МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ВАСИЛЬКІВ Владислав Олександрович

Алгоритми захисту даних Інтернет речей на основі блокчейн технології / Data Security Algorithms for Internet of Things Based on Blockchain Technology

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
В.О. Васильків

Науковий керівник
д.т.н., професор В.В.Яцків

Кваліфікаційну роботу
допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2023

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
«____» _____ 2022 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ВАСИЛЬКІВ Владислав Олександрович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Алгоритми захисту даних Інтернет речей на основі блокчейн технології / Data Security Algorithms for Internet of Things Based on Blockchain Technology
керівник роботи д.т.н., професор В.В. Яцків

затверджені наказом по університету від 1 грудня 2022 року № 491

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– провести аналіз алгоритмів консенсусу блокчейну;
– дослідити вразливості алгоритмів консенсусу в блокчейні;
– проаналізувати переваги та недоліки дерев Меркла для захисту цілісності даних Інтернет речей;
– дослідити рівні безпеки Інтернет-речей в Hyperledger-Fabric;
– розробити алгоритми захисту даних Інтернет -речей на основі технології блокчейн.

5. Перелік графічного матеріалу у роботі:
– Архітектура безпеки IoT на основі блокчейну;
– Застосування Hyperledger Fabric в IoT;
– Процес взаємодії вузлів;

– Процес шифрування та передачі даних IoT;
– Структура блоку даних;
– Розшифровування даних і процес перевірки IoT;
– Пропускна здатність транзакції та затримка зв’язку.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз технології блокчейн	12.2022 р. – 03.2023 р.	
2	Безпека Інтернет - речей на основі блокчейну	03.2023 р. – 05.2023 р.	
3	Архітектура Інтернет-речей на основі блокчейну	05.2023 р. – 11.2023 р.	

Студент _____ Васильків В.О.
(підпис)

Керівник роботи _____ д.т.н., професор В.В. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Алгоритми захисту даних Інтернет речей на основі блокчейн технології» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 81 сторінка і містить 17 ілюстрації, 1 додаток та 37 джерел за переліком посилань.

Метою роботи є підвищення захисту даних Інтернет речей на основі технології блокчейн.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи захисту даних, теорію алгоритмів та структур даних, алгоритми консенсусу, методи оцінки складності.

Результати дослідження: Розроблено алгоритм захисту даних Інтернету - речей в якому шифрування здійснюється відкритим ключем і тим самим немає необхідність зберігати на кінцевих пристроях закритий ключ.

Удосконалено алгоритм захисту даних Інтернет-речей заснований на технології блокчейн, який для зберігання, аналізу та обробки даних використовує шлюзові пристрої, що полегшує шифрування та зберігання даних сенсорів.

Результати роботи можуть бути застосовані при розгортанні захищеної системи збору даних з розподілених пристроїв з використанням технології блокчейн.

Ключові слова: ІНТЕРНЕТ-РЕЧЕЙ, БЛОКЧЕЙН, АЛГОРИТМ КОНСЕНСУСУ, ШИФРУВАННЯ ДАНИХ.

ABSTRACT

Qualification work on "Data Security Algorithms for Internet of Things Based on Blockchain Technology" for the degree of "Master" in the specialty 125 "Cybersecurity" educational and professional program "Cybersecurity" is written in 72 pages and contains 17 illustrations, 1 appendice and 37 source according to the list of links.

The purpose of the work is to improve data protection of the Internet of Things based on blockchain technology.

Research methods. To solve the set problems in this qualification work, the following methods were used: data protection methods, the theory of algorithms and data structures, consensus algorithms, complexity assessment methods.

Research results: An algorithm for data protection of the Internet of Things has been developed, in which encryption is carried out with a public key and thus there is no need to store a private key on the end devices.

The improved data protection algorithm of the Internet of Things is based on blockchain technology, which uses gateway devices for data storage, analysis and processing, which facilitates the encryption and storage of sensor data.

The results of the work can be applied when deploying a secure data collection system from distributed devices using blockchain technology.

Keywords: INTERNET OF THINGS, BLOCKCHAIN, CONSENSUS ALGORITHM, DATA ENCRYPTION.

ЗМІСТ

ВСТУП

Актуальність роботи. Технологія блокчейн є однією з найбільш масштабних технологій в наш час. Хоча першим застосуванням технології блокчейн був біткойн як криптовалюта, інші застосування цієї технології також привернули увагу урядового та промислового секторів. Згідно з опитуванням Всесвітнього економічного форуму до 2027 року блокчейн зберігатиме десять відсотків світового ВВП.

Технологія блокчейн була вперше представлена групою дослідників і до заснування Bitcoin Сатоші Накамото в 2008 році вона не мала звичайних застосувань. Однак слід зазначити, що останніми роками вона використовується в різних сферах, таких як біомедична, управління ланцюгом поставок і реєстрація інтелектуальних контрактів.

Зараз масштаби підключених пристроїв Інтернету речей (IoT) експоненціально зростають, і, за прогнозами, до 2025 року кількість підключених пристроїв досягне від 20 до 50 мільярдів [1]. Однак, оскільки все більше і більше інтелектуальних пристроїв підключаються до мережі, згенеровані дані вразливі до мережевих атак у процесі передачі та взаємодії, що призводить до модифікації даних або зловживання, таким чином загрожуючи безпеці всієї системи IoT. Крім того, завдяки різноманітній інтеграції пристроїв, мереж і служб дані, що зберігаються на пристрої, є вразливими до порушення конфіденційності через компрометацію вузлів, існуючих у мережі IoT [2]. Отже, захист даних Інтернет речей залишається актуальною задачею.

Мета і завдання дослідження. Метою роботи є підвищення захисту даних Інтернет речей на основі технології блокчейн.

Досягнення визначеної мети передбачає вирішення таких завдань:

– провести аналіз алгоритмів консенсусу блокчейну;

– дослідити вразливості алгоритмів консенсусу в блокчейні;

– проаналізувати переваги та недоліки дерев Меркла для захисту цілісності даних Інтернет речей;

– дослідити рівні безпеки Інтернет-речей в Hyperledger-Fabric;

– розробити алгоритми захисту даних Інтернет -речей на основі технології блокчейн.

Об’єкт дослідження – процеси забезпечення цілісності та конфіденційності даних Інтернет – речей;

Предмет дослідження – алгоритми забезпечення цілісності та конфіденційності даних Інтернет – речей на основі технології блокчейн.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи захисту даних, теорію алгоритмів та структур даних, алгоритми консенсусу, методи оцінки складності.

Наукова новизна одержаних результатів. Розроблено алгоритм захисту даних Інтернету - речей в якому шифрування здійснюється відкритим ключем і тим самим немає необхідність зберігати на кінцевих пристроях закритий ключ.

Практичне значення отриманих результатів. Удосконалено алгоритм захисту даних Інтернет-речей заснований на технології блокчейн, який для зберігання, аналізу та обробки даних використовує шлюзові пристрої, що полегшує шифрування та зберігання даних сенсорів.

Публікації та апробація КР.

1. Васильків В.О., Басістий В.П., Сидорчук Р.В. Блокчейн-платформа hyperledger fabric. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 96-97.

2. Голод Ю.В., Сидорчук Р.В., Васильків В.О. Аналіз вразливостей алгоритмів консенсусу. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С48 – 50.

1 АНАЛІЗ ТЕХНОЛОГІЇ БЛОКЧЕЙН

1.1 Технологія блокчейн

Блокчейн як розподілена та децентралізована база даних – це послідовність блоків, у кожному з яких зберігається список транзакцій. Кожен блок має три основні розділи: дані, хеш-блок і попередній хеш-блок. Хеш визначає ідентичність кожного блоку, як відбиток пальця, і є унікальним для кожного блоку. Інформація кожного блоку позначається хешем. Коли транзакція реєструється в блоці, її хеш-значення обчислюється в блоці шифрування, що містить інформацію, і отримується за математичними правилами. Кожен блок містить хеш попереднього блоку; таким чином блоки з'єднуються один з одним. Будь-які зміни, внесені в інформацію про блок, викликають зміни в його хеш-сумі. Таким чином, будь-які незаконні зміни в інформації в блоках можуть змінити його хеш-значення, і це зробить блок недійсним для наступних блоків. Структура блокчейну біткойна для трьох блоків приведено на рисунку 1.1 [4, 5].

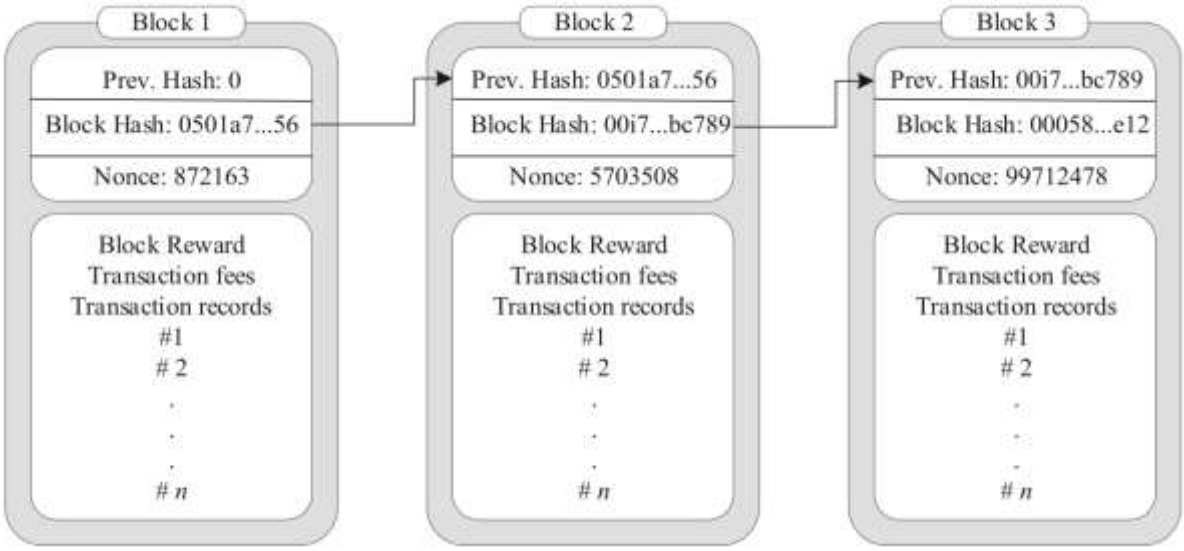


Рисунок 1.1 – Структура блокчейну біткойн

Як показано на рисунку 1.1, перший блок називається блоком Genesis, і оскільки перед ним немає іншого блоку, попередня хеш-сумма дорівнює нулю.

Кожен блок може містити тисячі записів транзакцій, які кодуються хеш-функцією перед трансляцією в мережу. Blockchain використовує функцію дерева Merkle для генерації остаточного хеш-значення як хеш-покажчика (хеш поточного блоку), і кожен блок містить хеш-суму попереднього блоку, щоб зберегти зв'язок блоків.

Дерево Merkle – це хеш-деревоподібна структура даних, яка зберігає транзакції у форматі двійкового дерева. Кожен листовий вузол дерева зберігає хеш-значення транзакцій, а нелистовий вузол містить хеш двох хешів відповідних дочірніх вузлів і, нарешті, корінь цього дерева, що називається Merkle root. Використання функції дерева Merkle зменшить вартість передачі даних і обчислювальних ресурсів [6]. Процес видобутку або перевірки нового блоку за допомогою алгоритму підтвердження роботи вимагає виконання вичерпного запиту до криптографічної хеш-функції, щоб знайти nonce таким чином, щоб він задовольняв попередньо визначену умову. Припустімо, що H() – хеш-функція, а x – корінь Merkle транзакцій у блоці. Отже, рішення PoW відбувається наступним чином [7]: «Враховуючи регульований параметр умови складності h, процес вирішення головоломки PoW спрямований на пошук рядка рішення, одноразового, такого, що для заданого рядка x, зібраного на основі даних блоку-кандидата, хеш-коду конкатенації x і nonce менше цільового значення $D(h)$»

$$Target\ hash\ =\ H(x\ nonce)\ \leq\ D(h),$$

де nonce абревіатура для «числа, що використовується лише один раз», визначеного як число, додане до хешованого блоку в блокчейні, щоб відповідати попередньо визначеному рівню складності та для деякої фіксованої довжини бітів L, $D(h) = 2L - h$.

Загальна класифікація поділяє блокчейн на три категорії, включаючи публічний блокчейн, блокчейн консорціуму та приватний блокчейн [8]:

Загальнодоступні блокчейни розглядаються як різновид децентралізованого блокчейну без дозволу, у якому інформація представлена для всіх учасників мережі, і всі можуть брати участь у її прийнятті. Bitcoin та Ethereum можна розглянути як приклади загальнодоступного блокчейну. Цей тип блокчейну є безпечним завдяки механізму консенсусу, який досягає згоди між усіма учасниками. Ці консенсусні алгоритми включають підтвердження роботи (PoW) і підтвердження частки (PoS).

Блокчейни консорціуму також відомі як федеративні блокчейни, в яких інформація представлена для всіх людей, але її зміна та прийняття можливі лише для певних груп. Наприклад, презентація маркетингових продуктів блокчейном.

Блокчейни консорціуму в основному використовуються в банківському секторі. Основна ідея полягає в тому, щоб розподілити повноваження між кількома органами влади замість того, щоб мати єдиний повний контрольний орган для прийняття колективного та неупередженого рішення. R3 (банки), EWF (енергія) і B3 (страхування) є деякими прикладами блокчейнів консорціуму.

Приватні блокчейни – це дозволені блокчейни, в яких інформація доступна для представлення спеціальної групи, а прийняття її змін можливе лише авторизованою групою. Наприклад, система оплати праці через блокчейн. Це централізований блокчейн, у якому є центральний орган, який визначає дозвіл на те, хто може читати, писати або брати участь у блокчейні. Таким чином, механізм консенсусу в приватних блокчейнах визначається одним центральним органом.

Ці три типи блокчейнів відрізняються залежно від того, як досягти консенсусу між учасниками. Наприклад, у загальнодоступному блокчейні всі майнери визначають консенсус, однак у блокчейні консорціуму та приватному блокчейні визначення консенсусу може здійснюватися вибраним набором вузлів або однією організацією відповідно.

Безпека та перевірка блоку є важливим завданням, яке виконується за допомогою певного механізму, який називається алгоритмом консенсусу. У розподілених системах спосіб досягнення консенсусу між ненадійними вузлами вважався проблемою візантійських генералів, де група армійських генералів оточила місто. Деякі генерали вважають за краще атакувати місто, а інші вважають за краще відступити. Атака міста деякими генералами не вдасться. Тому вони повинні домовитися про атаку або відступ. Досягнення консенсусу в розподіленому середовищі є проблемою. Для блокчейну, який має розподілену систему, консенсус також є проблемою [9].

Блокчейн це децентралізована мережа, у якій немає центрального вузла для спостереження та перевірки всіх транзакцій. Відповідно, існує потреба в розробці протоколів, які вказуватимуть, що всі транзакції правильні. Таким чином, алгоритми консенсусу вважаються ядром кожного блокчейну. У розподілених системах консенсус став проблемою, коли всі члени мережі (вузли) погоджуються прийняти або відхилити блок. Коли новий блок приймається усіма членами мережі, його можна додати до попереднього блоку.

Головна проблема блокчейнів полягає в тому, як досягти консенсусу між учасниками мережі. Розроблено широкий спектр алгоритмів консенсусу, кожен з яких має ряд переваг і недоліків. Така кількість існуючих алгоритмів консенсусу може спричинити плутанину при їх виборі та застосувати їх для вирішення проблем реального світу.

1.2 Аналіз алгоритмів консенсусу блокчейну

Алгоритми консенсусу є критично важливими в технології блокчейн з кількох причин. Вони забезпечують безпеку, перешкоджаючи зловмисникам отримати контроль над мережею, забезпечуючи дійсні транзакції та безперебійну роботу мережі. Вони допомагають досягти децентралізації,

гарантуючи, що всі вузли досягають консенсусу щодо дійсності транзакції, запобігаючи централізації.

Алгоритми консенсусу сприяють прозорості, роблячи всі транзакції видимими в блокчейні, полегшуючи відстеження та запобігання шахрайським діям. Вони підвищують ефективність, дозволяючи вузлам швидко погоджувати дійсність транзакцій і своєчасно додавати нові блоки в блокчейн.

Досягнення згоди в мережі блокчейн це складне і важливе завдання. Нові записи транзакцій будуть додані до блокчейну, оскільки новий блок перевіряється всіма вузлами в мережі. Слід зазначити, що після перевірки блоків їх неможливо змінити або видалити. Структура блокчейнів розроблена таким чином, щоб діяти в надійній та ненадійній мережі з ворожими користувачами. Кількість алгоритмів консенсусу зростає з кожним днем відповідно до розвитку блокчейна. Розглянемо найважливіші алгоритми консенсусу, які широко використовуються в мережах блокчейн, і обговоримо їхні переваги та недоліки.

1.2.1 Алгоритм підтвердження роботи

Найвідомішим методом консенсусу є підтвердження роботи (PoW), який був представлений Накомото [10] і використовується в біткойнах. Підтвердження роботи існує вже багато років як відповідний метод для цифрової валюти. У цьому методі комп’ютер виконує багато обчислень, щоб вирішити математичну головоломку. Ця головоломка виконується за допомогою функції хешування. Хеш функція – випадкова і складна математична формула, яка використовується для підтвердження транзакцій, що зберігаються в блоках. Кожен блок складається з хеш-значення попереднього блоку, історії транзакцій, nonce і поточного хешу блоку. Майнер, тобто комп’ютер, який намагається розв’язати хеш, намагатиметься знайти конкретне значення nonce таким чином, щоб хеш-значення відповідало заздалегідь визначеній умові; наприклад, знайти одноразовий номер, який зробить нульовими перших 30 бітів його хеш-значення. Змінюючи ці попередньо

визначені умови, мережа може бути дуже масштабованою та гнучкою для будь-яких умов. У PoW кожен вузол мережі обчислює хеш-значення заголовка блоку. Іншими словами, щоб досягти консенсусу в мережі, майнери намагаються знайти хеш-значення, що дорівнює або менше певного заданого значення. Коли один вузол знаходить цільове значення, він транслює блок у всю мережу, а всі інші вузли повинні підтвердити правильність хеш-значення. Отже, якщо блок перевірено, усі вузли додадуть цей новий блок до свого власного ланцюжка.

Перевагою алгоритму Proof of Work є висока безпека та децентралізація. Однак його основним недоліком є те, що функція видобутку та перевірки блоків витрачає багато енергії. Крім того, швидкість і рівень успіху цієї хеш-функції сильно залежать від обчислювальних можливостей апаратного забезпечення, яке виконує хеш. Крім того, хоча складність хеш-функції може бути масштабованою, через складність обчислення хеш-функції, вирішення цієї головоломки займає деякий час, і тому цей алгоритм не підходить для великих і швидкозростаючих мереж, які потребують величезної кількості транзакцій за секунду. Відповідно, його переваги полягають у децентралізованій структурі, високому рівні безпеки та прийнятному рівні масштабованості. До недоліків необхідно віднести меншу пропускну здатність, великий час створення блоку, енергоефективність, залежність від спеціального обладнання, висока вартість обчислення та великі вимоги до пропускної здатності.

1.2.2 Алгоритм доказу частки

Після Proof of Work наступним поширеним алгоритмом консенсусу в блокчейн є доказу частки (Proof of Stake, PoS). Основні проблеми в системах підтвердження роботи, такі як енергоефективність, стали причиною створення алгоритму підтвердження частки. Алгоритм PoS базується на ідеї, що творець наступного блоку має бути обраний за допомогою різних комбінацій випадкового вибору, його ставки та віку, що може забезпечити хорошу масштабованість.

Ця ідея була представлена в 2011 році для криптовалюти Peercoin, а потім була використана в інших, таких як Nxt і Blackcoin. Вибраний вузол для створення наступного блоку буде вибрано через квазівипадковий процес, у якому вибір залежить від активів, що зберігаються в гаманці (або пулі спільних ресурсів), пов'язаних із цим вузлом. Цей метод не потребує високої обчислювальної потужності для підтвердження будь-яких доказів, тому майнери не отримають жодної винагороди, окрім комісії за транзакцію. Хоча цей метод не потребує підтвердження обчислювальної потужності роботи, він сильно залежить від вузлів, які мають найбільшу частку, і блокчейн якимось чином стане централізованим. Крім того, існує ще одна поширена проблема для системи Proof-of-Stake, яка називається «нічого не поставлено на карту». Це означає, що якщо вузол не має нічого на своїй ставці під час неправильної поведінки, він не боїться нічого втратити. Тому для вузла не буде перешкод, щоб він не поводився неправильно. Прикладом неправильної поведінки може бути створення двох наборів нових блоків для отримання подвійної комісії за транзакції [11].

