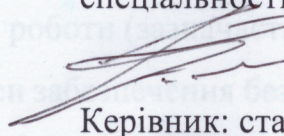


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра
на тему: «Методи та технології захисту IoT-пристроїв
на базі технологій блокчейн»

Виконав: студент 2-го курсу групи ЗКБм-24-1
спеціальності 125 «Кібербезпека та захист інформації»

 Володимир МОЛОДЕЦЬКИЙ

Керівник: старший викладач кафедри КІ та КБ

 Костянтин МАТВЕЄВ

Рецензент: доцент кафедри КІ та КБ, к.т.н., доцент

 Юрій РОССІНСЬКИЙ

Житомир – 2026

Область Державний університет «Житомирська політехніка»

Персонал Факультет інформаційно-комп'ютерних технологій

IoT-при Кафедра комп'ютерної інженерії та кібербезпеки

Програма Спеціальність 125 «Кібербезпека та захист інформації»

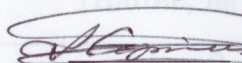
MQTT-брокер Освітня програма «Кібербезпека»

Блокчейн-платформа для реалізації смарт-контрактів

База даних для зберігання телеметричних даних ЗАТВЕРДЖУЮ

Консультанти з випускної кваліфікаційної роботи Завідувач
кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій ЄФІМЕНКО

3 03 2026 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу магістра

Студент: Молодецький Володимир Олександрович

Тема роботи: «Методи та технології захисту IoT-пристроїв на базі технологій блокчейн»

затверджена Наказом університету від 03 березня 2026 р. № 99/с

Термін здачі студентом закінченої роботи 25 04 2026 р.

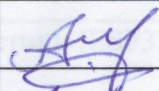
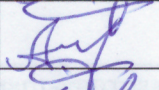
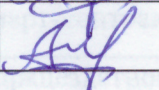
Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

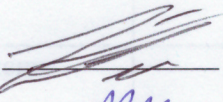
1. процеси забезпечення безпеки в розподілених IoT-системах, зокрема передача та обробка телеметричних даних від пристроїв із обмеженими ресурсами.
2. методи та технології забезпечення автентичності та цілісності даних IoT-пристроїв на основі криптографічних механізмів і блокчейн-інфраструктури
3. розроблення та програмна реалізація моделі захисту IoT-пристроїв, що забезпечує криптографічну автентифікацію, перевірку цілісності даних та їх незмінність за допомогою смарт-контрактів.

Обладнання та програмне забезпечення:

- Персональний комп'ютер або ноутбук
- IoT-пристрій (його програмна емуляція)
- Програмне забезпечення для розробки та тестування
- MQTT-брокер для передачі даних
- Блокчейн-платформа для реалізації смарт-контрактів
- База даних для зберігання телеметричної інформації

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Олексій ШЕЛУХА	 3.03.26	 3.03.26
2	Олексій ШЕЛУХА	 3.03.26	 3.03.26
3	Олексій ШЕЛУХА	 3.03.26	 3.03.26

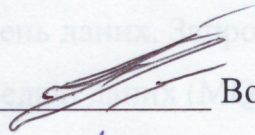
Студент  Володимир МОЛОДЕЦЬКИЙ

Керівник  Костянтин МАТВЄСЬ

Календарний план

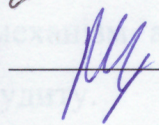
№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі	04.03.2026	Виконано
2	Аналіз проблем безпеки в IoT-мережах	08.03.2026	Виконано
3	Формування технічного завдання	12.03.2026	Виконано
4	Опрацювання літературних джерел	18.03.2026	Виконано
5	Проектування архітектури системи	23.03.2026	Виконано
6	Розробка моделі захисту IoT-пристроїв	28.03.2026	Виконано
7	Реалізація програмної частини (ESP32 + backend)	03.04.2026	Виконано
8	Інтеграція блокчейн та смарт-контракту	08.04.2026	Виконано
9	Тестування та перевірка працездатності системи	12.04.2026	Виконано
10	Написання пояснювальної записки	14.04.2026	Виконано
11	Підготовка до захисту	15.04.2026	Виконано

Студент



Володимир МОЛОДЕЦЬКИЙ

Керівник



Костянтин МАТВЄЄВ

Практична частина роботи включає розробку програмного сервісу для перевірки достовірності телеметричних даних, реалізацію взаємодії з MQTT-браузером, а також інтеграцію зі смарт-контрактом для збереження агрегованих даних у блокчейн-мережі. Проведене моделювання підтвердило ефективність запропонованого підходу та його стійкість до основних типів атак.

Ключові слова: IoT, BLOCKCHAIN, КІБЕРБЕЗПЕКА, КРИПТОГРАФІЯ, ДОСТОВІРНІСТЬ ДАНИХ, ЦИФРОВИЙ ПІДПИС.

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Методи та технології захисту IoT-пристроїв на базі технологій блокчейн» складається зі вступу, трьох розділів, висновків та списку використаної літератури. Текстова частина викладена на 57 сторінках друкованого тексту. Список використаних джерел містить 28 найменувань. У роботі наведено 7 рисунків. Загальний обсяг роботи – 57 сторінок.

Кваліфікаційна робота присвячена дослідженню та реалізації методів забезпечення безпеки IoT-пристроїв із використанням криптографічних механізмів та блокчейн-технологій. Основною метою роботи є розробка моделі захисту, що забезпечує достовірність, цілісність та автентичність телеметричних даних у розподіленому середовищі IoT.

У роботі проаналізовано сучасні архітектури IoT-систем, основні загрози безпеці, зокрема атаки типу Man-in-the-Middle, spoofing, replay та модифікацію даних, а також існуючі підходи до їх захисту. Обґрунтовано необхідність переходу від централізованих моделей безпеки до децентралізованих підходів із використанням блокчейн як механізму забезпечення довіри.

Розроблено модель захисту IoT-пристроїв, що базується на використанні криптографічного хешування (SHA-256), цифрового підпису та технології блокчейн для фіксації контрольних значень даних. Запропонована архітектура включає IoT-пристрої, мережевий рівень передачі даних (MQTT через TLS), сервер обробки та блокчейн-рівень. Реалізовано механізм автентифікації пристроїв, перевірки цілісності даних та їх подальшого аудиту.

Практична частина роботи включає розробку програмного сервісу для перевірки достовірності телеметричних даних, реалізацію взаємодії з MQTT-брокером, а також інтеграцію зі смарт-контрактом для збереження агрегованих хешів даних у блокчейн-мережі. Проведене моделювання підтвердило ефективність запропонованого підходу та його стійкість до основних типів атак.

Ключові слова: IoT, BLOCKCHAIN, КІБЕРБЕЗПЕКА, КРИПТОГРАФІЯ, ДОСТОВІРНІСТЬ ДАНИХ, ЦИФРОВИЙ ПІДПИС.

ANNOTATION

The explanatory note to the qualification work entitled «Methods and Technologies for Securing IoT Devices Based on Blockchain Technologies» consists of an introduction, three chapters, conclusions, and a list of references. The main text is presented on 57 pages of printed text. The list of references includes 28 sources. The work contains 7 figures. The total volume of the work is 57 pages.

The qualification work is devoted to the study and implementation of methods for securing IoT devices using cryptographic mechanisms and blockchain technologies. The main objective of the work is to develop a security model that ensures the reliability, integrity, and authenticity of telemetry data in a distributed IoT environment.

The work analyzes modern IoT architectures, major security threats such as Man-in-the-Middle, spoofing, replay, and data tampering attacks, as well as existing protection approaches. The necessity of transitioning from centralized security models to decentralized approaches using blockchain as a trust mechanism is substantiated.

A protection model for IoT devices is developed based on cryptographic hashing (SHA-256), digital signatures, and blockchain technology for fixing control values of data. The proposed architecture includes IoT devices, a data transmission layer (MQTT over TLS), a backend processing server, and a blockchain layer. Mechanisms for device authentication, data integrity verification, and further auditing are implemented.

The practical part of the work includes the development of a software service for validating telemetry data, implementation of interaction with an MQTT broker, and integration with a smart contract for storing aggregated data hashes in a blockchain network. The conducted simulation confirmed the effectiveness of the proposed approach and its resistance to major types of attacks.

Keywords: IoT, BLOCKCHAIN, CYBERSECURITY, CRYPTOGRAPHY, DATA INTEGRITY, DIGITAL SIGNATURE.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. Аналіз проблем безпеки IoT-систем та принципи їх захисту.....	10
1.1. Архітектура IoT-систем та особливості забезпечення їх безпеки	10
1.2. Принципи забезпечення безпеки IoT із використанням блокчейн-технологій	14
1.3. Криптографічні та мережеві методи захисту IoT-інфраструктури	17
Висновок до першого розділу	21
РОЗДІЛ 2. Методи та моделі захисту IoT-пристроїв на основі блокчейн-технологій	23
2.1. Криптографічні методи автентифікації у системах захисту IoT	23
2.2. Модель автентифікації IoT-пристроїв на основі блокчейн-інфраструктури	26
2.3. Аналіз загроз та оцінка ефективності запропонованої моделі захисту IoT-пристроїв.....	30
Висновок до другого розділу	34
РОЗДІЛ 3. Моделювання системи захисту IoT-пристроїв на основі криптографічних методів та блокчейн-технологій.....	36
3.1. Загальна модель системи забезпечення достовірності телеметричних даних IoT-пристроїв.....	36
3.2. Моделювання даних, потоків та структури зберігання інформації в системі	38
3.3. Реалізація системи забезпечення достовірності телеметричних даних IoT-пристроїв.....	42
Висновок до третього розділу	47
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	51
ДОДАТКИ.....	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACL – список контролю доступу (Access Control List)

API – програмний інтерфейс застосунку (Application Programming Interface)

Batch – група записів для агрегованої обробки

Blockchain – розподілений реєстр даних

DB – база даних (Database)

ESP32 – мікроконтролер для IoT-пристроїв

H – хеш-значення повідомлення

ID – ідентифікатор пристрою

IoT – Інтернет речей (Internet of Things)

MITM – атака «людина посередині» (Man-in-the-Middle)

MQTT – протокол обміну повідомленнями (Message Queuing Telemetry Transport)

PK – відкритий ключ (Public Key)

PKI – інфраструктура відкритих ключів (Public Key Infrastructure)

S – цифровий підпис

SHA-256 – криптографічна хеш-функція

SK – закритий ключ (Secret Key)

Smart Contract – смарт-контракт у блокчейн-мережі

TLS – протокол захищеної передачі даних (Transport Layer Security)

Web3 – інтерфейс взаємодії з блокчейн-мережею

ВСТУП

Стрімкий розвиток технологій Інтернету речей (IoT) зумовив широке впровадження розподілених систем, у яких велика кількість пристроїв здійснює збір, передачу та обробку телеметричних даних у режимі реального часу. Такі системи активно використовуються у промисловості, транспорті, енергетиці, а також у системах «розумного» середовища. Водночас специфічні особливості IoT-пристроїв, зокрема обмежені обчислювальні ресурси, автономність функціонування та велика масштабованість мереж, суттєво ускладнюють забезпечення їхньої безпеки.

Однією з ключових проблем сучасних IoT-систем є відсутність надійних механізмів перевірки автентичності джерела даних та забезпечення їх цілісності. У більшості реалізацій передача телеметрії здійснюється через брокери повідомлень, які виконують лише транспортні функції та не забезпечують криптографічної перевірки інформації. Централізовані сервери, що виконують обробку даних, можуть виступати єдиною точкою відмови, а також об'єктом атак, спрямованих на підміну або модифікацію даних.

