НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»

УДК 004.738:004.056.5

«До захисту допущено»
Завідувач кафедри
_____ Дмитро ЛАНДЕ
19 червня 2025 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»

зі спеціальності 125 «Кібербезпека»

на тему: Інтеграція блокчейну для безпечного децентралізованого управління даними IoT

Виконав(ла) здобувач вищої освіти IV курсу, групи ФБ-12
(шифр групи)

Юрченко Вікторія Богданівна
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент кафедри ІБ к.т.н. Коломицев М. В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доцент кафедри ІСТ, к.т.н., доцент Пасько В. П.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
19 червня 2025 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Юрченко Вікторії Богданівни
(прізвище, ім'я, по батькові)

1. Тема роботи Інтеграція блокчейну для безпечного децентралізованого управління даними IoT,
науковий керівник роботи Коломицев Михайло Володимирович, к.т.н., доцент, доцент кафедри інформаційної безпеки,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 26 травня 2025 р. № 1761-с

2. Термін подання здобувачем дипломної роботи 12 червня 2025 р.

3. Вихідні дані до роботи

Літературні джерела з тематики безпеки систем Інтернету речей, статті про протоколи консенсусу та типи блокчейну, технічна документація IoTeX, рекомендації ENISA щодо захисту IoT-систем.

4. Зміст роботи

Аналіз проблем централізованого управління даними в IoT-системах, класифікація типів блокчейнів, огляд і порівняння консенсусних алгоритмів, визначення переваг блокчейну для забезпечення безпеки IoT, опис механізмів смарт-контрактів для автоматизованої взаємодії між пристроями, розробка архітектури безпечної децентралізованої моделі управління даними IoT, реалізація прототипу на основі

PBFT, розробка смарт-контрактів для управління доступом і реакцією на події, тестування моделі.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація.

6. Дата видачі завдання: 16 вересня 2024 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Визначення теми ДР	16.09.2024 – 17.09.2024	Виконано
2	Огляд літератури по темі	18.09.2024 – 22.09.2024	Виконано
3	Підготовка плану ДР	23.09.2024 – 05.10.2024	Виконано
4	Визначення загроз безпеці IoT	14.04.2025 – 16.04.2025	Виконано
5	Визначення вимог до безпеки систем Інтернету речей	17.04.2025 – 20.04.2025	Виконано
6	Аналіз використання блокчейну для забезпечення безпеки	21.04.2025 – 24.04.2025	Виконано
7	Аналіз прикладів інтеграції блокчейну з IoT	25.04.2025 – 27.04.2025	Виконано
8	Аналіз викликів та обмежень такої інтеграції	28.04.2025 – 30.04.2025	Виконано
9	Розробка концепції моделі	01.05.2025 – 04.05.2025	Виконано
10	Реалізація прототипу блокчейну з PBFT	05.05.2025 – 06.05.2025	Виконано
11	Реалізація смарт-контрактів	07.05.2025 – 09.05.2025	Виконано
12	Реалізація головного модуля мережі	10.05.2025 – 11.05.2025	Виконано
13	Тестування розробленої системи	12.05.2025	Виконано
14	Написання першого розділу	13.05.2025 – 18.05.2025	Виконано
15	Написання другого розділу	19.05.2025 – 23.05.2025	Виконано
16	Написання третього розділу	24.05.2025 – 28.05.2025	Виконано
17	Написання четвертого розділу	29.05.2025 – 01.06.2025	Виконано
18	Оформлення ДР	02.06.2025 – 04.06.2025	Виконано
19	Створення презентації для передзахисту дипломної роботи	05.06.2025 – 07.06.2025	Виконано
20	Передзахист	12.06.2025	Виконано
21	Захист	19.06.2025	Виконано

Здобувач вищої освіти _____ Вікторія ЮРЧЕНКО
(підпис) (власне ім’я, ПРІЗВИЩЕ)

Науковий керівник _____ Михайло КОЛОМИЦЕВ
(підпис) (власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Обсяг роботи 73 сторінки, 17 ілюстрацій, 2 таблиці, 5 додатків, 29 джерел літератури.

Об’єкт дослідження: процес управління даними в системах Інтернету речей, що функціонують у розподіленому мережевому середовищі.

Предмет дослідження: архітектура та механізми безпечного децентралізованого управління даними в IoT-системах на основі консорціумного блокчейну, протоколу консенсусу PBFT та смарт-контрактів.

Мета дослідження: Підвищення безпеки управління даними в системах Інтернету речей за рахунок впровадження децентралізованої моделі на основі консорціумного блокчейну з використанням протоколу PBFT та смарт-контрактів для автоматизованого контролю доступу і фіксації подій.

Методи дослідження: Аналіз наукових публікацій та технічної документації з безпеки IoT і блокчейну, дослідження особливостей типів блокчейну та протоколів консенсусу, моделювання концепції мережі, побудова практичної моделі.

Отримані результати: Розроблено архітектуру моделі інтеграції блокчейну в середовище IoT з урахуванням принципів децентралізації, енергоефективності та керованості доступом. Реалізовано прототип консорціумної блокчейн-мережі з протоколом PBFT, у якій IoT-пристрої взаємодіють через edge-шлюзи. Розроблено смарт-контракти для автоматичного управління доступом, активації пристроїв, ізоляції скомпрометованих вузлів. Система протестована на сценаріях взаємодії вузлів, що підтвердило її функціональність і відповідність заявленим вимогам.

Результати роботи були представлені на XXIII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених.

Ключові слова: блокчейн, IoT, консенсус, смарт-контракт, PBFT, децентралізація, безпека

ABSTRACT

The volume of the thesis is 73 pages, 17 illustrations, 2 tables, 5 appendices, 29 sources of literature.

Research object: the process of data management in Internet of Things systems operating in a distributed network environment.

The subject of the research: the architecture and mechanisms of secure decentralized data management in IoT systems based on a consortium blockchain, the PBFT consensus protocol, and smart contracts.

The purpose of the research: Enhancing the security of data management in Internet of Things systems by implementing a decentralized model based on a consortium blockchain using the PBFT protocol and smart contracts for automated access control and event logging.

Research methods: Analysis of scientific publications and technical documentation on IoT and blockchain security, study of blockchain types and consensus protocols, modeling of the network concept, development of a practical system model.

Obtained results: An architectural model for blockchain integration into the IoT environment was developed, taking into account the principles of decentralization, energy efficiency, and access control. A prototype of a consortium blockchain network using the PBFT protocol was implemented, in which IoT devices interact via edge gateways. Smart contracts were developed for automatic access management, device activation, and isolation of compromised nodes. The system was tested in node interaction scenarios, which confirmed its functionality and compliance with the stated requirements.

The results of the work were presented at the XXIII All-Ukrainian Scientific and Practical Conference of Students, Postgraduates and Young Scientists.

Keywords: blockchain, IoT, consensus, smart contract, PBFT, decentralization, security

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IoT – Інтернет речей (Internet of Things)

IT – інформаційні технології

DDoS – розподілена відмова в обслуговуванні (Distributed Denial of Service)

DNS – система доменних імен (Domain Name System)

IP – Internet Protocol

DoS – відмова в обслуговуванні (Denial of Service)

MitM – людина посередині (Man-in-the-Middle)

ENISA – агентство Європейського Союзу з питань мережевої та інформаційної безпеки

RBAC – керування доступом на основі ролей

AES – Advanced Encryption Standard

ПЗ – програмне забезпечення

GDPR – загальний регламент про захист даних

ЄС – Європейський союз

PoW – протокол консенсусу Proof of Work

SHA – Secure Hash Algorithm

PoS – протокол консенсусу Proof of Stake

DPoS – протокол консенсусу Distributed Proof of Stake

PBFT – протокол консенсусу Practical Byzantine Fault Tolerance

PoA – протокол консенсусу Proof of Authority

PoET – протокол консенсусу Proof of Elapsed Time

TEE – Trusted Execution Environment

IoTeX – децентралізована блокчейн-платформа

VRF – перевірені випадкові функції (Verifiable Random Functions)

IOTX – нативний токен IoTeX

MQTT – Message Queue Telemetry Transport

TLS – Transport Layer Security

TPM – Trusted Platform Module

HSM – Hardware Security Module

CO_2 – вуглекислий газ

$PM_{2.5}$ – тверді частинки діаметром менше 2,5 мкм

PM_{10} – тверді частинки діаметром менше 10 мкм

TVOC – загальна кількість летких органічних сполук

CO – чадний газ

JSON – JavaScript Object Notation

CSV – Comma-Separated Values

ВСТУП

Актуальність роботи:

Інтернет речей (IoT) стрімко розвивається та охоплює дедалі більше сфер – від побутових пристроїв до критичної інфраструктури та промислових систем. Збільшення кількості IoT-пристроїв тягне за собою зростання обсягів даних, які передаються та обробляються в реальному часі, часто без належного рівня захисту. Більшість існуючих IoT-систем базуються на централізованих архітектурах, що створює залежність від окремих точок контролю, які можуть бути скомпрометовані або виведені з ладу.

Блокчейн як технологія розподіленого реєстру може усунути ці вразливості, однак її безпосередня інтеграція в IoT середовище ускладнюється обмеженими ресурсами пристроїв і вимогами до швидкодії. Це формує потребу у створенні адаптованих рішень, які б поєднували децентралізоване управління, контроль доступу та захист даних, зберігаючи при цьому продуктивність і енергоефективність. Робота спрямована саме на вирішення цього завдання, що робить її актуальною у контексті розвитку безпечних IoT-систем.

Метою дослідження: підвищення безпеки управління даними в системах Інтернету речей за рахунок впровадження децентралізованої моделі на основі консорціумного блокчейну з використанням протоколу PBFT та смарт-контрактів для автоматизованого контролю доступу і фіксації подій.

Завдання дослідження:

1. Проаналізувати особливості функціонування та загрози безпеці сучасних IoT-систем.
2. Обґрунтувати вибір консорціумної блокчейн-архітектури та протоколу PBFT у контексті забезпечення безпеки IoT.
3. Розробити прототип системи з використанням смарт-контрактів для автоматизації.
4. Провести тестування моделі в різних сценаріях взаємодії вузлів.

Об’єкт дослідження: процес управління даними в системах Інтернету речей, що функціонують у розподіленому мережевому середовищі.

Предмет дослідження: архітектура та механізми безпечного децентралізованого управління даними в IoT-системах на основі консорціумного блокчейну, протоколу консенсусу PBFT та смарт-контрактів.

Методи дослідження: Аналіз наукових публікацій та технічної документації з безпеки IoT і блокчейну, дослідження особливостей типів блокчейну та протоколів консенсусу, моделювання концепції мережі, побудова практичної моделі.

Новизна одержаних результатів: У роботі запропоновано архітектурну модель IoT-системи з децентралізованим управлінням даними, яка адаптована до обмежених ресурсів пристроїв шляхом використання edge-шлюзів, консорціумного блокчейну та протоколу PBFT. Особливістю моделі є поєднання смарт-контрактів для автоматизованого контролю доступу та ізоляції скомпрометованих вузлів з ефективною схемою взаємодії між компонентами системи без потреби в централізованому координаторі. Модель має прикладне спрямування, що дозволяє перевірити її функціональність на практиці.

Практичне значення одержаних результатів: Результати роботи можуть бути застосовані для побудови IoT-систем, які потребують безпечного обміну даними між пристроями без участі центрального контролера. Модель адаптована до обмежених ресурсів IoT-пристроїв і може впроваджуватись у промислові або інфраструктурні середовища з високими вимогами до безпеки. Створений прототип і структура смарт-контрактів можуть бути використані як основа для подальшої розробки більш масштабних рішень або для адаптації в реальних мережевих системах, де важлива децентралізація, контроль доступу та простежуваність дій.

Апробація результатів роботи: Результати дослідження були представлені на XXIII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (14 – 17.05.2025 р., м. Київ, Україна).

1 АНАЛІЗ БЕЗПЕКИ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Поняття та застосування IoT

Інтернет речей (англ. Internet of Things, IoT) – мережа фізичних об’єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію з зовнішнім середовищем, передавати відомості про свій стан і приймати дані ззовні [1]. До такої системи входять сенсори, датчики, мікроконтролери, радіомодулі, програмне забезпечення та засоби підключення до мережі. Завдяки взаємодії цих компонентів, кінцеві пристрої обмінюються даними між собою та з центральними системами управління, створюючи єдине інформаційне середовище для автоматизації процесів.

На сьогоднішній день сфера застосування IoT-пристроїв доволі широка. Хоча їх зазвичай використовують для збирання даних про навколишнє середовище, вони знайшли своє місце й у багатьох інших галузях [2]:

- Розумні будинки – керування освітленням, клімат-контроль, споживання енергії, безпека домівки.
- Медицина – дистанційне спостереження за станом пацієнтів.
- Промисловість – моніторинг стану обладнання, автоматизація виробничих процесів.
- Сільське господарство – контроль стану врожаю, температури та вологості в ґрунті.
- Транспорт – моніторинг дорожнього руху, допомога в паркуванні.
- Критична інфраструктура – моніторинг енергосистем, управління водопостачанням, вентиляцією.

Очікується, що кількість підключених пристроїв Інтернету речей перевищить 30 мільярдів до кінця десятиліття [3], що створює як нові можливості, так і виклики

щодо безпеки, енергозбереження, управління великими обсягами даних і сумісності між пристроями.

1.2 Edge-шлюзи

Edge-шлюз (також відомий як IoT-шлюз) – це пристрій, що функціонує на краю мережі та виконує роль посередника між IoT-пристроями та інфраструктурою обробки даних (локальною або хмарною). Основна функція edge-шлюзу полягає у перетворенні специфічних протоколів Інтернету речей у формати даних, придатних для обробки і агрегації.

На відміну від традиційної моделі, де всі дані надсилаються до хмари, edge-шлюзи дозволяють виконувати аналітику безпосередньо на краю мережі. Це означає, що незначущі або надлишкові дані можуть бути відфільтровані до того, як потраплять до хмари, знижуючи витрати на їх передачу та покращуючи продуктивність системи.

Завдяки своїй проміжній позиції між рівнем пристроїв та інфраструктурою обробки, edge-шлюз виступає не просто як засіб маршрутизації даних, а як самостійна обчислювальна одиниця з власними механізмами захисту, управління та попереднього аналізу інформації. Він дозволяє реалізувати концепцію edge computing, яка спрямована на перенесення обчислювальних процесів ближче до джерела генерації даних – тобто до самих IoT-пристроїв. Це має особливе значення в умовах, коли системи потребують мінімальних затримок у прийнятті рішень, наприклад, у виробництві, транспорті чи енергетиці.

Edge-шлюзи також відіграють важливу роль у вирішенні проблем сумісності між різними поколіннями пристроїв, оскільки здатні конвертувати протоколи обміну даними між застарілими системами та сучасними цифровими платформами. Це робить їх незамінним компонентом у гетерогенних IoT-мережах, де спостерігається велика різноманітність технічних протоколів, частот оновлення та типів даних.

