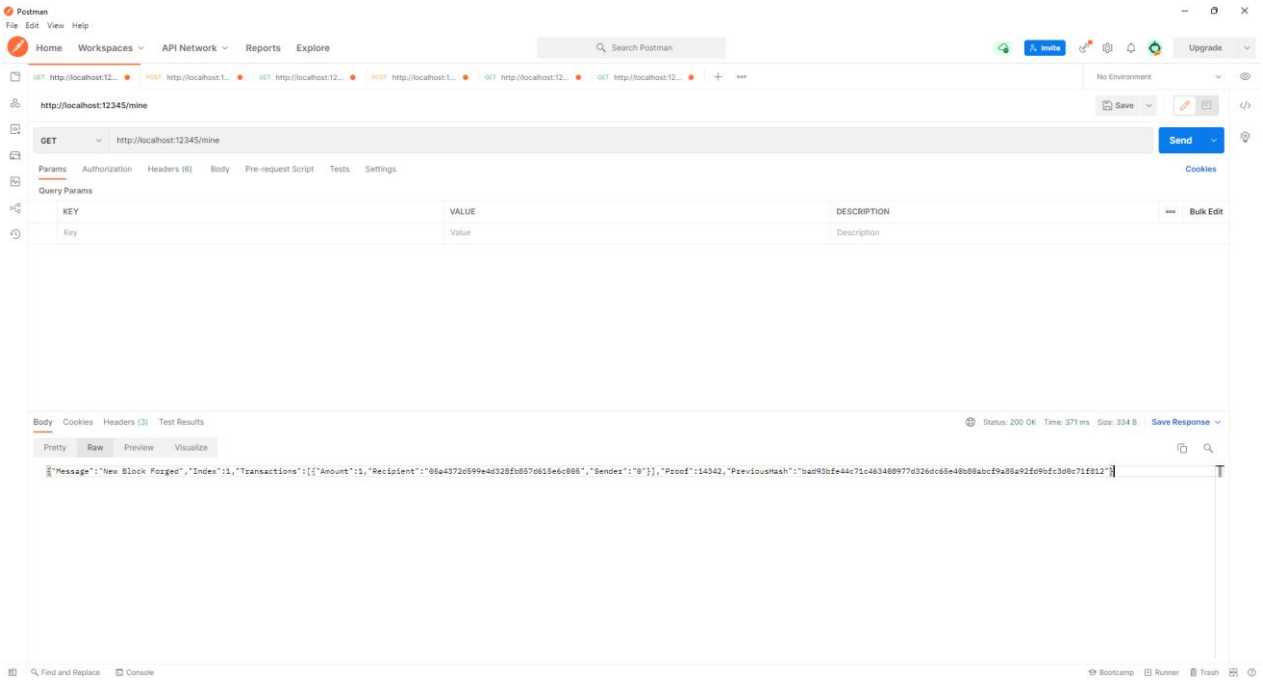


Рисунок 3.12 – Запит на майнінг блоку та результат виконання запиту

Для створення нової транзакції необхідно надіслати POST-запит до вузла мережі з тілом, яке містить параметри транзакції (рис. 3.13).