У таких умовах особливої актуальності набуває розроблення нових підходів до побудови систем захисту IoT, які дозволяють забезпечити достовірність даних на всіх етапах їх життєвого циклу – від моменту генерації на пристрої до зберігання та подальшої перевірки. Одним із перспективних напрямів є поєднання криптографічних механізмів із технологіями розподілених реєстрів (блокчейн), що дозволяє реалізувати модель децентралізованої довіри та забезпечити незмінність критичних даних.

В межах даної роботи запропоновано підхід до побудови захищеної IoT-системи, у якому пристрій (ESP32 або його програмна емуляція) формує телеметричні дані, обчислює їх хеш за алгоритмом SHA-256 та створює цифровий підпис із використанням приватного ключа. Дані передаються через протокол MQTT із використанням TLS, після чого на серверному рівні виконується перевірка їх цілісності та автентичності. Для підвищення масштабованості реалізовано механізм пакетної обробки, у межах якого формується агрегований хеш групи записів. Отримані

контрольні значення зберігаються у блокчейн-мережі за допомогою смарт-контракту, що забезпечує їх незмінність та можливість незалежної перевірки.

Такий підхід дозволяє усунути залежність від централізованих механізмів довіри, забезпечити криптографічну перевірку даних та створити надійну основу для аудиту IoT-систем.

Метою кваліфікаційної роботи є розроблення та дослідження моделі захисту IoT-пристроїв на основі криптографічних механізмів і блокчейн-технологій, спрямованої на забезпечення автентичності джерела даних, цілісності телеметричної інформації та її незмінності.

Об'єктом дослідження є процеси передачі, обробки та зберігання телеметричних даних у IoT-системах.

Предметом дослідження є методи та алгоритми забезпечення автентичності, цілісності та незмінності даних IoT-пристроїв із використанням криптографії та блокчейн-технологій.

Для досягнення поставленої мети у роботі визначено такі завдання:

- проаналізувати сучасні загрози безпеці IoT-систем та дослідити обмеження традиційних підходів до захисту даних;
- розробити модель автентифікації IoT-пристроїв і реалізувати механізм перевірки цілісності телеметричних даних;
- розробити систему серверної валідації повідомлень разом із механізмом пакетної обробки та формування агрегованих хешів;
- інтегрувати блокчейн для забезпечення незмінності даних та провести оцінку ефективності запропонованої моделі.

У результаті виконання роботи розроблено модель захисту IoT-системи, що поєднує криптографічну автентифікацію, серверну перевірку та блокчейн-якоріння даних. Запропонований підхід дозволяє підвищити рівень безпеки IoT-інфраструктури, забезпечити достовірність телеметричних даних та створити механізм їх незалежної перевірки без суттєвого зниження продуктивності системи.

РОЗДІЛ 1. Аналіз проблем безпеки IoT-систем та принципи їх захисту

1.1. Архітектура IoT-систем та особливості забезпечення їх безпеки

Інтернет речей (IoT) формує нову парадигму побудови інформаційних систем, у якій фізичні об'єкти інтегруються з обчислювальною інфраструктурою через мережеві технології обміну даними. У межах такої концепції сенсори, виконавчі механізми, контролери та інші пристрої отримують можливість автоматично збирати, передавати та обробляти інформацію, що забезпечує створення кіберфізичних систем нового покоління[4]. IoT активно застосовується у різних галузях, зокрема у промисловій автоматизації, системах «розумного» міста, медичних інформаційних системах та інтелектуальних енергетичних мережах.

На відміну від традиційних інформаційних систем, IoT-середовище характеризується високим рівнем розподіленості, значною кількістю взаємопов'язаних вузлів та безперервним потоком телеметричних даних. У таких умовах забезпечення безпеки стає складним завданням, оскільки компрометація окремого пристрою або каналу передачі може призвести не лише до локального порушення, але й до поширення недостовірних даних у межах всієї системи. Особливо критичним є те, що телеметричні дані часто використовуються для прийняття автоматизованих рішень, тому їхня достовірність безпосередньо впливає на коректність функціонування системи [14].

Архітектурні рівні IoT-систем

Більшість сучасних IoT-систем реалізуються відповідно до багаторівневої архітектурної моделі, що забезпечує розподіл функцій між різними компонентами інфраструктури [4]. Найбільш поширеною є трирівнева архітектура, яка включає:

1. Периферійний рівень (device layer)

На цьому рівні функціонують фізичні пристрої – сенсори, виконавчі механізми та мікроконтролери. Основним завданням таких пристроїв є збір даних із навколишнього середовища або виконання команд, що надходять із системи керування. Як правило, ці вузли мають обмежені апаратні ресурси, включаючи невелику обчислювальну потужність, обмежений обсяг пам'яті та низьке енергоспоживання.

Типовим прикладом таких пристроїв є мікроконтролери на базі ESP32, які широко застосовуються у прототипуванні та розробці IoT-рішень. У контексті забезпечення безпеки саме цей рівень виступає джерелом криптографічної довіри, оскільки пристрій володіє приватним ключем і формує цифровий підпис [2] для кожного повідомлення. Таким чином, достовірність даних повинна забезпечуватися ще на етапі їх генерації, а не лише під час передачі або обробки.

2. Проміжний рівень (edge або gateway layer)

Цей рівень виконує функцію агрегування даних, отриманих від периферійних пристроїв, та забезпечує їхню попередню обробку. Шлюзи або edge-вузли можуть здійснювати фільтрацію, буферизацію та первинний аналіз даних перед передаванням до центральних сервісів. Крім того, вони часто виступають точкою інтеграції між різними мережевими протоколами.

У сучасних IoT-системах на цьому рівні часто реалізується брокер повідомлень, наприклад Eclipse Mosquitto, який забезпечує маршрутизацію даних між пристроями та серверною частиною. Водночас важливо відзначити, що такі компоненти виконують лише функції транспорту та контролю доступу (ACL), але не забезпечують криптографічної перевірки цілісності та автентичності повідомлень. Таким чином, даний рівень не може розглядатися як повністю довірений.

3. Рівень сервісів та обробки даних (cloud layer)

На цьому рівні відбувається централізована обробка та зберігання даних, а також реалізуються сервіси керування пристроями, аналітики та візуалізації. Саме тут, як правило, виконується перевірка отриманих повідомлень, включаючи повторне обчислення хешу та верифікацію цифрового підпису.

У контексті запропонованої моделі саме серверний рівень виконує критичну функцію перевірки достовірності даних перед їх збереженням. Додатково після накопичення певної кількості записів формується агрегований хеш, який може бути зафіксований у розподіленому реєстрі, що забезпечує можливість подальшого аудиту.

Така архітектура забезпечує масштабованість і гнучкість системи, проте одночасно формує складну структуру взаємодій між вузлами, що створює додаткові

виклики для забезпечення безпеки, особливо у частині забезпечення наскрізної довіри до даних.

Протоколи взаємодії у IoT

Для забезпечення обміну даними між компонентами IoT-систем використовуються спеціалізовані мережеві протоколи, оптимізовані для пристроїв з обмеженими ресурсами. Найбільш поширеними є:

- MQTT (Message Queuing Telemetry Transport) – легкий протокол обміну повідомленнями за моделлю publish/subscribe, який широко застосовується у системах з низькою пропускнуою здатністю мережі [5];
- CoAP (Constrained Application Protocol) – протокол, орієнтований на пристрої з обмеженими ресурсами;
- HTTP/REST – застосовується для інтеграції IoT-сервісів із веб-додатками.

У сучасних реалізаціях MQTT використовується разом із протоколом TLS [7], що забезпечує захист каналу передачі даних. Проте TLS гарантує лише захист транспортного рівня і не забезпечує перевірку достовірності джерела даних після їх отримання. Таким чином, навіть при використанні захищених каналів залишається відкритою проблема наскрізної перевірки автентичності та цілісності повідомлень [15].

Типи IoT-пристроїв

IoT-інфраструктура складається з різних типів пристроїв, що виконують спеціалізовані функції:

- сенсори, які збирають інформацію з навколишнього середовища;
- виконавчі механізми, що здійснюють фізичний вплив;
- контролери та мікроконтролери, які забезпечують локальну обробку даних;
- шлюзи, що виконують роль посередників між локальною мережею та зовнішніми сервісами.

Гетерогенність таких пристроїв призводить до виникнення складної системи взаємозалежностей, у якій кожен компонент може стати потенційною точкою атаки або джерелом недостовірних даних.

Проблеми безпеки в архітектурі IoT

Попри переваги розподіленої архітектури, більшість існуючих IoT-рішень використовують централізовану модель керування, у якій ключові функції автентифікації та зберігання журналів подій зосереджені на центральному сервері або хмарній платформі. Такий підхід спрощує адміністрування системи, проте створює єдину точку відмови [14].

У разі компрометації центрального сервера можливі такі наслідки:

- підміна або модифікація журналів подій;
- несанкціонована зміна конфігурації пристроїв;
- підробка телеметричних даних;
- масове порушення політик доступу.

Додатковою проблемою є те, що транспортні протоколи (зокрема MQTT) та механізми контролю доступу (ACL) не забезпечують повноцінної перевірки достовірності даних. У результаті система може приймати та обробляти повідомлення, які є коректними з точки зору мережевої взаємодії, але не є достовірними з криптографічної точки зору.

Таким чином, ключовою проблемою сучасних IoT-архітектур є відсутність наскрізної моделі довіри до даних, яка б забезпечувала їхню автентичність і цілісність на всіх етапах обробки.

Використання блокчейн для побудови децентралізованої моделі довіри

Одним із перспективних підходів до вирішення зазначеної проблеми є використання технології блокчейн як механізму забезпечення незмінності даних та незалежного аудиту. Блокчейн являє собою розподілений реєстр, у якому записи зберігаються у вигляді криптографічно пов'язаних блоків [10].

У контексті IoT доцільно використовувати блокчейн не для зберігання всіх телеметричних даних, а як механізм фіксації їх агрегованих хешів (якорів довіри). Такий підхід дозволяє:

- забезпечити незмінність історичних даних;
- виконувати незалежну перевірку цілісності;

- зменшити навантаження на блокчейн-мережу;
- зберегти масштабованість системи.

Враховуючи обмежені ресурси IoT-пристроїв, безпосередня участь у блокчейн-мережі є недоцільною. Тому ефективною є архітектура, у якій блокчейн використовується як фінальний рівень фіксації достовірності даних після їх криптографічної перевірки на серверному рівні.

1.2. Принципи забезпечення безпеки IoT із використанням блокчейн-технологій

Безпека IoT-систем розглядається як комплекс властивостей, що забезпечують стійке та передбачуване функціонування розподіленої мережі пристроїв у потенційно ворожому середовищі. Особливістю IoT є поєднання кібернетичних і фізичних процесів: вплив на інформаційний компонент безпосередньо відображається на фізичних об'єктах керування. У зв'язку з цим теоретичні засади безпеки мають враховувати як класичні принципи інформаційної безпеки, так і специфіку кіберфізичних систем, зокрема необхідність гарантування достовірності телеметричних даних [2].