З технічного боку, edge-шлюзи зазвичай обладнані обчислювальними модулями, модулями зберігання даних, інтерфейсами підключення до різних мережевих середовищ (дротових та бездротових), а також засобами забезпечення криптографічного захисту. Поряд з апаратним забезпеченням, програмна складова включає засоби шифрування, засоби контролю автентичності, програмне забезпечення для обробки даних та можливість інтеграції з хмарними платформами для централізованого моніторингу та оновлення. [4-5]

1.3 Основні загрози безпеці Інтернету речей

На відміну від традиційних IT-систем, пристрої IoT характеризуються обмеженими обчислювальними ресурсами, відсутністю уніфікованих стандартів і значною різнорідністю. Через ці особливості системи Інтернету речей стають вразливими до різноманітних атак, які можуть впливати як на окремі пристрої, так і на всю інфраструктуру в цілому. Враховуючи це, загрози в IoT можна класифікувати за кількома основними групами. [6]

1.3.1 Несанкціонований доступ

Однією з найпоширеніших загроз є отримання зловмисниками несанкціонованого доступу до пристроїв Інтернету речей. Це часто обумовлено слабкою автентифікацією, використанням стандартних паролів і також тим, що багато пристроїв не отримують оновлень безпеки. Унаслідок цього атакуючі можуть з легкістю отримати контроль над пристроєм, змінити його конфігурацію, змусити працювати всупереч призначенню або повністю вивести з ладу.

Скомпрометовані пристрої можуть стати частиною мережі ботнету і використовуватись для атак на інші сервіси. На рисунку 1.1 зображено спрощений механізм функціонування IoT-ботнету, включаючи процес зараження пристроїв, їхнє об’єднання в мережу та використання для кібератак.

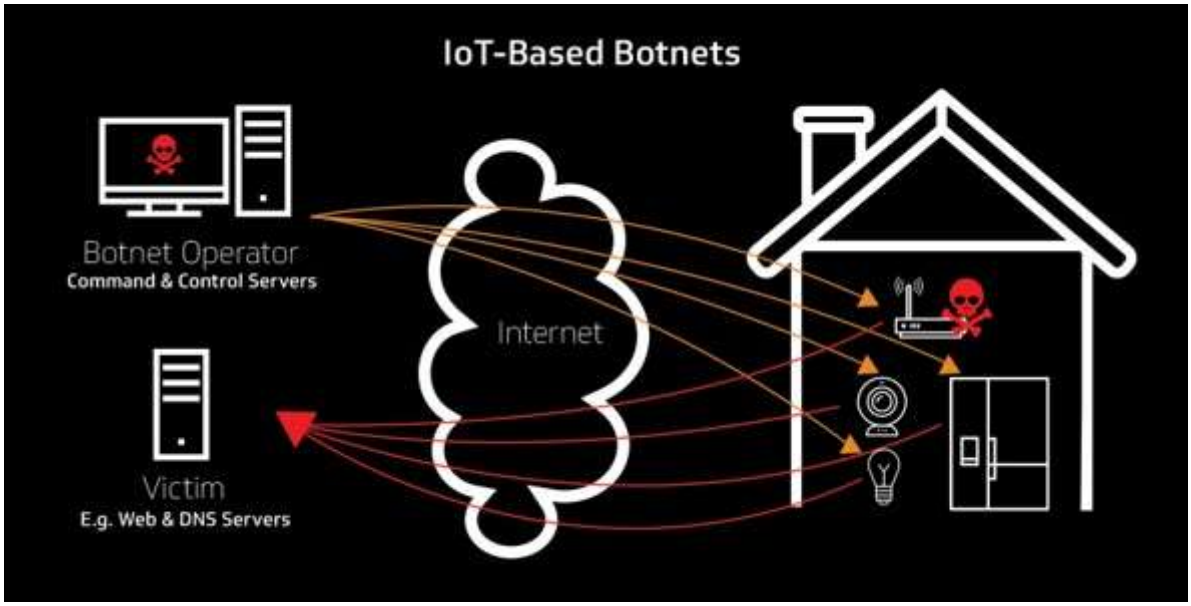

Рисунок 1.1 – Ботнети на основі IoT [7]

Відомим прикладом є мережа ботнету Mirai, що складалася з пристроїв IoT, які мали облікові дані за замовчуванням. У жовтні 2016 року ботнет Mirai спричинив масштабну DDoS-атаку на DNS-провайдера Dyn, що призвело до перебоїв у роботі таких сайтів, як Twitter, Netflix, Spotify і Reddit. Зловмисники використали десятки мільйонів IP-адрес для надсилання запитів, заблокувавши обслуговування клієнтів [8].

Загроза несанкціонованого доступу ускладнюється ще й тим, що багато IoT-пристроїв не мають можливості інтерактивного захисту, на кшталт блокування користувача після кількох невдалих спроб входу або використання двофакторної автентифікації. У поєднанні з великою кількістю пристроїв і слабким мережевим захистом, це робить Інтернет речей надзвичайно вразливим до масових експлуатацій.

1.3.2 Атаки типу «Відмова в обслуговуванні» (DoS)

Іншою загрозою для IoT-систем є атаки типу відмова в обслуговуванні (DoS) або їх розподілений варіант (DDoS). Такі атаки полягають у надсиланні надмірної

кількості запитів до пристрою або сервера з метою перевантаження його ресурсів і виведення з ладу.

IoT-пристрої, як правило, мають обмежену обчислювальну потужність та малий обсяг пам’яті. До того ж, враховуючи централізовану природу Інтернету речей, можна атакувати лише точку контролю, щоб паралізувати всю систему.

Окрім прямого впливу, DoS-атаки можуть виступати як засіб відволікання уваги або попередня фаза складніших вторгнень. Наприклад, виведення з ладу системи відеоспостереження може супроводжуватися паралельною спробою фізичного проникнення до об’єкта. Через це атаки типу DoS в IoT слід розглядати не лише як технічну проблему, а й як компонент складніших загроз для безпеки організацій.

1.3.3 Маніпуляція даними

Ще однією загрозою для систем Інтернету речей є маніпуляція даними. Зловмисники можуть перехопити, змінити або фальсифікувати дані, що передаються між пристроями. Особливо критичною ця загроза стає при обробці інформації з обмеженим доступом.

Використання слабких протоколів передачі даних, недостатнє шифрування, відсутність контролю цілісності інформації відкривають широкі можливості для атак типу «людина посередині» (MitM) (рисунок 1.2). Атакуючий, перехопивши трафік між сенсором і контролером, може змінити критичні параметри – наприклад, температуру, тиск, рівень забруднення – й викликати неправильну реакцію системи. Таке втручання може призвести до порушення конфіденційності, цілісності та навіть фізичної безпеки.

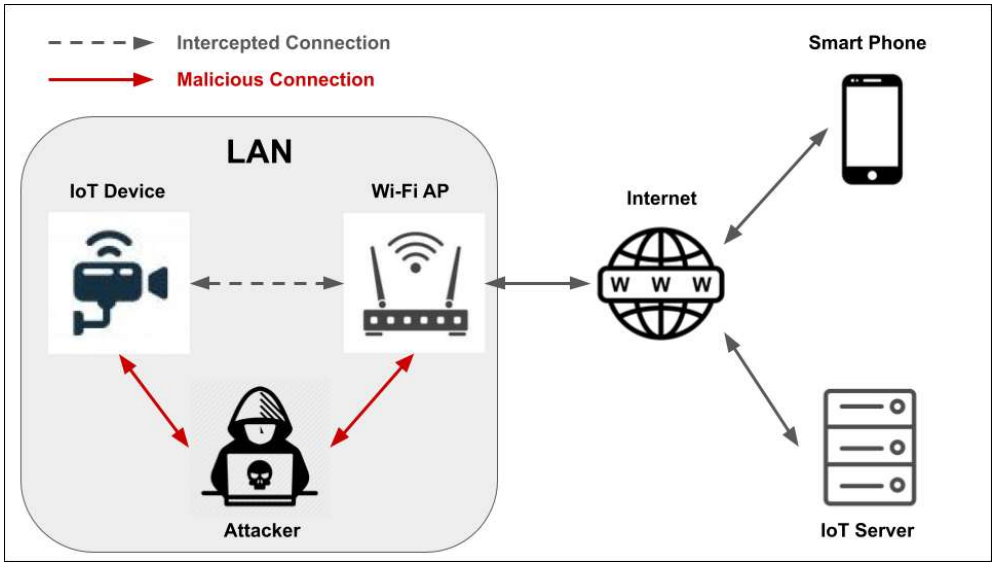

Рисунок 1.2 – MitM в IoT-системах [9]

1.3.4 Фізичні атаки

Фізичні атаки є особливо актуальними для середовищ, де IoT-пристрої розміщуються поза межами контрольованих приміщень. Відсутність постійного фізичного контролю дозволяє зловмисникам отримати безпосередній доступ до пристрою, що відкриває нові вектори загроз.

В умовах реального середовища однією з найтиповіших атак є заміна або модифікація пристрою на рівні сенсора. Наприклад, зловмисник може встановити фальшивий датчик руху або температури, який генерує штучно «нормальні» показники, маскуючи фактичну присутність сторонніх осіб або технічну несправність. Подібні атаки небезпечні тим, що можуть залишатися непоміченими тривалий час, особливо якщо система не має механізмів верифікації джерела даних.

1.4 Вимоги до безпеки IoT-систем

Для ефективного захисту IoT-інфраструктури необхідне впровадження багаторівневого підходу до безпеки. Відповідно до рекомендацій ENISA [10] та

враховуючи виклики середовищ Інтернету речей, можна сформувати наступні базові вимоги. [6]

1.4.1 Автентифікація та авторизація пристроїв

Однією з базових вимог до безпеки IoT-систем є забезпечення достовірності джерел даних і взаємодіючих пристроїв. Без надійної автентифікації система не здатна відрізнити легітимний вузол від зловмисно впровадженого, а без авторизації – контролювати обсяг дозволених дій для кожного суб’єкта.

У системах Інтернету речей автентифікація має особливе значення, оскільки пристрої часто працюють автономно, а їхня компрометація може залишитися непоміченою. Для забезпечення надійної автентифікації слід використовувати унікальні криптографічні ключі або цифрові сертифікати. Такі методи дозволяють не лише ідентифікувати пристрій, але й гарантувати, що він не був підмінений або скомпрометований у процесі розгортання.

Авторизація визначає набір дій, які дозволено виконувати автентифікованому об’єкту. Найпоширенішим підходом є розмежування доступу на основі ролей (RBAC), де кожному пристрою призначається певна роль і доступ до ресурсів надається відповідно до неї. Таким чином, навіть якщо один пристрій буде скомпрометовано, його можливості залишаться обмеженими згідно з наданими повноваженнями.

Жоден пристрій не повинен отримувати мережевий доступ або мати змогу ініціювати передачу даних, поки його ідентичність не буде підтверджена, а його права не будуть чітко визначені. Крім того, необхідно дотримуватися принципу мінімальних привілеїв, коли кожному пристрою надаються лише ті дозволи, які є критично необхідними для його функціонування.

1.4.2 Шифрування даних

Забезпечення конфіденційності та цілісності даних, що передаються між пристроями та інфраструктурою, має критичне значення. Протоколи, які використовуються для зв’язку, повинні бути доповнені транспортним шифруванням. Окрім цього, також варто забезпечувати шифрування даних на рівні сховища, особливо коли йдеться про чутливу інформацію – медичні показники, дані користувача тощо.

Важливо використовувати перевірені криптографічні алгоритми (наприклад, AES) для захисту даних. Однак, варто уникати самостійної реалізації криптографічних механізмів, оскільки це часто призводить до критичних помилок. Замість цього слід використовувати перевірені бібліотеки, оптимізовані для вбудованих систем.

1.4.3 Захищене завантаження та оновлення ПЗ

Оскільки пристрої IoT часто працюють без нагляду користувача, захищений процес завантаження та механізми безпечного оновлення ПЗ мають суттєве значення для підтримки їхньої надійності й захищеності. Уразливості у прошивці можуть надовго залишатися невиправленими, якщо не передбачено автоматизованого й захищеного механізму оновлення.

Процес захищеного завантаження гарантує цілісність та автентичність отриманого програмного забезпечення. Він базується на криптографічному підписанні коду: під час запуску вбудований механізм перевіряє цифровий підпис мікропрограми. Якщо перевірка не проходить, завантаження блокується. Це дозволяє запобігти запуску зміненого або шкідливого ПЗ, яке могло бути впроваджене через фізичний доступ або мережеву вразливість.

Безпечне оновлення ПЗ також має враховувати можливість віддаленого оновлення, але з попереднім підтвердженням цілісності файлу оновлення та перевіркою його походження. Бажаною є підтримка подвійного завантажувача, де

нова прошивка розміщується у другому розділі та активується лише після успішного проходження перевірок.

1.4.4 Безпечне управління життєвим циклом

Управління пристроєм протягом усього його життєвого циклу – від виробництва до зняття з експлуатації – має супроводжуватися належними заходами безпеки. Забезпечення безпеки на кожному з етапів є критично важливим для мінімізації загроз, збереження довіри до системи та запобігання несанкціонованому доступу після закінчення терміну служби.

На етапі **виробництва** важливо гарантувати, що в пристрій інтегруються лише перевірені компоненти й сертифіковане програмне забезпечення. Саме тут закладаються ключі автентифікації, сертифікати, механізми безпечного завантаження. Недостатній контроль на цьому етапі може призвести до появи закладок (backdoors) або несанкціонованого доступу з боку третіх сторін.

Під час **ініціалізації** та **розгортання** пристрою слід уникати використання облікових записів за замовчуванням, які часто залишаються незмінними і є легкою мішенню для атак. Процес має передбачати унікальну ідентифікацію кожного пристрою, генерацію індивідуальних криптографічних ключів та забезпечення налаштувань конфігурації, що відповідають політикам безпеки організації.

У фазі **експлуатації** пристрій повинен мати механізми безпечного оновлення, логування подій, захисту від змін налаштувань з боку неавторизованих користувачів. Важливо передбачити регулярний моніторинг поведінки пристрою для виявлення аномалій та ізоляцію скомпрометованих компонент.

Особливу увагу потрібно приділити етапу **виведення з експлуатації**. Пристрій, що більше не використовується, залишається джерелом ризику, якщо не видалено облікові дані, ключі або конфігураційні файли. Належне завершення життєвого циклу має включати: скидання до заводських налаштувань із гарантованим знищенням усіх даних; відкликання сертифікатів і токенів доступу;

анулювання авторизації у хмарних сервісах; контрольовану утилізацію пристрою або його фізичне знищення у випадку збереження чутливих даних.

1.4.5 Дотримання принципів GDPR і законодавства України

У випадках, коли IoT-система обробляє персональні дані громадян ЄС, вона має відповідати вимогам Загального регламенту захисту даних (GDPR). Це передбачає прозоре інформування користувачів про цілі збору інформації, отримання їх згоди, обмеження строків зберігання даних, а також забезпечення можливості реалізації прав суб'єкта даних. Зокрема, йдеться про право на доступ до своїх персональних даних, їх виправлення, стирання, обмеження обробки, перенесення та заперечення щодо певних операцій, включаючи профілювання [11].