Leased Proof of Stake – це вдосконалена версія Proof of Stake (PoS), яка використовується на платформі Waves. У Leased Proof of Stake вузол, який підтримує певну кількість криптовалюти, має право додати наступний блок до блокчейну. Однак у LPoS на платформі Waves користувачі можуть позичити частину свого облікового запису на повні вузли та отримати відсоток як бонус. Чим більша сума надається повному вузлу, тим більше шансів для повного вузла створити новий блок. Якщо повний вузол вибрано для створення нового блоку, кредитор отримує відсоток від транзакції, який збирає повний вузол пізніше [12].

Отже, існують деякі переваги консенсусних алгоритмів на основі PoS, такі як менший час створення блоку, висока пропускна здатність, енергоефективність, масштабованість (але менша, ніж PoW) і незалежність від спеціального обладнання. Однак ця група алгоритмів страждає від певної централізації та нижчої вартості неправильної поведінки в мережах блокчейн.

1.2.3 Делеговане підтвердження частки

Цей алгоритм був представлений Даніелем Ларімером [13]. Цей метод є вдосконаленням методу Proof of Stake, завдяки якому вузли обирають представників шляхом голосування для перевірки блоків. Кількість представників обмежена, що дозволить більш ефективно організувати мережу, і кожен представник може визначити достатній час для публікації кожного блоку. Цей метод використовувався в Bitshares. Однак це обмеження кількості представників зробило б мережу більш централізованою. Серед найважливіших особливостей цього механізму можна назвати масштабованість, енергоефективність і низьку вартість транзакцій. Незважаючи на всі ці переваги, це напівцентралізований механізм, і його краще використовувати в приватних блокчейнах. Однак, якщо обраний представник затримується або робить помилку в представленні необхідних звітів, вузли мережі можуть проголосувати, щоб визначити його заміну.

Доказ часу, що минув. Proof of Elapsed Time представлений Intel як один із консенсусних методів блокчейнів, який, подібно до PoW, вимагає від кожного майнера вирішення проблеми з хешуванням. Кожен затверджувач блоку (майнери) вибирається в найкоротший очікуваний час і з огляду на надійну функцію завдяки виробництву блоку. Ці вибори вибирають майнера випадковим чином у мережі та використовують довірене середовище виконання (TEE), щоб забезпечити безпеку свого виборчого процесу. TEE представлено спеціальним обладнанням Intel (SGX, Secure Guard Extension). Основною проблемою цього методу є його залежність від Intel, що суперечить філософії блокчейну, яка базується на децентралізації. Фактично, можна класифікувати цей метод як напівцентралізований алгоритм консенсусу.

1.2.4 Практична візантійська помилковість

Цей метод консенсусу використовується для вирішення візантійської загальної проблеми. Сучасні зловмисні атаки на програмне забезпечення

стають все більш поширеними. Зростаюча залежність галузі та урядів від інформаційних онлайн-сервісів зробить зловмисні атаки більш привабливими та зробить наслідки серйознішими. Крім того, кількість програмних помилок зросла через зростання розміру та складності програмного забезпечення. Оскільки зловмисні атаки та помилки програмного забезпечення можуть бути результатом довільної (візантійської) поведінки несправних вузлів, важливість практичного візантійського алгоритму відмовостійкості можна зрозуміти. Таким чином, цей алгоритм є формою адаптивного машинного режиму. Ця служба буде змодельована як машина режиму, яка відповідає за вузли в децентралізованій системі. У цьому методі всі вузли повинні брати участь у процесі голосування, щоб додати наступний блок, і консенсус досягається, коли більше двох третин вузлів мають сприятливу думку про блок. PBFT може нормально протистояти поведінці однієї третини платформ. Наприклад, у системі зі шкідливим вузлом принаймні чотири вузли повинні мати угоду, щоб досягти правильного результату. Інакше згоди та консенсусу не досягти. Таким чином консенсус досягається швидше та є економнішим порівняно з підтвердженням роботи. Крім того, на відміну від підтвердження частки, цей метод не вимагає жодних активів у частці для процесу консенсусу.

На завершення, енергоефективність і висока пропускна здатність вважаються його перевагами, а деякі моменти, такі як мало або відсутність параметрів, доступних для масштабування, і можливі затримки, оскільки мережа повинна чекати голосів усіх вузлів, відзначаються як недоліки.

1.2.5 Делегована візантійська відмовостійкість

Цей метод відповідає правилам PBFT, але не вимагає участі всіх вузлів у процесі голосування за додавання нового блоку. Кілька вузлів вибираються як представники інших вузлів і, на основі серії правил, дотримуються процесу консенсусу, як метод PBFT. У цьому методі деякі професійні вузли мають записувати транзакції для всіх вузлів. Цей метод використовується в алгоритмі NEO. Варто зазначити, що Delegated Byzantine Fault Tolerance має меншу

ймовірність затримок, ніж PBFT, але обмеження кількості виборців може загрожувати децентралізації мережі.

1.2.6 Підтвердження ваги (PoWeight)

Доказ ваги поєднує в собі широкий спектр дещо відмінних консенсусних алгоритмів на основі моделі консенсусу Algorand. Algorand досягає згоди за допомогою візантійського протоколу згоди, який здатний масштабувати користувачів відповідно до різних параметрів, які називаються вагами. У блокчейні, заснованому на доказі ваги, вага прикріплюється до кожного користувача. Вага обчислюється за різними факторами, які призведуть до різних алгоритмів консенсусу щодо підтвердження ваги. Зазвичай ці фактори залежать від того, скільки грошей у користувача на рахунку. Мережа залишатиметься захищеною, доки дві третини або більше користувачів є чесними. Атаки подвійних витрат також не можуть загрожувати безпеці мережі на основі підтвердження ваги.

Algorand має схожість з алгоритмом Proof of Stake. У PoS відсоток токенів, якими володіє користувач у мережі, визначає суму винагороди, яка підвищить прибутковість виявлення наступного блоку. У PoWeight буде використано інше зважене значення. Filecoin і Chia є прикладами криптовалют, які зараз використовують PoWeight. Filecoin обчислює зважений коефіцієнт, враховуючи кількість даних IPFS, якими володіє користувач, і називає цей алгоритм «Доказом простору-часу». Chia залежить від доказу простору та доказу часу для досягнення консенсусу. Доказ репутації також є ще одним зваженим фактором, який використовується в PoWeight Systems.

Отже, алгоритм PoWeight забезпечує суттєві можливості налаштування та масштабованості, дуже швидко підтверджує транзакції та є ефективним у використанні джерела живлення. З іншого боку, оскільки учасники не отримують винагород у цій мережі, користувачів важко стимулювати до участі. Хоча генерування пасивних потоків доходу не передбачено ядром PoWeight, цю проблему можна вирішити шляхом розробки креативних рішень.

1.2.7 Доказ вигоряння

Доказ вигоряння (Proof of burn, PoB) – це альтернативний метод досягнення згоди в мережі блокчейн. Ідея, що стоїть за цим, полягає в тому, що майнери не повинні витрачати енергію чи час, щоб довести, що вони зробили щось складне. У цьому алгоритмі майнери повинні спалити деякі зі своїх криптовалют, щоб отримати винагороду. Спалювання тут означає, що користувач повинен надіслати трохи криптовалюти на «адресу пожирача», щоб отримати монети, токени або привілеї майнінгу в мережі. Гроші, надіслані на адресу пожирача, не підлягають відновленню, і ніхто не може витратити їх знову, тому вони називаються спаленими та виходять з обігу. Як і процеси в PoW, спалювання монет є дорогою діяльністю для користувача, але не споживає ресурсів і енергії. Єдиним ресурсом, який використовується в PoB, є готовність користувача прийняти короткострокові втрати, щоб отримати довгострокову винагороду.

Як згадувалося, у випадку адрес пожирачів, адреса генерується випадковим чином і не пов’язана з жодним закритим ключем. Відсутність будь-якого відношення до жодного закритого ключа означає, що гроші, які зберігаються на адресі пожирача, в основному недоступні, і ніхто не може їх витратити. Слід зазначити, що всі криптовалюти PoB вимагають постійного підтвердження роботи видобутих криптовалют, таких як біткойн. Slimcoin (SLM) – це криптовалюта, яка використовує біткойни як метод майнінгу та консенсусний алгоритм. Чим більше монет спалює користувач, тим більше шансів знайти наступний блок. Це також схоже на PoS, в якому багаті, швидше за все, стануть багатшими.

Підсумовуючи його атрибути, це створює більшу стабільність, оскільки відомо, що хтось, хто ризикує короткостроковими збитками та витрачає свої гроші таким чином, залишатиметься в мережі довше, щоб отримати прибуток. Крім того, оскільки немає жодного фактора, який робить інвесторів

централізованими, PoB посилює децентралізацію та створює розподілену мережу.

З іншого боку, спалювання здобутих монет PoW витрачає енергію та час. Якщо одного разу вартість монет PoB стане більшою, ніж монети, спалені PoW, можна буде сказати, що PoB є більш енергоефективним, ніж PoW, і витрачені монети, енергія та час будуть якимось чином відновлені.

1.2.8 Підтвердження дієздатності

Концепція Proof of Capacity (PoC), також відома як Proof of Space (PoSpace), була введена Дзембовським у 2014 році. Тут майнери використовують вільний простір на жорсткому диску для видобутку безкоштовних монет. Першою криптовалютою, яка використовувала цей алгоритм, була Burstcoin. Алгоритм PoC складається з графіка жорсткого диска, що означає обчислення та зберігання рішень на вашому жорсткому диску до початку майнінгу. Деякі рішення швидші за інші. Якщо на жорсткому диску зберігається найшвидше (найближче) рішення головоломки останнього блоку, він виграє блок.

У Burstcoin реалізація алгоритму PoC складається з двох етапів. Перший етап називається графіком, на якому майнери створюють щось під назвою «Nonce». Nonce створюються шляхом повторного хешування даних, включаючи ідентифікатор майнера, за допомогою дуже повільної хеш-функції, відомої як Шабал. Оскільки хеші Шабала важко обчислити, вони обчислюються заздалегідь і зберігаються на жорсткому диску у вигляді одноразових номерів. Чим більше вільного простору виділяє майнер для побудови, тим більше буде створено одноразових номерів. Nonce містять 8192 хеші. Кожні два хеші створюють ковш, тому nonce містить 4096 ковшів, позначених від 0 до 4095 (рисунок 1.2).

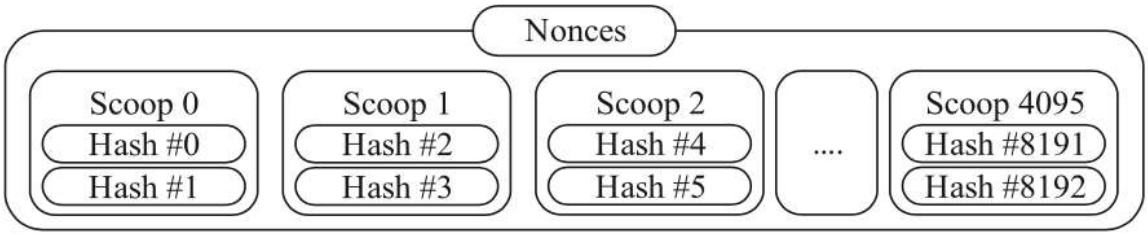


Рисунок 1.2 – Структура блокчейну на основі алгоритму підтвердження ємності

Перш ніж почати майнінг, майнер повинен заповнити весь свій бажаний вільний простір на жорсткому диску nonce. Ці одноразові номери діють як лотерейний квиток, який містить ряд цифр і літер. Якщо один із хешів у одноразовому номері є найближчим до нещодавньої головоломки в мережі, це означає перемогу в битві майнінгу.

Слід зазначити, що на відміну від біткойна, якому для майнінгу потрібне спеціальне обладнання, наприклад ASIC і графічні процесори, єдиним обладнанням, що використовується в PoC, є звичайний жорсткий диск, тому ніхто не може скористатися спеціальним обладнанням. Крім того, кажуть, що використання жорстких дисків є в 30 разів більш енергоефективним, ніж майнінг на основі ASIC, і немає необхідності постійно оновлювати ваше обладнання, оскільки старий диск також може зберігати nonces. Крім того, оскільки кожен має легкий доступ до жорстких дисків, мережа залишатиметься децентралізованою.

1.2.9 Доказ важливості

Доказ важливості (Proof of Importance, PoI) – це ще один алгоритм консенсусе, який вперше було представлено в проекті NEM, щоб усунути критику алгоритму Proof of Stake. У PoS чим більше вузол надає або зберігає суму валюти, тим більше він отримає балів за створення блоків як винагороду. Цей метод стимулював би власників рахунків заощаджувати монети, а не розповсюджувати їх, окрім того, щоб допомогти багатим стати ще багатшими. Проте в блокчейні NEM кожному обліковому запису або вузлу присвоюється оцінка важливості, яка впливає на шанси облікового запису отримати невелику

фінансову винагороду в обмін на додавання транзакцій користувачів у мережу. Щоб мати право на «збір», що означає додавання блоку в блокчейн, обліковий запис повинен мати принаймні 10 000 наданих XEM.

Нижче наведено три фактори, які визначають загальну оцінку облікового запису [14].

1. Передача прав: чим більша кількість наданих монет, тим вищий бал. Враховуються лише монети, які були на обліковому записі протягом певної кількості днів.

2. Транзакційне партнерство: хто зробить більше транзакцій з іншими обліковими записами NEM у мережі, отримає вищий бал.

3. Кількість і розмір транзакцій за останні 30 днів: кожна транзакція, яка перевищує мінімальний розмір, збільшить оцінку облікового запису. Більші та більш часті транзакції мають більший вплив.

Після підрахунку балів облікового запису обліковий запис отримає шанс, пов'язаний із досягнутим балом, для додавання блоку до мережі блокчейн. Цей метод забезпечує децентралізацію блокчейну, а також балансує між блокуванням грошей на рахунках та їх розподілом. Крім того, згідно з офіційним документом NEM, PoI – це стійкість до довільних маніпуляцій за допомогою NCDawareRank, наданого балансу, згаслих і зважених вихідних посилань, а також підсумовування до одиниці в обчисленні оцінки важливості. Ці контрзаходи роблять доказ важливості стійким до атак Sybil, які є помилковими або зловмисними, представляючи себе як кілька ідентифікаторів, щоб отримати контроль над системою. Циклічна атака, яка полягає в пересиланні XEM через транзакції передачі в циклі для підвищення оцінки важливості, також протистоїть у PoI.

Отже, алгоритм PoI може бути швидким і енергоефективним, оскільки йому не потрібен майнінг, а його система оцінки зробить його децентралізованим, масштабованим і безпечним. Для майнінгу не потрібне спеціальне обладнання, це значне вдосконалення традиційного алгоритму PoS.

1.2.10 Доказ активності

Підтвердження активності (PoA) є ще одним поширеним консенсусним алгоритмом, представленим у 2014 році. Автори заявили, що вони запропонували консенсусний алгоритм, заснований на поєднанні PoW і PoS. Це майже безпечний алгоритм проти ймовірних практичних атак на біткойн і має низькі штрафи щодо мережевого зв'язку та місця для зберігання.

Цей алгоритм запроваджено як захист від потенційних проблем у біткойнах, таких як «трагедія спільного майнингу», за якої майнери починають діяти лише у власних інтересах, і мережевих атак, таких як мережева відмова в обслуговуванні та ізоляція мережі. Що стосується біткойна, трагедія спільного майнингу може статися після того, як буде видобуто всі 21 мільйон монет із винагородою за майнінг, а майнери отримають винагороду лише за транзакції. Як згадував Хардін: «Коли зв'язок між вузлами низький, стає легше відмовити в обслуговуванні, переповнивши вузли-майнери, або здійснити атаку Sybil шляхом ізоляції та здійснення транзакцій з певним вузлом».

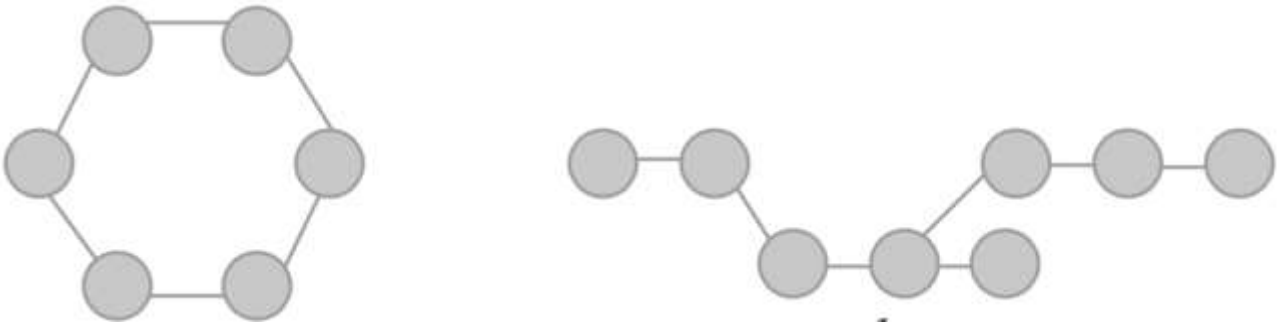
Крім того, у біткойнах зловмисник може спробувати маніпулювати ціною на біржах, де торгуються біткойнами, щоб спричинити втрату довіри. Однак за допомогою протоколів на основі Proof of Stake зацікавлені сторони мають меншу ймовірність падіння цін, оскільки монети, якими вони володіють, приносять дохід, пропорційний фактичній торгівлі, що відбувається. Ці потенційні проблеми в біткойнах згадуються як мотивація для створення алгоритму PoA, намагаючись отримати найкраще від PoW і PoS.

З точки зору безпеки, ймовірність атаки 51% в алгоритмі PoA падає майже до нуля, оскільки така атака вимагатиме від зловмисника 51% усіх монет, а також 51% потужності майнінгу одночасно, і, отже, PoA є більш безпечним у порівнянні з PoW та PoS. З іншого боку, PoA використовує майнінгову частину PoW, тому потребує величезної кількості енергії та обчислювальної потужності. Також згадується, що PoA може бути вразливим до атаки подвійного витрачання.

Отже, захищаючи мережу від деяких потенційних проблем, таких як атака 51%, підтвердження активності має значні недоліки. PoA потребує багато ресурсів та енергії, а також є сприйнятливим до хабарницьких атак подвійних витрат. Дві популярні криптовалюти, Decred і Espers, прийняли PoA у своєму блокчейні, тоді як Decred (DCR) має набагато кращі показники з точки зору ринкової ціни порівняно з Espers.

1.2.11 Орієнтовані ациклічні графи

Хоча спрямовані ациклічні графіки або DAG в основному є формою структур даних і не є справжніми блокчейн-мережами, але вони широко використовуються в успішних криптовалютах. Також знання про функції DAG може допомогти читачам краще зрозуміти блокчейни. NXT, IOTA та IoT Chain є одними з найуспішніших застосувань DAG. У реальних блокчейн-мережах транзакції зберігаються в ланцюжку мереж, але в DAG транзакції зберігаються топологічно в графі. Рисунок 1.3 а показує циклічний і неорієнтований граф, а рисунок 1.3 б представляє ациклічний граф.



а) б)

Рисунок 1.3 –Циклічний і неорієнтований граф (а), ациклічний граф (б).

Ациклічний граф – це граф, який не має циклу. В ациклічному графі інформація не може передаватися від одного вузла до іншого вузла і повертатися до вихідного вузла без зустрічі з вузлом більше одного разу. Спрямований ациклічний граф – це ациклічний граф, у якому інформація може проходити лише в попередньо визначеному напрямку.

Завдяки своїй безблоковій структурі DAG розглядаються як блокчейн без блоків. Криптовалютні транзакції перевіряються та додаються до мережі швидше, ніж мережі на основі PoW та PoS, оскільки немає необхідності зберігати їх у реальному блоці, а потім перевіряти весь блок. У блокчейні потрібно встановити довільний період часу, щоб забезпечити життєздатність основного ланцюга. Цей час очікування відомий як час блоку і дає мережі час для консолідації та перевірки правильної гілки ланцюга. Однак у групах DAG, якщо інформація спрямована однаково, вузли можуть існувати паралельно.

Цей тип мережі дасть нам можливість усунути потребу в блокуванні та швидше перевірити транзакції. Результатом є швидка, масштабована та повністю децентралізована мережа. Блокчейни чутливі до подвійних витрат, м'яких і навіть хардфорків, однак у DAG перевірка певної транзакції визначається кількістю транзакцій, що стоять за нею. Це робить систему DAG швидшою та захищеною від атак подвійного витрачання.

З точки зору ширини мережі, додавання транзакції до попередньої транзакції щоразу зробить мережу занадто широкою. У групі DAG кожна перевірена транзакція повинна зв'язуватися з наявною та новою транзакцією мережі. Коли транзакція відбувається в повній мережі DAG, мережа вибере існуючу пізнішу транзакцію для зв'язку. Цей підхід утримає ширину мережі в певному діапазоні, який може підтримувати швидку перевірку транзакцій. IOTA запропонувала власний алгоритм під назвою Tangle для контролю ширини мережі.