Базовими властивостями захищеності залишаються конфіденційність, цілісність, доступність та автентичність. Проте в IoT-середовищі особливого значення набувають додаткові вимоги, зокрема:

- забезпечення наскрізної достовірності даних (від моменту їх формування до зберігання);
- можливість перевірки історії змін;
- відсутність залежності від єдиного центру довіри;
- підтримка аудиту та верифікації у довгостроковій перспективі.

Традиційна модель безпеки IoT базується на централізованих сервісах керування ідентичностями та журналювання подій. Такий підхід спрощує реалізацію, однак створює критичні ризики. Компрометація центрального вузла може призвести до:

- підміни телеметричних даних;
- втрати достовірності журналів подій;
- несанкціонованого доступу до системи.

Крім того, централізована модель ускладнює масштабування, оскільки зі збільшенням кількості пристроїв зростає навантаження на серверну інфраструктуру. Це обумовлює необхідність переходу до моделей, у яких довіра формується не лише за рахунок централізованих компонентів, а через криптографічну перевірку та незалежну фіксацію результатів.

Однією з ключових задач є побудова моделі автентифікації, що забезпечує однозначну ідентифікацію пристрою протягом усього життєвого циклу. Використання спільних секретів або статичних ключів є недостатньо ефективним, оскільки створює передумови для клонування пристроїв. У зв'язку з цим доцільним є застосування криптографії з відкритим ключем, де кожен пристрій має власну пару ключів, а автентичність підтверджується за допомогою цифрового підпису [17].

Важливим аспектом є також забезпечення цілісності даних. У запропонованій моделі це досягається шляхом обчислення хеш-значення повідомлення безпосередньо на IoT-пристрої. Такий підхід дозволяє виявляти будь-які зміни даних незалежно від етапу їх обробки або передачі [16].

Однак навіть за умови використання криптографічних механізмів виникає проблема довгострокової довіри до даних. У централізованих системах записи можуть бути змінені або видалені, що унеможливорює проведення незалежного аудиту. Для вирішення цієї проблеми доцільним є використання блокчейн-технологій.

У межах даної роботи блокчейн розглядається не як середовище зберігання всіх телеметричних даних, а як інструмент фіксації їх контрольних значень. Зокрема, після накопичення певної кількості записів формується пакет (batch), для якого обчислюється агрегований хеш:

$$H_{\{batch\}} = H(R_1 \parallel R_2 \parallel \dots \parallel R_n)$$

де (R_i) – окремі записи телеметрії.

Отримане значення записується у смарт-контракт блокчейн-мережі, що забезпечує:

- незмінність контрольного значення;
- можливість незалежної перевірки даних;
- відсутність залежності від довіри до серверної частини.

Таким чином, блокчейн виступає як механізм "якоря довіри" (trust anchor), який фіксує стан даних у певний момент часу [12].

У запропонованій моделі процес забезпечення достовірності даних реалізується як багаторівнева процедура:

1. формування даних на IoT-пристрої;
2. обчислення хешу та створення цифрового підпису;
3. передача даних через захищений канал;
4. перевірка хешу та підпису на серверному рівні;
5. збереження валідованих даних;
6. агрегування та запис контрольного хешу у блокчейн.

Формально процес перевірки можна описати функцією валідації:

$$V(T) = \text{Verify}(PK, H, S) \wedge (H = \text{Hash}(M)) \wedge \text{Valid}(\text{Timestamp})$$

де:

- (M) – повідомлення;
- (H) – отриманий хеш;
- (S) – цифровий підпис;
- (PK) – відкритий ключ пристрою.

У такій моделі довіра формується як результат поєднання криптографічної перевірки та незалежної фіксації даних у блокчейн. Це дозволяє:

- гарантувати автентичність джерела даних;
- забезпечити їх цілісність;
- виключити можливість непомітної модифікації історичних записів;
- реалізувати незалежний аудит.

З огляду на обмежені ресурси IoT-пристроїв, зокрема мікроконтролерів ESP32, виконання операцій консенсусу є недоцільним. Тому у запропонованій

архітектурі пристрої виконують лише базові криптографічні операції (хешування та підпис), тоді як обробка, валідація та взаємодія з блокчейн-мережею реалізуються на серверному рівні.

Нижче наведено узагальнену класифікацію загроз, що враховується під час побудови моделі безпеки.

Таблиця 1.1

Основні категорії загроз IoT у контексті запропонованої моделі

Категорія	Характеристика
Ідентифікаційні	Підміна або клонування пристрою
Комунікаційні	Перехоплення або модифікація даних
Програмні	Компрометація прошивки
Дані	Підміна або викривлення телеметрії
Системні	Порушення роботи серверної частини

Проведений аналіз демонструє, що ефективне забезпечення безпеки IoT потребує інтеграції криптографічних механізмів, серверної валідації та розподіленої фіксації контрольних значень. Саме така комбінована модель дозволяє досягти наскрізної достовірності даних і забезпечити можливість їх перевірки незалежно від довіри до окремих компонентів системи, що й реалізовано у запропонованому підході.

1.3. Криптографічні та мережеві методи захисту IoT-інфраструктури

Забезпечення безпеки IoT-систем є складним багаторівневим завданням, оскільки такі системи поєднують значну кількість гетерогенних пристроїв, мережевих протоколів та програмних компонентів. На відміну від традиційних інформаційних систем, де більшість вузлів має достатні обчислювальні ресурси для реалізації повноцінних механізмів захисту, IoT-пристрої часто працюють у середовищі обмежених ресурсів. Це ускладнює використання стандартних криптографічних протоколів та створює додаткові ризики безпеки.

У більшості сучасних IoT-архітектур захист реалізується через поєднання кількох базових механізмів: криптографічних алгоритмів, інфраструктури керування ключами та мережових протоколів захищеної передачі даних. Такі механізми забезпечують автентифікацію пристроїв, захист конфіденційності інформації та контроль цілісності повідомлень. Водночас їх застосування, як правило, орієнтоване на захист окремих рівнів взаємодії (канал зв'язку, ідентифікація), але не формує цілісної моделі наскрізної довіри до даних у межах всієї системи.

Криптографічні механізми захисту

Основою безпеки IoT-систем є використання криптографічних алгоритмів, що дозволяють захистити дані від несанкціонованого доступу та модифікації. Найбільш поширеними є алгоритми симетричного та асиметричного шифрування, а також криптографічні хеш-функції.

Симетричне шифрування використовується для захисту каналів зв'язку між пристроями та сервером. Його перевагою є висока швидкість обробки даних та невеликі вимоги до обчислювальних ресурсів. Проте використання спільних секретних ключів створює проблему їхнього безпечного розповсюдження та ускладнює масштабування системи у разі збільшення кількості пристроїв.

Асиметрична криптографія дозволяє вирішити проблему розподілу ключів, використовуючи пари відкритого та закритого ключів. У такій моделі кожен пристрій має власний криптографічний ідентифікатор, що забезпечує можливість його автентифікації у мережі. Для ресурсно-обмежених пристроїв, зокрема на базі ESP32, доцільним є використання оптимізованих алгоритмів, що забезпечують баланс між безпекою та продуктивністю.

Криптографічні хеш-функції застосовуються для перевірки цілісності даних. Вони дозволяють сформувати унікальний цифровий відбиток повідомлення, який змінюється навіть при незначній модифікації даних. Однак сам по собі хеш не забезпечує підтвердження джерела даних, тому його ефективність визначається використанням у поєднанні з цифровим підписом.

Таким чином, криптографічні механізми забезпечують базові властивості безпеки, однак їх ефективність залежить від інтеграції у загальну архітектуру системи.

Інфраструктура керування ключами

Для забезпечення автентифікації пристроїв та керування криптографічними ключами у мережах IoT використовується інфраструктура відкритих ключів (PKI). Вона включає сертифікаційні центри, механізми видачі сертифікатів та процедури перевірки їх достовірності [2].

PKI дозволяє створити централізовану модель довіри, однак у масштабних IoT-системах така модель створює додаткові ризики. Компрометація сертифікаційного центру може призвести до порушення довіри до всієї системи. Крім того, підтримка великої кількості сертифікатів потребує значних ресурсів, що ускладнює застосування цього підходу у динамічних середовищах.

У зв'язку з цим у сучасних IoT-архітектурах доцільним є використання спрощених моделей керування ключами, де ідентичність пристрою формується на основі його криптографічного ключа, а перевірка виконується на серверному рівні.

Захищені мережеві протоколи

Для захисту каналів передачі даних у IoT-мережах використовуються протоколи захищеного з'єднання, що реалізують шифрування та автентифікацію під час передачі інформації. Одним із найбільш поширених є протокол TLS, який застосовується разом із MQTT для забезпечення конфіденційності та цілісності мережевого трафіку [8].

Захищені протоколи ефективно протидіють атакам типу перехоплення або модифікації повідомлень на рівні каналу зв'язку. Проте вони не забезпечують перевірку достовірності даних після їх доставки, оскільки не гарантують, що повідомлення не було змінено на етапах подальшої обробки.

Таким чином, TLS та MQTT ACL забезпечують лише базовий рівень захисту, але не формують механізм наскрізної перевірки даних.

Обмеження традиційних методів захисту

Незважаючи на ефективність криптографічних механізмів і захищених протоколів передачі даних, їх використання у IoT-системах має низку обмежень. Основною проблемою є централізований характер керування довірою. У разі

компрометації серверної частини система втрачає можливість перевірки достовірності даних.

Іншою проблемою є відсутність механізмів довгострокової перевірки. Дані, що зберігаються у базі, можуть бути змінені або видалені без можливості незалежної перевірки їх цілісності.

Крім того, навіть за наявності TLS та механізмів доступу, система не гарантує, що дані сформовані саме легітимним пристроєм і не були модифіковані після передачі.

Таким чином, традиційні методи забезпечують захист окремих рівнів системи, але не формують цілісної моделі довіри до даних.

Інтеграція блокчейн-технологій у модель безпеки IoT

Для усунення зазначених обмежень доцільним є використання блокчейн-технологій як механізму незалежної фіксації даних. У сучасних реалізаціях, зокрема на базі EVM-сумісних мереж (наприклад, Polygon), блокчейн використовується як інструмент збереження контрольних значень, а не повного обсягу телеметрії [11].

У такій моделі кожна критично важлива подія не зберігається безпосередньо у блокчейні, а агрегується у вигляді хешу, який записується у смарт-контракт. Це дозволяє:

- зменшити навантаження на мережу;
- підвищити масштабованість системи;
- забезпечити незмінність контрольних даних.

Такий підхід формує модель, у якій блокчейн виконує функцію незалежного механізму перевірки (anchor), що дозволяє підтвердити достовірність даних у будь-який момент часу.

Порівняльний аналіз методів захисту IoT

Для оцінки ефективності різних методів захисту доцільно провести їх порівняльний аналіз.

Таблиця 1.2.

Порівняння методів забезпечення безпеки IoT

Метод захисту	Основний принцип	Переваги	Недоліки	Доцільність використання
Симетричне шифрування	спільний секретний ключ	висока швидкість	складність розподілу ключів	захист каналу
Асиметрична криптографія	пара ключів	автентифікація	навантаження	ідентифікація
PKI	централізована сертифікація	керованість	єдина точка відмови	корпоративні мережі
TLS / DTLS	захищений канал	захист від MITM	не гарантує достовірність	транспорт
Blockchain	фіксація хешів	незмінність, аудит	складність інтеграції	контроль цілісності
Комбінована модель (запропонована)	підпис + хеш + batch + blockchain anchor	наскрізна довіра, аудит	складність реалізації	комплексний захист

Висновок до першого розділу

Аналіз особливостей IoT-мереж показав, що ключовими факторами, які ускладнюють забезпечення належного рівня безпеки, є обмежені обчислювальні ресурси пристроїв, різноманітність апаратних і програмних платформ, використання спрощених протоколів взаємодії, а також відсутність єдиної моделі довіри до даних у межах системи. Особливо критичною є проблема забезпечення достовірності телеметричних даних, які можуть бути змінені як під час передачі, так і на етапах подальшої обробки.