Подібні вимоги висуваються і на національному рівні – відповідно до Закону України «Про захист персональних даних». Закон вимагає, щоб обробка персональних даних здійснювалася з дотриманням принципів законності, прозорості, обмеження мети та обсягу збирання. Особлива увага приділяється обов'язковості згоди суб'єкта персональних даних, забезпеченню прав на доступ до власної інформації, внесення змін або її видалення, а також відповідальності володільця персональних даних у разі порушення безпеки або розголошення інформації [12].

Технічна реалізація відповідності цим вимогам в IoT-системах передбачає впровадження механізмів ідентифікації та автентифікації користувачів, логування запитів, шифрування інформації, а також контролю доступу до неї. Крім того, у разі витоку персональних даних відповідно до статті 33 GDPR організація зобов'язана повідомити контролюючий орган не пізніше ніж через 72 години [11]. У межах українського законодавства також передбачено обов'язки володільця персональних даних щодо вжиття необхідних заходів для захисту інформації та інформування суб'єкта про його права [12].

Висновки до розділу 1

Зважаючи на інформацію викладену в розділі 1, можна сформувати кілька ключових висновків щодо природи, структури та безпеки Інтернету речей.

IoT – це стрімко зростаюча галузь, що поєднує фізичні об’єкти з цифровим середовищем через мережі передачі даних. IoT-технології широко впроваджуються у побут, промисловість, медицину, транспорт та критичну інфраструктуру, забезпечуючи автоматизацію, моніторинг і прийняття рішень у режимі реального часу.

Одним з ключових елементів сучасних IoT-архітектур є edge-шлюзи, які виконують роль проміжного шару між малопотужними пристроями та системами обробки. Вони дозволяють здійснювати попередню обробку, фільтрацію та захист даних, зменшуючи затримки, мережеве навантаження та залежність від хмари.

Разом із розвитком Інтернету речей суттєво зростає кількість загроз безпеці. IoT-системи є вразливими до несанкціонованого доступу, атак типу DoS, маніпуляцій даними, фізичного втручання та експлуатації слабких або відсутніх механізмів автентифікації. Для зменшення ризиків визначено низку обов’язкових вимог до безпеки систем Інтернету речей.

Таким чином, ефективне функціонування IoT-систем вимагає поєднання технічних засобів захисту, організаційних підходів і правового регулювання. Лише комплексне врахування вказаних аспектів та пошук механізму їх гармонійної взаємодії дозволяє гарантувати довіру, надійність і стійкість Інтернету речей у сучасному цифровому середовищі.

2 АНАЛІЗ ТЕХНОЛОГІЙ БЛОКЧЕЙНУ І ЇХ ПЕРЕВАГИ

2.1 Принцип роботи блокчейну

Блокчейн – це децентралізований розподілений реєстр, який дозволяє здійснювати безпечні та прозорі транзакції без посередників [13]. Він має декілька ключових відмінностей від традиційних сховищ для зберігання інформації. Його архітектура унеможливлює змінення даних і забезпечує довіру між сторонами без необхідності централізованого управління.

Усі учасники мережі зберігають копію реєстру. Через що блокчейн часто називають децентралізованою базою даних. Оскільки немає єдиної точки відмови, система стабільніша та менш схильна до збоїв. Для забезпечення безпеки та конфіденційності мережі використовується криптографія, зокрема гешування та асиметричні криптосистеми [13].

Процес гешування перетворює інформацію на унікальний криптографічний геш, який використовується для перевірки правильності даних. Будь-яка зміна вхідної інформації призводить до зміни гешу, що дозволяє мережі миттєво виявляти спроби модифікації [13].

Асиметрична криптографія, забезпечує безпеку транзакцій у блокчейні. Кожен користувач має два ключі: відкритий, який доступний усім, і закритий, який знає лише власник. Закритий ключ використовується для підпису транзакцій, а відкритий – для перевірки їхньої легітимності мережею [13].

Кожен блок у блокчейні повинен містити [14] (рисунок 2.1):

- Геш попереднього блоку (у початковому блоці він нульовий);
- Часову мітку;
- Набір транзакцій (даних);
- Геш поточного блоку.

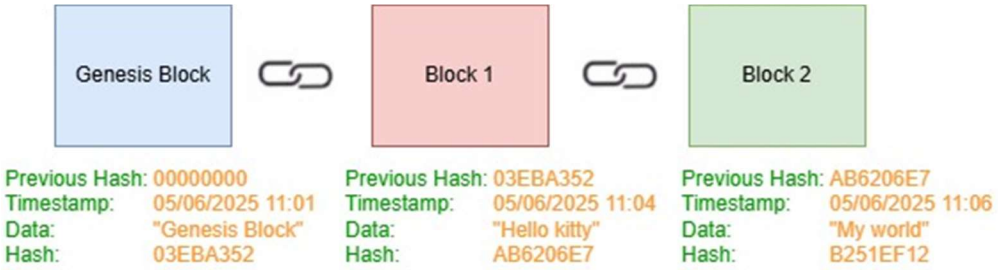

Рисунок 2.1 – Структура блоків

Така структура гарантує, що будь-яка зміна вмісту одного блоку порушить цілісність всього ланцюга, оскільки зміниться його геш, а отже – і геш наступного блоку, і так далі.

2.2 Протоколи консенсусу

У процесі роботи блокчейну ключову роль відіграє механізм досягнення консенсусу – це процес досягнення згоди щодо єдиного значення в розподіленому обчислювальному середовищі [15]. Іншими словами ці алгоритми визначають, як учасники мережі погоджуються щодо запису нових блоків. Від їх дизайну залежить як безпека системи, так і її продуктивність. Розглянемо деякі відомі методи [16].

2.2.1 Proof of Work

Proof of Work, PoW (доказ виконаної роботи) – один із перших і найпоширеніших алгоритмів консенсусу, вперше застосований у блокчейні Bitcoin. Його основна ідея полягає в тому, що вузли-учасники мережі (так звані майнери) повинні виконати складну обчислювальну задачу, щоб мати право додати новий блок до ланцюга.

Ці задачі – криптографічні головоломки – зазвичай формулюються як пошук такого значення nonce, яке після гешування (наприклад, з використанням SHA-256) утворює геш із певною кількістю початкових нулів як зазначено на рисунку 2.2. Цей

процес вимагає перебору великої кількості варіантів і не має оптимізації, окрім прямого перебору, що робить його ресурсномістким і часовитратним. Як тільки один із майнерів знаходить правильне рішення, інші вузли можуть легко перевірити його коректність, і блок додається до блокчейну.

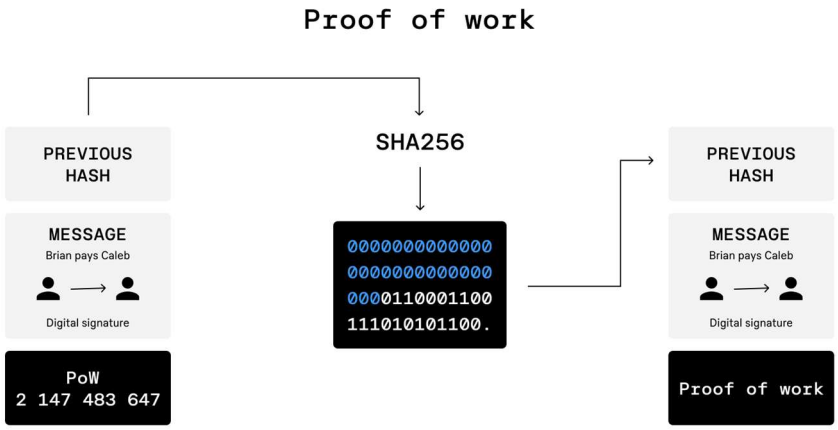

Рисунок 2.2 – Алгоритм PoW [17]

Алгоритм PoW забезпечує високий рівень безпеки, оскільки для успішного фальсифікування блокчейну зловмиснику довелося б перерахувати всі блоки з моменту змін – із тією ж складністю, яку мають усі чесні учасники разом. Таким чином, чим більше учасників працюють у мережі, тим безпечнішою вона стає.

Разом з тим, PoW має суттєві недоліки. Одним із головних є високе енергоспоживання, оскільки майнінг вимагає потужного обладнання та значних обсягів електроенергії. Ще одним важливим обмеженням є низька швидкість обробки транзакцій, через що нові блоки формуються повільно, наприклад, у Bitcoin – приблизно кожні 10 хвилин. Попри це, Proof of Work залишається основою багатьох публічних блокчейнів.

2.2.2 Proof of Stake

Proof of Stake, PoS (доказ частки володіння) – це консенсусний механізм, який був запропонований як альтернатива енерговитратному Proof of Work. Його головна

ідея полягає в тому, що вузли мережі (валідатори) отримують право додавати новий блок до блокчейну не через обчислювальну роботу, а залежно від того, яку частку криптовалюти вони утримують на своїх гаманцях.

У системі PoS кожен учасник, який володіє певною кількістю токенів, може «застейкати» їх, тобто заблокувати як гарантію своєї чесної поведінки. Чим більше активів задіяно, тим вища ймовірність того, що саме цей вузол буде обрано для створення наступного блоку. Такий підхід стимулює валідаторів до чесної роботи, адже у разі зловживань їх частка може бути зменшена або конфіскована.

Порівняно з PoW, алгоритм PoS має низку переваг. Він використовує значно менше енергії, оскільки не потребує інтенсивних обчислень. Також тут вища швидкість підтвердження транзакцій та відсутня потреба у дороговартісному обладнанні.

Проте PoS має і певні ризики. Головний із них – схильність до централізації: великі власники токенів мають більший шанс стати валідаторами, а отже, можуть поступово зосередити контроль над мережею. Це може призвести до створення «криптовалютної олігархії», що суперечить принципам децентралізації. Також існує загроза атаки «нічого не коштує» (nothing-at-stake), коли валідатори можуть підтримувати кілька конкуруючих гілок блокчейну без витрат, що потенційно підриває єдність реєстру.

Proof of Stake активно використовується в новітніх блокчейн-платформах, таких як Ethereum, Cardano тощо, і поступово витісняє Proof of Work як більш екологічно та економічно стійкий механізм досягнення консенсусу.

2.2.3 Delegated Proof of Stake

Delegated Proof of Stake, DPoS (делегований доказ частки володіння) – це вдосконалений варіант алгоритму Proof of Stake, який був розроблений для підвищення ефективності, швидкості транзакцій та масштабованості блокчейн-систем. Основна ідея DPoS полягає в тому, що власники токенів не беруть участі в

генерації блоків напряму, а голосують за делегатів, які отримують повноваження валідувати транзакції та формувати нові блоки від імені спільноти.

Голосування, як правило, є безперервним і відкритим: учасники можуть у будь-який момент переобрати делегата, якщо його дії не відповідають очікуванням або порушують довіру. Такий підхід створює систему репутаційного стимулювання, де делегати повинні підтримувати високу якість своєї роботи, аби залишатися обраними.

DPoS забезпечує високу продуктивність – у типових реалізаціях лише невелика фіксована кількість делегатів беруть участь у процесі підтвердження транзакцій, що значно знижує час на досягнення консенсусу. Як результат, такі блокчейни можуть досягати сотень або тисяч транзакцій на секунду, на відміну від класичного PoW чи PoS.

Однак із підвищенням ефективності виникає доволі високий ризик централізації – якщо більшість учасників не беруть активної участі в голосуванні, то лише невелика група делегатів фактично контролює мережу. У таких умовах зловживання повноваженнями або змова між делегатами може призвести до порушення принципу децентралізації.

2.2.4 Practical Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance, PBFT (практична візантійська відмовостійкість) – це консенсусний алгоритм, призначений для забезпечення узгодженості розподіленої системи навіть у разі, якщо частина її учасників поводиться некоректно або ворожо. PBFT є одним із найвідоміших рішень класичної проблеми «візантійських генералів» (рисунок 2.3) і передбачає стійкість до 1/3 недоброчесних вузлів, що робить його придатним для середовищ із підвищеними вимогами до довіри.

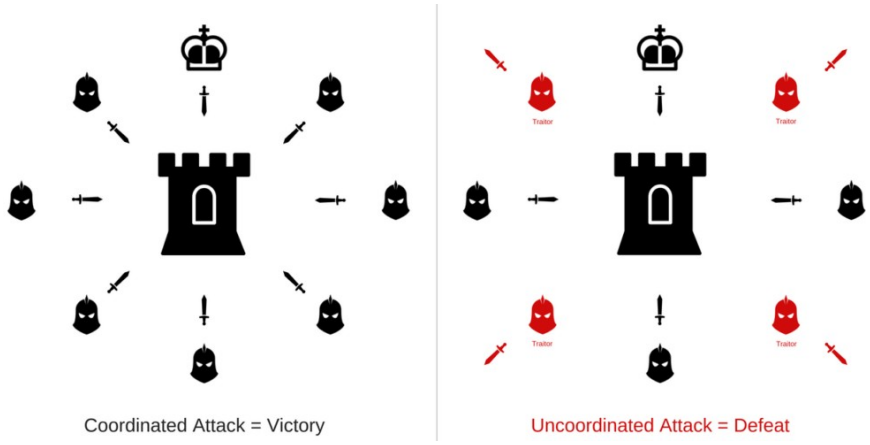

Рисунок 2.3 – Проблема візантійських генералів [18]

На відміну від PoW чи PoS, PBFT не використовує складних обчислень або фінансових ставок. Замість цього, досягнення консенсусу відбувається шляхом багатокрокового голосування між вузлами мережі. Алгоритм працює в умовах попередньо відомої кількості учасників і вимагає, щоб щонайменше дві третини з них погодилися з запропонованим блоком або транзакцією, перш ніж вона буде остаточно прийнята.

Протокол складається з трьох основних фаз:

1 **Pre-prepare** – первинна пропозиція блоку.

2 **Prepare** – розсилка цієї пропозиції до інших вузлів і збір підтверджень.

3 **Commit** – остаточне узгодження та фіксація блоку після досягнення необхідної кількості голосів.

Завдяки такій схемі PBFT забезпечує швидке та передбачуване підтвердження транзакцій, що не залежить від обчислювальної потужності або вартості токенів. Це робить його ефективним і енергозберігаючим варіантом консенсусу, особливо в середовищах із обмеженою кількістю вузлів.

Проте головний недолік PBFT – погана масштабованість. Оскільки кожен вузол повинен комунікувати з кожним іншим на кожному етапі, кількість повідомлень сильно зростає. Це робить протокол малоефективним у великих мережах з великою кількістю вузлів. Тому PBFT найкраще працює у невеликих або консорціумних системах, де кількість учасників є обмеженою і відомою наперед.

2.2.5 Proof of Authority

Proof of Authority, PoA (доказ повноважень) – це консенсусний механізм, у якому право підтвердження транзакцій і створення нових блоків належить обмеженій групі заздалегідь авторизованих вузлів. Учасники цієї групи, так звані валідаційні вузли, обираються на основі свого «авторитету» - тобто репутації, довіри, ідентифікації або юридичного статусу в межах мережі.

На відміну від Proof of Work чи Proof of Stake, де консенсус залежить від обчислювальних ресурсів або обсягу активів, у PoA достатньо мати підтверджену особу або статус, щоб брати участь у валідації. Завдяки цьому PoA забезпечує високу швидкість обробки транзакцій і низьке енергоспоживання, що робить його надзвичайно привабливим для корпоративних, приватних або регульованих блокчейн-систем.