У мережі DAG немає процесу майнінгу, тому немає залежності від спеціального обладнання, відповідно енергоспоживання дуже низьке. Перевірка транзакцій відбувається майже миттєво.

Комісія за транзакцію може бути низькою та швидкою. Отже, це робить мережу DAG зручною для невеликих і навіть мікротранзакцій або платежів. Наприклад, мережа IoT може обробляти понад 10 000 транзакцій на секунду. Ці характеристики мережі DAG, а також її здатність захищатися від атаки 51%

зробили її ідеальним підходом для Інтернету речей і комунікацій між машинами.

1.3 Вразливості алгоритмів консенсусу

Безпека блокчейну залежить від надійності та потужності алгоритму консенсусу, який використовується для перевірки транзакцій і блоків. Розглянемо найпоширеніші атаки на безпеку, які теоретично можуть загрожувати майже всім типам алгоритмів консенсусу. Існують також інші типи атак і вразливостей у протоколах блокчейнів, але цим загальним і фундаментальним вразливостям завжди слід приділяти найбільшу увагу при порівнянні різних типів блокчейнів [15–17].

1.3.1 Атака подвійних витрат

Атака подвійних витрат відбувається, коли людина намагається витратити певну суму грошей на блокчейн двічі. Це може статися, коли зловмисник намагається створити звичайну транзакцію, щоб включити її в блок, а потім через деякий час створює шахрайську конфліктну транзакцію та вставляє її в новий розгалужений шахрайський блок, намагаючись скасувати здійснену транзакцію. Потім зловмисник повинен спробувати розширити створену ним шахрайську гілку мережі, доки шахрайська гілка не буде перевірена та прийнята як правильна гілка, яка включає шахрайську транзакцію.

Незважаючи на те, що різні алгоритми консенсусу намагаються пом’якшити цю вразливість і мають різні механізми для її усунення, подвійних витрат теоретично неможливо повністю уникнути в системах блокчейн.

1.3.2 Атака 51%

Цей тип атаки вперше був застосований у мережі блокчейну PoW біткойна, але його також можна запустити в інших системах блокчейну. Атаки 51% також теоретично неможливо уникнути. Протоколи блокчейну намагаються збільшити вартість цієї атаки, щоб захистити її, але можуть бути не в змозі повністю запобігти їй. Коли зловмисник може контролювати понад 50% потужності (наприклад, потужності майнінгу або потужності перевірки) у блокчейні, він може виконувати зловмисні дії, як-от подвійне витрачання або перешкоджати іншим вузлам отримувати їхні чесні транзакції. Цей тип атаки називається атакою 51%. Зловмисник не завжди повинен володіти 51% потужності мережі, тоді як він може підкупити інші вузли, щоб вони пішли за ним, або він може тимчасово орендувати необхідну йому потужність. Таким чином, цей тип атаки завжди повинен приділяти увагу порівнянню безпеки блокчейну. З точки зору порівняння, стверджується, що алгоритми PoW, PoS і DPoS, хоча і діють по-різному проти атаки 51%, але є вразливі до неї. Однак алгоритм PoA підвищує вартість 51% атаки, оскільки зловмиснику потрібно мати 51% усіх монет і 51% потужності майнінгу одночасно.

1.3.3 Атака Sybil

Атака Sybil – це загальна форма атаки, під час якої зловмисник намагається контролювати однорангову мережу, створюючи низку шахрайських ідентифікаторів у блокчейні. Ці особи видають себе за унікальних користувачів або вузлами, які фактично контролюють зловмисник. Ці ідентифікаційні дані використовуються для отримання права голосу, блокування повноважень перевірки або навіть трансляції фальшивого повідомлення в мережі соціальних повідомлень блокчейну. Успішна атака Sybil може надати зловмиснику непропорційний контроль над мережею або оточити чесний вузол і спробувати вплинути на інформацію, що доходить до нього, а потім поступово впливати на блокчейн.

Атаки Sybil важко виявити та запобігти, але блокчейни намагаються застосувати власні підходи, щоб запобігти цьому. Нижче наведено деякі з підходів, які блокчейн може використовувати для запобігання даної атаки.

1. Збільшення вартості створення вузла. Підвищення вартості створення ідентичності є першим підходом до зменшення ризику атаки Sybil. Наприклад, в алгоритмі Proof-of-burn користувачам потрібно купити, а потім спалити монети, надіславши монети на незмінну адресу, щоб підтвердити свою особу. У proof of stack користувачам потрібно мати кілька монет у своєму стеку, а в proof of work користувачі повинні мати та витратити певну обчислювальну потужність.

Завдання тут полягає в тому, щоб знайти ідеальну вартість для створення ідентифікаційної інформації, яка ефективно зменшує ризик атаки Sybil і також не обмежує звичайних користувачів у приєднанні до мережі. Ця вартість також може змінюватися з часом, і блокчейн завжди повинен бути готовий прийняти ідеальну вартість. Криптовалюта, яка найрозумніше знайде цю ідеальну вартість, а також найшвидше змінить її залежно від потреб мережі буде переможцем.

2. Вимагання певного типу довіри. Другим поширеним способом боротьби з атаками Sybil є вимога форми довіри перед тим, як дозволити вузлу приєднатися до блокчейну. Цією формою довіри може стати проста двоетапна перевірка електронною поштою/SMS або запит на підтвердження від групи адміністраторів. Для забезпечення цієї довіри також використовуються деякі обмеження на основі IP-адреси користувача та інші поширені способи захисту від ботнетів.

3. Надання неоднакової влади ідентичностям. Надання різних повноважень користувачам і вузлам є ще одним способом захисту від атаки Sybil. Наприклад, в алгоритмі підтвердження ваги кожному обліковому запису надається вага на основі кількох параметрів, таких як вік облікового запису користувача, кількість унікальних користувачів, які здійснили з ним транзакції, кількість монет, якими володіють користувачі і кількість транзакцій. Ця задана

вага дає кожному користувачеві відповідну кількість повноважень для голосування або участі. Однак такий нерівний розподіл влади робить систему меритократією (кожному – за здатностями), а не чистою демократією, і може бути нецікавою для нових користувачів.

2 БЕЗПЕКА ІНТЕРНЕТ- РЕЧЕЙ НА ОСНОВІ БЛОКЧЕЙНУ

2.1 Захист цілісності даних на основі дерева Меркла

Дерева Merkle або Hash Trees – це конструкція, яка використовується для створення хешів великих обсягів даних. Вона працює незалежно від того, чи йдеться про один великий файл, про велику кількість маленьких файлів чи будь-який інший сценарій, де є значна кількість даних [18, 19].

Дерево Merkle – це дерево хеш-значень. Кінцевим результатом є структура дерева, в якій кожен проміжний вузол є хешем своїх дочірніх вузлів, доки листові вузли дерева не стануть хешами вихідних даних. Це означає, що кожен кінцевий вузол є хешем лише невеликої частини даних.

Ця структура даних має дві основні мети.

1. Ефективна перевірка, що велика кількість даних залишається незмінною, і визначення того, де саме могли відбутися зміни

2. Ефективний доказ того, що частина даних присутня у більшому наборі даних.

Використання хеш-функції для створення кожного з цих вузлів має ряд переваг. Хеші зазвичай мають фіксований розмір, незалежно від кількості вхідних даних. Це означає, що дерево Меркла з відомою кількістю листкових вузлів матиме відомий розмір, незалежно від фактичного розміру даних, які хешуються. Криптографічні хеш-функції також, як правило, стійкі до зіткнень. Це означає, що важко змінити вихідні дані та отримати ті самі хеші, а це означає, що ми можемо покладатися на них для перевірки цілісності даних.

Вони мають широкий спектр використання в програмному забезпеченні, починаючи з різних файлових систем, таких як Btrfs і ZFS, протоколів передачі даних, таких як IPFS, механізмів баз даних, таких як Cassandra і Riak, і навіть реалізацій блокчейнів, таких як Bitcoin і Ethereum.

3. Побудова дерева Меркла. Побудова дерева Меркла починається з розбиття даних на блоки. Листові вузли дерева є хешами цих блоків. Потім

обчислюються всі інші вузли в нашому дереві, комбінуючи їхні безпосередні гілки та генеруючи їх хеш:

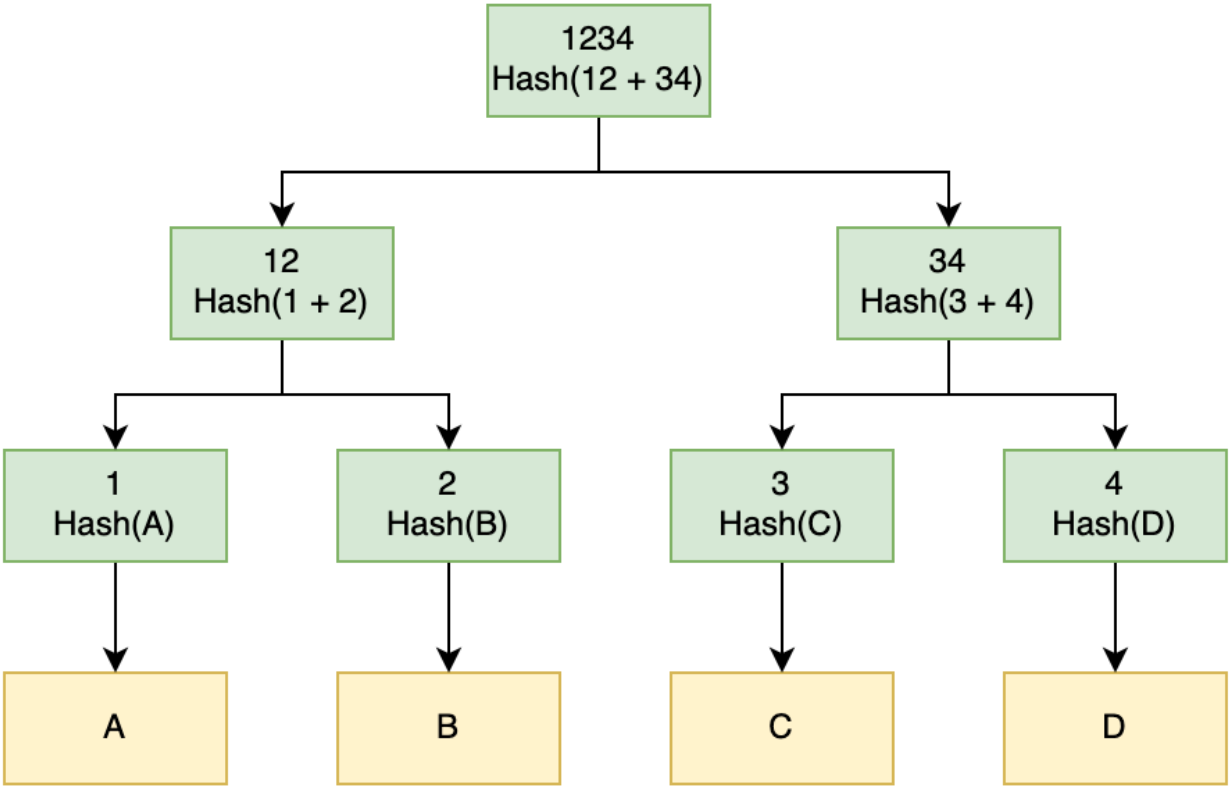


Рисунок 2.4 – Структура дерева Merkle

На рисунку 2.4 присутні чотири блоки даних – «A», «B», «C» і «D». Перше, що робимо, це хешуємо кожен із них у хеші «1», «2», «3» і «4». Далі об'єднуємо та хешуємо їх. Отже, згенеруємо хеш «12», об'єднавши хеші «1» і «2», а потім хешуючи цей результат. Повторюємо це по всьому дереву, поки не отримаємо кореневий вузол.

Те, як саме ми визначаємо блоки даних, не має значення, якщо це можна відтворити. Якби ми генерували це для однієї величезної частини даних, ми могли б розділити її на частини однакового розміру. Наприклад, можна розділити файл розміром 1 ГБ на 64 блоки розміром по 16 МБ кожен. Як альтернатива, якби ми генерували це для великої колекції даних, тоді ми могли б мати кожен блок. Наприклад, ми можемо розділити каталог файлів так, щоб кожен блок представляв окремий файл.

Також не важливо, яка хеш-функція використовуємо, і як ми об’єднуємо попередні хеші перед тим, як генерувати наступний рівень хешів. Єдине, що важливо, щоб воно було абсолютно однаковим як при створенні, так і при використанні дерева Merkle.

2.1.1 Перевірка цілісності даних

Дерева Merkle можуть бути ефективним способом перевірки цілісності великої кількості даних. Простий спосіб зробити це – створити те саме дерево, а потім порівняти отриманий і обчислений хеш із вихідних даних. Якщо вони збігаються, то всі дані збігаються, а якщо вони відрізняються, то в даних є певна невідповідність.

Однак ми можемо зробити краще, ніж це. Якщо ми можемо отримати все Дерево Merkle з вихідних даних, тоді можна перевірити цілісність даних, а також точно визначити, де могли відбутися будь-які зміни (рисунок 2.5).

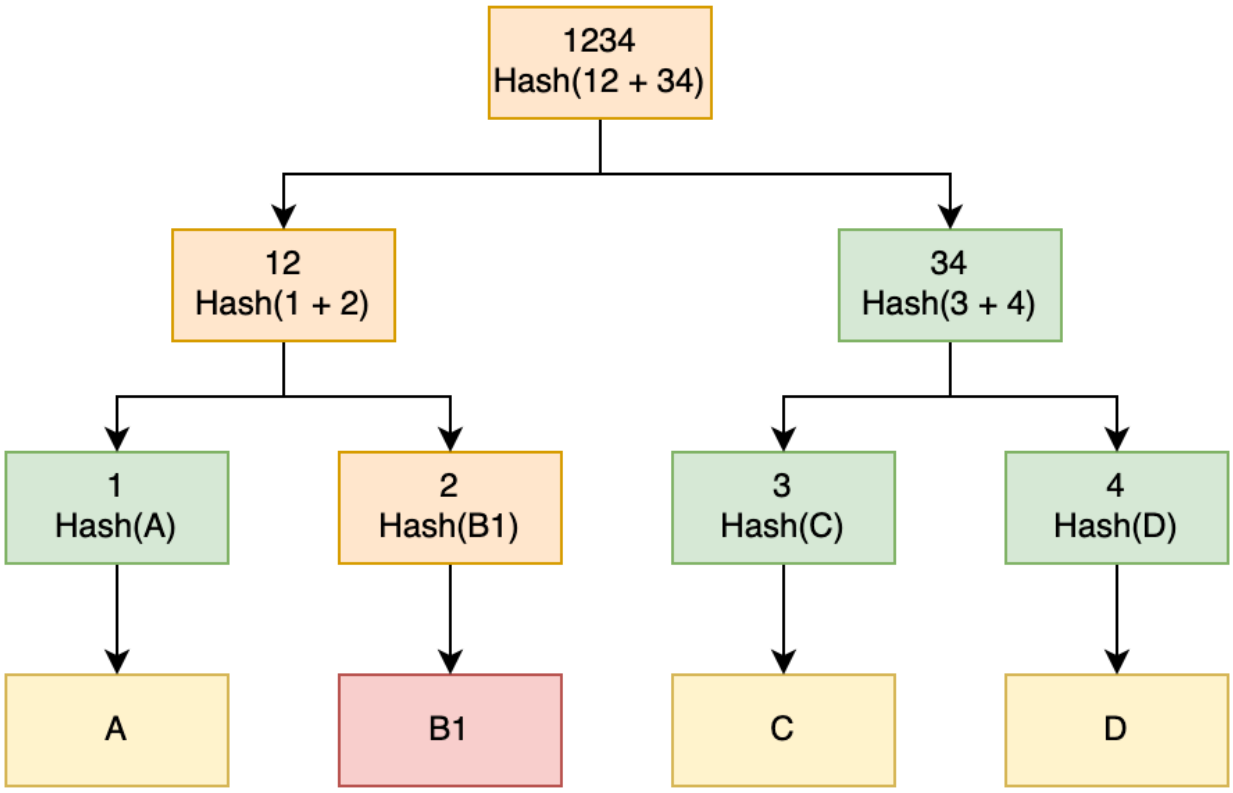


Рисунок 2.5 – Зміна даних в дереві Merkle

При завантаженні того самого файлу, що й раніше, блок «B» був якимось чином пошкоджений. При обчисленні хешу для цього нового файлу, ми побачимо, що він відрізняється від оригінального файлу, що говорить нам про те, що відбулось порушення цілісності.

Враховуючи це, можна порівняти наступний рівень дерева. Тут можна побачити, що вузол «12» відрізняється в двох деревах, але «34» є однаковим. Це говорить про те, що на стороні дерева «34» все добре, тобто блоки даних «C» і «D» правильні, а натомість щось не так на стороні дерева «12».

Знову ж таки, можна порівняти наступний рівень цього піддерева. Тут видно, що вузол «1» той самий, але вузол «2» інший. Це говорить про те, що блок даних «A» правильний, але блок даних «B1» дещо відрізняється від того, що було в оригінальному файлі.

На основі початкових блоків даних можна визначити, які саме з них пошкоджено. Це означає, що можна повторно завантажити лише ці блоки та замінити їх, і тепер у нас є правильні дані. Якби у нас був лише верхній хеш, ми були б змушені знову завантажити весь файл, оскільки ми б не знали, де саме було пошкодження.

Чим глибше наше дерево, тим більше хешів нам потрібно обчислити, щоб дістатися до цієї точки, але менше даних нам потрібно буде завантажити, щоб виправити це. Це вигідно, оскільки обробка вводу-виводу зазвичай ефективніша, ніж мережевий ввід-вивід. Наприклад, якби у файлі розміром 1 ГБ був один пошкоджений блок розміром 16 МБ, ми б обчислили 127 хешів, щоб заощадити на передачі додаткових 1008 МБ даних.

Ненадійні джерела. Якщо хеш верхнього рівня отриманий з надійного джерела, ми можемо отримати все дерево з ненадійного джерела та довести, що воно правильне, оскільки все дерево Merkle хешується. Наприклад, якщо ми завантажуємо файли з однорангової мережі, ми можемо отримати верхній хеш із надійного джерела та фактично завантажити Merkle Tree із самої мережі. Ми не можемо довіряти цьому джерелу, але якщо ми можемо довіряти верхньому хешу, ми можемо довести, що він правильний.

Поки наші хеші достатньо стійкі до колізійних атак, ненадійному джерелу буде важко створити інше дерево з таким же верхнім хешем. Це означає, що ми можемо завантажити дерево з ненадійного джерела, а потім самостійно згенерувати всі відповідні хеші та перевірити їх збіг. Поки це створює той самий верхній хеш, ми знаємо, що всі інші хеші в дереві також правильні. І якщо він не дає той самий верхній хеш, ми можемо відкинути його та спробувати знову з іншого джерела.

Підтвердження наявності даних. Інше використання дерева Merkle полягає в тому, щоб ефективно довести, що частина даних присутня у вихідному наборі. Наприклад, щоб довести, що один конкретний файл був присутній у більшому наборі файлів або транзакцій у межах блокчейну. Для цього обчислюємо всі хеші у верхній частині протилежних піддерев у дереві Merkle, які потім можна об'єднати, щоб продемонструвати, що вони створюють однаковий верхній хеш (рисунок 2.6).

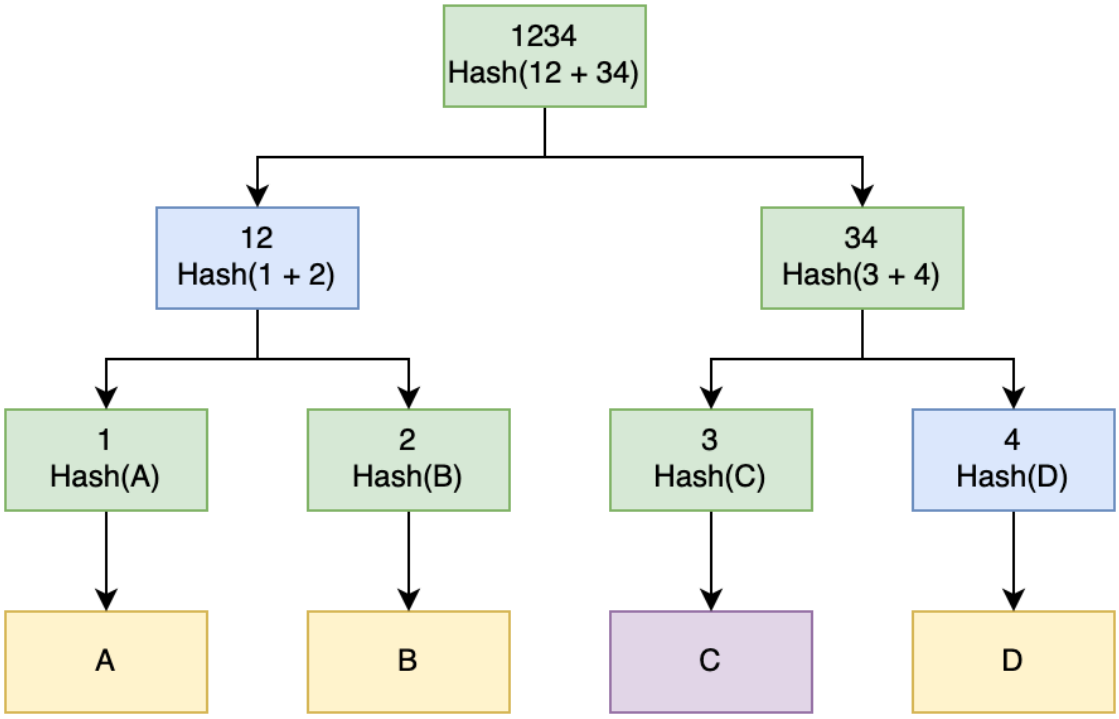


Рисунок 2.6 – Підтвердження наявності даних

Щоб довести, що є блок C, необхідно надати хеш для цього блоку та хеші «4» і «12». Потім інша система може об'єднати хеш для блоку C із наданим

хешем «4», щоб створити хеш «34», а потім об’єднати це з хешем «12», щоб отримати хеш «1234». Якщо це збігається, ми успішно довели, що у є блок С.