У межах аналізу типових загроз встановлено, що IoT-мережі є вразливими до широкого спектра атак, серед яких особливу небезпеку становлять атаки типу Man-in-the-Middle, підміна ідентичності пристрою (spoofing), повторне відтворення повідомлень (replay), а також модифікація переданих даних. Зазначені загрози можуть призводити до викривлення телеметрії, несанкціонованого доступу до системи та порушення її функціонування. При цьому навіть використання захищених каналів передачі даних не гарантує повної довіри до отриманої інформації.

Огляд сучасних методів і технологій кіберзахисту IoT-пристроїв показав, що традиційні підходи, зокрема використання криптографічних алгоритмів, інфраструктури відкритих ключів та протоколів захищеного з'єднання, забезпечують захист окремих рівнів системи, але не дозволяють реалізувати механізм наскрізної перевірки достовірності даних. Основними обмеженнями таких підходів є централізований характер керування довірою та відсутність можливості незалежного аудиту історичних даних.

У результаті проведеного аналізу обґрунтовано доцільність використання комбінованого підходу до захисту IoT-інфраструктури, який поєднує криптографічні механізми (хешування та цифровий підпис), серверну валідацію даних та використання блокчейн-технологій як інструменту фіксації контрольних значень. Такий підхід дозволяє забезпечити автентичність джерела даних, їх цілісність та можливість перевірки у довгостроковій перспективі.

Таким чином, результати першого розділу формують теоретичне підґрунтя для розроблення моделі захисту IoT-пристроїв, яка забезпечує наскрізну достовірність телеметричних даних шляхом інтеграції криптографічних методів та механізмів розподіленої фіксації, що буде детально розглянуто у наступному розділі роботи.

РОЗДІЛ 2. Методи та моделі захисту IoT-пристроїв на основі блокчейн-технологій

2.1. Криптографічні методи автентифікації у системах захисту IoT

Забезпечення безпеки IoT-інфраструктури значною мірою залежить від використання ефективних криптографічних механізмів, що дозволяють гарантувати достовірність даних, автентичність джерела повідомлення та неможливість несанкціонованої модифікації інформації. У розподілених середовищах із великою кількістю автономних пристроїв ці вимоги стають особливо критичними, оскільки вузли мережі можуть функціонувати у відкритих або частково недовіренних мережах [2]. У контексті проблематики, визначеної у попередньому розділі, ключовим завданням є забезпечення наскрізної криптографічної перевірки телеметричних даних [15] – від моменту їх формування на IoT-пристрої до зберігання та подальшого аудиту. Це означає, що механізми захисту повинні працювати не лише на рівні каналу передачі, але й забезпечувати підтвердження достовірності даних незалежно від довіри до окремих компонентів системи (зокрема MQTT-брокера або серверної інфраструктури).

Особливістю IoT-систем є те, що значна частина пристроїв має обмежені апаратні ресурси. Це означає, що складні криптографічні алгоритми, які широко застосовуються у традиційних інформаційних системах, можуть бути надмірно ресурсомісткими для периферійних вузлів. Зокрема, мікроконтролери на базі ESP32 мають обмежений обсяг оперативної пам'яті та обчислювальної потужності, що потребує оптимального вибору криптографічних алгоритмів [3].

У запропонованій моделі захисту IoT-інфраструктури доцільним є використання комбінації криптографічного хешування та цифрового підпису. Такий підхід дозволяє реалізувати базовий принцип системи – довіра формується на рівні даних, а не інфраструктури їх передачі.

Криптографічне хешування

Одним із базових криптографічних механізмів є використання хеш-функцій. Хеш-функція перетворює довільне повідомлення у фіксовану послідовність бітів –

хеш-значення. Основною властивістю такої функції є те, що навіть незначна зміна вхідних даних призводить до суттєвої зміни результату.

У запропонованій системі використовується алгоритм SHA-256, який формує 256-бітове хеш-значення для будь-якого набору даних. Даний алгоритм широко застосовується у системах інформаційної безпеки та блокчейн-інфраструктурах завдяки високій стійкості до криптографічних атак та відносно низьким обчислювальним витратам [16].

Основні властивості SHA-256:

- детермінованість – однакові вхідні дані завжди формують однаковий хеш;
- стійкість до колізій – ймовірність отримання однакового хешу для різних повідомлень є надзвичайно малою;
- незворотність – відновлення вихідних даних із хеш-значення є обчислювально неможливим. [16]

У контексті запропонованої архітектури хешування виконується безпосередньо на IoT-пристрої, що дозволяє:

- зафіксувати початковий стан даних;
- забезпечити контроль цілісності ще до передачі у мережу;
- використовувати хеш як основу для подальшого підпису та агрегування.

Таким чином, хеш виступає первинним елементом формування довіри до даних.

Цифровий підпис як механізм автентифікації пристроїв

Окрім забезпечення цілісності даних, важливою задачею є підтвердження того, що повідомлення було сформовано саме тим пристроєм, який заявлений як джерело інформації. Для цього використовується механізм цифрового підпису.

Цифровий підпис базується на використанні криптографії з відкритим ключем, де кожен пристрій має пару ключів [2]:

- закритий ключ, який зберігається безпосередньо на пристрої;
- відкритий ключ, який використовується іншими учасниками мережі для перевірки підпису. [17]

У запропонованій моделі саме IoT-пристрій виступає джерелом криптографічної довіри, оскільки тільки він володіє закритим ключем, необхідним для формування підпису.

Процедура створення цифрового підпису включає:

1. формування хеш-значення повідомлення;
2. підпис хешу за допомогою закритого ключа пристрою;
3. передавання повідомлення разом із підписом.

На стороні серверу виконується перевірка:

- повторне обчислення хешу;
- перевірка підпису за відкритим ключем;
- прийняття рішення про валідність даних.

Такий підхід дозволяє забезпечити автентичність навіть у випадку, якщо транспортна інфраструктура (наприклад MQTT-брокер) є недовіреною.

Використання криптографії у блокчейн-орієнтованій моделі безпеки

Криптографічні механізми хешування та цифрового підпису є фундаментальними елементами блокчейн-систем [9]. Проте у запропонованій роботі блокчейн використовується не як основне сховище даних, а як механізм фіксації їх цілісності у часі.

У рамках запропонованої моделі криптографічний процес має наступну послідовність:

1. IoT-пристрій формує телеметричні дані;
2. обчислюється хеш повідомлення (SHA-256);
3. хеш підписується закритим ключем пристрою;
4. дані передаються через MQTT поверх TLS; [7]
5. backend-сервер виконує перевірку хешу та підпису;
6. валідні дані зберігаються у базі даних;
7. після накопичення пакету записів формується агрегований хеш;
8. агрегований хеш записується у блокчейн як якір достовірності (trust anchor).

Такий підхід дозволяє:

- забезпечити перевірку кожного повідомлення;
- гарантувати незмінність історичних даних;
- реалізувати масштабовану модель без перевантаження блокчейн-мережі.

Проблеми використання криптографії у IoT-середовищі

Попри ефективність криптографічних механізмів, їх застосування у IoT-системах має певні обмеження. Основною проблемою є ресурсні обмеження периферійних пристроїв, які можуть ускладнювати виконання складних криптографічних операцій.

Крім того, виникає проблема безпечного зберігання криптографічних ключів. У разі компрометації закритого ключа зломисник отримує можливість формувати валідні підписані повідомлення.

Іншою важливою проблемою є необхідність перевірки великого обсягу даних у реальному часі. Саме тому у запропонованій моделі використовується розподіл функцій:

- IoT-пристрій – формування хешу та підпису;
- сервер – перевірка достовірності;
- блокчейн – фіксація цілісності.

2.2. Модель автентифікації IoT-пристроїв на основі блокчейн-інфраструктури

Одним із ключових завдань забезпечення безпеки IoT-інфраструктури є формування надійної моделі ідентифікації та автентифікації пристроїв, що беруть участь у процесі обміну даними [2]. У розподілених IoT-середовищах велика кількість автономних вузлів взаємодіє між собою через відкриті мережеві канали, що створює додаткові вимоги до механізмів перевірки достовірності джерела інформації. За відсутності ефективних механізмів ідентифікації зломисник може імітувати роботу легітимного пристрою, що призводить до спотворення телеметричних даних або несанкціонованого доступу до системи.

У більшості існуючих IoT-рішень управління ідентичностями реалізується за допомогою централізованих сервісів, які відповідають за реєстрацію пристроїв, зберігання ключів та перевірку автентичності повідомлень. Попри відносну простоту такої моделі, її застосування у великих розподілених системах супроводжується низкою обмежень. Централізовані компоненти можуть створювати вузькі місця у системі, знижувати її масштабованість та підвищувати ризик порушення безпеки у разі компрометації серверної інфраструктури. [14]

Альтернативним підходом є використання розподілених технологій, які дозволяють зберігати інформацію про ідентичності пристроїв у незмінному реєстрі. Однією з таких технологій є блокчейн, що забезпечує децентралізоване зберігання даних та криптографічну перевірку транзакцій. У запропонованій моделі блокчейн використовується як реєстр ідентичностей пристроїв та механізм фіксації довіри, тоді як перевірка автентичності повідомлень виконується на серверному рівні. [10]

Формування ідентичності IoT-пристрою

Основою моделі автентифікації є формування унікальної криптографічної ідентичності для кожного IoT-вузла. У запропонованій архітектурі така ідентичність визначається двома основними параметрами:

- DeviceID – унікальний ідентифікатор пристрою;
- PublicKey – відкритий криптографічний ключ пристрою.

DeviceID може формуватися на основі апаратного ідентифікатора мікроконтролера або генеруватися під час процесу реєстрації. Відкритий ключ використовується для перевірки цифрового підпису повідомлень, сформованих пристроєм.

У рамках даної роботи як периферійний вузол розглядається мікроконтролер ESP32, який має достатні обчислювальні ресурси для виконання базових криптографічних операцій. [3]

Кожен пристрій генерує криптографічну пару ключів:

- приватний ключ SK
- відкритий ключ PK [17]

Приватний ключ зберігається безпосередньо на пристрої та не передається мережею. Таким чином, саме пристрій виступає джерелом криптографічної довіри у системі.

Алгоритм реєстрації IoT-пристрою у блокчейн-інфраструктурі

Для забезпечення можливості перевірки ідентичності пристрою необхідно виконати процедуру його реєстрації у системі. У запропонованій моделі реєстрація виконується через серверний рівень із подальшою фіксацією у блокчейн.

Алгоритм реєстрації:

1. Пристрій генерує криптографічну пару ключів (SK, PK).
2. Формується унікальний ідентифікатор DeviceID.
3. Пристрій передає до backend-сервера дані: {DeviceID, PublicKey}
4. Backend-сервер виконує перевірку коректності отриманих даних.
5. Після валідації сервер формує реєстраційний запис, доповнюючи його часо-вою міткою:
6. $DeviceRecord = \{DeviceID, PublicKey, Timestamp\}$
7. Сформований запис використовується для створення транзакції у блокчейн-мережі.
8. Після підтвердження транзакції запис зберігається у розподіленому реєстрі.