Алгоритм PoA має певні обмеження. Основна критика цього підходу полягає у зниженні рівня децентралізації – оскільки обмежене коло учасників фактично контролює блокчейн, система втрачає частину властивостей публічного довірчого середовища. Крім того, довіра до валідаторів є ключовим фактором, і компрометація хоча б одного з них може призвести до спотворення даних або нав’язування небажаних змін.

2.2.6 Proof of Elapsed Time

Proof of Elapsed Time, PoET (доказ витраченого часу) – це консенсусний алгоритм, розроблений компанією Intel, який ґрунтується на випадковому відборі вузлів для створення блоків за допомогою таймерів, що генеруються у захищеному середовищі. Ідея полягає в тому, що кожен учасник мережі створює таймер випадкової тривалості, і вузол, у якого таймер завершиться першим, отримує право згенерувати наступний блок.

Головною технічною вимогою для PoET є використання довіреного виконуваного середовища (Trusted Execution Environment). Усі таймери

формуються всередині TEE, що гарантує чесність, випадковість і неможливість підробки. Завдяки цьому забезпечується справедливий розподіл шансів між усіма учасниками, без потреби в енерговитратних обчисленнях чи великих фінансових ставках.

Разом з тим, PoET має і низку обмежень. Найголовніше з них – потреба в довірі до апаратної платформи, зокрема до виробника TEE. Якщо TEE буде скомпрометоване або штучно змінене, система втратить надійність. Крім того, алгоритм погано масштабований до відкритих середовищ, оскільки неможливо перевірити доброчесність кожного нового вузла без централізованої сертифікації його обладнання.

У таблиці 2.1 наведено порівняння описаних протоколів консенсусу.

Таблиця 2.1 – Порівняння протоколів консенсусу [16]

Консенсус	**Доступність**	**Децентралізація**	**Масштабованість**	**Пропуска здатність**	**Затримка**	**Обчислювальне навантаження**	**Мережеве навантаження**	**Витрати на сховище**
PoW	Публі-чний	Висока	Висока	Низька	Висока	Високе	Низьке	Високі
PoS	Публі-чний	Висока	Висока	Низька	Середня	Низьке	Низьке	Високі
DPoS	Публі-чний	Середня	Висока	Висока	Середня	Середнє	Неві-домо	Високі
PBFT	Прива-тний	Середня	Низька	Висока	Низька	Низьке	Високе	Високі
PoA	Публі-чний	Середня	Висока	Висока	Середня	Середнє	Низьке	Високі

Кінець таблиці 2.1

PoET	Прива-тний	Середня	Висока	Висока	Низька	Низьке	Низьке	Високі

2.3 Типи блокчейну та особливості їх застосування

Блокчейн-системи поділяються на кілька основних типів залежно від рівня доступу до мережі, правил управління та участі у валідації транзакцій. Найпоширенішими є публічні, приватні та консорціумні блокчейни (рисунок 2.4). Кожен із цих типів має свої переваги та недоліки, що визначають доцільність їх застосування в різних контекстах.

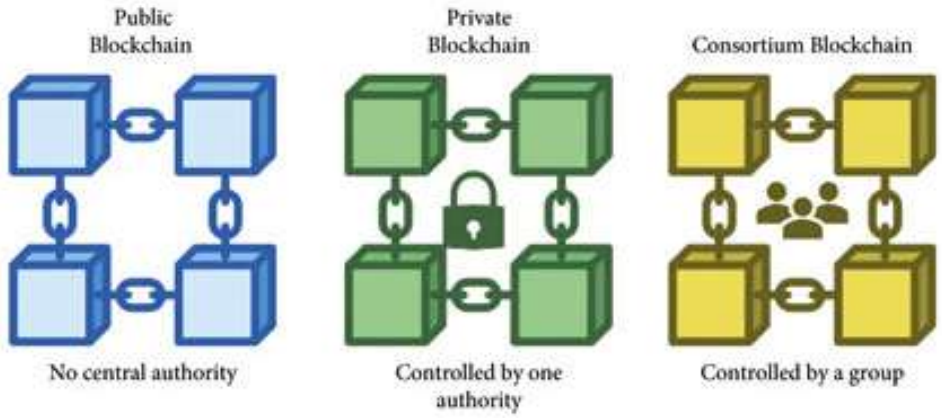

Рисунок 2.4 – Типи блокчейну [19]

2.3.1 Публічні блокчейни

Публічний блокчейн є найпопулярнішою та найвідкритішою формою розподіленого реєстру, в якій будь-хто може стати учасником мережі, створювати транзакції, перевіряти їх або навіть брати участь у валідації блоків. Такі системи не вимагають попередньої авторизації для приєднання, що робить їх повністю

децентралізованими. Типовими прикладами публічних блокчейнів є Bitcoin, Ethereum та інші криптовалютні мережі [13].

Даний тип блокчейну забезпечує максимально можливу прозорість і довіру за рахунок повного відкриття журналу транзакцій усім користувачам мережі. Це дозволяє будь-якому учаснику незалежно перевірити автентичність записів і механізмів обробки даних. Такий підхід усуває потребу в централізованих посередниках, адже довіра забезпечується не авторитетом організації, а криптографією та механізмом узгодження блоків [13].

Публічні блокчейни характеризуються високим рівнем децентралізації, оскільки контроль над мережею розподілений між великою кількістю незалежних вузлів. Завдяки цьому досягається висока стійкість до цензури та маніпуляцій. Проте така децентралізація має і зворотний бік – вона часто супроводжується низькою швидкістю транзакцій і високими витратами на енергоспоживання.

Відкритість публічних блокчейнів також створює специфічні вектори атак. До найпоширеніших загроз належать:

- **Атака подвійного витрачання (double-spending)** – коли одна й та сама криптовалюта витрачається більше одного разу до підтвердження транзакцій.
- **Атака 51%** – коли зловмисник або група отримує контроль над більшістю обчислювальної потужності мережі, що дає змогу контролювати додавання нових блоків.
- **Sybil-атаки, eclipse-атаки, фішинг і соціальна інженерія** – всі вони пов’язані з відкритістю мережі, через що сторонні особи можуть маніпулювати інформаційними потоками.

Оскільки публічні блокчейни відкриті для всіх і не мають централізованого управління, забезпечення безпеки залежить виключно від досконалості криптографічних протоколів і механізмів узгодження блоків, а також активного моніторингу спільнотою. Саме тому ці системи, хоч і повністю прозорі, потребують складних технічних рішень для збереження цілісності й надійності у відкритому середовищі.

2.3.2 Приватні блокчейни

Приватний блокчейн – це тип розподіленого реєстру, доступ до якого обмежений визначеним колом учасників [13]. На відміну від публічних блокчейнів, які є відкритими для будь-кого, приватні вимагають попереднього схвалення чи запрошення від адміністратора мережі або керівного консорціуму. Це забезпечує суворий контроль над доступом, що особливо цінується в корпоративному середовищі.

Учасники приватного блокчейну заздалегідь визначені й перевірені, що дозволяє забезпечити більш надійний контроль над даними та взаємодією всередині системи. Такий контроль відкриває можливості для гнучкого керування ролями, обмеженням доступу до транзакцій або до окремих частин даних.

Приватні блокчейни вирізняються вищою продуктивністю порівняно з публічними. Завдяки меншій кількості вузлів та ефективнішим механізмам консенсусу, вони забезпечують швидшу обробку транзакцій і менше енергоспоживання [13]. Такі алгоритми досягають узгодженості між вузлами за рахунок попередньо визначеного кворуму, що знижує обчислювальне навантаження та латентність системи.

Ще однією ключовою перевагою є підвищена конфіденційність – дані в приватних блокчейнах доступні лише авторизованим учасникам [13], що робить їх придатними для застосування в системах, які потребують суворого захисту інформації.

Попри свої переваги, приватні блокчейни мають і певні обмеження. Оскільки всі учасники мають бути заздалегідь відомими й схваленими, це обмежує рівень прозорості. У таких мережах менше децентралізації, адже централізоване управління або довірені вузли відіграють вирішальну роль у функціонуванні системи.

Централізація управління породжує ще одну проблему – зменшення довіри до системи [13], оскільки користувачам потрібно покладатися на адміністратора

блокчейну. Такі системи не завжди відповідають ідеології децентралізації, хоча й забезпечують високу ефективність та сумісність із корпоративними стандартами.

2.3.3 Консорціумні блокчейни

Консорціумні блокчейни поєднують риси як публічних, так і приватних блокчейнів, створюючи гібридну модель. У такій мережі група довірених організацій – учасників консорціуму – об'єднується для спільного керування мережею, обміну даними та перевірки транзакцій [13]. На відміну від повністю відкритих або централізованих рішень, консорціумні блокчейни дозволяють досягнути балансу між прозорістю, ефективністю та контролем доступу.

Цей тип блокчейну забезпечує баланс між децентралізацією та контрольованістю. Він добре підходить для систем, де важливо мати ефективну синхронізацію даних між кількома учасниками, але при цьому контролювати, хто саме бере участь у прийнятті рішень. Така модель добре масштабується, дозволяє централізовано оновлювати політики безпеки й підтримує алгоритми консенсусу з низьким навантаженням, наприклад, PBFT.

Така архітектура має певні недоліки, зокрема обмежена кількість вузлів створює ризик змов між учасниками консорціуму, що може впливати на прийняття рішень. Також у випадку конфлікту інтересів, між організаціями можуть виникати труднощі при досягненні консенсусу, що ускладнює ефективне функціонування системи.

Так як консорціумний блокчейн це поєднання двох інших типів, то і певні загрози також успадковуються. А саме:

- **Sybil-атаки** – створення фальшивих вузлів для маніпулювання голосуванням.
- **Eclipse-атаки** – ізоляція вузлів та підміна інформації.
- **Маніпуляція консенсусом** – контроль великої частини вузлів з метою впливу на підтвердження транзакцій.

- **Внутрішні загрози** – неналежне використання привілеїв співробітниками або учасниками.
- **Фішинг та соціальна інженерія** – атаки через людський фактор, що може призвести до компрометації ключових елементів системи.

Консорціумні блокчейни є ефективним компромісом між безпекою, продуктивністю та децентралізацією, дозволяючи організаціям спільно працювати над перевіркою транзакцій і керуванням системою без надмірної залежності від централізованих авторитетів.

У таблиці 2.2 наведено порівняння описаних типів блокчейнів.

Таблиця 2.2 – Порівняння типів блокчейнів [20]

	Тип блокчейну		
	Публічний	**Приватний**	**Консорціумний**
Інклюзивність	Так	Ні	Ні
Читання	Будь-хто	Відомі користувачі	В залежності від ситуації
Запис	Будь-хто	Затверджені учасники	Затверджені учасники
Право власності	Ніхто	Одна організація	Декілька організацій
Ідентифікація учасників	Ні	Так	Так
Швидкість транзакцій	Повільно	Швидко	Швидко

2.4 Смарт-контракти

Смарт-контракти – це самовиконувані програми з попередньо визначеними умовами [21]. У технічному сенсі смарт-контракт – це код, що зберігається на блокчейні, і виконується розподіленими вузлами в момент ініціації відповідною

транзакцією. Його основна функція полягає в тому, щоб забезпечити надійне, автономне та незмінне виконання угод без потреби в посередниках чи довірених третіх сторонах.

Основні властивості смарт-контрактів:

- **Автоматизація**: виконуються безпосередньо при настанні заданих умов, без ручного втручання.
- **Незмінність**: після розміщення в блокчейні змінити код або логіку неможливо.
- **Детермінованість**: результат виконання є однаковим для всіх учасників.
- **Прозорість**: усі користувачі можуть переглядати код та історію виконання.
- **Самодостатність**: можуть ініціювати транзакції, обробляти дані, викликати інші контракти.

Проте разом із перевагами смарт-контракти мають і певні ризики. Основні загрози включають [13]:

- **Вразливості в коді**: помилки в логіці, можуть призвести до порушення безпеки мережі.
- **Проблеми з верифікацією**: труднощі в доведенні коректності та безпечності складних контрактів.
- **Ризик зловживання**: якщо контракт надає надмірні права або має недоцільно широкі умови виконання, він може бути використаний зловмисниками.

Через це при впроваджені смарт-контрактів важливими є аудит безпеки, формальна верифікація, обмеження доступу та використання перевірених бібліотек.

2.5 Переваги блокчейну для забезпечення безпеки

Технологія блокчейн надає низку важливих переваг у контексті забезпечення безпеки інформації та побудови надійних розподілених систем. Її застосування дозволяє кардинально змінити традиційні підходи до зберігання, обробки та перевірки даних, усуваючи потребу у довірених посередниках і мінімізуючи ризики централізованих атак.

2.5.1 Незмінність даних

Однією з ключових переваг блокчейн-технології є незмінність записів, що означає неможливість редагування або видалення даних після їх додавання до ланцюга блоків. Кожен блок містить геш попереднього, а також власний набір транзакцій. У разі будь-якої зміни хоча б одного байта інформації в будь-якому блоці буде змінено його геш, що автоматично порушить цілісність усієї наступної частини ланцюга. Таким чином, спроба змінити дані не залишиться непоміченою й потребуватиме перебудови всієї структури блокчейну.

Це створює високий рівень захисту від несанкціонованих змін, оскільки будь-яка модифікація вимагатиме або згоди більшості учасників мережі, або неможливої в реальних умовах обчислювальної потужності для переписування всієї історії. Така властивість забезпечує цілісність даних, зберігає історичну достовірність транзакцій та дозволяє використовувати блокчейн як надійне джерело доказів у критичних сферах.

2.5.2 Децентралізація

Децентралізація є фундаментальною властивістю блокчейну, яка усуває потребу в центральному органі управління. У традиційних системах дані зазвичай зберігаються на центральному сервері або контролюються однією організацією, що

створює єдину точку відмови. Якщо такий центр буде скомпрометовано або атаковано, уся система може зазнати краху.

У блокчейн-мережі кожен вузол зберігає повну або часткову копію реєстру, що забезпечує не лише доступність, а й стійкість до зовнішніх атак та внутрішніх зловживань. Ніхто не має виняткової влади над мережею: всі рішення приймаються колективно на основі протоколу консенсусу. Завдяки цьому навіть у разі виходу з ладу частини вузлів мережа продовжує функціонувати, а спроби підміни даних стають значно складнішими. Децентралізація підвищує довіру до системи, оскільки гарантії надійності закладені в архітектурі самої технології, а не в окремому суб'єкті.

2.5.3 Аудит і простежуваність

Блокчейн забезпечує високий рівень аудиту та простежуваності завдяки структурі, в якій кожна транзакція містить унікальний ідентифікатор, часову мітку та цифровий підпис джерела. Ці параметри фіксуються незмінно в реєстрі, що дозволяє точно відстежити, хто, коли і яку дію здійснив у системі. Такий механізм унеможливлює несанкціоновану зміну історії подій без залишення цифрового сліду.

Це особливо важливо в системах із підвищеними вимогами до прозорості та контролю, таких як критична інфраструктура, фінансові платформи або державні реєстри. Блокчейн дозволяє будувати доказову базу змін, легко виявляти аномалії або спроби втручання, а також проводити аналіз інцидентів без потреби в зовнішніх журналах. Простежуваність у блокчейні – це не просто механізм журналювання, а основа для формування довіри, відповідальності та відповідності нормативним вимогам.