Відповідно, чим глибше дерево, тим більше хешів нам потрібно, але тим більш дрібнозернистим він може бути. З приведеним деревом потрібно надати два додаткові хеші та підтвердити блок, який становить 1/4 вихідного набору даних. Якби було дерево глибини шість, тоді потрібно було б надати 5 додаткових хешів і підтвердити блок, який становить 1/64 вихідного набору даних.

Перевага полягає в тому, що можна довести це іншій стороні, яка має доступ лише до верхнього хешу, на відміну від можливості довести це лише іншим сторонам, які мають доступ до вихідних даних. Це може бути значно ефективнішим, якщо вихідні дані дуже великі або задіяні ненадійні засоби зв’язку.

2.1.2 Переваги дерева Merkle

Використання дерев Merkle має кілька переваг для блокчейну. По-перше, це дозволяє ефективно перевіряти транзакції. Замість того, щоб перевіряти кожну транзакцію окремо, вузол може перевіряти корінь Merkle, який представляє весь набір даних [19].

По-друге, це дозволяє компактно зберігати транзакції. Дерево Merkle дозволяє зберігати лише корінь Merkle замість того, щоб зберігати всі транзакції в одному блоці, який набагато менший. Нарешті, дерева Merkle дозволяють швидко синхронізувати вузли. Коли новий вузол приєднується до мережі, він запитує лише корінь Merkle замість усього набору даних, що забезпечує швидшу та ефективнішу синхронізацію.

2.1.3 Недоліки дерева Меркла

Хоча дерева Merkle надають кілька переваг технології блокчейн, є деякі потенційні проблеми, які слід розглянути.

Однією з проблем є можливість атаки зіткнення, коли два різні набори даних призводять до того самого кореня Merkle. Це потенційно може дозволити зловмисникам втручатися в дані, не будучи виявленими.

Іншою проблемою є можливість атаки 51%, коли один об'єкт або група контролює більшу частину обчислювальної потужності мережі. У цьому сценарії зловмисник може потенційно змінити транзакції та їхні відповідні хеші в дереві Merkle, які потім можуть бути прийняті мережею як дійсні.

Важливо зазначити, що хоча ці проблеми існують, вони не обов'язково роблять дерева Merkle неефективними. Правильне впровадження та заходи безпеки можуть допомогти зменшити ці ризики та забезпечити цілісність блокчейну.

Окрім блокчейнів, дерева Merkle також можна використовувати для виявлення зловмисних або випадкових змін будь-яких ваших файлів під час завантаження зі шкідливих каналів.

Хоча дерева Merkle є широко використовуваним і ефективним методом зберігання та перевірки даних у технології блокчейн, існують також деякі альтернативні підходи.

Одним із таких підходів є використання розріджених дерев Merkle, які схожі на звичайні дерева Merkle, але зберігають лише непорожні листки дерева. Це може зменшити вимоги до зберігання та підвищити ефективність у деяких випадках.

Дерево Меркла з n листків має $O(log2\, n)$ – розмір доказів. У великих деревах надсилання доказів може домінувати над споживанням пропускної здатності. Векторні зобов'язання (VC) представляють собою потенційну альтернативу деревам Меркла з доказами постійного розміру.

2.2 Алгоритм консенсусу PBFT

Алгоритм консенсусу PBFT (Practical Byzantine Fault Tolerance) дозволяє розподіленій системі досягти консенсусу, навіть якщо невелика кількість вузлів демонструє зловмисну поведінку (наприклад, фальсифікацію інформації). Під час передачі інформації PBFT використовує криптографічні алгоритми, такі як підпис, перевірка підпису та хешування, щоб гарантувати, що все залишається непідробним, незаперечним і від нього неможна відмовитися. Він також оптимізує алгоритм BFT, зменшуючи його складність від експоненціальної до поліноміальної. У розподіленій системі, яка складається з $3f+1$ вузлів (f представляє кількість візантійних вузлів), консенсусу можна досягти, якщо не менше $2f+1$ невізантійних вузлів функціонують нормально [20 - 22].

Доведення алгоритму PBFT. Розподілена система, яка містить $3f+1$ вузлів, може мати щонайбільше f скомпрометованих вузлів (візантійські вузли). Коли $2f+1$ вузлів досягають консенсусу щодо певного повідомлення, вся система також досягне консенсусу.

По-перше, досягнення консенсусу це питанням більшості. Вузли в системі діятимуть на основі консенсусу, досягнутого більшістю вузлів. У системі є певна кількість скомпрометованих вузлів, і ці скомпрометовані вузли можуть транслювати дані частини «чесних» вузлів (невізантійські вузли), які підтримали певний запит під час трансляції на іншу групу вузлів. У гіршому випадку f скомпрометованих вузлів разом заважатимуть процесу консенсусу. Однак система все одно зможе досягти консенсусу за найгірших обставин.

Якщо в певний час «чесні» вузли розділені на дві частини через розділення мережі: «чесні» вузли f стоять на стороні повідомлення A, тоді ці вузли складають $Set(A,f)$; тим часом f скомпрометованих вузлів перебувають на боці повідомлення B, тому ці вузли складають $Set\ (B,f)$; і несправні вузли f повідомляють несправним вузлам на стороні A, що вони підтримують повідомлення A, і повідомляють вузлам на стороні B, що вони підтримують повідомлення B.

Тепер, з точки зору $Set\,(B,f)$, повідомлення A зібрало $2f$ голосів. З точки зору $Set(B,f)$, повідомлення B набрало $2f$ голосів. Тоді після того, як перший вузол, що залишився без помилок, займе позицію, один із наборів стане більшістю, а інший стане меншістю. Повідомлення A отримає $2f+1$ голосів, а повідомлення B отримає $2f$ голосів і навпаки. У системі є лише f вузлів з помилками, тому вузли без помилок, які є на стороні повідомлення A, повинні мати $f+1$ голосів, а ті, хто на стороні B, повинні мати f голосів. Відповідно до правила більшості система здатна досягти консенсусу. Після фіксації розділу мережі $Set(B,f)$ також знатиме, що досягнуто консенсусу, і тому виконає його.

Іншими словами, у розподіленій системі з максимальною кількістю скомпрометованих вузлів f, поки існує $3f+1$ вузлів, більшість несправних вузлів завжди зможуть досягти консенсусу незалежно від того, як f скомпрометованих вузлів заважатимуть процесу.

Алгоритму PBFT. В алгоритмі PBFT є три типи вузлів: клієнти, головні вузли і репліки.

Клієнт: клієнтські вузли відповідають за надсилання запитів на транзакції.

Головні вузли відповідають за упаковку транзакцій у блоки та фінальні блоки. Кожен процес досягнення консенсусу має один і тільки один первинний вузол.

Репліки: вузли-репліки відповідають за фіналізацію блоків. Кожен процес досягнення консенсусу включає кілька вузлів репліки, і всі вони відбуваються подібним чином.

Як головний, так і репліковий вузли є консенсусними вузлами.

Алгоритм консенсусу PBFT складається з наступних кроків (рисунок 2.7):

1) клієнт надсилає запити;

2) система виконує трифазний процес консенсусу;

3) клієнт отримує відповідь і підтверджує, що консенсус досягнуто.

Рисунок 2.7 – Консенсус PBFT

Клієнт надсилає запит. Клієнт D надсилає запит до системи, а вузли консенсусу (R0, R1, R2, R3) отримують запит. Після того, як основний вузол (у цьому випадку R0) транслює повідомлення попередньої підготовки, система починає виконувати трифазний консенсус. Запит від клієнта тут можна розглядати як набір кількох транзакцій (рисунку 2.7).

2.2.1 Трифазний протокол консенсусу

Консенсус PBFT складається з трьох етапів [23]:

1) попередня підготовка;

2) підготовка;

3) фіксація.

Разом вони утворюють ядро консенсусного алгоритму PBFT. Розглянемо вказані етапи детально.

1. Попередня підготовка. Основний вузол відповідає за перевірку запитів і генерацію відповідних повідомлень попередньої підготовки. Потім основний вузол транслюватиме попередньо підготовлені повідомлення всім вузлам-реплікам. Після отримання повідомлень вузли-репліки перевірять легітимність цих попередньо підготовлених повідомлень, а потім передадуть відповідне підготовче повідомлення.

2. Підготовка. Збір підготовлених повідомлень. Після того, як певний вузол збере 2f+1 повідомлення про підготовку, він оголосить, що готовий до подання блоку, і почне транслювати повідомлення фіксації;

Збір повідомлень комітів. Після того, як певний вузол збере 2f+1 повідомлення комітів, він обробить рідні запити, збережені локально, і внесе відповідні зміни в стан системи.

3. Попередня підготовка. Основний вузол (такий як R0 на рисунку 2.7) надсилає повідомлення попередньої підготовки (PRE_PREPARE$, v, n, d >$ $, m$) до інших вузлів репліки (таких як R1, R2, R3, показані на рисунку 2.7). V – номер перегляду, n – порядковий номер, d – підсумок повідомлення, а m – вихідні дані повідомлення.

Вузли-копії отримують повідомлення попередньої підготовки та перевіряють наступне [24]:

а) легітимність підпису m і сумісність d з m: $d = hash(m)$;

б) якщо вузол зараз знаходиться у v;

в) вузол не має інших попередньо підготовлених повідомлень на тій же сторінці (перегляд v, послідовність n). А саме, немає інших m’ і d’, де d’$= hash(m')$;

г) $h \leq n \leq H$, де H і h представляють високий і низький пороги n.

Після успішного завершення перевірки вузли-репліки надсилають відповідні підготовчі повідомлення (PREPARE$, v, n, d >, i$), де i представляє ідентифікатор вузла рекліки.

2. Підготовка. Вузол консенсусу i отримує $2f$ підготовлених повідомлень від інших вузлів консенсусу (загалом $2f + 1$, включаючи його власний) і перевіряє, чи всі v, n, d цих повідомлень відповідають тим, які він надіслав. Після завершення перевірки консенсус-вузол i встановлює значення $prepared(m, v, n)$ у значення $true$. $Prepared(m, v, n)$ показує, чи вважає вузол консенсусу фазу підготовки повідомлення m у (v, n) завершеною. Нарешті консенсус-вузол i надсилає повідомлення фіксації ($COMMIT,\ v,\ n,\ d,\ i$) і переходить у фазу фіксації.

3. Фіксація. Вузол консенсусу i отримує $2f$ повідомлення про фіксацію від інших вузлів консенсусу (загалом $2f + 1$, включаючи його власний) і перевіряє, чи всі v, n, d цих повідомлень узгоджені. Після завершення перевірки вузол консенсусу встановить для $committed - local(m, v, n)$ значення $true$. $Committed - local(m, v, n)$ показує, що вузол консенсусу підтверджує, що за повідомлення m проголосували принаймні $2f + 1$ вузлів або $f + 1$ несправних вузлів.

Відповідь на запити вузла клієнта. Коли клієнт D отримує $f + 1$ ідентичне повідомлення про фіксацію, це підтверджує, що досягнуто консенсусу щодо його запиту. Раніше було доведено, що $f + 1$ ідентичних повідомлень коміту містить принаймні один з непомилкового вузла, а непомилковий вузол надішле повідомлення коміту лише тоді, коли принаймні $2f + 1$ вузлів проголосували за запит. Таким чином клієнт може підтвердити досягнення консенсусу щодо його запиту.

Перегляд змін. Протокол перегляду зміни має вирішальне значення для алгоритму консенсусу PBFT. Якщо протягом обмеженого часу немає консенсусу щодо запиту, старі дані зберігають узгодженість, а система зберігає поточний статус. Щоб забезпечити систематичну доступність, потрібна нова схема. Алгоритм консенсусу PBFT застосовує протокол перегляду зміни, щоб знову зробити систему доступною. Коли виконується протокол перегляду змін, буде обрано новий основний вузол для досягнення консенсусу та відповіді клієнту протягом обмеженого часу, щоб виконати вимогу доступності.

Основною причиною запуску протоколу перегляду змін є те, що вузол-репліка підтверджує, що протягом обмеженого часу поточний основний вузол не може досягти консенсусу щодо запиту на обмін. Це може бути тому, що основний вузол тимчасово недоступний, або це скомпрометований вузол, або мережа нестабільна для поточної розподіленої системи тощо. Протокол зміни перегляду повинен враховувати кілька можливостей для реалізації толерантності для візантійських вузлів. Наприклад, візантійські вузли можуть

ініціювати протокол зміни перегляду, а основний вузол може бути візантійським вузлом у новому поданні.

Слід зазначити, що зміна перегляду не призведе до відкоту повідомлення коміту. Іншими словами, консенсус у старому вигляді (n) все ще дійсний у новому вигляді.

У трифазному протоколі PBFT основний вузол ініціює процедуру консенсусу, тоді як основний і репліковий вузли беруть участь у перевірці повідомлень і голосуванні. Якщо не вдається досягти консенсусу між усіма вузлами консенсусу в розподіленій системі, буде виконано протокол зміни перегляду, і новообраний основний вузол ініціює нову процедуру досягнення консенсусу. Якщо є консенсус щодо повідомлення, усі вузли виконають повідомлення, і статус системи буде змінено. Оскільки зміна статусу неможлива без консенсусу, система підтримує узгодженість, іншими словами, поточний статус є незворотним. Водночас протокол зміни перегляду у консенсусному алгоритмі PBFT гарантує безпеку та активність усієї розподіленої системи, щоб система могла швидко відновитися після недоступності, щоб продовжувати надавати послуги.

2.2.2 Переваги і недоліка протоколу PBFT

Переваги. Алгоритм консенсусу PBFT допускає несправності певної кількості візантійських вузлів, щоб забезпечити безпеку та активність асинхронної розподіленої системи. Вдосконалений BFT (Byzantine Fault Tolerance), алгоритм PBFT знижує систематичну складність з експоненційного рівня до поліноміального, так що BFT можна застосовувати до реальних систем.

Механізм консенсусу PBFT гарантує, що розподілена система пропонує сильну узгодженість. Він підходить для сценаріїв у приватних і корпоративних мережах [24, 25].

Недоліки. PBFT відрізняється складним зв’язком і низькою масштабованістю. Коли кількість консенсусних вузлів у розподіленій системі

достатньо велика, функціональність різко знизиться. У разі значного розбиття мережі ефективність консенсусу буде знижена, що призведе до величезної затримки.

Алгоритм консенсусу PBFT застосовує шаблон, коли статус змінюється лише після досягнення консенсусу, що забезпечує стійку узгодженість розподіленої системи в реальному часі. Це також є прикладом для застосування алгоритму консенсусу DpoS, оскільки наразі потрібно щонайменше 18 часових інтервалів, щоб блоки в мережі досягли консенсусу, і це створює труднощі для деяких децентралізованих програм, які вимагають узгодженості статусу високого рівня.

2.3 Блокчейн-платформа Hyperledger Fabric

Технологія блокчейн стала важливим словом у світі бізнесу. Його здатність захищати дані, зменшувати витрати та підвищувати ефективність робить його привабливим варіантом для підприємств у різних галузях.

Hyperledger Fabric – блокчейн-платформа з відкритим кодом, яка привернула значну увагу в останні роки завдяки своїм надійним функціям і гнучкості. Hyperledger Fabric створена Linux Foundation у 2015 році. Вона розроблена як гнучка, модульна та масштабована структура блокчейну, яку можуть використовувати підприємства для створення децентралізованих програм. Hyperledger Fabric використовує дозволену мережеву архітектуру, що означає, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн. Це робить її ідеальною платформою для підприємств, яким потрібна сувора конфіденційність і безпека даних [26].

Однією з ключових особливостей Hyperledger Fabric є її високий ступінь гнучкості. Це дозволяє підприємствам налаштовувати алгоритм консенсусу, мову смарт-контракту та керування ідентифікацією відповідно до своїх конкретних потреб. Hyperledger Fabric також підтримує кілька мов

програмування, включаючи Go, Java і Node.js, що полегшує розробникам створення програм на основі мережі блокчейн.

Підприємства завжди шукають способи покращити свою діяльність і зменшити витрати. Hyperledger Fabric може допомогти досягти цих цілей, надаючи безпечну та прозору платформу для здійснення транзакцій компаніями. Архітектура мережі Hyperledger Fabric із дозволами гарантує, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн, що означає гарантовану конфіденційність і безпеку даних. Крім того, гнучка архітектура Hyperledger Fabric полегшує підприємствам розробку та розгортання децентралізованих програм. Це дозволяє компаніям налаштовувати алгоритм консенсусу, мову розумного контракту та керування ідентифікацією відповідно до своїх конкретних потреб. Це означає, що підприємства можуть створювати блокчейн-рішення, які адаптовані до їхніх унікальних бізнес-вимог.

Ключові характеристики Hyperledger Fabric. Hyperledger Fabric має кілька ключових особливостей, які роблять її привабливим варіантом для підприємств. До них належать [27].

1. Масштабованість. Hyperledger Fabric розроблено як масштабована блокчейн-платформа. Вона використовує модульну архітектуру, яка дозволяє створювати кілька каналів в одній мережі блокчейн. Це означає, що підприємства можуть створювати окремі канали для різних підрозділів або відділів, які можна масштабувати незалежно.

2. Гнучкість. Hyperledger Fabric – дуже гнучка блокчейн-платформа. Це дозволяє підприємствам налаштовувати алгоритм консенсусу, мову смарт-контракту та керування ідентифікацією відповідно до своїх конкретних потреб. Hyperledger Fabric також підтримує кілька мов програмування, включаючи Go, Java і Node.js, що полегшує розробникам створення програм на основі мережі блокчейн.

3. Конфіденційність і безпека. Hyperledger Fabric використовує дозволену мережеву архітектуру, що означає, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн. Це забезпечує конфіденційність і безпеку

даних, що робить його ідеальною платформою для підприємств, яким потрібні суворі заходи конфіденційності та безпеки.

4. Модульна архітектура. Hyperledger Fabric використовує модульну архітектуру, яка дозволяє створювати кілька каналів в одній мережі блокчейн. Це означає, що підприємства можуть створювати окремі канали для різних підрозділів або відділів, які можна масштабувати незалежно.

5. Механізм консенсусу. Hyperledger Fabric підтримує підключаються алгоритми консенсусу, дозволяючи учасникам мережі вибирати найбільш підходящий алгоритм для свого випадку використання, будь то візантійський відмовостійкий алгоритм або практичний візантійський відмовостійкий алгоритм.

Компоненти Hyperledger Fabric. Однорангові вузли. Ці вузли підтримують спільну книгу, виконують ланцюжковий код, схвалюють транзакції та беруть участь у процесі консенсусу.

Вузли-замовники. Вузли-замовники встановлюють порядок транзакцій, упаковують їх у блоки та розповсюджують на однорангові вузли для перевірки та виконання.

Центри сертифікації (ЦС). ЦС відповідають за керування ідентифікацією в мережі. Вони видають криптографічні сертифікати учасникам мережі, забезпечуючи безпечне спілкування та авторизацію транзакцій.

Канали. Канали дозволяють створювати підмережі в межах основної мережі Hyperledger Fabric, забезпечуючи приватні та конфіденційні транзакції між конкретними учасниками.

Chaincode. Chaincode містить бізнес-логіку програми та визначає правила перевірки та виконання транзакцій.

У Hyperledger Fabric існує концепція каналів, яка дозволяє організаціям-учасникам приєднуватися та спілкуватися одна з одною. Канал можна розглядати як тунель для однієї організації для таємного спілкування з іншими організаціями-учасниками, які приєднуються до того самого каналу. Будь-які інші особи, які не беруть участі в каналі, про який йде мова, ніколи не матимуть

доступу до жодної транзакції чи інформації, пов'язаної з цим каналом. Одна організація може брати участь у кількох каналах одночасно. На рисунку 2.8 зображено найпростішу мережу Hyperledger Fabric із двома організаціями (Org1 та Org2), які приєднуються до одного каналу. Hyperledger Fabric включає наступні компоненти: Peer, Orderer, CA та Client [27].