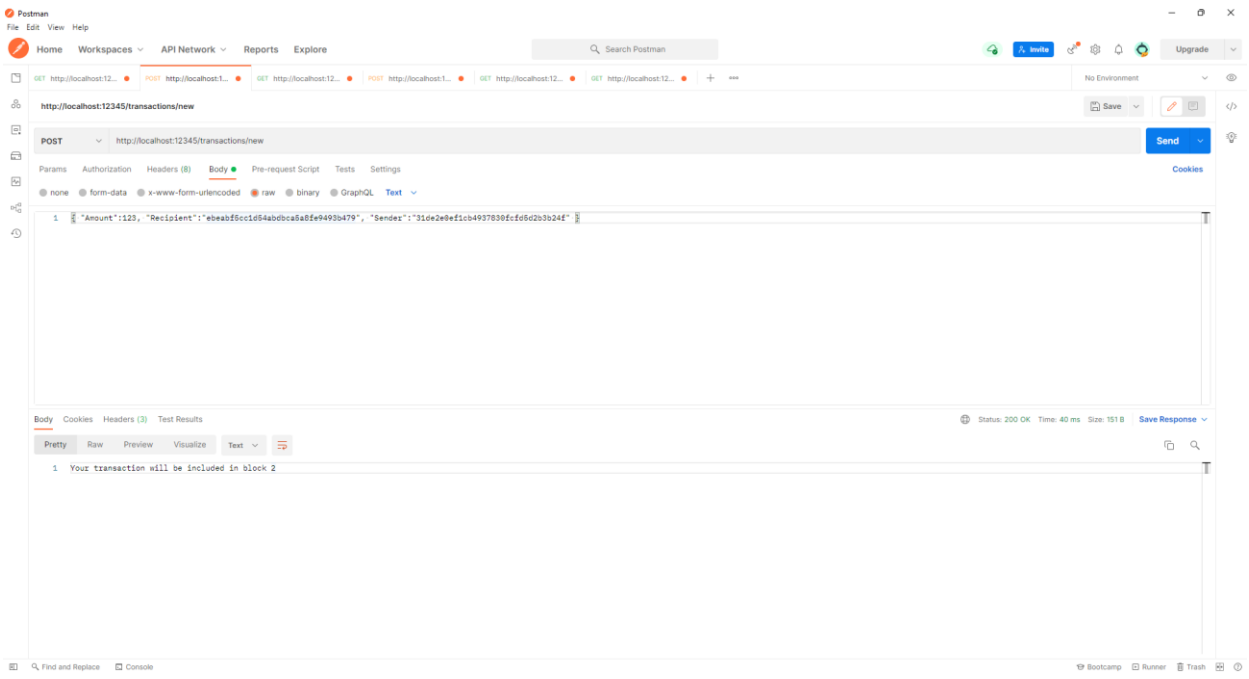


Рисунок 3.13 – Запит на виконання транзакції

Для втілення розподіленої мережі вузлів необхідно розробити програмний код, який відповідає за реєстрацію нових вузлів у мережі та втілення алгоритмів консенсусу. Код реалізації цих 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

Відповідно до алгоритму консенсусу PoW валідним ланцюжком є найдовший ланцюжок. Цей ланцюжок є авторитетним на цьому вузол. Метод ResolveConflicts дозволяє конфлікти та замінює ланцюг 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

Далі реєструються кінцеві точки 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

Для проведення транзакцій та перевірки роботи алгоритмів консенсусу необхідно запустити два нових вузли на комп’ютерах мережі та виконати їх реєстрацію (рис. 3.17 – 3.18).