2.5.4 Криптографічний захист

Блокчейн широко застосовує сучасні криптографічні механізми, які забезпечують автентичність, цілісність та незаперечність усіх транзакцій. Одним із ключових компонентів є цифрові підписи, що створюються за допомогою приватного ключа власника даних. Тільки той, хто володіє відповідним закритим ключем, може підписати транзакцію, а її справжність легко перевіряється за допомогою відкритого ключа. Це виключає можливість підробки авторства або змін без виявлення.

Ще однією основою безпеки є геш-функції, які перетворюють вхідні дані у фіксованого розміру унікальний код. Будь-яка зміна навіть одного біта у транзакції призводить до повністю іншого гешу, що дозволяє миттєво виявити спробу маніпуляції. Крім того, асиметричне шифрування дозволяє захищати самі дані, забезпечуючи конфіденційність обміну між сторонами. Завдяки цим криптографічним засобам блокчейн гарантує, що лише уповноважені учасники можуть здійснювати транзакції, а записана інформація залишатиметься цілісною та достовірною впродовж усього часу існування системи.

2.5.5 Контроль доступу

У приватних та консорціумних блокчейнах реалізовано гнучкі механізми контролю доступу, які дозволяють точно визначати, хто має право переглядати, змінювати або підтверджувати інформацію. На відміну від публічних мереж, де всі дані відкриті, у таких системах можна розмежовувати повноваження між учасниками залежно від їхньої ролі, рівня довіри чи організаційної приналежності.

Завдяки використанню політик доступу у смарт-контрактах, система забезпечує мінімізацію прав – кожен вузол чи користувач виконує лише ті дії, які прямо дозволено. Це дозволяє обмежити коло учасників, що мають доступ до критичних даних, і зменшує ризик зловживань або витоків. Контроль доступу також є важливою умовою відповідності нормативним вимогам.

2.5.6 Інтеграція смарт-контрактів

Інтеграція смарт-контрактів у блокчейн-систему дозволяє автоматизувати взаємодії між вузлами на основі заздалегідь визначених правил. Такі контракти виконуються лише за умови дотримання чітко встановлених логічних умов, що виключає необхідність людського втручання у процес прийняття рішень. Завдяки цьому зменшується ризик людських помилок, неправомірних дій або упереджених рішень, а також підвищується надійність системи.

Автоматизація через смарт-контракти дозволяє забезпечити стандартизовану, прозору та прогнозовану поведінку мережі. Наприклад, такі контракти можуть використовуватись для управління доступом, активації пристроїв, обробки транзакцій або виконання бізнес-логіки в умовах децентралізованої взаємодії. Усе це робить смарт-контракти потужним інструментом для створення самодостатніх безпечних екосистем, у яких довіра базується не на людях, а на коді. [21]

Висновки до розділу 2

Зважаючи на інформацію, викладену в розділі 2, можна сформулювати низку ключових висновків щодо принципів роботи блокчейну, його типів, консенсусних механізмів та ролі у забезпеченні безпеки.

Блокчейн – це розподілена цифрова структура зберігання інформації, що забезпечує незмінність, прозорість та довіру між учасниками без потреби у централізованому посереднику. Його функціонування ґрунтується на зв’язку між блоками, де кожен новий блок залежить від попереднього за допомогою криптографічних геш-функцій. Це робить підробку чи модифікацію даних практично неможливою без згоди більшості мережі.

Існують різні типи блокчейнів: публічні, приватні та консорціумні. Кожен з них має свої особливості застосування. Публічні – відкриті для будь-кого, але менш масштабовані; приватні – орієнтовані на внутрішні процеси організацій з високим

рівнем контролю; консорціумні – комбінують переваги обох, забезпечуючи баланс між децентралізацією та керованістю.

Значну роль у роботі блокчейн-систем відіграють протоколи консенсусу: від класичних Proof of Work та Proof of Stake до спеціалізованих механізмів, як-от PBFT, DPoS або PoET. Кожен з них має свої переваги і обмеження залежно від вимог до безпеки, швидкодії та довіри між учасниками.

Окреме значення мають смарт-контракти, які дозволяють автоматизувати взаємодію між учасниками мережі. Вони виконуються виключно за заздалегідь визначених умов, забезпечуючи передбачуваність, прозорість і мінімізацію людського фактора.

У підсумку, технологія блокчейн надає потужні інструменти для побудови безпечних і довірених інформаційних систем. Завдяки незмінності даних, децентралізації, вбудованій криптографії, можливості контролю доступу та автоматизації через смарт-контракти, блокчейн створює нову парадигму інформаційної безпеки, що є надзвичайно актуальною в умовах цифровізації й зростання кіберзагроз.

3 ОСОБЛИВОСТІ ВИКОРИСТАННЯ БЛОКЧЕЙНУ В ІОТ

3.1 Забезпечення децентралізованого управління даними Інтернету речей

Інтеграція блокчейн-технологій у системи Інтернету речей відкриває нові можливості для безпечного, децентралізованого та прозорого управління даними. У традиційних IoT-архітектурах дані, зібрані сенсорами та пристроями, зазвичай передаються до централізованих хмарних платформ для подальшої обробки та зберігання. Через це можливі певні ризики: залежність від єдиної точки доступу, ризик втрати даних через відмову сервера, можливість зловживання доступом або модифікації даних зі сторони адміністратора системи.

Блокчейн дозволяє подолати ці обмеження, забезпечивши децентралізовану інфраструктуру зберігання та обміну даними, у якій кожен вузол мережі володіє копією розподіленого реєстру. Завдяки цьому IoT-пристрої можуть реєструвати свої дані безпосередньо в блокчейні, де кожен запис має криптографічну перевірку достовірності, часову мітку та унікальний цифровий підпис. У результаті зникає потреба в центральному посереднику, що суттєво знижує ризики компрометації.

Кожен вузол або пристрій може взаємодіяти з іншими безпосередньо через розподілений реєстр, що значно підвищує стійкість і надійність системи. Дані, що зберігаються у блокчейні, є незмінними та прозорими, що робить їх придатними для подальшої перевірки, аудиту або машинного аналізу. Крім того, у блокчейн-мережах дані можна шифрувати, забезпечуючи конфіденційність навіть у публічних середовищах.

Завдяки використанню блокчейну, управління даними в IoT стає децентралізованим, стійким до відмов і маніпуляцій, прозорим і контрольованим. Це створює передумови для побудови довірених IoT-систем, які не залежать від жодного єдиного оператора або провайдера сервісів, і можуть ефективно працювати навіть у гетерогенних або масштабованих середовищах. [21]

3.2 Роль смарт-контрактів в управлінні IoT пристроями

Смарт-контракти, як самовиконувані програми, записані у блокчейн, відіграють ключову роль в автоматизації управління пристроями IoT. Завдяки своїй децентралізованій природі та незмінності, вони забезпечують надійне виконання наперед визначених дій, коли виконуються відповідні умови. Це дозволяє системам Інтернету речей автономно виконувати операції, такі як управління доступом, обробка транзакцій або регулювання параметрів роботи, без необхідності централізованого контролю.

Кожен такий тригер або дія записується в блокчейн, що формує надійний і незмінний журнал подій. Це забезпечує повну простежуваність, дає змогу перевіряти правильність виконання дій, знижує ризик фальсифікацій та спрощує аудит. Смарт-контракти також можуть реалізовувати політики доступу, контроль прав взаємодії між пристроями та реакцію на аномальні умови, що є особливо цінним у системах із високими вимогами до безпеки та автономності. [21]

Наприклад, смарт-контракт може бути налаштований так, щоб реагувати на вхідні дані від сенсорів. Якщо сенсор якості повітря фіксує перевищення допустимого рівня забруднення, контракт автоматично активує інші пристрої в системі – такі як системи очищення повітря чи вентиляційні установки. Схема такої мережі показана на рисунку 3.1. Уся ця логіка виконується автоматично, без необхідності в централізованому сервері, диспетчері чи API.

Таким чином, смарт-контракти у блокчейні перетворюють IoT-інфраструктуру на самокеровану систему, у якій пристрої можуть не лише збирати та передавати дані, а й ініціювати дії, приймати рішення та координувати взаємодію між собою без потреби в централізованій координації. Це значно підвищує ефективність, надійність і масштабованість розподілених IoT-архітектур.

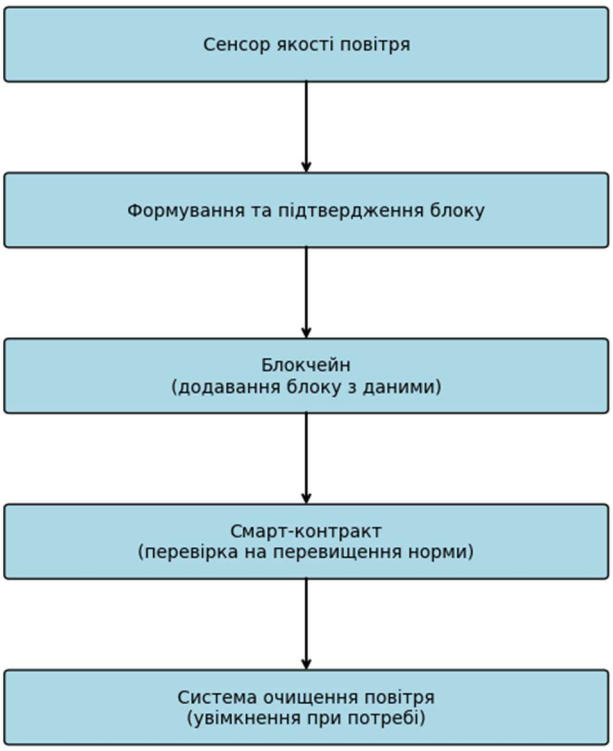

Рисунок 3.1 – Схема IoT-системи зі смарт-контрактом

3.3 Приклади інтеграції блокчейну в IoT

Одним із найяскравіших прикладів інтеграції блокчейн-технологій в Інтернет речей є IoTeX – приватна та масштабована платформа, спеціально розроблена для забезпечення безпеки, конфіденційності та децентралізованого управління IoT-даними. Запущена у 2017 році, ця система орієнтована на побудову екосистеми «Internet of Trusted Things», у якій люди та пристрої можуть безпечно взаємодіяти, залишаючись власниками своїх даних [22].

IoTeX використовує Roll-DPoS – модифіковану версію консенсусу Delegated Proof of Stake, що доповнена механізмами PBFT та Verifiable Random Functions (VRF). Це забезпечує високу швидкодію, масштабованість і децентралізацію, навіть в умовах роботи з обмеженими обчислювальними ресурсами. Для підвищення ефективності та гнучкості IoTeX також впроваджує сайдчейни, кожна з яких

обслуговує окремі пристрої або додатки, що дозволяє розвантажити основний блокчейн та забезпечити сумісність з різноманітними IoT-рішеннями [22].

Окрім архітектурної гнучкості, IoTeX робить акцент на конфіденційності користувачів. Наприклад, для захисту транзакцій використовуються кільцеві підписи, а кожен пристрій у мережі має децентралізований ідентифікатор. Всі транзакції здійснюються через токен IOTX, який використовується для реєстрації пристроїв, оплати послуг, участі в голосуванні та розподілу стимулів між учасниками мережі. Сама мережа підтримує взаємодію між людьми, машинами та децентралізованими додатками, створюючи безпечний і керований обмін даними [22].

Інші приклади:

- **Helium** – децентралізована мережа для пристроїв IoT, яка використовує блокчейн для стимулювання людей розгортати вузли, які забезпечують покриття мережі. Дані, що передаються через Helium, записуються у блокчейн, що гарантує їхню прозорість та захист від маніпуляцій.
- **Grid+** – використовує блокчейн для оптимізації енергетичних систем, відстеження та управління відновлюваними джерелами енергії. Дозволяє споживачам купувати та продавати електроенергію без посередників.
- **VeChain** – впроваджує блокчейн у логістичні IoT-системи для гарантування автентичності та забезпечення прозорості ланцюгів постачання. [23]

Дані приклади демонструють, як децентралізована блокчейн-інфраструктура може стати основою безпечного, масштабованого та автономного управління даними в IoT-середовищі, з урахуванням обмежених ресурсів пристроїв, вимог до конфіденційності та потреб у гнучкій інтеграції. Різноманітність цих прикладів свідчить про широкий потенціал блокчейну як базової технології для розвитку надійної та децентралізованої інфраструктури Інтернету речей.

3.4 Потенційні виклики та обмеження

Інтеграція блокчейн-технологій у системи Інтернету речей, попри значні переваги, пов’язана з низкою технічних викликів і обмежень, які необхідно враховувати під час проєктування реальних рішень.

3.4.1 Обмежені ресурси пристроїв

Більшість IoT-пристроїв проєктуються як малопотужні енергоефективні модулі, які мають обмеження щодо обчислювальних можливостей, пам’яті та енергоспоживання. У той час як традиційні блокчейн-системи, зокрема публічні, потребують складних криптографічних операцій, перевірки цифрових підписів, обробки транзакцій і збереження копій блоків, IoT-пристрої фізично не здатні ефективно виконувати ці задачі.

Окрім цього, тривала робота пристрою в умовах частого підключення до мережі та обміну блокчейн-даними збільшує витрати енергії та скорочує термін автономної експлуатації. Також стандартна реалізація блокчейну передбачає зберігання хоча б частини історії транзакцій, що є неприйнятним для пристроїв із обмеженим обсягом постійної пам’яті.

3.4.2 Масштабованість

Одним із головних викликів для інтеграції блокчейну в IoT є здатність обробляти велику кількість транзакцій, яка стрімко зростає зі збільшенням кількості підключених пристроїв. У великих IoT-системах тисячі сенсорів можуть генерувати події щосекунди, що створює величезне навантаження на мережу та блокчейн-інфраструктуру.

Традиційні блокчейни не здатні ефективно обробляти потік даних у реальному часі через обмежену пропускну здатність та повільні механізми валідації

транзакцій. Зокрема, спільна валідація кожного блоку усіма вузлами мережі створює значне навантаження і уповільнює роботу системи. Щоб подолати ці обмеження, необхідне використання адаптивних консенсус-алгоритмів та інфраструктури з високою пропускною здатністю.

3.4.3 Затримки

У багатьох випадках системи Інтернету речей вимагають обробки інформації та прийняття рішень у режимі реального часу. Пристрої, що реагують на зміни навколишнього середовища – наприклад, датчики температури, димові сенсори або системи безпеки – не можуть дозволити собі очікування кілька секунд або хвилин до підтвердження транзакції в блокчейні. Навіть короткі затримки в комунікації або обробці можуть призвести до небажаних наслідків, зниження ефективності або навіть загроз безпеці.

Блокчейн-системи, особливо публічні, часто мають порівняно високий час підтвердження транзакцій, зумовлений природою консенсусу, який вимагає узгодження між великою кількістю вузлів. Цю проблему частково можна вирішити за допомогою локального попереднього аналізу за допомогою edge-шлюзів.