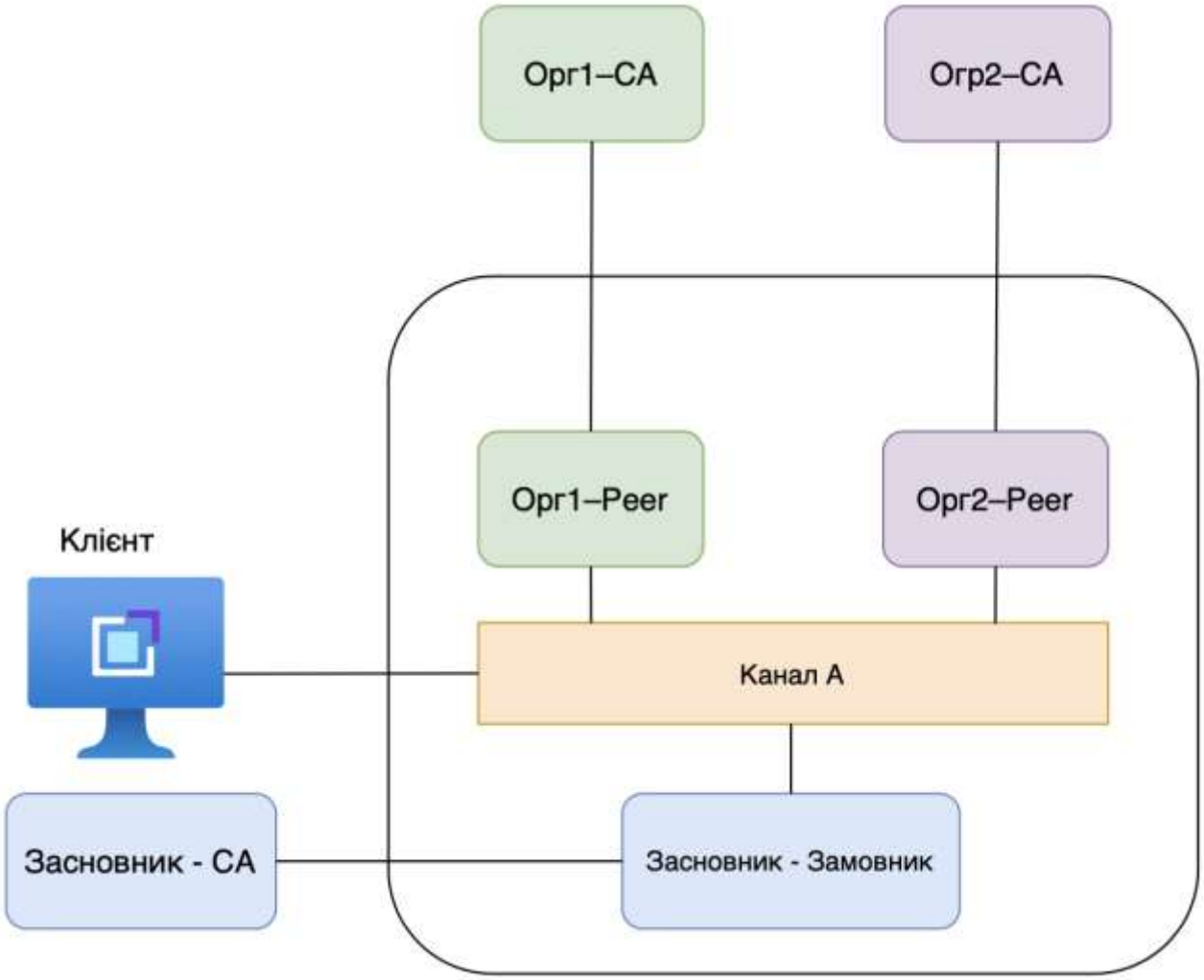


Рисунок 2.8 – Мережа Hyperledger Fabric із двома організаціями

1. Peer – це вузол блокчейну, який зберігає всі транзакції на каналі приєднання. Кожен пір може приєднатися до одного або кількох каналів за потреби. Однак сховище для різних каналів на одному вузлі буде окремим. Таким чином, організація може гарантувати, що конфіденційна інформація буде передана лише дозволеним учасникам певного каналу.

2. Orderer є одним із найважливіших компонентів, які використовуються в механізмі консенсусу Fabric. Orderer – це служба, яка відповідає за впорядкування транзакцій, створення нового блоку замовлених транзакцій і розповсюдження новоствореного блоку всім одноранговим користувачам у відповідному каналі.

3. CA – центр сертифікації, який відповідає за керування сертифікатами користувачів, такими як реєстрація користувачів, відкликання користувачів тощо.

Hyperledger Fabric це дозволена мережа блокчейн. Це означає, що лише авторизовані користувачі можуть запитувати (отримувати доступ до інформації) або викликати (створювати нову транзакцію) транзакцію на наданому каналі. Hyperledger Fabric використовує стандартний сертифікат X.509 для представлення дозволів, ролей і атрибутів для кожного користувача. Іншими словами, користувач може запитувати або викликати будь-яку транзакцію на будь-якому каналі на основі дозволів, ролей і атрибутів, якими він володіє.

4. Клієнт вважається додатком, який взаємодіє з мережею блокчейнів Fabric. Тобто Клієнт може взаємодіяти з мережею Fabric відповідно до своїх дозволів, ролей і атрибутів, як зазначено в його сертифікаті, отриманому від сервера CA пов’язаної організації.

Щоб розробляти програми за допомогою Hyperledger Fabric, важливо добре розуміти архітектуру.

3 АРХІТЕКТУРА ІНТЕРНЕТ-РЕЧЕЙ НА ОСНОВІ БЛОКЧЕЙНУ

3.1 Рівні безпеки Інтернет-речей в Hyperledger-Fabric

З точки зору Hyperledger-Fabric, архітектуру безпеки IoT можна розділити на п’ять рівнів: рівень сприйняття даних, рівень мережевої передачі, рівень консенсусу, рівень контракту та рівень додатків. Кожен рівень працює разом, щоб сформувати базову архітектуру IoT. Архітектура безпеки IoT на основі блокчейну показана на рисунку 3.1.

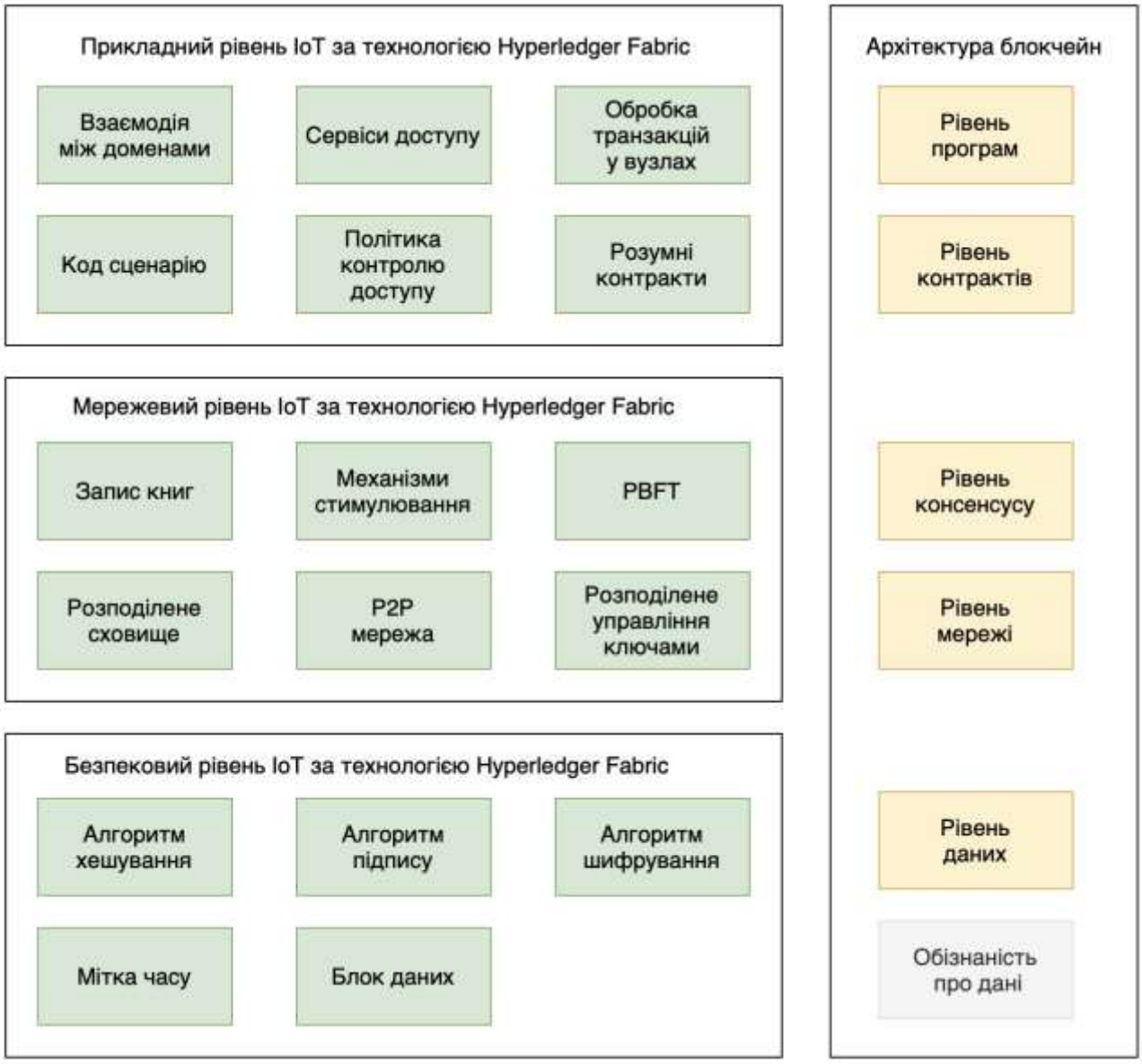


Рисунок 3.1 – Архітектура безпеки IoT на основі блокчейну

3.1.1 Рівень сприйняття IoT

Рівень сприйняття – це фізична область і базовий модуль IoT. Його основна функція полягає в ідентифікації об’єктів, зборі інформації та вимірюванні різноманітних фізичних властивостей, таких як температура, вологість, атмосферний тиск і освітленість. Після того, як інформація зібрана та вилучена, ключова інформація різних типів даних формується за допомогою спеціальної хеш-функції, асиметричного шифрування, дерева Merkle та інших технічних елементів і перетворюється в математичну базу фіксованої довжини для усунення неоднорідності даних. Після уніфікації формату даних сенсора зібрана інформація інкапсулюється в блоки, які зберігаються в блокчейні з міткою часу та технологією асиметричного шифрування та пов’язуються з найдовшим основним блокчейном для формування нового блоку.

1. Шифрування підпису означає, що коли інформація про дані сенсора передається до конкретної клієнтської програми, відправник використовує відкритий ключ одержувача для виконання асиметричного шифрування та надсилає зашифрований текст одержувачу. Після декодування зашифрованого тексту у відкритий текст за допомогою закритого ключа одержувач може використовувати інформацію з даних для реалізації різноманітних прикладних послуг. Без закритого ключа дані датчика в блоці не можуть бути розшифровані у відкритий текст, що гарантує, що дані датчика не будуть незаконно привласнені зловмисниками, що призведе до проблеми розкриття конфіденційності.

2. Процес автентифікації виглядає наступним чином: коли інтелектуальні пристрої реєструються в системі, сервери перевіряють, чи має програма відповідні права контролю доступу відповідно до політики контролю доступу, попередньо визначеної в смарт-контракті, і система перевіряє ідентифікаційну інформацію та IP-адресу пристроїв через реєстраційну інформацію, що зберігається в блокчейні. Після перевірки система може увійти в систему для керування мережею та інтелектуальних служб.

Коли дані сенсора з міткою часу упаковані в блоки, відповідний вузол шлюзу отримає права на ведення книги обліку. Позначка часу зазвичай являє собою послідовність символів, вміст яких записує важливу інформацію, таку як джерело даних сенсора, цифровий підпис і час видачі, що робить дані датчика добре відстежуваними та забезпечує високу безпеку мережі блокчейн.

3.1.2 Мережевий рівень IoT

Мережевий рівень містить режим мережу блоків та протокол автентифікації даних та інші технічні елементи, тому він може гарантувати, що всі вузли блоків у всій мережі можуть брати участь у передачі та гарантувати надійність даних. На мережевому рівні кожен вузол має рівний статус і взаємодіє один з одним у структурі топології P2P (рівний – рівному). Кожен вузол не тільки здійснює передачу даних вимірювання, але й виконує автентифікацію блокової інформації, яка знаходиться в блоці а також реалізує протокол мережевої маршрутизації.

З точки зору структури мережевого рівня, платформа IoT, заснована на блокчейні, є типовою платформою обробки великих даних з децентралізованим і розподіленим сховищем. Перевага цього режиму полягає в тому, що будь-який вузол може автентифікувати, аналізувати та зберігати дані датчиків, не покладаючись на центральні сервери. Якщо кількість недійсних або незаконних вузлів $f < (n - 1)/3$ (n є загальною кількістю вузлів), це не вплине на зберігання та оновлення основного блокчейну. Розподілена архітектура IoT на основі технології Hyperledger-Fabric показана на рисунку 3.2.

Різні сенсорні пристрої завантажують усі види зібраних даних у блокчейн і транслюють на всі вузли мережі через мережу P2P. Коли інші вузли зв’язуються з новою інформацією блоку, автентичність і дійсність даних сенсорів буде перевірено відповідно до структури даних, інструкції ключа, джерела адреси, мітки часу та іншої інформації. Якщо дані сенкора правдиві та надійні, вузол зберігатиме дані датчиків у тілі блоку відповідно до часового ряду та продовжуватиме пересилати до сусідніх вузлів.

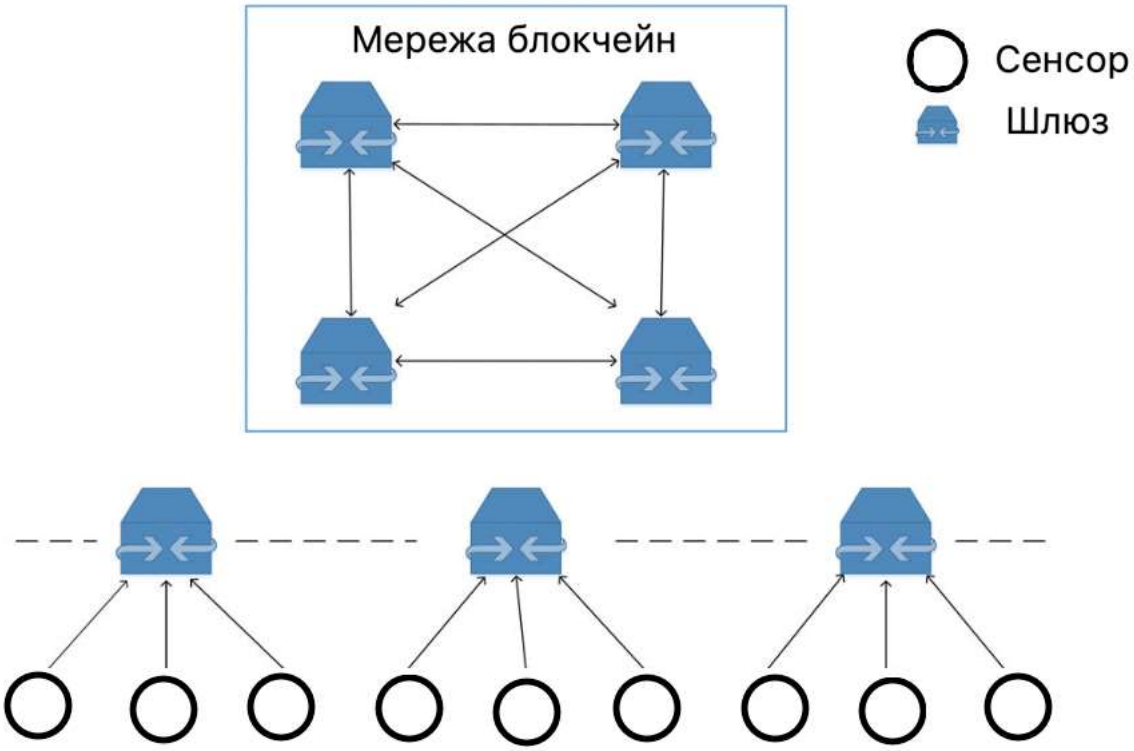


Рисунок 3.2 – Архітектура розподіленого сховища IoT

Якщо блок отримує незаконні дані сенсора, мережа блокчейну негайно припинить передачу даних, щоб гарантувати, що незаконні дані не будуть перенаправлені в мережу IoT. З дизайну мережевого рівня видно, що режим зберігання розподілених вузлів є високозахищеним від втручання, що забезпечує безпеку всієї системи IoT.

3.1.3 Рівень консенсусу IoT

Технологія Hyperledger-Fabric принципово вирішує проблему консенсусу блоків та довіри, дозволяючи деяким розподіленим вузлам швидко досягати консенсусу щодо різних типів даних сенсорів. Рівень консенсусу використовує механізм консенсусу PBFT, який забезпечує швидку автентифікацію, консенсус вузлів і гарантує, що дані не можуть бути сфабриковані. На рисунку 3.3 показано процес застосування Hyperledger-Fabric в IoT.

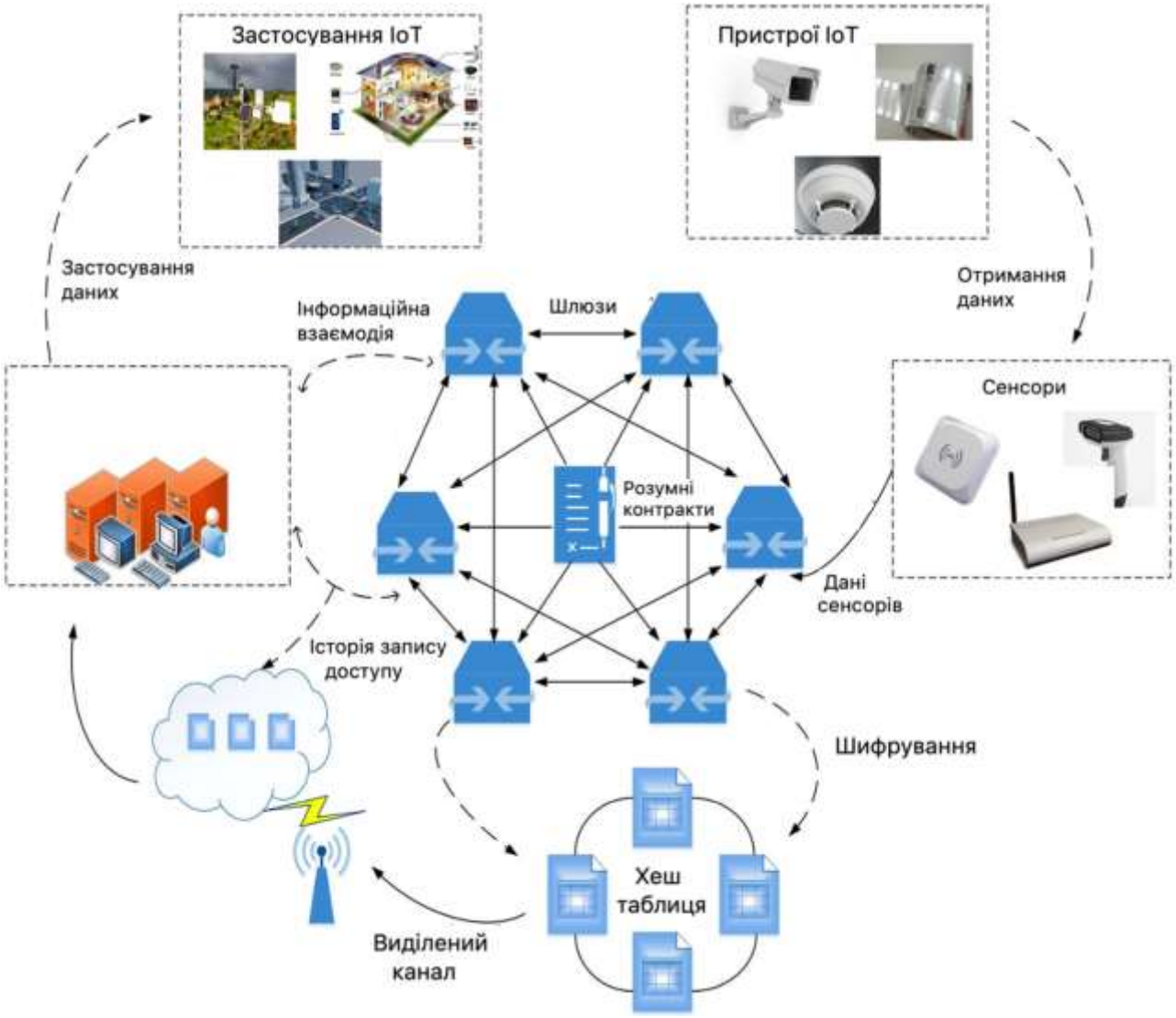


Рисунок 3.3 – Застосування Hyperledger Fabric в IoT

Механізм консенсусу PBFT має нижчі витрати на обчислення та споживання енергії, тому він може досягти автентифікації даних з найменшими витратами енергії та обчислювальних ресурсів у системі IoT з обмеженими ресурсами. У середовищі IoT майже в реальному часі впровадження PBFT може уникнути роздвоєння блокчейну, що призводить до затримки підтвердження транзакції. Крім того, PBFT має хорошу відмовостійкість, яка може містити майже одну третину помилок вузлів, при умові, що більше двох третин вузлів беруть участь в автентифікації, платформа IoT може працювати нормально.