Таким чином формується незмінний реєстр відповідності DeviceID ↔ PublicKey, який використовується для подальшої автентифікації. [11]

Процедура автентифікації пристрою

Після реєстрації пристрій може передавати телеметричні дані до системи. У запропонованій моделі використовується підхід наскрізної перевірки, у якому криптографічна обробка виконується на пристрої, а валідація – на сервері.

Алгоритм автентифікації:

1. Пристрій формує повідомлення M.
2. Обчислюється хеш:

$$H = \text{SHA256}(M) \text{ [16]}$$

3. Хеш підписується:

$$S = \text{Sign}(SK, H)$$

4. Передається пакет:

$$\{DeviceID, M, S\}$$

5. Дані передаються через MQTT поверх TLS.

6. Backend-сервер отримує повідомлення.

7. Отримується відкритий ключ PK із блокчейн-реєстру.

8. Виконується перевірка:

$$Verify(PK, H, S) [17]$$

Якщо перевірка успішна, повідомлення вважається автентичним і може бути збережене.

Схема автентифікації пристрою

Архітектуру взаємодії між компонентами системи можна представити у вигляді наступної схеми:

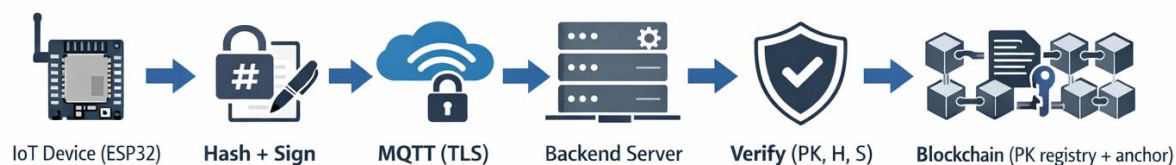


Рисунок 2.1 – Автентифікація пристрою

У даній схемі:

- MQTT-брокер виконує лише транспортну функцію;
- backend є точкою валідації;
- блокчейн використовується як джерело довіри до ключів та механізм аудиту.

Математична модель автентифікації пристрою

Для формалізації процесу введемо:

D – множина пристроїв

PK – множина відкритих ключів

SK – множина закритих ключів

M – повідомлення

H – хеш

Функції:

$$H = \text{Hash}(M)$$

$$S = \text{Sign}(SK, H)$$

$$\text{Auth} = \text{Verify}(PK, H, S)$$

Автентифікація:

$$\text{Auth}(\text{DeviceID}, M, S) \rightarrow \{0,1\}$$

де:

1 – пристрій автентичний

0 – неавтентичний

Додатково вводиться умова:

$PK \in \text{BlockchainRegistry}(\text{DeviceID})$

Це забезпечує зв'язок між ідентичністю та ключем.

Підготовка моделі до аналізу атак

Запропонована модель автентифікації дозволяє протидіяти ряду поширених атак:

- MITM – не може змінити підписані дані;
- Spoofing – неможливо підробити підпис;
- Replay – може бути виявлена через timestamp;
- Firmware attack – виявляється через перевірку хешу.

Ключовою перевагою є те, що перевірка виконується незалежно від транспортної інфраструктури, а історія ідентичностей зберігається у незмінному реєстрі.

2.3. Аналіз загроз та оцінка ефективності запропонованої моделі захисту IoT-пристроїв

Основні типи атак на IoT-інфраструктуру

У розподілених IoT-системах, де пристрої передають телеметричні дані через мережеву інфраструктуру, основні загрози пов'язані з можливістю підміни джерела даних, їх модифікації або повторного відтворення. У контексті запропонованої архітектури, що включає IoT-пристрій, MQTT-брокер, backend-сервер і блокчейн-інфраструктуру, доцільно виділити такі типи атак:

- **MITM (Man-in-the-Middle)** – перехоплення трафіку між IoT-пристроєм та сервером із можливістю його модифікації;
- **Spoofing-атака** – підміна ідентичності пристрою;
- **Replay-атака** – повторне передавання раніше перехоплених повідомлень;
- **Firmware attack** – модифікація програмного забезпечення пристрою;
- **Data tampering** – зміна телеметричних даних під час передачі або зберігання [14].

У запропонованій системі передавання даних здійснюється через MQTT поверх TLS, при цьому MQTT-брокер виконує лише функції маршрутизації повідомлень і контролю доступу (ACL), але не забезпечує криптографічної перевірки їх достовірності. Це означає, що довіра до даних формується на рівні криптографічної обробки та серверної валідації. [7]

Таблиця 2.1.

Основні атаки на IoT-систему

Тип атаки	Опис атаки	Потенційні наслідки
MITM	Перехоплення та модифікація трафіку	Порушення цілісності даних
Spoofing	Підміна ідентичності пристрою	Несанкціонований доступ
Replay	Повторне відтворення повідомлень	Імітація активності
Firmware attack	Зміна прошивки пристрою	Компрометація вузла
Data tampering	Модифікація даних	Викривлення телеметрії

Механізми захисту у запропонованій моделі

Запропонована модель базується на поєднанні таких компонентів:

- хешування даних на пристрої (SHA-256); [16]
- цифровий підпис повідомлення;
- передавання через TLS;
- серверна перевірка підпису та хешу;
- збереження даних у базі;
- формування батчів і запис їх хешів у блокчейн.

Відповідність «атака – механізм захисту»

Тип атаки	Механізм захисту
MITM	TLS + цифровий підпис
Spoofing	DeviceID + PublicKey + підпис
Replay	Timestamp / унікальність повідомлення
Firmware attack	контроль хешу прошивки
Data tampering	SHA-256 + перевірка на сервері

Аналіз ефективності запропонованої моделі

Автентичність пристроїв

У запропонованій моделі кожен IoT-пристрій має унікальну криптографічну ідентичність, що визначається парою значень DeviceID та PublicKey. Приватний ключ використовується для формування цифрового підпису повідомлення, тоді як відкритий ключ зберігається у блокчейн-реєстрі. [2]

Умова автентичності визначається як:

$$Verify(PK, Hash(M), S) = TRUE$$

де PK отримується з реєстру ідентичностей.

Таким чином, підміна пристрою без доступу до приватного ключа є неможливою, що забезпечує ефективний захист від spoofing-атак.

Цілісність даних

Перед передачею даних IoT-пристрій обчислює хеш повідомлення:

$$H = SHA256(M)$$

На backend-сервері виконується повторне обчислення хешу та його порівняння. У разі невідповідності повідомлення відхиляється.

Це забезпечує виявлення будь-якої модифікації даних як під час передачі, так і на етапі обробки.

Стійкість до MITM-атак

Навіть у випадку перехоплення трафіку зломисник не може змінити повідомлення без порушення його хешу та цифрового підпису. Перевірка на сервері виявляє будь-яку невідповідність:

$\text{Verify}(\text{PK}, \text{Hash}(M'), S) = \text{FALSE}$

Таким чином, модифіковані дані не проходять валідацію. [14]

Стійкість до replay-атак

Для протидії replay-атакам у структурі повідомлення використовується часовий параметр або унікальний ідентифікатор повідомлення. Сервер перевіряє актуальність даних та відхиляє повторно передані повідомлення.

Стійкість до firmware-атак

Контроль цілісності програмного забезпечення пристрою може здійснюватися через перевірку хешу прошивки. У разі зміни програмного коду змінюється і результат криптографічного хешування, що дозволяє виявити компрометацію пристрою.

Незмінність даних та аудит

Особливістю запропонованої моделі є використання блокчейн-інфраструктури для фіксації агрегованих даних.

Після накопичення певної кількості записів формується батч:

$$H_{batch} = \text{SHA256}(M_1 || M_2 || \dots || M_n)$$

Отриманий хеш записується у смартконтракт блокчейн-мережі як незмінний якір.

У разі перевірки достовірності даних виконується повторне обчислення хешу батчу та його порівняння із значенням у блокчейні:

$$H'_{batch} = H_{batch}$$

Це забезпечує неможливість непомітної зміни історичних даних.

Порівняльна оцінка ефективності моделі

Таблиця 2.3.

Ефективність захисту

Тип атаки	Рівень захисту	Обґрунтування
MITM	високий	TLS + підпис
Spoofing	високий	РК + підпис
Replay	середній	залежить від timestamp
Firmware attack	середній	залежить від контролю прошивки
Data tampering	високий	хеш + перевірка

Узагальнення результатів

Запропонована модель забезпечує:

- криптографічну автентифікацію кожного повідомлення;
- перевірку цілісності даних на серверному рівні;
- незалежність від MQTT-брокера як довіреного вузла;
- можливість перевірки історичних даних через блокчейн;
- масштабованість за рахунок використання батчів.

Таким чином, поєднання криптографічних механізмів, серверної валідації та блокчейн-якоріння дозволяє реалізувати модель, у якій забезпечується достовірність телеметричних даних IoT-пристроїв на всіх етапах їх обробки – від моменту генерації до довгострокового зберігання.

Висновок до другого розділу

У другому розділі магістерської роботи було розроблено та обґрунтовано модель забезпечення безпеки IoT-пристроїв, яка базується на поєднанні криптографічних механізмів, серверної валідації даних та використання блокчейн-технологій як інструменту фіксації їх цілісності.

У ході дослідження визначено, що використання алгоритму хешування SHA-256 дозволяє забезпечити контроль цілісності телеметричних даних безпосередньо на рівні IoT-пристрою, тоді як цифровий підпис гарантує автентичність джерела

інформації. Такий підхід забезпечує формування первинного рівня довіри ще до передачі даних у мережу, що є критично важливим для розподілених IoT-систем.

Запропонована модель автентифікації передбачає використання унікальної криптографічної ідентичності пристрою, яка формується на основі відкритого ключа. При цьому перевірка достовірності повідомлень виконується на серверному рівні шляхом повторного обчислення хешу та перевірки цифрового підпису. Це дозволяє забезпечити незалежність процесу валідації від мережевої інфраструктури, зокрема MQTT-брокера.

Особливістю розробленого підходу є використання механізму пакетної обробки даних, у межах якого після накопичення визначеної кількості записів формується агрегований хеш. Такий хеш фіксується у блокчейн-мережі як незмінний якір, що забезпечує можливість подальшої перевірки достовірності історичних даних без необхідності їх зберігання у блокчейні.

Проведений аналіз загроз показав, що запропонована модель ефективно протидіє основним типам атак на IoT-інфраструктуру, зокрема підміні ідентичності пристрою, модифікації даних, перехопленню трафіку та replay-атакам. Це досягається за рахунок поєднання криптографічної автентифікації, перевірки цілісності та використання незмінного реєстру контрольних значень.

Таким чином, у другому розділі сформовано комплексну модель захисту IoT-систем, яка забезпечує наскрізну достовірність даних – від моменту їх формування на пристрої до довгострокового зберігання. Отримані результати створюють основу для практичної реалізації системи, що розглядається у наступному розділі роботи.