3.4.4 Інтероперабельність

Інфраструктура Інтернету речей є надзвичайно гетерогенною – IoT-пристрої можуть використовувати різні протоколи зв’язку, формати передачі даних і моделі адресації. Це ускладнює їх безпосередню інтеграцію в єдину блокчейн-систему, яка зазвичай покладається на уніфіковані формати та інтерфейси.

Забезпечення взаємодії таких пристроїв через блокчейн-мережі можливе через створення проміжного програмного забезпечення, уніфікованих API, а також гнучких шлюзів, які здатні адаптувати дані з одного середовища до іншого. Крім того, необхідна стандартизація механізмів ідентифікації, обміну повідомленнями та управління доступом. Без урахування цих аспектів міжпротокольна взаємодія

залишатиметься фрагментованою, а повноцінне децентралізоване управління IoT-екосистемами через блокчейн – обмеженим. [21]

Висновки до розділу 3

На основі матеріалу, викладеного в розділі 3, можна сформулювати ключові висновки щодо можливостей і особливостей використання блокчейну для децентралізованого управління даними в Інтернеті речей.

Інтеграція блокчейн-технологій у IoT-системи дозволяє відійти від централізованих моделей обробки даних і створити стійку до зловживань та збоїв архітектуру. Завдяки розподіленому реєстру, IoT-пристрої можуть взаємодіяти безпосередньо один з одним, уникаючи залежності від єдиної точки контролю, що підвищує надійність, довіру та безперервність роботи всієї системи.

Особливу роль у цьому процесі відіграють смарт-контракти, які забезпечують автоматизовану та незалежну взаємодію між пристроями. Вони дозволяють не лише фіксувати події, але й приймати рішення у відповідь на зовнішні умови, що робить IoT-системи більш гнучкими та самокерованими.

Разом з тим, впровадження блокчейну в IoT пов’язане з низкою викликів, зокрема обмеженими ресурсами пристроїв, проблемами масштабованості, затримками та труднощами інтероперабельності. Ці обмеження вимагають ретельного технічного опрацювання архітектурних рішень та адаптації протоколів взаємодії.

Таким чином, поєднання блокчейну з Інтернетом речей має великий потенціал для формування надійних, автономних і безпечних систем, однак вимагає глибокого розуміння технічних бар’єрів і методів їх подолання.

4 РОЗРОБКА І РЕАЛІЗАЦІЯ МОДЕЛІ ІНТЕГРАЦІЇ БЛОКЧЕЙНУ В ІОТ

4.1 Концепція моделі

Основною метою цієї роботи є побудова безпечної децентралізованої моделі управління даними в середовищі Інтернету речей. З урахуванням наявних загроз, обмежень пристроїв та вимог до довіри між вузлами, запропоновано архітектуру, що поєднує переваги блокчейн-технології з децентралізованим контролем доступу та узгодженням дій між пристроями.

Для побудови моделі було обрано *консорціумний блокчейн*. Такий підхід передбачає, що до мережі входить обмежена кількість довірених вузлів – наприклад, контролери, шлюзи або сервери в рамках однієї організації та партнерської екосистеми. Це дозволяє зберегти децентралізованість, але при цьому мати можливість контролю доступу.

Досягнення консенсусу між вузлами відбувається за допомогою *протоколу PBFT*. Його особливість – швидке досягнення консенсусу без потреби у складних обчисленнях. Протокол побудований на механізмі голосування: щоб додати новий блок до ланцюга, необхідна згода щонайменше двох третин учасників.

Архітектурно модель виглядає так (рисунок 4.1):

- Дані від пристроїв Інтернету речей передаються через протокол MQTT з TLS [24].
- IoT-пристрої не є повноцінними вузлами блокчейну, а взаємодіють з ним через edge-шлюзи.
- Edge-шлюзи відповідають за формування транзакцій на основі даних з пристроїв та передають їх до блокчейн-мережі для узгодження.
- Усі вузли мережі мають заздалегідь визначені ключі, що унеможливлює участь сторонніх або невідомих учасників.

- Вузли виконують ролі учасників PBFT-протоколу: ініціюють обробку запиту, беруть участь у фазах pre-prepare, prepare та commit, після чого блок із транзакціями записується в блокчейн.
- Для забезпечення автоматичного управління доступом, активації пристроїв, перенесення скомпрометованих вузлів до карантину застосовуються смарт-контракти, які спрацьовують при виконанні певних умов.

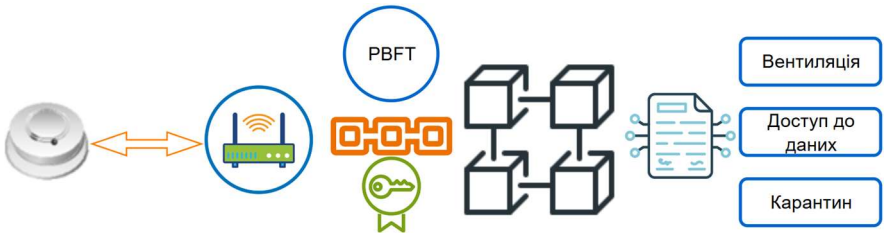

Рисунок 4.1 – Схема роботи мережі

Доцільність архітектури обґрунтовується такими факторами:

- **Консорціумна архітектура**: менша потреба в анонімності, підвищена контрольованість і надійність.
- **Протокол PBFT**: висока узгодженість і стійкість до наявності певної кількості скомпрометованих пристроїв.
- **Смарт-контракти**: гнучке управління доступом і реагування на події.
- **Енергоефективність**: відсутність високих енергетичних витрат у процесі підтвердження транзакцій робить модель придатною для пристроїв із обмеженими ресурсами.

Кожен компонент системи виконує чітко визначені функції в межах довіреного середовища, що дозволяє забезпечити контрольований обмін інформацією, фіксацію подій у незмінному реєстрі та динамічне управління доступом без централізованого контролера. Взаємозалежність елементів архітектури гарантує, що управління даними залишається розподіленим, контрольованим і захищеним навіть за умов виникнення часткових збоїв або скомпрометованих вузлів.

4.2 Прототип блокчейну з PBFT

Для перевірки працездатності запропонованої архітектури було реалізовано прототип блокчейн-системи, що функціонує на основі консенсусу PBFT. Реалізація виконана у вигляді модуля Python BL.py, що моделює функціонування вузлів, блокчейну та мережі загалом. На рисунку 4.2 зображено схему роботи прототипу. Повний код програми міститься у додатку А.

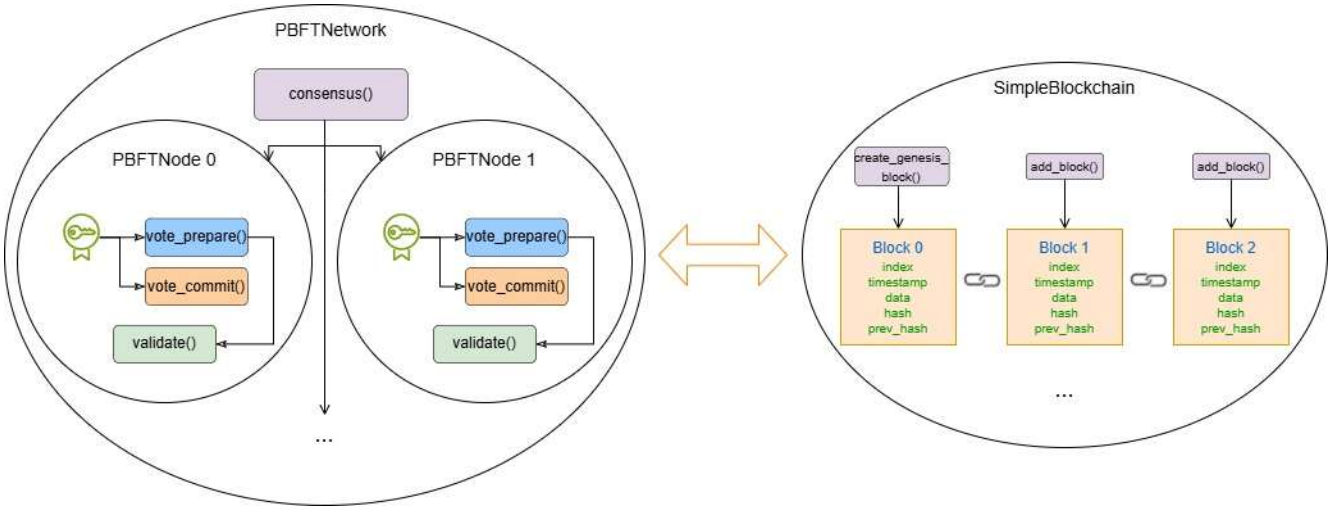

Рисунок 4.2 – Схема прототипу блокчейн-мережі

4.2.1 Клас PBFTNode

Даний клас моделює окремий вузол PBFT-мережі. Кожен учасник має унікальний ключ, що використовується як ідентифікатор довіри. Функція key() повертає істинне значення ключа.

У процесі консенсусу вузол бере участь у двох основних фазах – vote_prepare() та vote_commit(). На етапі prepare відбувається валідація блоку та перевірка справжності ключа вузла. Валідація реалізована доволі просто: перевіряється приналежність блоку до заданого типу та наявність обов’язкових атрибутів таких

як геш і дані. На етапі commit перевіряється тільки справжність ключа як еквівалент згоди та підпису легітимного вузла.

В реальних системах захист ключів базується на апаратно-криптографічних механізмах (наприклад, TPM, HSM) [6], де ключ неможливо прочитати навіть за повного контролю над ПЗ. У даній реалізації, навпаки, доступ до ключа спрощено – що відображає виключно логіку консенсусу PBFT, а не повноцінну модель захисту.

4.2.1 Клас PBFTNetwork

Клас PBFTNetwork реалізує модель PBFT-мережі, що складається з вказаної кількості вузлів, кожен з яких здатен брати участь у голосуванні щодо додавання блоку до блокчейну. Метою цього класу є симуляція процесу досягнення консенсусу відповідно до протоколу Practical Byzantine Fault Tolerance.

У конструкторі мережа створюється як список об’єктів PBFTNode. Кожному вузлу присвоюється унікальний ідентифікатор, який зберігається у відповідному вузлі. Кількість вузлів задається параметром num_nodes, який за замовчуванням дорівнює 4, що відповідає мінімальній кількості учасників для консенсусу PBFT.

Метод consensus() координує основні фази погодження блоку, обчислюючи кількість позитивних голосів на кожному етапі. Спочатку кожен вузол у мережі викликає метод vote_prepare(). Після цього підраховується кількість голосів «за», і якщо вона перевищує порогове значення, що відповідає двом третинам мережі, відбувається перехід до фази commit. Якщо й на цьому етапі набрано достатню кількість позитивних голосів, метод повертає True, що означає успішне досягнення консенсусу. У разі недостатньої кількості голосів на будь-якому етапі блок вважається відхиленим і виводиться повідомлення про неможливість досягнення згоди.

4.2.1 Клас SimpleBlockchain

Представлений клас реалізує базову модель блокчейну – лінійного ланцюга блоків, що зберігаються у вигляді списку. Основною функцією цього класу є послідовне збереження блоків даних, які було погоджено в мережі через консенсус. Клас забезпечує створення початкового (genesis) блоку та додавання нових блоків із збереженням зв'язків між ними.

Кожен блок містить:

- `index` – порядковий номер;
- `timestamp` – мітка часу створення блоку;
- `data` – збережені дані;
- `hash` – геш цих даних;
- `prev_hash` – геш даних попереднього блоку.

Генезис-блок – це перший блок у системі, який не має попередника, тому для нього явно задаються значення `hash: '0'` та `prev_hash: '0'`. Він також містить фіксований запис `'data': 'Genesis Block'` і позначається індексом 0.

Метод add_block() відповідає за формування нового блоку на основі вхідних даних і гешу. Спочатку визначається останній блок у ланцюгу (last_block), зчитується його геш, який буде збережений у полі prev_hash нового блоку, а сам новий блок формується за визначеною вище структурою. Блок додається до списку chain, після чого виводяться повідомлення про успішне додавання та поточну кількість блоків у ланцюгу.

4.3 Реалізація смарт-контрактів

Смарт-контракти в запропонованій моделі відіграють ключову роль у забезпеченні гнучкого, автоматизованого управління діями в мережі IoT. Їх виконання відбувається автоматично за наперед визначеними умовами, без потреби

в ручному втручанні. У системі реалізовано три окремі контракти, кожен з яких відповідає за певний аспект взаємодії.

4.3.1 Контракт контролю якості повітря

У даному смарт-контракті реалізовано реакцію системи на перевищення показників якості повітря в приміщенні. Код програми наведено у додатку Б.

Словник LIMITS містить порогові значення для таких показників як:

- **CO_2**: 1000 ppm [25];
- **$PM_{2.5}$**: 35 мкг/м^3 [26];
- **PM_{10}**: 154 мкг/м^3 [26];
- **TVOC**: 220 ppb [25];
- **CO**: 9 ppm [27].

При виклику функції check_exceedances() створюється новий словник exceedances за умови наявності перевищень в наданих даних. Якщо такі виявлено, перевіряється також статус роботи вентиляційної системи. У реальній мережі смарт-контракт надсилає команду на увімкнення вентиляції, якщо вона була вимкнена. Проте в даній реалізації, інформація про її запуск просто виводиться у консоль. Якщо ж вентиляція вже була увімкнена, натомість виводиться інформація тільки про це.

4.3.2 Контракт керування доступом

Контракт DataAccessControl відповідає за авторизацію учасників мережі, які зможуть отримати доступ до збережених у блокчейні даних. Система базується на моделі з єдиним адміністратором – власником контракту, який має право викликати функції для надання доступу – grant_access() – та його відкликання – revoke_access(). Повний код програми міститься у додатку В.

При реалізації смарт-контрактів на мові програмування Solidity, об’єктам можна призначити тип private. Це нікому не дає змоги читати та модифікувати їх вміст. Також за допомогою модуля Ownable з бібліотеки OpenZeppelin, можна визначити дії, які може робити лише власник смарт-контракту [28]. Перенесення цієї логіки на Python було виконано наступним чином.

На початку файлу «зашито» ключ адміністратора, за допомогою якого він має право на керування доступом. При ініціалізації класу цей ключ визначається як `self.__owner` та додається у список авторизованих користувачів `self.__authorized_users`, що мають право на доступ до даних у блокчейні. Використання подвійного підкреслення перед змінними означає, що вони не будуть доступні безпосередньо за своїми іменами. Це не повноцінний механізм захисту від їх читання та перезапису, але все ж таки шанс цього зменшується.

Методи класу grant_access() та revoke_access() може викликати лише власник цього контракту. Вони використовуються для додавання користувачів у список авторизованих та їх видалення звідти відповідно. Через метод get_data() надається доступ до даних шляхом перевірки наявності користувача у списку авторизованих. Повідомлення про надання або ненадання доступу виводяться у консоль.

4.3.3 Контракт карантину вузлів

Останній контракт NodeQuarantineContract моделює механізм виявлення та ізоляції вузлів, які є скомпрометованими. Автоматично виявляються випадки підміни ключа вузла, що може свідчити про зловмисне втручання, і виводити такий вузол з мережі в карантин до моменту з’ясування обставин. Код смарт-контракту наведено у додатку Г.