3.1.4 Контрактний рівень IoT

Контрактний рівень інкапсулює різні механізми заохочення, зокрема код сценарію та комбінацію більш складних розумних контрактів. Смарт-контракт розгортається в центрі сертифікації як код без збереження стану, керований подіями та повний автоматичний код виконання з підтримкою Тьюринга.

Після того, як інформація про подію передається в смарт-контракт, кінцевий автомат запускається для оцінки. Відповідно до попередньо встановленої інформації, якщо одна або кілька дій відповідають умові запуску, кінцевий автомат вибирає відповідну політику контролю доступу для автоматичного виконання.

У великомасштабному гетерогенному середовищі IoT, такому як Industry 4.0, центри сертифікації кожної спільноти керують своїми смарт-контрактами, а центри сертифікації в кожному регіоні з’єднані між собою, щоб сформувати внутрішню розподілену мережу, яка, нарешті, підключена до великої магістральної мережі для доступу до інших доменів. Завдяки ланцюжковій архітектурі можна також досягти взаємодії в різнорідних доменах, впроваджуючи смарт-контракти для різних сценаріїв для задоволення різних потреб.

Розумний контракт може ефективно обробляти інформацію та гарантувати, що одна із сторін договору може примусово виконувати контракт без залучення третьої сторони, щоб уникнути виникнення дефолту.

3.1.5 Прикладний рівень IoT

Прикладний рівень інкапсулює різноманітні прикладні служби, що забезпечує ефективне та надійне агрегування, обчислення та обробку різноманітної інформації про дані та реалізує інтелектуальне застосування IoT.

Рівень програми включає дві частини: підрівень підтримки програми та конкретну програму. Програми IoT підтримують підрівень для динамічного збору, зберігання, дешифрування, аналізу та перевірки зашифрованих блоків даних.

Для частин робочої області IoT, де вузли мають справу з великою кількістю транзакцій, можна використовувати розподілені характеристики блокчейну, щоб повністю використовувати неактивні вузли, розподілені в інших місцях, де вузли мають певну обчислювальну потужність, ємність для зберігання та пропускну здатність. Термінал програми IoT зчитує інформацію про дані через інтерфейс, наданий мережею блокчейну, встановлює обробку транзакцій вузла та службу доступу в процесі та завершує інформаційну взаємодію між серверами проміжного програмного забезпечення.

Наприклад, після того, як вузли шлюзу хешують дані, відповідна хеш-таблиця шифрується та зберігається в блоці. Однак різні вузли шлюзу належать до різних серверів керування проміжним програмним забезпеченням. Коли серверам проміжного програмного забезпечення потрібен доступ до даних з інших доменів, їм потрібно визначити, чи має він відповідні повноваження контролю доступу, запустивши смарт-контракт. Якщо так, потрібні дані будуть зчитані з вузлів розподіленого шлюзу; якщо ні, запит буде відхилений. У той же час, через інформацію про стан пристроїв, прикладний рівень може судити про наявність проблем з пристроями IoT в процесі роботи, щоб вчасно знайти та визначити місцезнаходження несправного пристрою, таким чином усуваючи несправності та забезпечуючи нормальну роботу платформи IoT.

3.2 Алгоритм функціонування Інтернет-речей за технологією Hyperledger-Fabric

Розроблено алгоритм роботи IoT за технологією Hyperledger-Fabric, який складається з шифрування даних, консенсусу, дешифрування даних та застосування. Крім того, через різний розподіл вузлів використано гібридну структуру мережі, тобто розглядаємо мережеву структуру між пристроєм і пристроєм, шлюзом і шлюзом, і приймаємо ієрархічну структуру мережі між пристроєм і шлюзом, шлюзом і проміжним програмним забезпеченням.

Протокол зв'язку – це стільниковий зв'язок між шлюзом і шлюзом, шлюзом і проміжним програмним забезпеченням. Крім того, між пристроями та пристроями, пристроями та шлюзами вони передають дані один одному через Wi-Fi. Алгоритм роботи запропонованої архітектури представлений наступним чином.

3.2.1 Шифрування даних і консенсус IoT

Інформація, зібрана сенсорами, зазвичай містить деяку важливу конфіденційну та особисту інформацію. Щоб забезпечити безпечну та надійну передачу даних Інтернету речей, ідентифікаційні дані пристроїв потрібно перевірити, а зібрані дані мають бути зашифровані. Процес поділу конкретних вузлів і їх взаємодія показано на рисунку 3.4.

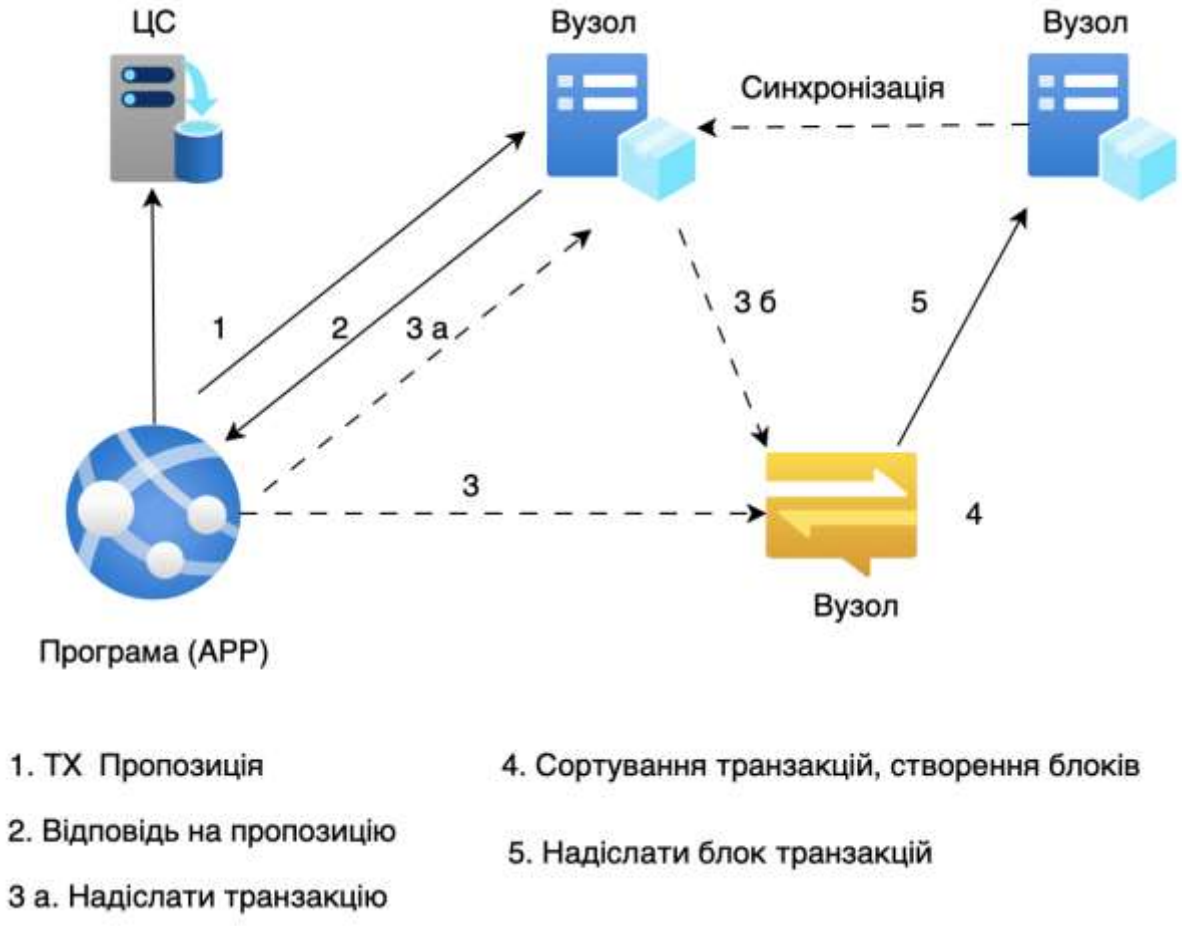


Рисунок 3.4 – Процес взаємодії вузлів

Як показано на рисунку 3.4, коли клієнт ініціює транзакцію в мережу Fabric, спочатку клієнту потрібно отримати законну ідентичність від центру

сертифікації, щоб приєднатися до виділеного каналу в мережі. Незаконним пристроям забороняється доступ до мережі. У той же час зібрана вихідна інформація про дані обробляється партнерами у вузлах шлюзу.

Партнери використовують алгоритм хешування SHA-256 для генерації дайджесту повідомлення та шифрують дайджест за допомогою відповідного приватного ключа, який підписаний цифровим підписом. Замовлення відповідають за глобальне сортування всіх законних транзакцій у мережі, а пакет упорядкованих комбінацій транзакцій генерується в структуру блоку, яка потім надсилається до вузлів для перевірки.

Після того, як вузли комітера регулярно отримують блок, вони використовують протокол консенсусу PBFT, щоб перевірити структуру повідомлення транзакції, цілісність підпису, чи повторюється він тощо. Після проходження перевірки законний запит виконується, а результат записується в книгу, тоді як новий блок будується та зв'язується з найдовшим основним блокчейном. Процес шифрування та передачі даних IoT показано на рисунку 3.5.

Рисунок 3.5 – Процес шифрування та передачі даних IoT

Кожен блок складається із заголовка блоку та тіла блоку. Заголовок блоку в основному зберігає номер версії поточного блоку, адресу попереднього блоку (Prev-block), позначку часу, кореневе значення дерева Merkle тощо. Тіло блоку в основному зберігає різні типи даних. Структура блоку даних приведена на рисунку 3.6.

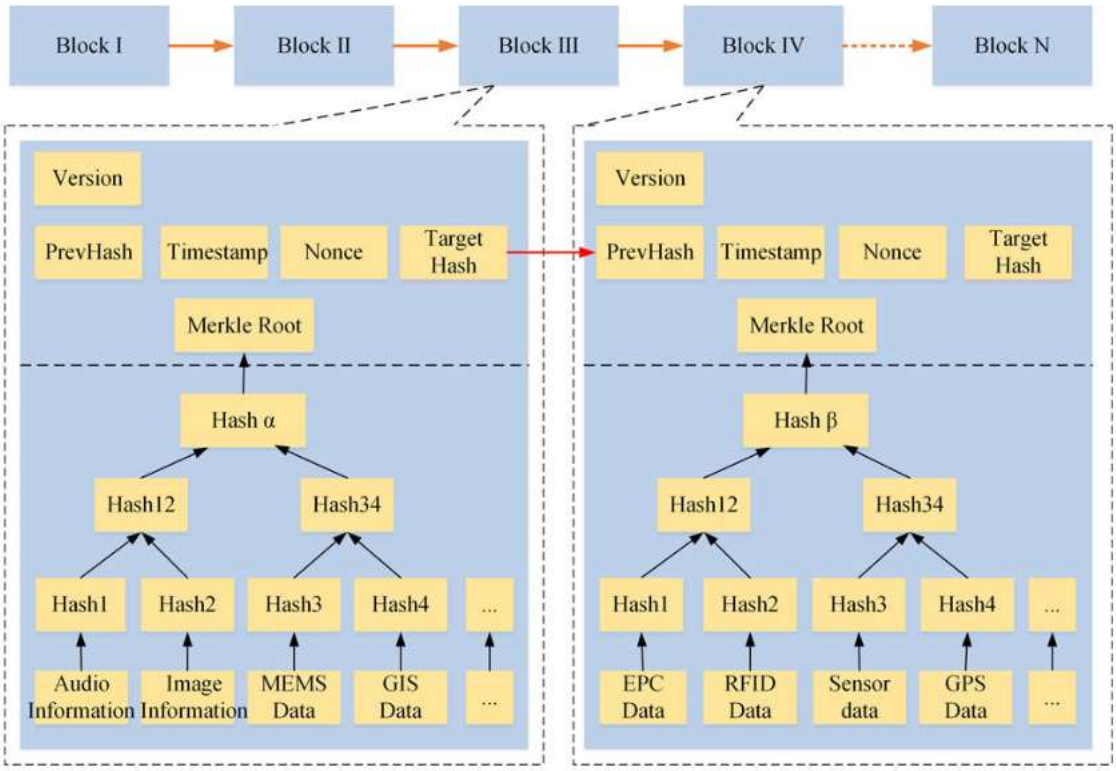


Рисунок 3.6 – Структура блоку даних

Блоки генеруються і з'єднуються в ланцюжок по одному в хронологічному порядку. Кожен блок пов’язаний один з одним через хеш-адресу заголовка Prev-block. Тіло блоку містить всю зібрану та сприйняту інформацію. Мережа блокчейну, що складається з вузлів шлюзу, формується за допомогою мережевої технології P2P, а зібрана інформація групується та нумерується розподіленою хеш-таблицею та зберігається у відповідних вузлах шлюзу, тому дані можуть швидко знаходити точне положення в процесі передачі повідомлень, щоб підвищити ефективність пошуку ресурсів і уникнути надсилання зловмисними вузлами великої кількості марних запитів,

які спричиняють виснаження центрального процесора або пропускної здатності атакованого вузла шлюзу. Як відомо, механізм верифікації та консенсусу блокчейна може ефективно уникнути доступу таких шкідливих вузлів.

Тим часом щойно згенеровані блоки проходять автентифікацію за протоколом консенсусу PBFT, перш ніж зв'язуватися з найдовшим основним ланцюжком блоків. PBFT – це алгоритм реплікації кінцевого автомата. По-перше, кінцевий автомат копіює інформацію про дані на різних вузлах шлюзу. Набір усіх копій інформації про дані представлено літерою U, а кожна копія представлена цілим числом від 0 до $|U| - 1$. Припускається, що кількість несправних з'єднань мережі дорівнює f, а кількість з'єднань мережі всієї служби дорівнює $3f + 1$. Потім випадковим чином вибераємо головний вузол з усіх вузлів шлюзу, щоб відсортувати запити автентифікації даних. Інші підлеглі вузли виконують запити автентифікації в порядку, наданому головним вузлом. Алгоритм автентифікації даних виглядає наступним чином [25]:

1. Головний вузол транслює вибране повідомлення іншим підлеглим вузлам через повідомлення про призначення серійного номера, а інші підлеглі вузли надсилають інтерактивні повідомлення, якщо вони приймають повідомлення, інакше вони не відповідають.

2. Після того, як $2f$ вузли отримають повідомлення, які взаємодіють один з одним, кожен вузол надсилає серійний номер для підтвердження повідомлення.

3. Коли $2f + 1$ вузли отримують порядкове повідомлення підтвердження, це означає, що інформацію про дані підтверджено.

Коли інші підлеглі вузли виявляють, що головний вузол є шкідливим вузлом, алгоритм вибирає інші підлеглі вузли як головний вузол. Як показано на рисунку 3.7, представлено чотири вузли: C – клієнт, N0 – головний вузол, N1, N2 – підлеглий вузол, а N3 – зловмисний вузол, який не відповідає та не надсилає жодних повідомлень.

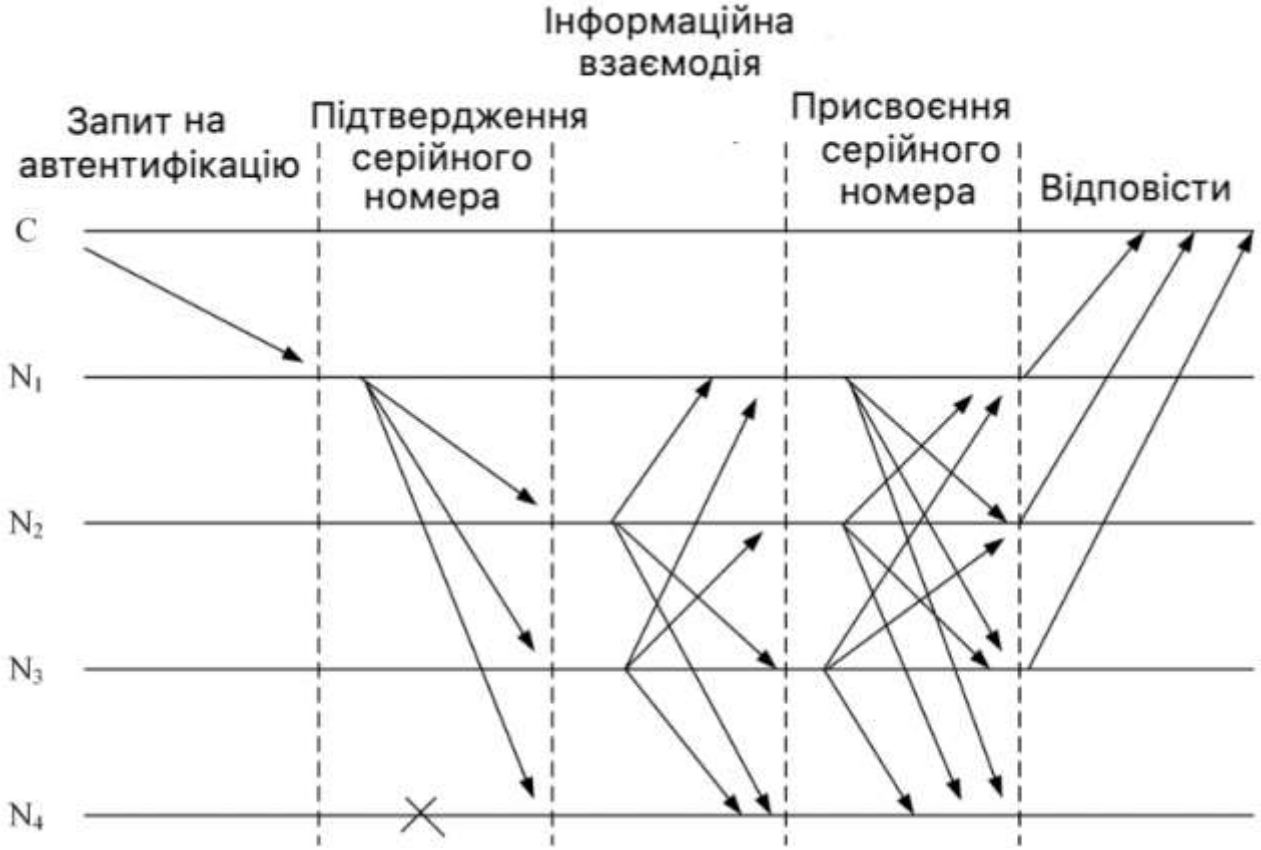


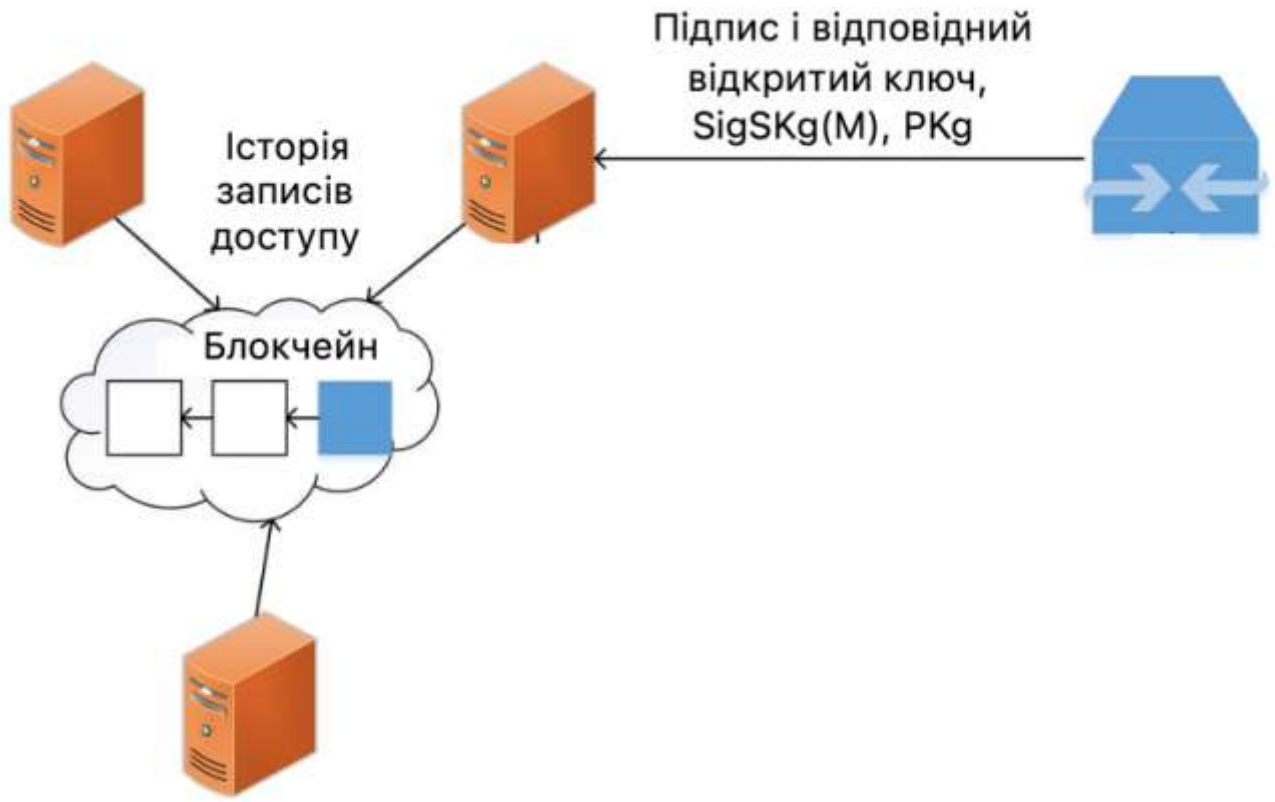
Рисунок 3.7 – Процес консенсусу блоку даних

Коли кінцевий стан вузла досягає підтвердження серійного номера, це означає, що в цьому раунді досягнуто консенсусу.