РОЗДІЛ 3. Моделювання системи захисту IoT-пристроїв на основі криптографічних методів та блокчейн-технологій

3.1. Загальна модель системи забезпечення достовірності телеметричних даних IoT-пристроїв

У сучасних IoT-системах ключовою проблемою є забезпечення достовірності телеметричних даних, що генеруються розподіленими пристроями та передаються через потенційно недовірені мережеві середовища. Традиційні централізовані підходи не дозволяють гарантувати незмінність даних та створюють ризики компрометації у разі порушення роботи центрального вузла. [1]

У даній роботі запропоновано модель, що забезпечує наскрізний контроль достовірності даних шляхом поєднання криптографічної обробки на рівні пристрою, серверної валідації та фіксації контрольних значень у блокчейн-мережі. [9]

Архітектура системи

Система реалізована як багаторівнева структура (рис.3.1), що включає:

- IoT-пристрої (ESP32);
- мережевий рівень передачі даних (MQTT через TLS); [7]
- сервер обробки даних (backend);
- підсистему зберігання;
- блокчейн-рівень.

Принцип функціонування системи

Функціонування системи базується на послідовному проходженні даних через усі рівні архітектури з виконанням перевірок на кожному етапі.

На рівні IoT-пристрою формуються телеметричні дані (M), для яких обчислюється криптографічний хеш:

$$H(M) = \text{SHA256}(M)$$

Після цього хеш підписується приватним ключем пристрою:

$$S = \text{Sign}(SK, H(M))$$

Сформований пакет:

$$\{ID, M, H(M), S\}$$

передається до мережі за допомогою протоколу MQTT через захищений канал TLS.

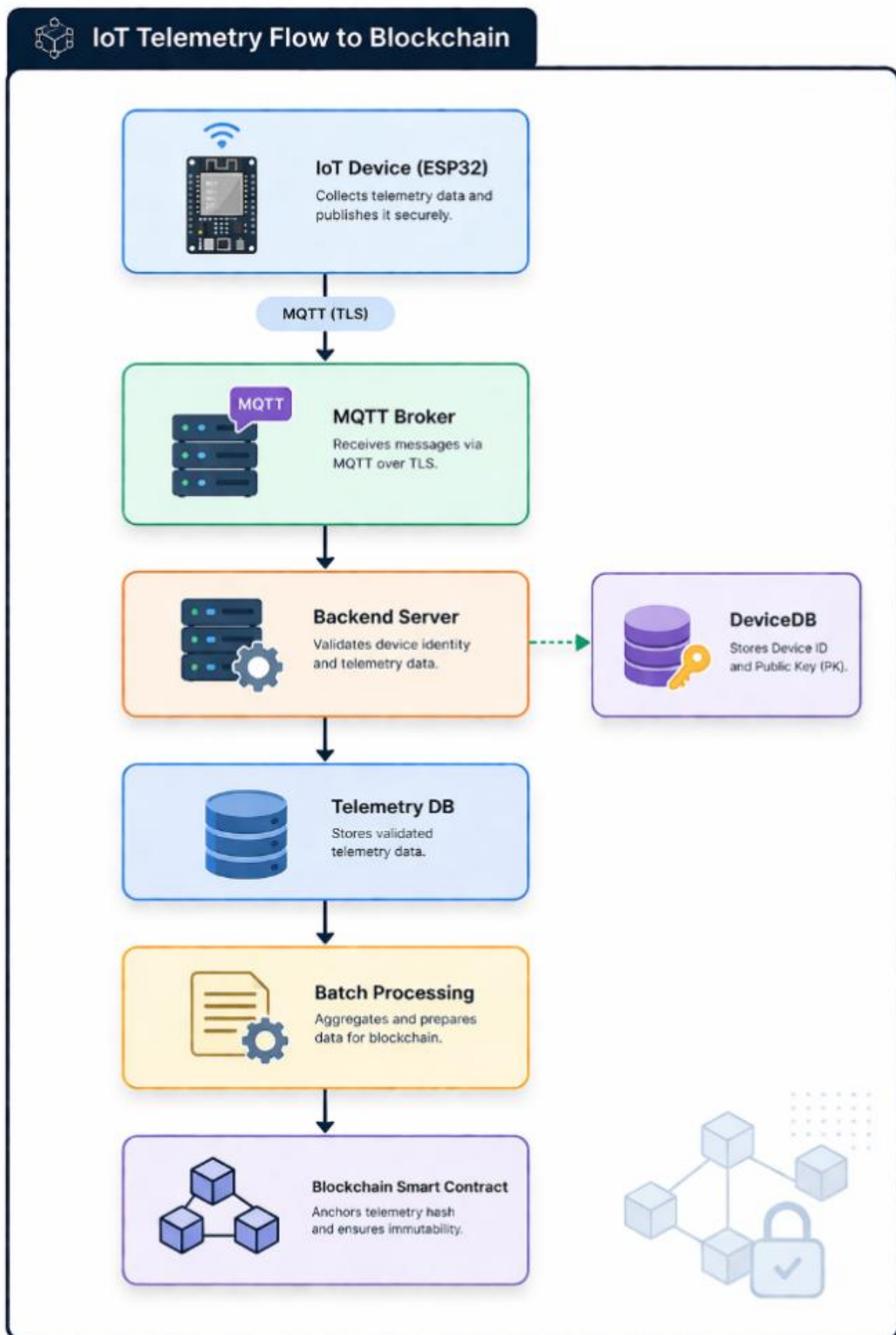


Рисунок 3.1 – Загальна модель системи

MQTT-брокер виконує лише функції маршрутизації повідомлень та контролю доступу, не здійснюючи перевірки їх достовірності. Додатково на цьому рівні формується часова мітка, яка використовується для подальшого аналізу даних. [6]

Основна логіка забезпечення безпеки реалізується на backend-сервері. Тут виконується перевірка цілісності даних шляхом повторного обчислення хешу, а також перевірка цифрового підпису з використанням відкритого ключа пристрою. У разі успішної валідації дані зберігаються у базі телеметрії.

Для оптимізації роботи системи використовується механізм пакетної обробки. Після накопичення визначеної кількості записів формується батч, для якого обчислюється агрегований хеш:

$$H_{batch} = H(R_1 || R_2 || \dots || R_n) [9]$$

Отримане значення записується у блокчейн-мережу за допомогою смарт-контракту. У такій моделі блокчейн використовується як незмінний реєстр контрольних значень, що дозволяє перевіряти достовірність даних у майбутньому.

Модель довіри

Запропонована система базується на принципі мінімальної довіри до окремих компонентів:

- IoT-пристрій є джерелом даних та формує криптографічний доказ їх достовірності;
- MQTT-брокер не вважається довіреним елементом;
- backend-сервер виконує перевірку даних;
- блокчейн забезпечує незмінність контрольних значень.

Такий підхід дозволяє усунути залежність від єдиного центру довіри та підвищити стійкість системи до атак.

3.2. Моделювання даних, потоків та структури зберігання інформації в системі

У запропонованій системі забезпечення достовірності телеметричних даних ключову роль відіграє узгоджена організація інформаційних потоків,

формалізована структура даних та раціональна модель їх зберігання. На відміну від традиційних IoT-рішень, де основна увага приділяється лише передачі та обробці даних, у даній роботі особливий акцент зроблено на забезпеченні їх цілісності та можливості подальшої перевірки достовірності. [15]

Функціонування системи передбачає послідовне перетворення даних – від моменту їх формування на IoT-пристрої до етапу фіксації контрольних значень у блокчейн-мережі. Кожен етап супроводжується зміною структури даних та додаванням нових атрибутів, що забезпечують їх криптографічний захист і можливість валідації.

Формалізація структури даних

На початковому етапі IoT-пристрій формує телеметричне повідомлення, яке можна подати у вигляді:

$$M = \{m_1, m_2, \dots, m_k\}$$

де (m_i) – окремі параметри сенсорних вимірювань (наприклад, температура, вологість тощо).

Для забезпечення цілісності даних виконується обчислення криптографічного хешу:

$$H(M) = SHA256(M)$$

Отримане значення є унікальним цифровим відбитком повідомлення і змінюється навіть при незначній модифікації вхідних даних.

З метою підтвердження автентичності джерела інформації використовується механізм цифрового підпису. Хеш повідомлення підписується приватним ключем пристрою:

$$S = Sign(SK, H(M))$$

Таким чином формується повний пакет даних:

$$Packet = \{ID, M, H(M), S\}$$

де (ID) – ідентифікатор пристрою, що визначається як хеш його відкритого ключа. Сформована структура дозволяє забезпечити як цілісність даних (через ($H(M)$)), так і їх автентичність (через (S)).

Модель потоків даних у системі

Рух даних у системі реалізується як послідовність взаємопов'язаних етапів, на кожному з яких виконується відповідна обробка або перевірка інформації.

На рівні IoT-пристрою формується криптографічно захищене повідомлення, яке передається через мережеву інфраструктуру за допомогою протоколу MQTT. Передача здійснюється через захищений канал TLS, що дозволяє запобігти перехопленню або модифікації даних на транспортному рівні.

Після надходження до MQTT-брокера повідомлення маршрутизується до backend-сервера. На цьому етапі до повідомлення додається часова мітка, що використовується для подальшої перевірки актуальності даних та виявлення повторних передач (replay-атак). [14]

Основна обробка виконується на серверному рівні, де здійснюється перевірка цілісності даних шляхом повторного обчислення хешу, а також перевірка цифрового підпису з використанням відкритого ключа пристрою. У разі успішного проходження обох перевірок повідомлення вважається достовірним і передається до підсистеми зберігання. [17]

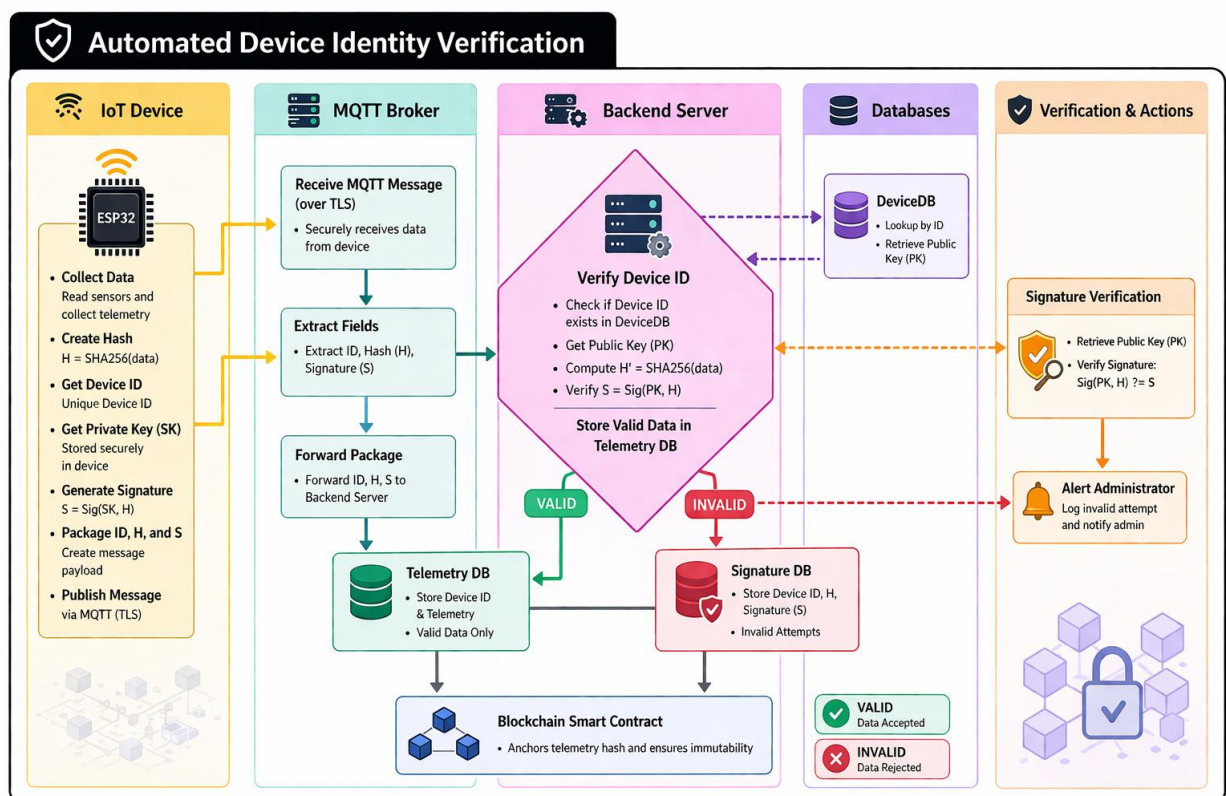


Рисунок 3.2 – Поток даних та трансформація структури повідомлення

Подальший етап передбачає накопичення записів у базі даних телеметрії та формування пакетів (батчів), що використовуються для оптимізації взаємодії з блокчейн-мережею. Для кожного батчу обчислюється агрегований хеш, який виступає контрольним значенням для всієї групи даних.