Через метод register_node() відбувається збереження всіх вузлів з їх автентичними ключами у словнику valid_keys. Метод print_status() виводить інформацію про стан мережі. Якщо у множині quarantined_nodes знаходяться вузли, їх список також виводиться у консоль.

У методі check_node_key() представлена головна логіка виявлення аномалій. Порівнюється наданий вузлом ключ reported_key із ключем, що був зареєстрований під час ініціалізації. Якщо ключі не збігаються, система вважає вузол скомпрометованим і додає його до множини карантину. Натомість, якщо вузол уже був у карантині, але передає правильний ключ, що свідчить про усунення проблеми, він виводиться з ізоляції. Повідомлення про всі події виводяться у консоль.

4.4 Тестування системи

Для перевірки працездатності запропонованої моделі було реалізовано тестовий сценарій через модуль Python main.py, повний вміст якого представлено у додатку Ґ. Цей файл виконує роль мережі, що координує взаємодію між блокчейном, вузлами PBFT, смарт-контрактами та вхідними даними від IoT-пристроїв (рисунок 4.3). У програмі наявні закоментовані частини для симуляції зловмисних дій.

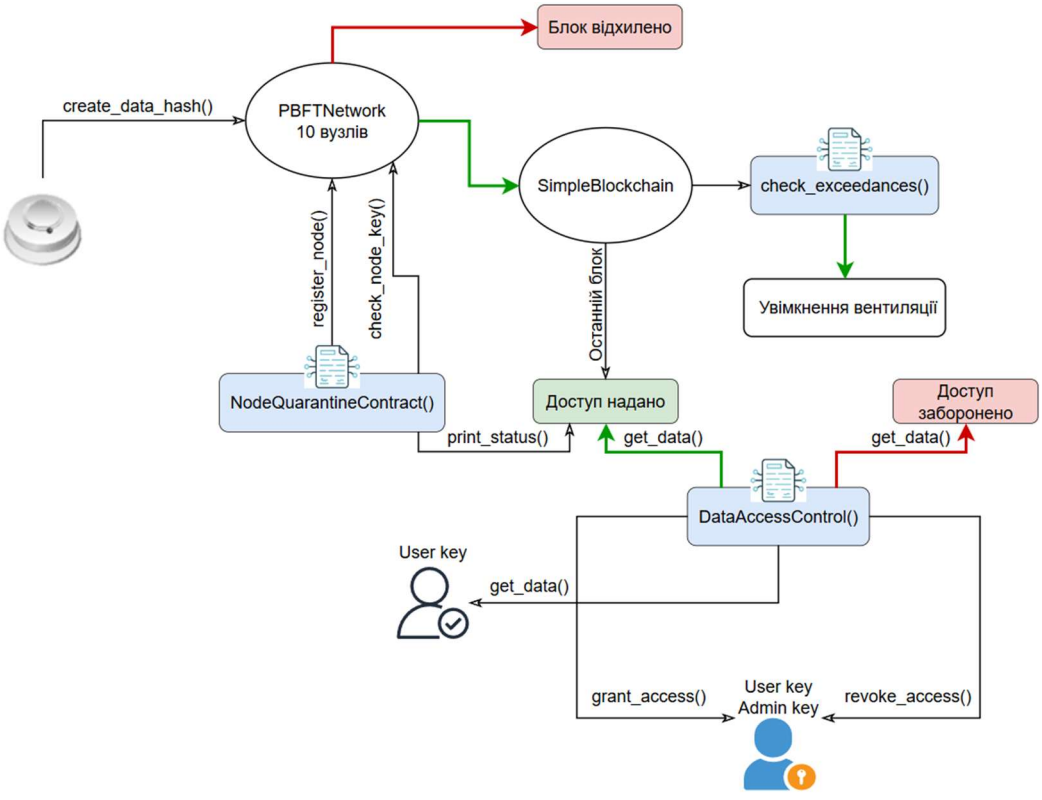

Рисунок 4.3 – Схема програми

Дані для тестування взяті з публічного датасету «IoT Indoor Air Quality» [29]. Були вручну відібрані деякі записи з січня поточного року та перетворені у більш зручніший формат JSON за допомогою коду на рисунку 4.4.

```
import pandas as pd

df = pd.read_csv('IoT_data.csv')
df.to_json('data.json', orient='records', date_format='iso', force_ascii=False, indent=2)
```

Рисунок 4.4 – Перетворення CSV у JSON

У коді ініціалізується блокчейн, мережа PBFT з 10 вузлами, смарт-контракти. Кожен рядок даних гешується функцією create_data_hash(), формується блок і перевіряється консенсусом. При успішному погодженні блок додається до блокчейну і викликається перевірка на якість повітря (рисунок 4.5).

```
[Edge] Геш даних: dc863549e7332f9103926fc3caec21de6380c251ee9cf6b21f82e8e619a393cd
[Blockchain] Блок додано до блокчейну
[Blockchain] Кількість блоків у ланцюгу: 42

[Edge] Геш даних: 4fbc8c3a29d645b4ebb8aa1f07a5942bbca7105b89551044dddfec03982ba2eb
[Blockchain] Блок додано до блокчейну
[Blockchain] Кількість блоків у ланцюгу: 43
[Smart contract] Перевищення норм: {'TVOC (ppb)': 250.32}
[Smart contract] Увімкнено вентиляцію

[Edge] Геш даних: f41f8f825375f797135838ddf39c15cf73570761dba5e162474c7e796f2e344a
[Blockchain] Блок додано до блокчейну
[Blockchain] Кількість блоків у ланцюгу: 44
[Smart contract] Перевищення норм: {'PM2.5 (?g/m?)': 61.26, 'TVOC (ppb)': 226.26}
[Smart contract] Вентиляція вже увімкнена

[Edge] Геш даних: cafa6ba3eef6899a6d9f703ebd7dbfe718cbbe63def5bb16aad57a3ee845c416
[Blockchain] Блок додано до блокчейну
[Blockchain] Кількість блоків у ланцюгу: 45

[Edge] Геш даних: bc34d53aee07e0b8f1bbb3d6394345e3caf8f0a02f8950edf312a7b19fe75204
[Blockchain] Блок додано до блокчейну
[Blockchain] Кількість блоків у ланцюгу: 46
[Smart contract] Перевищення норм: {'TVOC (ppb)': 242.65}
[Smart contract] Вентиляція вже увімкнена
```

Рисунок 4.5 – Процес додавання блоків

Додано простий інтерфейс з командами для надання/відкликання доступу, отримання даних з блокчейну, перевірки статусу карантину вузлів. Для надання доступу, адміністратор повинен ввести ключ користувача та свій ключ для

підтвердження автентичності. Після цього користувач вже матиме змогу отримати дані з блокчейну. Цю процедуру зображено на рисунку 4.6.

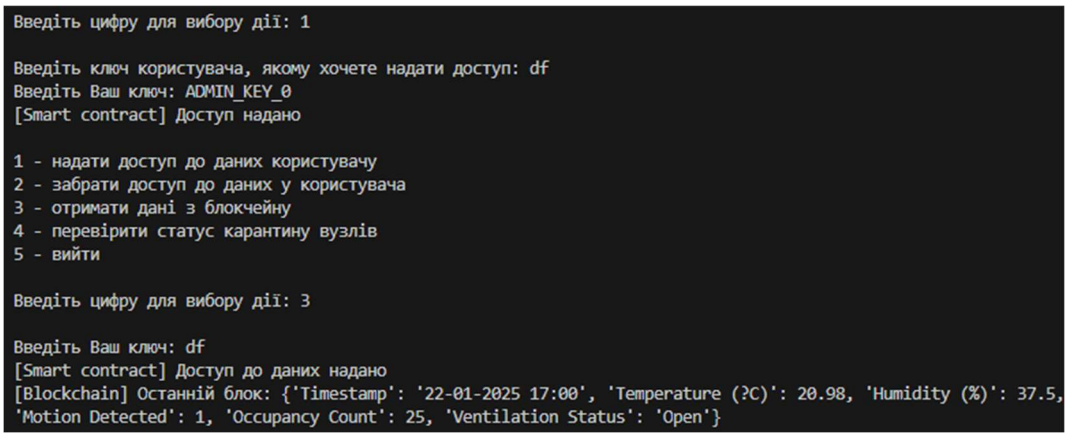

Рисунок 4.6 – Надання доступу користувачу

Для відкликання доступу власник повинен ввести ті ж самі дані. Після чого обраний користувач вже не матиме доступу до системи. На рисунку 4.7 зображено цей процес, і також продемонстровано заборону цієї дії, якщо користувач не адміністратор.

Рисунок 4.7 – Відкликання доступу до даних

Перевірити статус карантину вузлів можуть тільки авторизовані користувачі. Тому спочатку потрібно ввести свій ключ доступу. Даний процес зображено на рисунку 4.8. Для демонстрації роботи відповідного смарт-контракту реалізовано симуляцію підміни ключів вузлів та відновлення одного з них після перевірки карантину (рисунок 4.9).

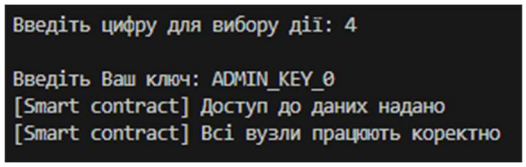

Рисунок 4.8 – Перевірка карантину вузлів

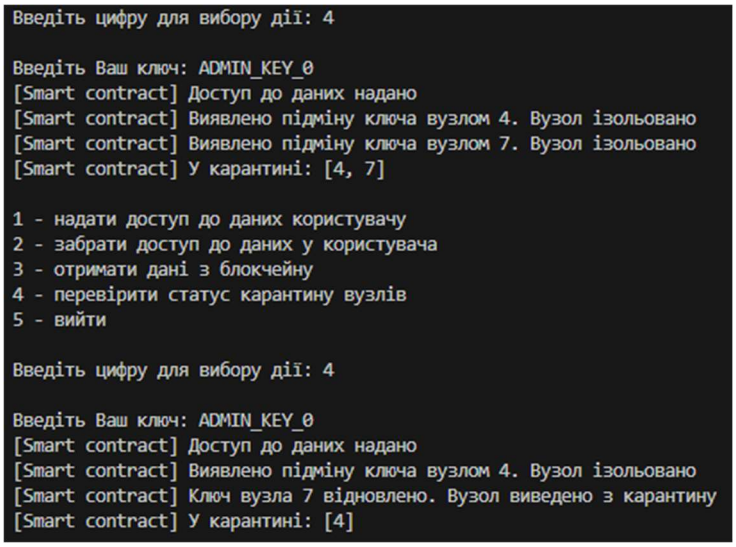

Рисунок 4.9 – Перевірка карантину вузлів з симуляцією підміни

На відміну від традиційної моделі системи з консенсусом PBFT, запропонована мережа не буде працювати при наявності більше ніж 1/3 зловмисних вузлів. Відповідно, нові блоки у такій системі не будуть додаватися, що потребує негайного втручання адміністратора мережі для відновлення скомпрометованих вузлів. Симуляцію такої ситуації представлено на рисунку 4.10, мережу ініціалізовано з 4 вузлами.

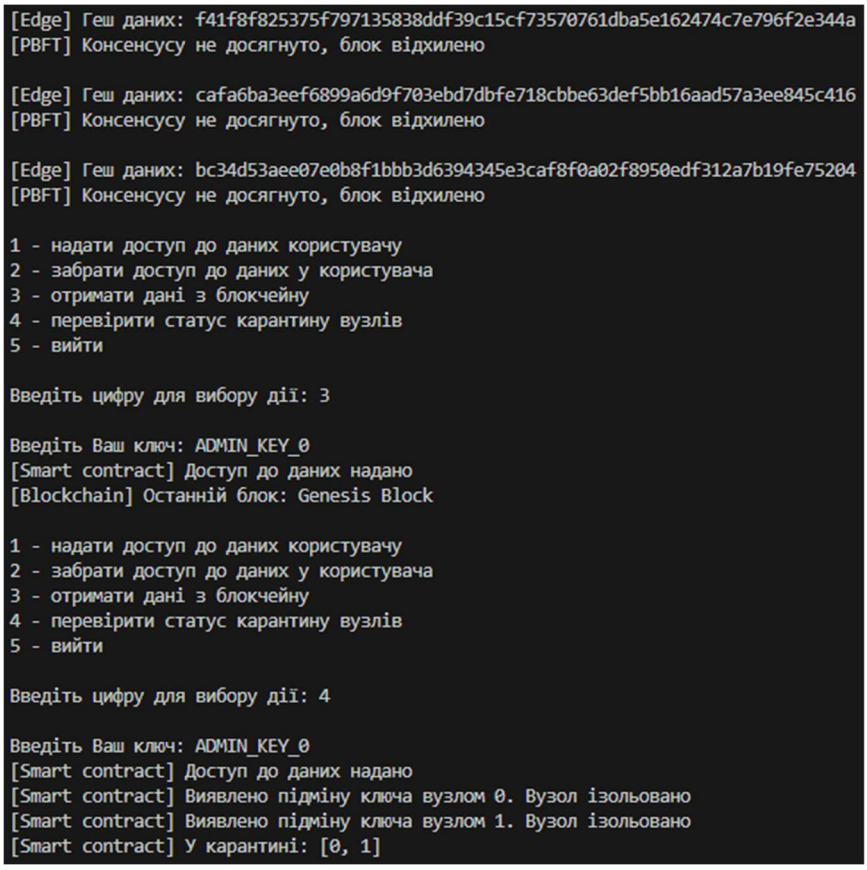

Рисунок 4.10 – Система з більше ніж 1/3 зловмисними вузлами

Висновки до розділу 4

На основі розробленої та реалізованої моделі, описаної в розділі 4, можна сформулювати висновки щодо практичної доцільності та ефективності запропонованого підходу до децентралізованого управління даними в IoT за допомогою блокчейну.

Застосування консорціумної блокчейн-мережі з протоколом PBFT дозволяє створити середовище, в якому дані з IoT-пристроїв зберігаються у незмінному вигляді та узгоджуються лише після підтвердження більшістю довірених учасників. Така архітектура забезпечує високий рівень стійкості до компрометації окремих вузлів, прозорість взаємодії та відсутність єдиної точки відмови.

Смарт-контракти відіграють ключову роль у забезпеченні безпеки та автоматизації процесів, оскільки дозволяють реалізувати контроль доступу до даних, автоматичне реагування на перевищення критичних показників, а також виявлення скомпрометованих вузлів у мережі. Це робить систему не лише безпечною, але й самокерованою – здатною адаптуватися до змін без зовнішнього втручання.

Результати тестування підтвердили працездатність моделі: блоки даних успішно формуються та додаються лише після досягнення консенсусу, смарт-контракти спрацьовують відповідно до заданих умов, а зловмисна поведінка вузлів виявляється і блокується автоматично. Така поведінка системи є показовою з точки зору практичного впровадження безпечної децентралізованої логіки управління у середовищах з підвищеними вимогами до довіри та цілісності даних.