3.2.2 Розшифрування даних в IoT

Після отримання підписаної інформації сервери проміжного програмного забезпечення повинні перевірити повідомлення, щоб переконатися в його цілісності під час передачі. Відповідні сервери проміжного програмного забезпечення отримують прямий доступ до вузлів шлюзу, які надсилають повідомлення, використовують закритий ключ, який він зберігає, щоб розшифрувати інформацію, а потім використовують ту саму хеш-функцію SHA-256 для обчислення інформаційних дайджестів. Потім перевіряється отриманий цифровий підпис за допомогою відкритого ключа, який зберігається вузлом шлюзу. Якщо два дайджести однакові, сервери проміжного програмного забезпечення на стороні приймання можуть підтвердити, що цифровий підпис

зберігається чесним вузлом шлюзу та не був підроблений. Якщо передані дані незначно підроблені, рядок, отриманий за допомогою операції хешування, значно зміниться, щоб гарантувати, що блок даних можна швидко ідентифікувати під час підробки, і, нарешті, забезпечити цілісність блоку. Процес розшифрування та перевірки даних в IoT показано на рисунку 3.8.



Рисунку 3.8 – Розшифрування і процес перевірки даних IoT

Перевірка даних здійснюється за наступним алгоритмом, який представлений наступною послідовністю кроків.

1. Використовуйте приватний ключ SKm, щоб розшифрувати SigSKg(M) і отримати хеш повідомлення A.

2. Використовуйте закритий ключ SKm, щоб розшифрувати зашифрований текст та отримати відкритий текст.

3. Використовуйте ту саму функцію SHA-256 для обчислення хешу відкритого тексту B.

4. Перевірте, чи хеші A та B є однакові.

Для перевірки безпеки процесу підпису та дешифрування даних у системі IoT на основі блокчейну використано модель Biba [28]. Це модель безпеки цілісності для проведення формальної перевірки безпеки. Оскільки різні засоби шифрування при передачі даних можуть гарантувати їх безпеку, в даній роботі основна увага зосереджена на цілісності даних і не обговорюється рівень конфіденційності інформації. Процес перевірки відбувається наступним чином.

Відповідно до стратегії цілісності моделі Biba, система в стані цілісності, повинна відповідати таким вимогам [29 – 32]:

1) дозволяти серверам проміжного програмного забезпечення переглядати шлюзи з вищим або рівним рівнем цілісності;

2) змінювати дозволи шлюзів з рівним або нижчим рівнем цілісності;

3) викликати інші сервери з заданим рівнем або нижчим рівнем цілісності.

Тобто: якщо $I(s) \leq I(g)$, то $\{r\} \in SG$, якщо $I(s) \geq I(g)$, тоді$\{w\} \in SG$, де рівень цілісності серверів s_i та шлюзів g_i виражається як $I(s_i)$ та $I(g_i)$, діапазон проміжного ПЗ шлюз доступу – це SG, де $\{r\},\{w\}$ представляють операції читання та запису на шлюзи відповідно.

Оскільки інформація, що зберігається у вузлах шлюзу, зашифрована, а сервери проміжного програмного забезпечення використовують незворотність і захист від зіткнень хеш-функції SHA-256, щоб уникнути відмови, підробки та уособлення під час перевірки доступу та дешифрування, а мітка часу також може забезпечити цілісність даних. Таким чином, запропонована схема може гарантувати конфіденційність, цілісність і неспростовність даних.

3.3 Аналіз і порівняння безпеки

Розглянемо моделі загроз і стійкість розробленої схеми до атак. На етапі шифрування алгоритм цифрового підпису еліптичної кривої (ECDSA) і хеш-алгоритм SHA-256 можуть уникнути таких атак. Випадковість і анонімність

вибору головного вузла в консенсусному протоколі PBFT можуть зменшити такі атаки [34].

3.3.1 Безпека шифрування

В алгоритмі ECDSA (цифровий підпис еліптичної кривої) ми використовуємо еліптичну криву для створення цифрового підпису. Загалом ми беремо хеш повідомлення, а потім створюємо підпис за допомогою закритого ключа. Потім відкритий ключ можна використовувати для перевірки підпису.

Порівняно з іншими алгоритмами цифрового підпису (такими як RSA та DSA), він має абсолютні переваги щодо захисту від атак та найвищої безпеки одиничних бітів, наприклад 160-бітний ECC має таку саму силу безпеки, як і 1024-бітні RSA, DSA та має переваги невеликих обчислень і високої швидкості обробки [35, 36]. Тому дані, зашифровані ECDSA, зловмисник не може розшифрувати, через відсутність закритого ключа, щоб отримати вихідні дані. Якщо зловмисник запускає словникову атаку, щоб вгадати секретний ключ, у схемі секретний ключ обчислюється центром автентифікації на основі ідентичності пристроїв, тому зловмисник не може отримати повідомлення, не знаючи закритого ключа. Крім того, одностороння хеш-функція SHA-256, яка використовується в даній роботі, забезпечує додаткову надійність системи.

3.3.2 Безпека етапу перевірки

Як описано в попередньому пункті, кожен запит перевірки клієнта наосліп і випадковим чином відображається на одному головному вузлі для впорядкування запитів. Крім того, для кожного новоствореного блоку n вузлів випадковим чином вибираються як верифікатори, де значення n генерується випадковим чином, але має певний зв'язок із кількістю вузлів помилок f, тобто $n \geq 2f + 1$.

DDoS-атака. DDoS-атака більш імовірна, якщо набір вузлів автентифікації відомий заздалегідь. Такі атаки можуть порушити роботу блокчейну або бути здійсненими з мережі чи поза нею. Можливість заміни та

випадковий вибір вузлів перевірки можуть значно пом’якшити цю атаку. Це пояснюється тим, що набір вузлів перевірки є випадковим і анонімним до того, як він бере участь у консенсусному голосуванні. Крім того, кожен крок процесу голосування отримуватиме відгуки від інших підлеглих вузлів. Таким чином, практично неможливо запустити DDoS-атаку, яка вимагає атаки на всі вузли мережі, щоб знищити систему.

Підкуп або пошкодження головних або підлеглих вузлів: прикладом такої атаки є зловмисний верифікатор, який підкупив інші головні або підлеглі вузли, щоб вони прийняли недійсний блок і проголосували за нього. Виконання такої атаки вимагає знання ідентичності цільового вузла. Протокол PBFT анонімізує взаємодію між стороною консенсусу та стороною верифікатора. Крім того, навіть якщо випадково вибраний головний вузол є зловмисним, система може виявити його завдяки відмовостійкості алгоритму PBFT, щоб подолати цей вид атаки.

3.3.3 Аналіз безпеки системи на основі теорії ігор

У даній роботі блокчейн був розроблений як розподілена система з декількома вузлами шлюзу, що може гарантувати, що атаки не будуть зосереджені на централізованих службах. Для DDoS-атаки зловмисник повинен отримати доступ до кількох вузлів одночасно, що робить атаку більш складною, довготривалою та дорогою. Щоб краще оцінити архітектуру системи, використаємо теорію ігор для аналізу захисту від деяких реальних атак. Проведемо моделювання взаємодії між шлюзовими пристроями та сенсорними пристроями як неповну інформаційну гру для вивчення процесу наступальної та оборонної гри під зовнішніми атаками.

Для представлення неповної інформаційної атаки та гри в захист використаємо: $G[N, \{T_i\}, p, \{S_i(t_i)\}, \{u_i\}]$,

де N – набір гравців у грі, для цієї роботи N = {шлюзові вузли, вузли сприйняття},

T_i – набір типів учасника i, існує лише один тип для шлюзового вузла, а для сенсорних вузлів T_i= {легальні вузли, шкідливі вузли},

p – розподіл ймовірностей учасників у всіх просторах типів,

$S_i(t_i)$ – набір політик, доступних, коли учасник i має тип t_i, коли i є вузлом шлюзу, S_i(вузол шлюзу) = {автентифікація, та неавтентифікація}; коли i є допустимим вузлом, S_i (легальний вузол) = {не атакувати}; коли i є зловмисним вузлом, S_i(зловмисний вузол) = {атака, не атакувати},

u_i – виграш гравця і в грі.

Значення u_i пов'язане не лише зі стратегіями, прийнятими обома сторонами гри, але й із типом гри, до якої вони належать.

3.4 Оцінка обчислювальної складності

Алгоритм PBFT зменшив операційну складність оригінального візантійського протоколу з експоненціальної до поліноміальної, таким чином, з $o\left(n^{(f+1)}\right)$ до $o(n^2)$, і зробив можливим застосування в розподіленій системі IoT з обмеженими ресурсами [37].

Запропонована схема повністю використовує швидкість обробки та простір для зберігання шлюзових пристроїв. Розроблена схема може швидко відповідати на запити, оскільки шлюз може попередньо обробляти та фільтрувати дані, тим самим зменшуючи затримку інформації про транзакції та проблеми з блокуванням, що викликано непотрібною автентифікацією даних. Крім того, оскільки всі операції обробки даних, такі як шифрування та зберігання, виконуються шлюзовими пристроями, споживання енергії основними сенсорними вузлами також зменшується, а час роботи батареї покращується. Таким чином, запропонована схема має відносно надійну комунікаційну здатність.

3.4.1 Застосування Hyperledger-Fabric для IoT

Hyperledger-Fabric використовує модульну архітектуру для побудови шлюзових вузлів у загальну дозволену блокчейн-мережу для надання послуг plug-and-play для різних додатків IoT. Hyperledger-Fabric має менші потреби в енергії та обчислювальних ресурсах, високу продуктивність транзакцій і низьку затримку підтвердження транзакцій, вона більше підходить для пристроїв IoT з обмеженими ресурсами.

Мережу блокчейну з одноранговими вузлами $n = 4$, вузлом замовлення $o = 1$, $CA = 1$ і вузлом помилки $f = 1$ за допомогою служби Bluemix від IBM із запуском програми IoT, розробленої командою IBM Watson. Пропускна здатність транзакції та затримка зв'язку запропонованого рішення показані на рисунку 3.9.

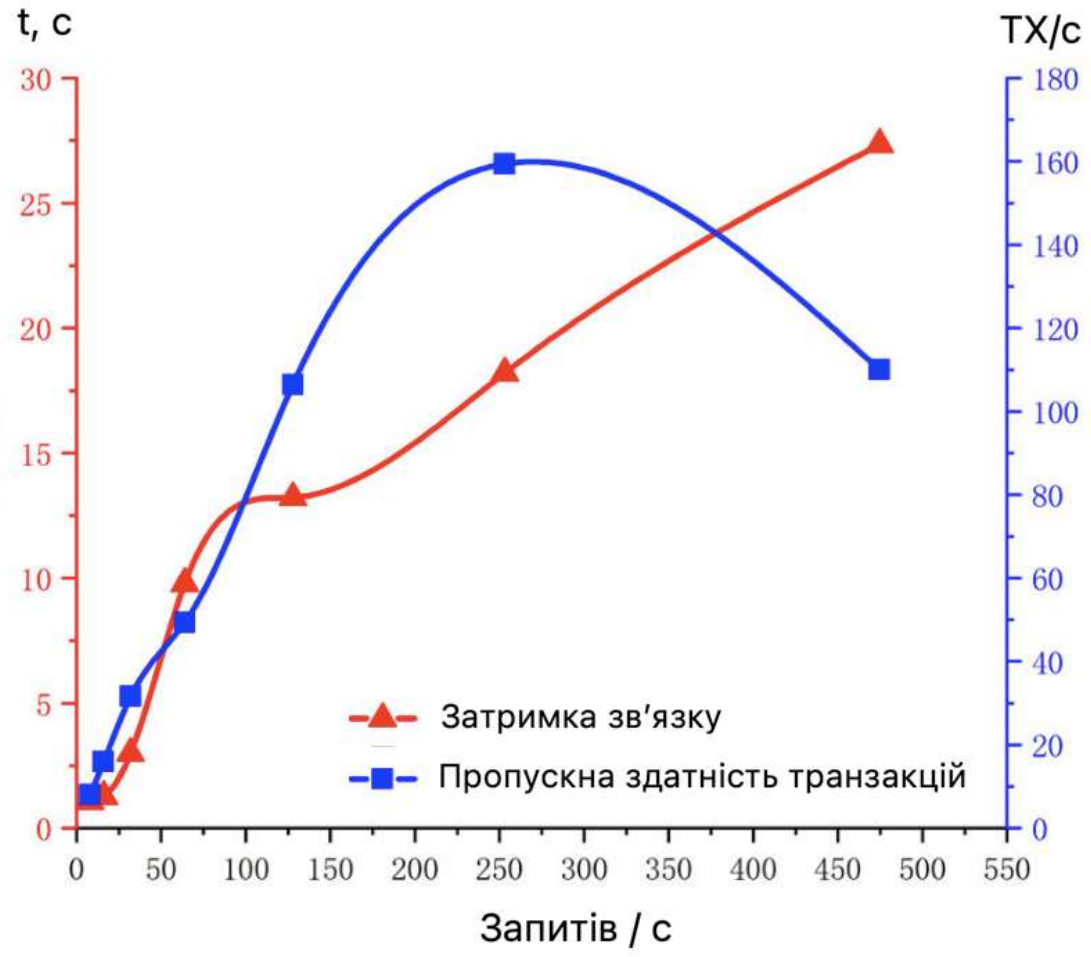


Рисунок 3.9 – Пропускна здатність транзакції та затримка зв'язку

Як показано на рисунку 3.9, зі збільшенням запитів за секунду, хоча час досягнення консенсусу відповідно збільшується, збільшення є незначним. Тоді як пропускна здатність транзакцій має відносно високий приріст. Коли кількість запитів на секунду досягає приблизно 250, пропускна здатність системи знижується через збільшення затримок у черзі для запитів на етапах підготовки та фіксації PBFT. Однак із довгостроковою роботою системи IoT кількість вузлів, які беруть участь у мережі, поступово збільшується, а візантійські вузли, які допускає система, динамічно зростають, однак щоб вирішити цю задачу можна застосувати паралельну обробку транзакцій.

Традиційні мережі можуть бути захищені міжмережевими екранами, статичними мережами IDS/IPS та іншими периферійними механізмами захисту. Однак для пристроїв IoT з обмеженими ресурсами традиційні механізми захисту мережі важко застосувати для захисту пристрої IoT від внутрішніх атак і несанкціонованого пошкодження.

Удосконалено алгоритм захисту даних Інтернет-речей заснований на технології блокчейн. Оскільки більшість пристроїв IoT значною мірою обмежені для зберігання, аналізу та обробки даних, розроблена схема повністю використовує шлюзові пристрої, що полегшує шифрування та зберігання даних сенсорів, а також дозволяє адаптувати технологію блокчейну до поточної системи IoT.

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу розробки алгоритмів захисту даних в середовищі Інтернет речей з використанням технології блокчейн. При цьому отримано такі результати.

1. Проведено аналіз технології блокчейн та можливостей її використання для захисту даних Інтернет речей.

2. Проведено аналіз найбільш відомих розроблених алгоритмів консенсусу блокчейну. Таким чином, було обговорено короткий огляд механізмів консенсусу та їх переваги та недоліки.

3. Проаналізовано вразливості алгоритмів консенсусу в блокчейні та основні типи атак на блокчейн.

4. Розроблено алгоритм захисту даних Інтернету - речей в якому шифрування здійснюється відкритим ключем і тим самим немає необхідність зберігати на кінцевих пристроях закритий ключ.

5. Удосконалено алгоритм захисту даних Інтернет-речей заснований на технології блокчейн, який для зберігання, аналізу та обробки даних використовує шлюзові пристрої, що полегшує шифрування та зберігання даних сенсорів.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Herlihy. Blockchains from a distributed computing perspective. *Commun. ACM*, 2019, vol. 62, pp. 78-85,.

3. J. Jin, J. Gubbi, S. Marusic and M. Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112-121, April 2014.

4. E. Fernandes, A. Rahmati, K. Eykholt and A. Prakash. Internet of things security research: A rehash of old ideas or new intellectual challenges?. *IEEE Security Privacy*, 2017, vol. 15, no. 4, pp. 79-84.

5. A. Dorri, S.S. Kanhere, R. Jurdak and P. Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 618-623, March 2017.

6. Yu, L. Tan, M. Aloqaily, H. Yang and Y. Jararweh. Blockchain-enhanced data sharing with traceable and direct revocation in IioT. *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7669-7678, Nov. 2021.

7. K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang and T. Sato. A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid. *IEEE Trans. Instrum. Meas.*, vol. 64, no. 8, pp. 2072-2085, Aug. 2015.

8. K.-K. R. Choo, S. Gritzalis and J. H. Park. Cryptographic solutions for industrial Internet-of-Things: Research challenges and opportunities. *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3567-3569, Aug. 2018.

9. C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin and K.-K. R. Choo. Blockchain-based cross-domain authentication for intelligent 5G-enabled Internet of drones. *IEEE Internet Things J.*, Sep. 2021.

10. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 2008.

11. W. Chen et al. Cooperative and distributed computation offloading for blockchain-empowered industrial Internet of Things. *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8433-8446, Oct. 2019.

12. A. Bahga and V. K. Madisetti. Blockchain platform for industrial Internet of Things. *J. Softw. Eng. Appl.*, 2016, vol. 9, no. 10, pp. 533-546.

13. Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng and Y. Zhang. Consortium blockchain for secure energy trading in industrial Internet of Things. *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690-3700, Aug. 2018.

14. P. W. Khan and Y. Byun. A blockchain-based secure image encryption scheme for the industrial Internet of Things. *Entropy*, vol. 22, no. 2, pp. 175, Feb. 2020.

15 I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM*, 2018, vol. 61, pp. 95-102.

16. S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo and K. Ren. Searching an encrypted cloud meets blockchain: A decentralized reliable and fair realization. *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2018, pp. 792-800,.

17. C. M. Chen, X. Deng, W. Gan, J. Chen, and S. K. I. Hafizul, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, 2021, vol. 21, no. 1, pp. 3021–3033.

18. H. Yuan, X. Chen, J. Wang, J. Yuan, H. Yan, and W. Susilo. Blockchain-based public auditing and secure deduplication with fair arbitration. *Information Sciences*, 2020, vol. 41, no. 9, pp. 209–217.

19. Q. Hu, M. R. Asghar, and S. Zeadally, "Blockchain-based public ecosystem for auditing security of software applica- tions," *Computing*, 2021, vol. 12, no. 3, pp. 12–19.

20. L. Perez, L. Ibarra, G. F. Alejandro, R. Agustin, and L. A. Carlos, "A loyalty program based on Waves blockchain and mobile phone interactions," *The Knowledge Engineering Review*, 2020, vol. 35, no. 22, Article ID 098e231.

21. S. Biswas, K. Sharif, F. Li, and S. Mohanty. Blockchain for E-health-care systems: easier said than done," *Computer*, 2020, vol. 53, no. 7, pp. 57–67.

22. K. T. Tsung, Z. R. Hugo, and O. M. Lucila. Comparison of blockchain platforms: a systematic review and healthcare examples. *Journal of the American Medical Informatics As- sociation*, 2019, vol. 26, no. 5, pp. 5–12.

23. Y. Chen, H. Xie, K. Lv, S. Wei, and C. Hu. DEPLEST: a blockchain-based privacy-preserving distributed database toward user behaviors in social networks. *Information Sci- ences*, 2019, vol. 501, pp. 100–117.