Організація зберігання даних

У системі використовується розділена модель зберігання, яка передбачає окреме збереження оперативних даних та контрольних значень.

На серверному рівні реалізовано дві основні бази даних. Перша з них – реєстр пристроїв – містить інформацію про ідентифікатори та відповідні їм відкриті ключі. Ця база використовується для забезпечення автентифікації повідомлень.

Друга база – телеметрична – призначена для зберігання перевірених даних. Кожен запис у цій базі включає ідентифікатор пристрою, отримані дані, часову мітку та хеш повідомлення. Така структура дозволяє у будь-який момент перевірити цілісність збереженої інформації.

З метою зменшення навантаження на блокчейн-мережу використовується підхід пакетної обробки даних. Після накопичення визначеної кількості записів формується батч:

$$Batch = \{R_1, R_2, \dots, R_n\}$$

Для якого обчислюється агрегований хеш:

$$H_{\{batch\}} = H(R_1 || R_2 || \dots || R_n)$$

Отримане значення записується у блокчейн і виступає як незмінний якір для відповідного набору даних.

Математична модель перевірки даних

Перевірка достовірності повідомлення здійснюється за умовою:

$$Valid = \left(H(M)_{\{calc\}} = H(M) \right) \wedge Verify(PK, H(M), S)$$

де:

- перша складова забезпечує цілісність;
- друга – автентичність.

У свою чергу, перевірка цілісності батчу визначається як:

$$\begin{aligned} &Integrity_{\{batch\}} = \\ &\begin{cases} 1, & H'\{batch\} = H\{blockchain\} \setminus \\ 0, & \text{інакше} \end{cases} \\ &\end{cases} \end{aligned}$$

3.3. Реалізація системи забезпечення достовірності телеметричних даних IoT-пристроїв

З метою детального обґрунтування запропонованої архітектури системи доцільно виконати її функціональну декомпозицію на окремі компоненти. Такий підхід дозволяє не лише визначити роль кожного елемента, але й формалізувати процеси обробки даних, структуру їх зберігання та механізми забезпечення безпеки на кожному рівні. [15]

У межах розробленої моделі виділено чотири ключові компоненти: IoT-пристрій, мережевий рівень передачі даних (MQTT), серверний рівень обробки (backend) та блокчейн-інфраструктура. Кожен із цих елементів виконує окрему функцію в загальному процесі забезпечення автентичності та цілісності телеметричних даних.

IoT-пристрій (ESP32) як джерело довірених даних

IoT-пристрій є початковим елементом системи, на якому формується первинна інформація. У контексті даної роботи пристрій не лише генерує телеметричні дані, але й виконує базові криптографічні операції, що забезпечують їхню цілісність та автентичність.

На цьому рівні реалізується наступна послідовність дій:

1. формування телеметричного повідомлення;
2. обчислення хешу повідомлення;
3. створення цифрового підпису.

Формально це можна описати як:

$$H = \text{SHA256}(M)$$

$$S = \text{Sign}(SK, H)$$

У результаті формується структура повідомлення:

$$\text{Packet} = \{ID, M, H, S\}$$

де:

- (M) – телеметричні дані;
- (H) – хеш повідомлення;
- (S) – цифровий підпис;
- (ID) – ідентифікатор пристрою.



Рисунок 3.3 – Обробка даних на рівні IoT-пристрою

MQTT-брокер як транзитний рівень передачі даних

Передача даних у системі реалізується за допомогою протоколу MQTT, який забезпечує ефективний обмін повідомленнями у мережах із обмеженими ресурсами. Важливою особливістю є те, що MQTT-брокер не виконує перевірку достовірності даних, а лише забезпечує їх доставку.

Для підвищення безпеки використовується TLS, який захищає канал передачі. Крім того, брокер застосовує механізм ACL для контролю доступу до топіків.

На цьому етапі до повідомлення додається службова інформація, зокрема часова мітка:

$$\text{Packet}' = \{ID, M, H, S, \text{Timestamp}\}$$



Рисунок 3.4 – Обробка повідомлення на рівні MQTT

Backend-сервер як центр перевірки та зберігання даних

Серверний рівень є ключовим елементом системи, оскільки саме тут виконується перевірка достовірності отриманих даних. Архітектурно backend реалізовано як два логічно розділені сервіси, що працюють над одними базами даних.

1. Сервіс керування ідентичністю

Цей сервіс відповідає за реєстрацію пристроїв і формування їхніх ідентифікаторів. Ідентифікатор обчислюється як хеш відкритого ключа:

$$ID = H(PK)$$

Перед записом у базу даних виконується валідація:

- перевірка формату ключа;
- перевірка унікальності;
- перевірка структури запиту.

2. Сервіс обробки телеметрії

Основною функцією цього сервісу є перевірка отриманих повідомлень.

Перевірка виконується у два етапи:

1. Перевірка цілісності:

$$H_{\{calc\}} = SHA256(M)$$

$$H_{\{calc\}} == H$$

2. Перевірка автентичності:

$$Verify(PK, H, S)$$

Лише у разі виконання обох умов повідомлення вважається валідним.

3. Структура бази TelemetryDB



Рисунок 3.5 – Логіка backend-обробки

Блокчейн як механізм забезпечення незмінності даних

Блокчейн у запропонованій системі використовується як механізм забезпечення довготривалої цілісності даних. Замість зберігання всіх телеметричних записів у блокчейні використовується підхід агрегування.

Після накопичення (n) записів формується батч:

$$Batch = \{R_1, R_2, \dots, R_n\}$$

Для нього обчислюється агрегований хеш:

$$H_{\{batch\}} = SHA256(R_1 || R_2 || \dots || R_n)$$



Рисунок 3.6 – Формування та запис батчу

У межах даного дослідження було реалізовано програмний сервіс, призначений для забезпечення достовірності телеметричних даних, що надходять від IoT-пристроїв у розподіленому середовищі. Запропоноване рішення базується на інтеграції криптографічних методів, протоколу MQTT та блокчейн-технології, що дозволяє сформувати багаторівневу модель довіри.

Особливістю підходу є те, що перевірка достовірності виконується не лише на рівні цифрового підпису, але й підкріплюється використанням блокчейн як незалежного та незмінного джерела істини. При цьому самі телеметричні дані не зберігаються у блокчейн, що забезпечує масштабованість системи, а фіксуються лише їх контрольні хеші.

Ключовим елементом системи виступає смарт-контракт, який реалізує функції децентралізованого реєстру. Він забезпечує збереження відповідності між ідентифікаторами пристроїв та їх відкритими ключами, а також зберігає контрольні хеші пакетів телеметричних даних.

Таким чином, смарт-контракт виконує роль глобального джерела довіри [11], яке не залежить від окремого сервера та гарантує незмінність збережених даних. Основна логіка обробки даних реалізована у вигляді сервісу перевірки, який функціонує як MQTT-клієнт. Він виконує збір сертифікатів (відкритих ключів), перевірку їх достовірності та валідацію телеметричних повідомлень, підписаних приватним ключем пристрою.

Для інтеграції з блокчейн у роботі використовується бібліотека Web3.py, яка дозволяє взаємодіяти зі смарт-контрактом безпосередньо з Python-середовища. У межах реалізації змодельовано отримання відкритого ключа з блокчейн та запис контрольного хешу:

```
# Отримання ключа з блокчейн
device_key = contract.functions.getDeviceKey(DEVICE_ID).call()

# Запис хешу пакетних даних
tx = contract.functions.storeBatchHash(batch_id, batch_hash).transact({
    'from': account
})
```

З метою оптимізації взаємодії з блокчейн у системі використовується підхід пакетної обробки даних. Валідні записи накопичуються, після чого для них обчислюється агрегований хеш, який і записується у блокчейн. Це дозволяє значно зменшити кількість транзакцій та підвищити ефективність системи.

У межах дослідження також було змодельовано IoT-пристрій, який генерує телеметричні дані, підписує їх приватним ключем та передає через MQTT-брокер. Це дозволило відтворити повний цикл взаємодії у контрольованому середовищі та перевірити працездатність запропонованого підходу.

Таким чином, реалізований сервіс поєднує механізми криптографічної перевірки, передачі даних через MQTT та децентралізованого збереження контрольних значень у блокчейн. У результаті формується система, здатна забезпечити високий рівень довіри до телеметричних даних навіть у недовіреному середовищі, що підтверджує доцільність використання запропонованого підходу у сучасних IoT-рішеннях.

Висновок до третього розділу

У третьому розділі було розроблено та реалізовано систему забезпечення достовірності телеметричних даних IoT-пристроїв, яка базується на поєднанні криптографічних методів, протоколу MQTT та блокчейн-технологій. На відміну від традиційних підходів, запропоноване рішення орієнтоване на побудову розподіленої моделі довіри, що не залежить від централізованих компонентів системи.

У процесі дослідження було сформовано загальну модель системи, яка охоплює всі етапи обробки даних – від їх генерації на рівні IoT-пристрою до фіксації контрольних значень у блокчейн. Детально проаналізовано структуру даних, їх трансформацію в межах системи, а також механізми забезпечення цілісності та автентичності на кожному етапі передачі.

Особливу увагу приділено реалізації серверного рівня, який виконує ключову функцію перевірки достовірності телеметричних даних. Розроблений сервіс забезпечує збір сертифікатів, їх валідацію, а також перевірку цифрових підписів

повідомлень, що дозволяє виявляти підроблені або скомпрометовані дані. Використання механізму ланцюга довіри для перевірки відкритих ключів підвищує стійкість системи до атак підміни.

Важливим результатом є інтеграція блокчейн-технології, яка використовується як децентралізований механізм забезпечення незмінності даних. Запропоновано підхід до збереження агрегованих хешів пакетів телеметрії у смарт-контракті, що дозволяє значно зменшити навантаження на мережу та забезпечити ефективність роботи системи. Реалізований смарт-контракт виконує функції реєстру відкритих ключів пристроїв та сховища контрольних значень, що формує незалежне джерело довіри.

У результаті реалізації підтверджено, що поєднання криптографічних методів із блокчейн-технологіями дозволяє створити ефективну систему захисту IoT-інфраструктури, яка забезпечує як автентичність джерела даних, так і їх цілісність у довгостроковій перспективі. Запропонований підхід може бути використаний як основа для побудови масштабованих і захищених систем збору та обробки телеметричних даних у сучасних розподілених середовищах.