Таким чином, запропонована модель інтеграції блокчейну в IoT-середовище була реалізована у вигляді функціонального прототипу як ефективної та контрольованої архітектури, здатної працювати в умовах обмежених ресурсів і водночас забезпечувати ключові вимоги до безпеки, узгодженості та автономності.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено архітектурну модель безпечного децентралізованого управління даними в системах Інтернету речей, яка поєднує блокчейн, протокол консенсусу PBFT та смарт-контракти для автоматизації дій. Запропоноване рішення дозволяє усунути залежність від централізованих координаторів і забезпечити цілісність, прозорість і стійкість до компрометації вузлів у розподілених середовищах.

У результаті аналізу сучасних IoT-систем виявлено, що найбільш критичними з точки зору безпеки є централізація управління, слабка автентифікація пристроїв, відсутність шифрування та низька стійкість до атак. Це створює передумови для втрати даних, перехоплення управління та збоїв у роботі інфраструктури. Було обґрунтовано необхідність переходу до децентралізованих моделей управління з підвищеним рівнем довіри між вузлами.

У процесі обґрунтування вибору архітектури встановлено, що консорціумний блокчейн надає оптимальний баланс між децентралізацією та контрольованим доступом. Протокол PBFT було обрано як ефективне рішення для досягнення консенсусу в умовах обмежених ресурсів IoT-систем, оскільки він не потребує енерговитратних обчислень і забезпечує відносно високу стійкість до наявності зловмисних вузлів.

Було реалізовано прототип системи, у якому IoT-пристрої передають дані через edge-шлюзи до блокчейн-мережі. Усі транзакції обробляються згідно з протоколом PBFT, а смарт-контракти забезпечують автоматизоване управління доступом, ізоляцію скомпрометованих вузлів і реакцію на визначені події. Структура смарт-контрактів дозволяє гнучко налаштовувати правила доступу без участі людини, що знижує ризик помилок або зловживань.

За результатами тестування моделі підтверджено її працездатність у сценаріях обміну даними, додавання нових блоків, фіксації подій і зміни статусу вузлів. Система демонструє стабільну взаємодію між компонентами, відсутність

критичних збоїв і здатність зберігати послідовність дій навіть у разі часткової компрометації мережі. Прототип показав, що модель може бути адаптована до реальних середовищ з обмеженими обчислювальними ресурсами та високими вимогами до безпеки.

У порівнянні з відомими рішеннями, запропонована модель поєднує концепції консорціумного блокчейну, протоколу PBFT і edge-архітектури, що дозволяє ефективно інтегрувати блокчейн у IoT-системи без значного перевантаження пристроїв. Результати роботи відповідають сучасним вимогам до інформаційної безпеки, мають прикладний характер і можуть бути використані як основа для подальших наукових досліджень та практичного впровадження. Надалі перспективним є розширення функціональності смарт-контрактів, інтеграція з механізмами виявлення аномалій у поведінці пристроїв, а також розробка більш гнучкої політики керування доступом на основі ролей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. IT-Enterprise. Internet of Things, IoT. — URL: https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot
2. Evolution of IoT in Various Application Domains / V. Sridhar, S. Rani, P. K. Pareek, P. Bhambri. — 2024. — DOI: 10.1201/9781003460367-2.
3. Vailshery L. S. Number of IoT connections worldwide 2022-2034. — URL: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
4. Yin Z. Edge gateways: Their role and importance in edge computing. — URL: https://stlpartners.com/articles/edge-computing/edge-gateways-their-role-and-importance/
5. Amnimo Inc. Edge gateway – YouTube. — URL: https://www.youtube.com/watch?v=QnK0rf3y69s
6. Aditya K. S., Mohan S. G. Internet of Things Security. — 2024. — DOI: 10.1201/9781003460367-4.
7. TyeYeah. Build Mirai Botnet and Try It. — 03/12/2020. — URL: https://tyeyeah.github.io/2020/12/03/2020-12-03-Build-Mirai-Botnet-and-Try-It/
8. Statt N. How an army of vulnerable gadgets took down the web today. — 10/21/2016. — URL: https://www.theverge.com/2016/10/21/13362354/dyn-dns-ddos-attack-cause-outage-status-explained
9. rogerlin. IoT SSL/TLS MITM Attack. — 06/08/2021. — URL: https://hackmd.io/@rogerlin/S1UeTQocO
10. ENISA. Baseline Security Recommendations for IoT. — 2017. — DOI: 10.2824/03228. — URL: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot
11. Парламент ЄС. Загальний регламент про захист даних (GDPR). — URL: https://gdpr-text.com/uk/
12. Верховна Рада України. Про захист персональних даних | від 01.06.2010 № 2297-VI. — URL: https://zakon.rada.gov.ua/laws/show/2297-17

13. Dave C. Security Challenges with Blockchain. — 2024 — ISBN: 978-81-96862-08-4.
14. richarddavis. Structure of a Block in Blockchain. — 06/21/2021. — URL: https://www.theengineeringprojects.com/2021/06/structure-of-a-block-in-blockchain.html
15. Uddin M., Muzammal M., Hameed M. K., Javed I. T., Alamri B., Crespi N. CBCIoT: A Consensus Algorithm for Blockchain-Based IoT Applications. Applied Sciences. 2021; 11(22):11011. https://doi.org/10.3390/app112211011
16. Salimitari M., Chatterjee M., Fallah Y. P. A survey on consensus methods in blockchain for resource-constrained IoT networks // Internet of Things. — 2020. — T. 11. — C. 100212. — ISSN 2542-6605. — DOI: https://doi.org/10.1016/j.iot.2020.100212 — URL: https://www.sciencedirect.com/science/article/pii/S2542660520300482
17. Anna. What is Proof of work (PoW)? | CoinLoan Blog. — 09/02/2022. — URL: https://coinloan.io/blog/what-is-proof-of-work-pow/
18. Bitcoin and The Byzantine Generals Problem - The Wolf of All Streets. — URL: https://www.thewolfofallstreets.io/bitcoin-and-the-byzantine-generals-problem/
19. Alotaibi, Sara. (2022). The Blockchain Framework for the Sharing of Data between Training Providers Based on the Internet of Things. Scientific Programming. 2022. 10.1155/2022/3944200.
20. У чому різниця між публічними, приватними та консорціумними блокчейнами? | Binance Academy. — URL: https://academy.binance.com/uk-UA/articles/private-public-and-consortium-blockchains-whats-the-difference
21. Exploring Integration of the Blockchain and Internet of Things (IoT) Computing for Secure and Decentralized Data Management / T. K. Vashishth, V. Sharma, K. K. Sharma, B. Kumar, S. Chaudhary, R. Panwar. — 2024. — DOI: 10.1201/9781003460367-10.
22. What is IoTeX: Privacy-Centric Blockchain for the IoT - Phemex Academy. — URL: https://phemex.com/academy/what-is-iotex-iotx

23. Singh Y. Blockchain in IoT Use Cases. — URL: https://ideausher.com/blog/blockchain-in-iot-use-cases/
24. HiveMQ Team. TLS/SSL - MQTT Security Fundamentals. — 03/06/2024. — URL: https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl/
25. Air quality shield – IBM Documentation. — URL: https://www.ibm.com/docs/en/mwi?topic=shields-air-quality-shield
26. Метеопост - Що таке PM2.5 та PM10. — URL: https://meteopost.com/info/PM/
27. Carbon Monoxide Levels Chart – CO2 Meter. — URL: https://www.co2meter.com/blogs/news/carbon-monoxide-levels-chart?srsltid=AfmBOoq1kNX9glP28WkU-FyE6qJJV9N6pcpDcVNkfGNE7z7N1fdOKG6w
28. Access Control – OpenZeppelin Docs. — URL: https://docs.openzeppelin.com/contracts/5.x/api/access#Ownable
29. Nusrath Ibrahim, and Praveen Veeramachaneni. (2025). IoT Indoor Air Quality [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/11462898

ДОДАТОК А PYTHON 3 КОД BL.PY

```
from time import time

#PBFT вузол
class PBFTNode:
    def __init__(self, node_id):
        self.node_id = node_id
        self.node_key = self.key()

    def key(self):
        return f"NODE_KEY_{self.node_id}"

    def vote_prepare(self, block, key):
        return self.validate_block(block) and self.key() == key

    def vote_commit(self, key):
        return self.key() == key

    def validate_block(self, block):
        return isinstance(block, dict) and 'hash' in block and 'data' in block

#PBFT симулятор
class PBFTNetwork:
    def __init__(self, num_nodes=4):
        self.nodes = [PBFTNode(i) for i in range(num_nodes)]

    def consensus(self, block):
        prepare_votes = sum(n.vote_prepare(block, n.node_key) for n in self.nodes)
        if prepare_votes >= (2 * (len(self.nodes) + 1) / 3):
            commit_votes = sum(n.vote_commit(n.node_key) for n in self.nodes)
            if commit_votes >= (2 * (len(self.nodes) + 1) / 3):
                return True
        print("[PBFT] Консенсусу не досягнуто, блок відхилено")
        return False

#Блокчейн
class SimpleBlockchain:
    def __init__(self):
        self.chain = []
        self.create_genesis_block()

    def create_genesis_block(self):
        self.chain.append({
            'index': 0,
            'timestamp': time(),
            'data': 'Genesis Block',
            'hash': '0',
```

```
            'prev_hash': '0'
        })

    def add_block(self, data, tx_hash):
        last_block = self.chain[-1]
        block = {
            'index': len(self.chain),
            'timestamp': time(),
            'data': data,
            'hash': tx_hash,
            'prev_hash': last_block['hash']
        }
        self.chain.append(block)
        print("[Blockchain] Блок додано до блокчейну")
        print(f"[Blockchain] Кількість блоків у ланцюгу: {len(self.chain)}")
        return block
```

ДОДАТОК Б PYTHON 3 КОД SMART_CONTRACT_AIR.PY

```
LIMITS = {
    "CO2 (ppm)": 1000,
    "PM2.5 (?g/m?)": 35,
    "PM10 (?g/m?)": 154,
    "TVOC (ppb)": 220,
    "CO (ppm)": 9
}

#Перевірка на перевищення норм показників якості повітря
def check_exceedances(data_row):
    exceedances = {
        param: value
        for param, value in data_row.items()
        if param in LIMITS and value > LIMITS[param]
    }
    if exceedances:
        print("[Smart contract] Перевищення норм:", exceedances)
        if data_row.get("Ventilation Status") == "Closed":
            print("[Smart contract] Увімкнено вентиляцію")
        else:
            print("[Smart contract] Вентиляція вже увімкнена")
```

ДОДАТОК В PYTHON 3 КОД SMART_CONTRACT_ACCESS.PY

```
ADMIN_KEY = "ADMIN_KEY_0"

class DataAccessControl:
    def __init__(self):
        self.__owner = ADMIN_KEY
        self.__authorized_users = {ADMIN_KEY}

    def grant_access(self, user, caller):
        if caller == self.__owner:
            self.__authorized_users.add(user)
            print("[Smart contract] Доступ надано")
        else:
            print("[Smart contract] Ви не можете надати доступ")

    def revoke_access(self, user, caller):
        if caller == self.__owner:
            self.__authorized_users.discard(user)
            print("[Smart contract] Доступ відкликано")
        else:
            print("[Smart contract] Ви не можете відкликати доступ")

    def get_data(self, caller):
        if caller in self.__authorized_users:
            print("[Smart contract] Доступ до даних надано")
            return True
        else:
            print("[Smart contract] Доступ до даних заборонено")
```

ДОДАТОК Г PYTHON 3 КОД SMART_CONTRACT_QUARANTINE.PY

```
class NodeQuarantineContract:
    def __init__(self):
        self.valid_keys = {}
        self.quarantined_nodes = set()

    def register_node(self, node_id, key):
        self.valid_keys[node_id] = key

    def check_node_key(self, node_id, reported_key):
        valid_key = self.valid_keys.get(node_id)
        if valid_key != reported_key:
            self.quarantined_nodes.add(node_id)
            print(f"[Smart contract] Виявлено підміну ключа вузлом {node_id}. Вузол
ізольовано")
            return
        if node_id in self.quarantined_nodes:
            self.quarantined_nodes.remove(node_id)
            print(f"[Smart contract] Ключ вузла {node_id} відновлено. Вузол виведено з
карантину")

    def print_status(self):
        if not self.quarantined_nodes:
            print("[Smart contract] Всі вузли працюють коректно")
        else:
            print(f"[Smart contract] У карантині: {list(self.quarantined_nodes)}")
```

ДОДАТОК Ґ PYTHON 3 КОД MAIN.PY

```
import json
import hashlib
from BL import PBFTNetwork, SimpleBlockchain
import smart_contract_air
from smart_contract_access import DataAccessControl
from smart_contract_quarantine import NodeQuarantineContract

HELP = """
1 - надати доступ до даних користувачу
2 - забрати доступ до даних у користувача
3 - отримати дані з блокчейну
4 - перевірити статус карантину вузлів
5 - вийти
"""

with open("data.json", "r") as f:
    sensor_data_batch = json.load(f)

def create_data_hash(data_row):
    data_string = json.dumps(data_row, sort_keys=True)
    return hashlib.sha256(data_string.encode()).hexdigest()

blockchain = SimpleBlockchain()
pbft_network = PBFTNetwork(num_nodes=10)

#Симуляція системи з більше ніж 1/3 зловмисними вузлами
#pbft_network = PBFTNetwork(num_nodes=4)
#pbft_network.nodes[0].node_key = "NODE_KEY_FAKE"
#pbft_network.nodes[1].node_key = "NODE_KEY_FAKE1"

#Додавання блоків до блокчейну
for row in sensor_data_batch:
    tx_hash = create_data_hash(row)

    print("\n[Edge] Геш даних:", tx_hash)

    new_block = {
        'data': row,
        'hash': tx_hash,
    }

    if pbft_network.consensus(new_block):
        blockchain.add_block(new_block['data'], new_block['hash'])
        smart_contract_air.check_exceedances(blockchain.chain[-1]['data'])

access_control = DataAccessControl()
```

```
quarantine = NodeQuarantineContract()

#Реєстрація вузлів у контракті карантину
for node in pbft_network.nodes:
    quarantine.register_node(node.node_id, node.key())

#Симуляція підміни ключа вузлів
#pbft_network.nodes[4].node_key = "NODE_KEY_FAKE"
#pbft_network.nodes[7].node_key = "NODE_KEY_FAKE1"

while True:
    #Вибір дії
    print(HELP)
    choice = input("Введіть цифру для вибору дії: ")

    if choice == '1':
        access_control.grant_access(input("\nВведіть ключ користувача, якому хочете
надати доступ: "), input("Введіть Ваш ключ: "))
    elif choice == '2':
        access_control.revoke_access(input("\nВведіть ключ користувача, у якого хочете
забрати доступ: "), input("Введіть Ваш ключ: "))
    elif choice == '3':
        if access_control.get_data(input("\nВведіть Ваш ключ: ")):
            blockchain_data = blockchain.chain[-1]['data']
            print("[Blockchain] Останній блок:", blockchain_data)
    elif choice == '4':
        if access_control.get_data(input("\nВведіть Ваш ключ: ")):
            for node in pbft_network.nodes:
                quarantine.check_node_key(node.node_id, node.node_key)
            quarantine.print_status()
            #Відновлення справжнього ключа вузла для тестування
            #pbft_network.nodes[7].node_key = pbft_network.nodes[7].key()
    elif choice == '5':
        break
    else:
        print("Такого режиму немає")
```