23. M. Rosenfeld, C. Lee, I. Bentov, and A. Mizrahi, "Proof of activity: extending bitcoin's proof of work via proof of stake," *Performance Evaluation Review*, 2014, vol. 122, no. 2, pp. 89–94.

24. I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Proceedings of the financial cryptography and data security*, vol. 23, no. 11, February 2016, Article ID 09902e23122.

25. Castro, M., & Liskov, B. Practical byzantine fault tolerance. In OsDI, Vol. 99, No. 1999, pp. 173-186

26. Understanding Hyperledger Fabric's Architecture. [Електронний ресурс]. – Режим доступу: https://medium.com/hyperlegendary/understanding-hyperledger-fabrics-architecture-3b37d81c3e96

27. Unlocking the Power of Blockchain with Hyperledger Fabric: A Guide for Enterprises. [Електронний ресурс]. – Режим доступу: https://medium.com/@spydra/unlocking-the-power-of-blockchain-with-hyperledger-fabric-a-guide-for-enterprises-bd55570612a3

28. Balon, N., & Thabet, I. The Biba security model, 2004.

29. R. Wei, A. Jh, D. Tza, R. Yi, and R. C. K. Kim. A flexible method to defend against computationally resourceful miners in blockchain proof of work. Information Sciences, 2020, vol. 507, pp. 161–171.

30. A. O. Bang and U. P. Rao. A novel decentralized security architecture against sybil attack in RPL-based IoT networks: a focus on smart home use case. The Journal of Super- computing, 2021, vol. 77, pp. 1–36.

31. T. B. M. de Sales, A. Perkusich, L. M. de Sales, H. O. de Almeida, G. Soares, and M. de Sales, "Asap -V: a privacy-preserving authentication and sybil

detection pro- tocol for VANETs," Information Sciences, vol. 372, pp. 208– 224, 2016.

32. M. Yokoo, Distributed Consistency Algorithm, 2001, Springer, Berlin Heidelberg, pp. 123-124.

33. Z. Yu, B. Wang, R. Lu, and Y. Yong. DRBFT: delegated randomization Byzantine fault tolerance consensus protocol for blockchains. Information Sciences, 2021, vol. 559, no. 2, pp. 67–78.

34. Y. Li, L. Qiao, and Z. Lv. An optimized Byzantine fault tolerance algorithm for consortium blockchain. Peer-to-Peer Networking and Applications, 2021, vol. 14, no. 10, pp. 9899–9995.

35. O. Green. Hash Graph-scalable hash tables using A sparse graph data structure. ACM Transactions on Parallel Computing, 2019, vol. 8, no. 2,

36. J.C. Bronski, L. Deville. Spectral theory for networks with attractive and repulsive interactions. Mathematics, 2013, vol. 128, no. 11, pp. 128–136.

37. T. Lasy. From Hashgraph to a family of atomic broadcast algorithms. The Knowledge Engineering. 2019, vol. 12, no. 10, pp. 1609–1612.

38.

39.

ДОДАТОК А

Копії публікацій

ГРОМАДСЬКЕ ОБЄДНАННЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2023
Тернопіль

ЗМІСТ

УДК 004.056.5

ВАСИЛЬКІВ В.О., БАСІСТИЙ В.П., СИДОРЧУК Р.В.
Західноукраїнський національний університет

БЛОКЧЕЙН-ПЛАТФОРМА HYPERLEDGER FABRIC

Вступ. Технологія блокчейн стала важливим словом у світі бізнесу. Його здатність захищати дані, зменшувати витрати та підвищувати ефективність робить його привабливим варіантом для підприємств у різних галузях.

Hyperledger Fabric – блокчейн-платформа з відкритим кодом, яка привернула значну увагу в останні роки завдяки своїм надійним функціям і гнучкості. Hyperledger Fabric створена Linux Foundation у 2015 році. Вона розроблена як гнучка, модульна та масштабована структура блокчейну, яку можуть використовувати підприємства для створення децентралізованих програм. Hyperledger Fabric використовує дозволену мережеву архітектуру, що означає, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн. Це робить її ідеальною платформою для підприємств, яким потрібна сувора конфіденційність і безпека даних.

1. Характеристики Hyperledger Fabric

Однією з ключових особливостей Hyperledger Fabric є її високий ступінь гнучкості. Це дозволяє підприємствам налаштовувати алгоритм консенсусу, мову смарт-контракту та керування ідентифікацією відповідно до своїх конкретних потреб. Hyperledger Fabric також підтримує кілька мов програмування, включаючи Go, Java і Node.js, що полегшує розробникам створення програм на основі мережі блокчейн.

Ключові характеристики Hyperledger Fabric. Hyperledger Fabric має кілька ключових особливостей, які роблять її привабливим варіантом для підприємств. До них належать [1].

1. Масштабованість. Hyperledger Fabric розроблено як масштабована блокчейн-платформа. Вона використовує модульну архітектуру, яка дозволяє створювати кілька каналів в одній мережі блокчейн. Це означає, що підприємства можуть створювати окремі канали для різних підрозділів або відділів, які можна масштабувати незалежно.

2. Гнучкість. Hyperledger Fabric –гнучка блокчейн-платформа. Це дозволяє підприємствам налаштовувати алгоритм консенсусу, мову смарт-контракту та керування ідентифікацією відповідно до своїх конкретних потреб. Hyperledger Fabric також підтримує кілька мов програмування, включаючи Go, Java і Node.js, що полегшує розробникам створення програм на основі мережі блокчейн.

3. Конфіденційність і безпека. Hyperledger Fabric використовує дозволену мережеву архітектуру, що означає, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн. Це забезпечує конфіденційність і безпеку даних, що робить його ідеальною платформою для підприємств, яким потрібні суворі заходи конфіденційності та безпеки.

4. Модульна архітектура. Hyperledger Fabric використовує модульну

архітектуру, яка дозволяє створювати кілька каналів в одній мережі блокчейн. Це означає, що підприємства можуть створювати окремі канали для різних підрозділів або відділів, які можна масштабувати незалежно.

5. Механізм консенсусу. Hyperledger Fabric підтримує підключаються алгоритми консенсусу, дозволяючи учасникам мережі вибирати найбільш підходящий алгоритм для свого випадку використання, будь то візантійський відмовостійкий алгоритм або практичний візантійський відмовостійкий алгоритм.

2. Компоненти платформи Hyperledger Fabric

До основних компоненти платформи Hyperledger Fabric відносяться:. однорангові вузли, вузли-замовники, центри сертифікації, канали та Chaincode.

Однорангові вузли підтримують спільну книгу, виконують ланцюжковий код, схвалюють транзакції та беруть участь у процесі консенсусу.

Вузли-замовники. Вузли-замовники встановлюють порядок транзакцій, упаковують їх у блоки та розповсюджують на однорангові вузли для перевірки та виконання.

Центри сертифікації (ЦС). ЦС відповідають за керування ідентифікацією в мережі. Вони видають криптографічні сертифікати учасникам мережі, забезпечуючи безпечне спілкування та авторизацію транзакцій.

Канали. Канали дозволяють створювати підмережі в межах основної мережі Hyperledger Fabric, забезпечуючи приватні та конфіденційні транзакції між конкретними учасниками.

Chaincode. Chaincode містить бізнес-логіку програми та визначає правила перевірки та виконання транзакцій.

У Hyperledger Fabric також існує концепція каналів, яка дозволяє організаціям-учасникам приєднуватися та спілкуватися одна з одною. Канал можна розглядати як тунель для однієї організації для таємного спілкування з іншими організаціями-учасниками, які приєднуються до того самого каналу. Будь-які інші особи, які не беруть участі в каналі, про який йде мова, ніколи не матимуть доступу до жодної транзакції чи інформації, пов'язаної з цим каналом. Одна організація може брати участь у кількох каналах одночасно.

На рисунку 1 зображено найпростішу мережу Hyperledger Fabric із двома організаціями (Org1 та Org2), які приєднуються до одного каналу.

Hyperledger Fabric включає наступні компоненти: Peer, Orderer, CA та Client [2].

1. Peer це вузол блокчейну, який зберігає всі транзакції на каналі приєднання. Кожен пір може приєднатися до одного або кількох каналів за потреби. Однак сховище для різних каналів на одному вузлі буде окремим. Таким чином, організація може гарантувати, що конфіденційна інформація буде передана лише дозволеним учасникам певного каналу.

2. Orderer є одним із найважливіших компонентів, які використовуються в механізмі консенсусу Fabric. Orderer – це служба, яка відповідає за впорядкування транзакцій, створення нового блоку замовлених транзакцій і розповсюдження новостворено блоку всім одноранговим користувачам у відповідному каналі.

3. CA – центр сертифікації, який відповідає за керування сертифікатами користувачів, такими як реєстрація користувачів, відкликання користувачів тощо.

4. Клієнт вважається додатком, який взаємодіє з мережею блокчейнів Fabric. Тобто Клієнт може взаємодіяти з мережею Fabric відповідно до своїх дозволів, ролей і атрибутів, як зазначено в його сертифікаті, отриманому від сервера CA пов'язаної організації.

Щоб розробляти програми за допомогою Hyperledger Fabric, важливо добре розуміти архітектуру.

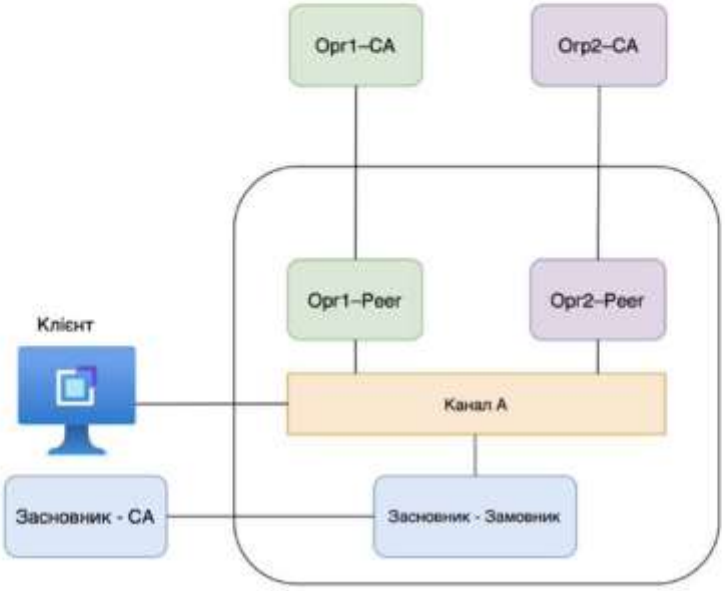


Рисунок 1 – Мережа Hyperledger Fabric із двома організаціями

Висновки. Hyperledger Fabric це дозволена мережа блокчейн. Це означає, що лише авторизовані користувачі можуть запитувати (отримувати доступ до інформації) або викликати (створювати нову транзакцію) транзакцію на наданому каналі. Hyperledger Fabric використовує стандартний сертифікат X.509 для представлення дозволів, ролей і атрибутів для кожного користувача. Іншими словами, користувач може запитувати або викликати будь-яку транзакцію на будь-якому каналі на основі дозволів, ролей і атрибутів, якими він володіє.

Hyperledger Fabric може допомогти підприємствам покращити свою діяльність і зменшити витрати, надаючи безпечну та прозору платформу для здійснення транзакцій. Архітектура мережі Hyperledger Fabric із дозволами гарантує, що лише авторизовані учасники можуть отримати доступ до мережі блокчейн, що означає гарантовану конфіденційність і безпеку даних.

Перелік використаних джерел.

1. Understanding Hyperledger Fabric's Architecture. [Електронний ресурс]. – Режим доступу: https://medium.com/hyperlegendary/understanding-hyperledger-fabrics-architecture-3b37d81c3e96

2. Demystifying Hyperledger Fabric (1/3): Fabric Architecture. [Електронний ресурс]. – Режим доступу: https://www.serial-coder.com/post/demystifying-hyperledger-fabric-fabric-architecture/



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2023)

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2023
Тернопіль

ЗМІСТ

СИСТЕМИ ТА ТЕХНОЛОГІЇ КІБЕРБЕЗПЕКИ

УДК 004.056.5

Голод Ю.В., Сидорчук Р.В., Васильків В.О.

Західноукраїнський національний університет

АНАЛІЗ ВРАЗЛИВОСТЕЙ АЛГОРИТМІВ КОНСЕНСУСУ

Вступ. Блокчейн – це децентралізована мережа, у якій немає центрального вузла для спостереження та перевірки всіх транзакцій. Таким чином, існує потреба в розробці протоколів, які вказуватимуть, що всі транзакції правильні. У розподілених системах консенсус став проблемою, коли всі члени мережі (вузли) погоджуються прийняти або відхилити блок. Коли новий блок приймають усі члени мережі, його можна додати до попереднього блоку.

Програми на основі блокчейну зараз стають поширеними в усіх аспектах людського життя для сприяння безпеці, довірі та надійності. Безпека блокчейну залежить від надійності та потужності алгоритму консенсусу, який використовується для перевірки транзакцій і блоків.

Мета: аналіз враливостей алгоритмів консесусу.

1. Типи блокчейн

На даний час виділяють три типи блокчейну: публічний, приватний і гібридний [1].

1. Публічний блокчейн – це тип мережі блокчейну, яка є відкритою та доступною для всіх, хто хоче взяти участь. Він працює децентралізовано, тобто жодна особа чи орган влади не контролює мережу. Натомість учасники мережі, які називають вузлами, спільно підтримують і перевіряють блокчейн. У загальнодоступному блокчейні будь-хто може приєднатися до мережі, додати обчислювальну потужність і брати участь у таких діях, як перевірка транзакцій і створення блоків. Прозорість публічних блокчейнів дозволяє кожному переглядати та перевіряти всю історію транзакцій, забезпечуючи високий рівень довіри та незмінності. Добре відомі приклади публічних блокчейнів включають Bitcoin, Ethereum і Litecoin. Ці загальнодоступні блокчейни поширені та відкриті для будь-кого, забезпечуючи прозору та безпечну інфраструктуру для різноманітних децентралізованих програм і цифрових активів.

2. Приватний блокчейн – це тип мережі блокчейну, яка працює в межах певної організації або групи відомих і надійних організацій. На відміну від загальнодоступних блокчейнів, приватні блокчейни мають обмежений доступ, тобто права участі та перевірки зазвичай надаються обмеженій кількості попередньо схвалених учасників. У приватному блокчейні контроль і управління мережею централізовано або розподілено між певним набором учасників. Рівень децентралізації може змінюватися залежно від конкретного дизайну та вимог приватного блокчейна. Приватні блокчейни зазвичай використовуються в різних галузях, включаючи фінанси, управління ланцюгами поставок і охорону здоров'я, де учасникам необхідно співпрацювати та безпечно обмінюватися інформацією в закритій екосистемі. Приклади приватних структур блокчейну включають Hyperledger Fabric, R3 Corda тощо.

3. Гібридний блокчейн – це комбінація публічних і приватних елементів

блокчейну, спрямована на використання переваг обох моделей. Він прагне забезпечити рішення, яке поєднує в собі прозорість і децентралізацію публічних блокчейнів із конфіденційністю та контролем приватних блокчейнів. У гібридному блокчейні певні аспекти мережі залишаються відкритими, що забезпечує відкриту участь, прозору перевірку та консенсус серед більшої кількості учасників. У той же час певні елементи або транзакції в межах блокчейну можна обмежити та залишити приватними, обмежуючи доступ до вибраної групи відомих учасників.

2. Алгоритми консенсусу блокчейн

Алгоритм консенсусу (АК) – це набір правил або протоколів, які дозволяють вузлам у мережі блокчейн узгодити спільний стан мережі. Вони використовуються для забезпечення того, щоб усі вузли в мережі дійшли згоди щодо дійсності транзакцій і порядку їх додавання до блокчейну [2].

АК відповідає за підтримку цілісності блокчейна, гарантуючи, що жоден вузол або група вузлів не зможе маніпулювати мережею. АК є критично важливими в технології блокчейн з кількох причин. Вони забезпечують безпеку, перешкоджаючи зловмисникам отримати контроль над мережею, забезпечуючи дійсні транзакції та безперебійну роботу мережі.

АК допомагають досягти децентралізації, гарантуючи, що всі вузли досягають консенсусу щодо дійсності транзакції, запобігаючи централізації. АК сприяють прозорості, роблячи всі транзакції видимими в блокчейні, полегшуючи відстеження та запобігання шахрайським діям. Вони підвищують ефективність, дозволяючи вузлам швидко погоджувати дійсність транзакцій і своєчасно додавати нові блоки в блокчейн.

Проблеми з АК виникають через основні недоліки в дизайні. Важливо розуміти, що проблеми з АК можуть заважати довгостроковому застосуванню рішень на основі блокчейну. У більшості випадків АК блокчейнів дотримуються принципу забезпечення безпеки через дефіцит. Контроль над обмеженими ресурсами може визначати владу над створенням блоків і розширенням мережі блокчейнів.

3. Атаки та алгоритми консенсусу

Атака подвійних витрат відбувається, коли людина намагається витратити певну суму грошей в блокчейні двічі. Це може статися, коли зловмисник намагається створити звичайну транзакцію, щоб включити її в блок, а потім через деякий час створює шахрайську конфліктну транзакцію та вставляє її в новий розгалужений шахрайський блок, намагаючись скасувати раніше здійснену транзакцію. Потім зловмисник повинен спробувати розширити створену ним шахрайську гілку мережі, доки шахрайська гілка не буде перевірена та прийнята як правильна гілка, яка включає шахрайську транзакцію.

Незважаючи на те, що різні АК намагаються пом'якшити цю вразливість і мають різні механізми для її усунення, подвійних витрат неможливо повністю уникнути в системах блокчейн, і теоретично це можливо постійно.

Тип атаки 51% вперше був застосований у мережі блокчейну біткойна, але його також можна запустити в інших системах блокчейну. Атаки 51% також

теоретично неможливо уникнути. Протоколи блокчейну намагаються збільшити вартість цієї атаки, щоб захистити її, але можуть бути не в змозі повністю запобігти їй. Коли зловмисник може контролювати понад 50% потужності у блокчейні, він може виконувати зловмисні дії, такі як подвійне витрачання або перешкоджати іншим вузлам отримувати їхні чесні транзакції. Цей тип атаки називається атакою 51%. Зловмисник не завжди повинен володіти 51% потужності мережі, тоді як він може підкупити інші вузли, щоб вони пішли за ним, або він може тимчасово орендувати необхідну йому потужність. Таким чином, цей тип атаки завжди повинен приділяти увагу порівнянню безпеки блокчейну. З точки зору порівняння, стверджує, що алгоритми PoW, PoS і DPoS, хоча і діють по-різному проти атаки 51%, але є вразливі до неї. Однак алгоритм PoA підвищує вартість 51% атаки, оскільки зловмиснику потрібно мати 51% усіх монет і 51% потужності майнінгу одночасно.

Атака Sybil. Це загальна форма атаки, під час якої зловмисник намагається контролювати однорангову мережу, створюючи низку шахрайських ідентифікаторів у блокчейні. Ці особи здаються унікальними користувачами або вузлами, які фактично контролює зловмисник. Ці ідентифікаційні дані використовуються для отримання права голосу, блокування повноважень перевірки або навіть трансляції фальшивого повідомлення в мережі соціальних повідомлень блокчейну. Успішна атака Sybil може надати зловмиснику непропорційний контроль над мережею або оточити чесний вузол і спробувати вплинути на інформацію, що доходить до нього, а потім впливати на блокчейн.

Атаки Sybil важко виявити та запобігти, але блокчейни намагаються застосувати власні підходи, щоб запобігти цьому. Наведено деякі з підходів, які блокчейн може використовувати для запобігання атаці Sybil.

1. Збільшення вартості створення вузла. Підвищення вартості створення ідентичності є першим підходом до зменшення ризику атаки Sybil.

2. Вимагання певного типу довіри. Другим поширеним способом боротьби з атаками Sybil є вимога форми довіри перед тим, як дозволити вузлу приєднатися до блокчейну. Цією формою довіри може стати проста двоетапна перевірка електронною поштою/SMS або запит на підтвердження від групи адміністраторів.

3. Надання неоднакової влади ідентичностям. Надання різних повноважень користувачам і вузлам є ще одним способом захисту від атаки Sybil.

Висновок. Розглянуто найпоширеніші атаки на безпеку, які теоретично можуть загрожувати майже всім типам алгоритмів консенсусу. Існують також інші типи атак і вразливостей у блокчейн протоколах, приведеним загальним і фундаментальним вразливостям завжди необхідно приділяти найбільшу увагу при порівнянні оцінці різних типів блокчейнів.

Перелік використаних джерел.

1. Beginner's Guide to Consensus Algorithms in Blockchain Technology. [Електронний ресурс]. – Режим доступу: https://medium.com/@learnwithwhiteboard_digest/beginners-guide-to-consensus-algorithms-in-blockchain-technology-34c2026b2b36

2. Challenges and Security issues in Consensus Algorithm. [Електронний ресурс]. – Режим доступу: https://101blockchains.com/consensus-algorithm-issues/