ВИСНОВКИ

У межах даної магістерської роботи було досягнуто поставленої мети – розроблено та реалізовано систему забезпечення достовірності телеметричних даних IoT-пристроїв на основі криптографічних методів та блокчейн-технологій з урахуванням особливостей розподілених середовищ і обмежених ресурсів периферійних пристроїв.

У ході дослідження було проаналізовано основні загрози безпеці IoT-систем, зокрема атаки типу Man-in-the-Middle, підміну ідентичності пристрою (spoofing), повторне відтворення повідомлень (replay), а також модифікацію телеметричних даних. Проведений аналіз дозволив сформулювати вимоги до системи захисту, серед яких ключовими є забезпечення автентичності джерела даних, їх цілісності та можливості незалежної перевірки у довгостроковій перспективі.

Розроблено модель системи, у якій реалізовано наскрізний контроль достовірності даних – від моменту їх формування на IoT-пристрої до фіксації контрольних значень у блокчейн-мережі. На рівні пристрою реалізовано механізм криптографічного хешування (SHA-256) та цифрового підпису, що забезпечує формування довіри до даних ще до їх передачі у мережу.

Передача даних у системі здійснюється за допомогою протоколу MQTT, який використовується як транспортний рівень без покладання на нього функцій довіри. Основна логіка перевірки реалізована на серверному рівні, де виконується валідація цифрового підпису та перевірка цілісності повідомлень. Додатково впроваджено механізм перевірки відкритих ключів на основі ланцюга довіри, що дозволяє виявляти спроби їх підміни.

У межах роботи реалізовано програмний сервіс, який забезпечує збір сертифікатів, їх перевірку та валідацію телеметричних даних у реальному часі. Результати перевірки розділяються на валідні та скомпрометовані, що дозволяє здійснювати подальший аналіз та аудит даних.

Для забезпечення незмінності інформації впроваджено блокчейн-рівень, у межах якого використовується смарт-контракт для збереження відповідності між ідентифікаторами пристроїв та їх відкритими ключами, а також для фіксації

агрегованих хешів пакетів телеметричних даних. Застосування механізму пакетної обробки дозволило значно зменшити кількість транзакцій у блокчейн-мережі та підвищити ефективність системи.

Проведене моделювання роботи системи підтвердило коректність реалізованих механізмів, зокрема перевірки цифрових підписів, виявлення підроблених даних та забезпечення незмінності контрольних значень. Отримані результати свідчать про високу ефективність запропонованого підходу у протидії основним типам атак на IoT-інфраструктуру.

Таким чином, розроблена система забезпечує формування децентралізованої моделі довіри, у якій достовірність телеметричних даних гарантується за рахунок поєднання криптографічних методів, серверної валідації та блокчейн-якоріння. Запропонований підхід може бути використаний при побудові масштабованих та захищених IoT-систем, де критично важливою є гарантія цілісності та автентичності даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Tanenbaum A., Wetherall D. *Computer Networks*. – 5th ed. – Pearson, 2011.
2. Stallings W. *Cryptography and Network Security: Principles and Practice*. – 7th ed. – Pearson, 2017.
3. Kahn J. M., Katz R. H., Pister K. S. J. Next century challenges: Mobile networking for smart dust // *Proceedings of ACM MobiCom*. – 1999.
4. Atzori L., Iera A., Morabito G. The Internet of Things: A survey // *Computer Networks*. – 2010.
5. MQTT Version 3.1.1. OASIS Standard. – 2014.
6. Banks A., Gupta R. *MQTT Version 5.0*. OASIS Standard. – 2019.
7. Dierks T., Rescorla E. *The Transport Layer Security (TLS) Protocol Version 1.2*. – RFC 5246.
8. Rescorla E. *The Transport Layer Security (TLS) Protocol Version 1.3*. – RFC 8446.
9. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. – 2008.
10. Wood G. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. – 2014.
11. Buterin V. A Next-Generation Smart Contract and Decentralized Application Platform. – 2013.
12. Antonopoulos A. M. *Mastering Ethereum*. – O'Reilly Media, 2018.
13. Bonneau J. et al. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies // *IEEE Symposium on Security and Privacy*. – 2015.
14. Singh K., Kapoor D. A survey of IoT security: Issues, challenges, and solutions // *Journal of Network and Computer Applications*. – 2017.
15. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. Security, privacy and trust in Internet of Things // *Computer Networks*. – 2015.
16. Eastlake D., Hansen T. *US Secure Hash Algorithm 2 (SHA-256)* – RFC 6234.
17. Rivest R., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems // *Communications of the ACM*. – 1978.
18. Ferguson N., Schneier B. *Practical Cryptography*. – Wiley, 2003.

19. Web3.py Documentation – <https://web3py.readthedocs.io/>
20. Solidity Documentation – <https://docs.soliditylang.org/>
21. Dorri A., Kanhere S. S., Jurdak R. Blockchain in internet of things: challenges and solutions. IEEE Communications Magazine. 2017. Vol. 55, no. 12. P. 119–125.
22. Christidis K., Devetsikiotis M. Blockchains and Smart Contracts for the Internet of Things. IEEE Access. 2016. Vol. 4. P. 2292–2303.
23. Reyna A., Martín C., Chen J., Soler E., Díaz M. On blockchain and its integration with IoT. Challenges and opportunities. Future Generation Computer Systems. 2018. Vol. 88. P. 173–190
24. Minoli D., Occhiogrosso B. Blockchain mechanisms for IoT security. Internet of Things. 2018. Vol. 1-2. P. 1–13.
25. ETSI EN 303 645. Cyber Security for Consumer Internet of Things: Base Requirements. V2.1.1. 2020.
26. NIST Special Publication 800-183. Networks of 'Things'. Gaithersburg : National Institute of Standards and Technology, 2016.
27. Makhdoom I., Abolhasan M., Abbas H., Ni W. Blockchain's adoption in IoT: The next step towards personalized and sustainable ecosystem. Journal of Network and Computer Applications. 2019. Vol. 129. P. 5–26.
28. ISO/IEC 20924:2024. Internet of Things (IoT) – Vocabulary. (Міжнародний стандарт, що визначає термінологію в галузі IoT).

ДОДАТКИ

Смарт-контракт децентралізованого реєстру IoT-пристроїв та хешів телеметричних даних.

```
pragma solidity ^0.8.0;

contract IoTRegistry {

    // Відповідність DeviceID → відкритий ключ (у вигляді адреси)
    mapping(string => address) public deviceKeys;

    // Збереження хешів пакетів телеметричних даних
    mapping(uint256 => bytes32) public batchHashes;

    // Реєстрація пристрою у системі
    function registerDevice(string memory deviceId, address pubKey) public {
        deviceKeys[deviceId] = pubKey;
    }

    // Отримання відкритого ключа пристрою
    function getDeviceKey(string memory deviceId) public view returns(address) {
        return deviceKeys[deviceId];
    }

    // Збереження хешу пакету даних
    function storeBatchHash(uint256 batchId, bytes32 hash) public {
        batchHashes[batchId] = hash;
    }

    // Отримання хешу пакету
    function getBatchHash(uint256 batchId) public view returns(bytes32) {
        return batchHashes[batchId];
    }
}
```

Програмна реалізація сервісу валідації телеметричних даних IoT із перевіркою цифрового підпису та керуванням довіреними ключами

```
import json
import base64
import csv
import os
import paho.mqtt.client as mqtt
from cryptography.hazmat.primitives import serialization, hashes
from cryptography.hazmat.primitives.asymmetric import padding

# Параметри MQTT-брокера
BROKER = "broker.mqtt.cool"
PORT = 1883

# Ідентифікатор пристрою
DEVICE_ID = "id_100"

# Поточний довірений відкритий ключ пристрою
public_key = None

def save_to_csv(filename, value, timestamp, signature):
    """
    Збереження результатів перевірки у файл.
    good.csv – валідні дані
    bad.csv – підроблені або скомпрометовані
    """
    file_exists = os.path.exists(filename)

    with open(filename, "a", newline="") as f:
        writer = csv.writer(f)

        # Додаємо заголовок при першому записі
        if not file_exists:
            writer.writerow(["timestamp", "value", "signature"])

        writer.writerow([timestamp, value, signature])
```

```

def load_last_good():
    """
    Завантаження останнього валідного запису.
    Використовується для побудови ланцюга довіри до ключа.
    """
    if not os.path.exists("good.csv"):
        return None

    with open("good.csv", "r") as f:
        rows = f.readlines()

        if len(rows) <= 1:
            return None

        # Беремо останній запис
        last = rows[-1].strip().split(",")
        return last

def validate_new_key(new_public_key):
    """
    Перевірка нового відкритого ключа.
    Ідея: новий ключ має підтвердити підпис попередніх валідних даних.
    Це дозволяє виявити підміну ключа.
    """
    last = load_last_good()

    # Якщо історії немає – довіряємо першому ключу
    if last is None:
        return True

    timestamp, value, signature_b64 = last

    value = float(value)
    value_str = f"{value:.2f}"

    # Декодуємо підпис
    signature = base64.b64decode(signature_b64)

    try:
        # Перевіряємо, чи новий ключ може підтвердити старі дані

```

```

        new_public_key.verify(
            signature,
            value_str.encode(),
            padding.PKCS1v15(),
            hashes.SHA256()
        )
        return True

except Exception:
    # Якщо перевірка не пройдена – ключ підозрілий
    return False

def on_message(client, userdata, msg):
    """
    Основний обробник MQTT-повідомлень.
    В залежності від топіку:
    – приймає сертифікат (ключ)
    – або перевіряє телеметричні дані
    """
    global public_key

    # Отримання відкритого ключа (сертифіката)
    if msg.topic == f"KI-22-1/{DEVICE_ID}/CA":
        new_key = serialization.load_pem_public_key(msg.payload)

        # Перевірка ключа
        if validate_new_key(new_key):
            public_key = new_key
        else:
            # Запит на повторну публікацію ключа
            client.publish(f"KI-22-1/{DEVICE_ID}/key_request", "republish")

    # Отримання телеметричних даних
    elif msg.topic == f"KI-22-1/{DEVICE_ID}/data":

        # Якщо ключ ще не отримано – ігноруємо
        if public_key is None:
            return

        data = json.loads(msg.payload.decode())

```

```

value = data["value"]
timestamp = data["timestamp"]
signature_b64 = data["signature"]

value_str = f"{value:.2f}"
signature = base64.b64decode(signature_b64)

try:
    # Перевірка цифрового підпису
    public_key.verify(
        signature,
        value_str.encode(),
        padding.PKCS1v15(),
        hashes.SHA256()
    )

    # Дані валідні
    save_to_csv("good.csv", value, timestamp, signature_b64)

except Exception:
    # Дані підроблені або пошкоджені
    save_to_csv("ban.csv", value, timestamp, signature_b64)

# Ініціалізація MQTT-клієнта
client = mqtt.Client()

# Прив'язка обробника повідомлень
client.on_message = on_message

# Підключення до брокера
client.connect(BROKER, PORT, 60)

# Підписка на топіки
client.subscribe(f"KI-22-1/{DEVICE_ID}/CA")
client.subscribe(f"KI-22-1/{DEVICE_ID}/data")

# Запуск циклу обробки повідомлень
client.loop_forever()

```