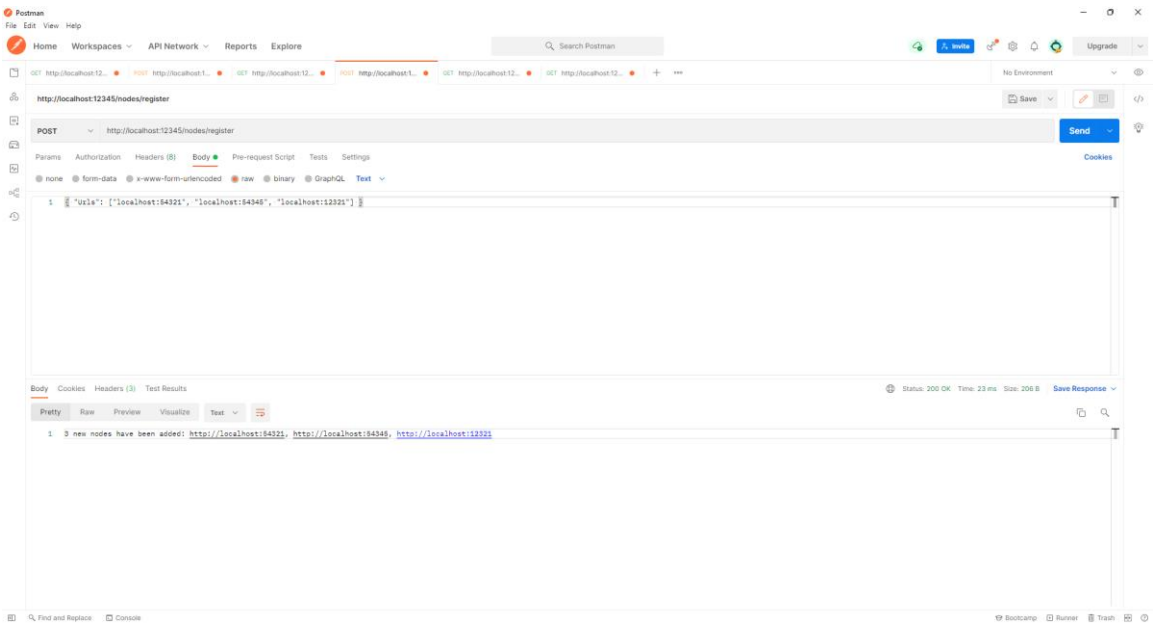


Рисунок 3.17 – Реєстрація нового вузла мережі

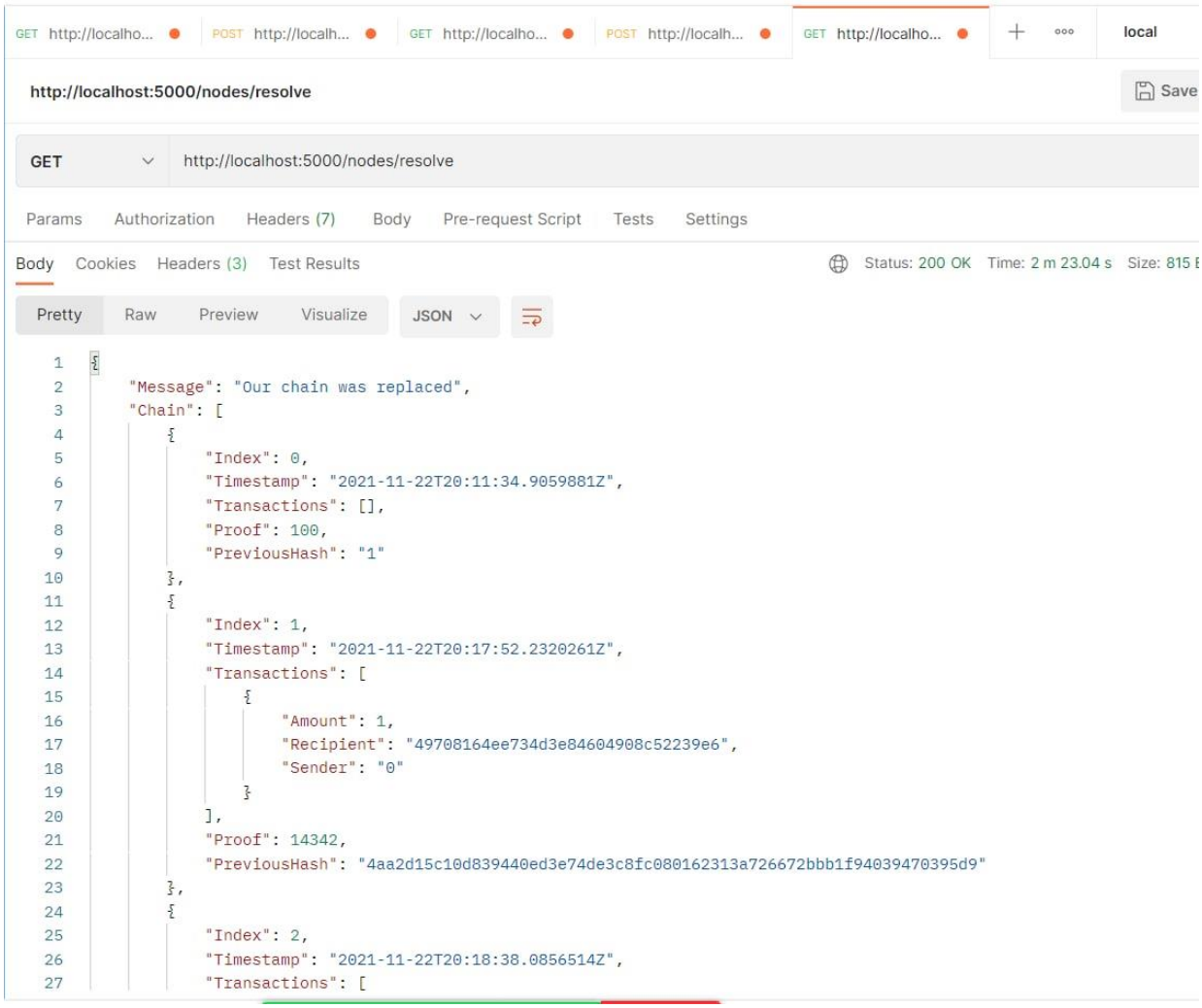


Рисунок 3.18 – Виконання транзакції та перевірка алгоритму консенсусу

Висновки до розділу 3

У цьому розділі було втілено процес створення та використання криптовалют на основі децентралізованих систем. Основною метою цього розділу було створення імітації процесу майнінгу криптовалют та розробка методу створення розподіленої криптовалюти, який уникає проблем централізації та вузьких місць, що часто виникають у зв'язку з впровадженням технології blockchain. Було продемонстровано, що цей процес гарантує захист кожної попередньої транзакції після короткого періоду зближення для кожної окремої операції перевірки. Це створило наочний приклад того, як виглядає структура blockchain та як протікає процес майнінгу.

ВИСНОВКИ

Метою даної роботи є проведення аналізу системи методів оцінки поточного рівня безпеки децентралізованих систем, що базується на аналізі сучасних загроз та механізмів їх запобігання, а також програмна реалізація процесу створення криптовалют на основі децентралізованих систем з використанням технології блокчейн. Робота включає в себе створення криптовалют, аналіз безпеки в децентралізованих системах, і вивчення механізмів забезпечення конфіденційності та автентичності.

У ході дослідження розглядаються аспекти аналізу безпеки децентралізованих систем, включаючи механізми забезпечення конфіденційності та автентичності. Також докладно розглядаються питання, пов'язані з побудовою криптовалютних бірж, створенням технології блокчейн та смарт-контрактів, їх функціонуванням, а також тренди розвитку та можливі ризики. Особлива увага приділяється реалізації програмного продукту для створення криптовалют.

Перша частина роботи фокусувалася на аналізі безпеки децентралізованих систем і охоплювала теми, такі як основні принципи їх конструкції, потенційні загрози та застосування різних алгоритмів гешування для забезпечення безпеки.

У другому розділі досліджувалися механізми, які гарантують конфіденційність, автентичність та цілісність в децентралізованих системах.

Третя частина включала в себе практичне втілення процесу створення та використання криптовалют на основі децентралізованих систем, а також оцінку поточного рівня їх безпеки.

Технологія блокчейн все ще знаходиться на етапі розвитку. Початковою метою було вирішити проблему подвійних витрат у цифрових валютах, що веде до створення цифрових валют, що не потребують централізованого контролю. Bitcoin, в основі якого лежала концепція прямих цифрових транзакцій, став першим успішним прикладом вирішення цієї проблеми. Проте, після його запуску стало зрозуміло, що потенціал технології блокчейн виходить за межі області цифрових

валют. Основна сила полягає в здатності блокчейну уникати ризиків, пов'язаних із контрагентами та необхідністю фінансового посередництва в цифрових транзакціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

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

3. 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.

4. 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.

5. 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

6. 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.

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

8. 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.)

9. 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.

10. 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.).

11. 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.).

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

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

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

15. 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.

16. 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.

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

18. 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.