МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ГОНЧАРОВ Юрій Вікторович

Метод збереження конфіденційності в Інтернеті речей / Method for ensuring privacy in the Internet of Things

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КНм-21
Ю. В. Гончаров
_____

Науковий керівник:
к.т.н., доцент О.Р. Осолінський
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
В.о. завідувача кафедри
_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ГОНЧАРОВ Юрій Вікторович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Метод збереження конфіденційності в Інтернеті речей / Method for ensuring privacy in the Internet of Things
керівник роботи к.т.н., доцент О.Р. Осолінський
затверджені наказом по університету від 20 грудня 2024 року № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2025 р.

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

4. Основні питання, які потрібно розробити
– аналіз архітектурнних моделей систем Інтернету речей;
– поняття конфіденційності та загрози безпеці в IoT-середовищі;
– сучасні методи збереження конфіденційності та приватності в IoT-системах;
– вибір перспективного шляху і постановка задачі дослідження;
– формальна модель IoT-системи та моделі загроз конфіденційності;
– концепція та архітектура запропонованого методу збереження конфіденційності в IoT-системах;
– основні алгоритми функціонування;
– вибір програмно-апаратної платформи та інструментальних засобів;
– реалізація програмних модулів;
– сценарії оцінки методу.

5. Перелік графічного матеріалу у роботі
– Загальна архітектури системи;
– Блок-схема загального алгоритму роботи всіх модулів.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 20.05.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 28.10. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 11.11.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 25.11.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту.	до 1.12.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 04.12.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.12. 2025 р.	

Студент _____ Ю. В. Гончаров
підпис

Керівник роботи _____ к.т.н., доцент О.Р.Осолінський
підпис

РЕЗЮМЕ

Кваліфікаційна робота на тему «Метод збереження конфіденційності в Інтернеті речей» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 80 сторінки і містить 23 ілюстрації, 4 додатки та 34 використаних джерел.

Метою даної кваліфікаційної роботи є розроблення методу збереження конфіденційності в Інтернеті речей.

Методи досліджень: засоби та методи інформаційної безпеки в IoT, математичне моделювання потоків даних і політик доступу, проєктування програмно-апаратних систем.

Результати дослідження: розроблено метод збереження конфіденційності даних в IoT-системах на основі формальної моделі потоків даних та атрибутивного керування доступом, реалізовано трикомпонентну архітектуру, створено експериментальний стенд на базі ESP32 та Raspberry Pi з протоколом MQTT і проведено кількісну оцінку запропонованого методу за показниками затримки, пропускної здатності, завантаження ресурсів і ймовірності успішної атаки.

Результати роботи можуть застосовуватись для побудови систем моніторингу та керування, що забезпечує конфіденційність телеметрії та стійкість до типових атак на трафік.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, КОНФІДЕНЦІЙНІСТЬ ДАНИХ, ШИФРУВАННЯ, АТРИБУТИВНЕ КЕРУВАННЯ ДОСТУПОМ, MQTT, ESP32, RASPBERRY PI, БЕЗПЕКА IOT, ПОТОКИ ДАНИХ, ЕКСПЕРИМЕНТАЛЬНИЙ СТЕНД.

ABSTRACT

Qualification work on the topic «Method for ensuring privacy in the Internet of Things» for Master's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 80 pages and it contains 23 figures, 4 annexes and 34 references.

The aim of this thesis is to develop a method for preserving confidentiality in the Internet of Things.

Research methods: tools and methods of information security in IoT, mathematical modelling of data flows and access policies, design of hardware–software systems.

Research results: a method for preserving data confidentiality in IoT systems has been developed on the basis of a formal model of data flows and attribute-based access control; a three-component architecture has been implemented; an experimental testbed based on ESP32 and Raspberry Pi with the MQTT protocol has been created; and a quantitative evaluation of the proposed method has been carried out in terms of latency, throughput, resource load and probability of a successful attack.

The results of the thesis can be used for building monitoring and control systems that ensure telemetry confidentiality and resilience to typical traffic attacks.

Keywords: INTERNET OF THINGS, DATA CONFIDENTIALITY, ENCRYPTION, ATTRIBUTE-BASED ACCESS CONTROL, MQTT, ESP32, RASPBERRY PI, IOT SECURITY, DATA FLOWS, EXPERIMENTAL TESTBED.

ЗМІСТ

Вступ 7
1 Теоретичні основи забезпечення конфіденційності даних в системах інтернету речей 10
1.1 Поняття та архітектурні моделі систем інтернету речей 10
1.2 Поняття конфіденційності та загрози безпеці в IoT-середовищі 14
1.3 Сучасні методи збереження конфіденційності та приватності в IoT-системах 19
1.4 Вибір перспективного шляху і постановка задачі дослідження 21
Висновки до розділу 1 23
2 Метод забезпечення приватності даних в IoT-системах 24
2.1 Формальна модель IoT-системи та моделі загроз конфіденційності 24
2.2 Концепція та архітектура запропонованого методу збереження конфіденційності в IoT-системах 30
2.3 Основні алгоритми функціонування 34
Висновки до розділу 2 40
3 Програмна реалізація та експериментальна оцінка методу збереження конфіденційності в IoT-системах 42
3.1 Вибір програмно-апаратної платформи та інструментальних засобів 42
3.2 Реалізація програмних модулів 46
3.3 Сценарії оцінки методу 51
Висновки до розділу 3 55
Висновки 56
Список використаних джерел 57
Додаток А Загальна архітектури системи 61
Додаток Б Загальний алгоритм роботи всіх модулів 62
Додаток В Код програми модуля підготовки та передавання захищених повідомлень IoT-пристрою 63
Додаток Г Апробація отриманих результатів 66

ВСТУП

Інтернет речей стає базовою технологією для розумного дому, промислової автоматизації, медицини, енергетики та міської інфраструктури [5, 20, 24, 25, 31]. В таких системах сенсори та вбудовані пристрої постійно збирають і передають великі обсяги даних про стан середовища, обладнання та поведінку користувачів. Значна частина цієї інформації є чутливою. Порушення конфіденційності таких даних може призвести до фінансових, репутаційних втрат, а також загрози безпеці людей та критичної інфраструктури [5, 13, 20–22, 26].

Особливість IoT-платформ полягає у поєднанні великої кількості ресурсно обмежених пристроїв, різнорідних протоколів зв’язку та хмарних сервісів для зберігання і аналітики. Дані проходять шлях від сенсорів через шлюзи та периферійні вузли до серверів і прикладних застосунків. На кожному з цих етапів можливі перехоплення, модифікація чи несанкціонований доступ. Тому завдання забезпечення конфіденційності не зводиться до шифрування одного каналу зв’язку, а потребує узгодженого підходу, який охоплює всі рівні архітектури та весь життєвий цикл даних [2, 5, 10, 24, 25, 31].

Більшість існуючих рішень зосереджені або на протоколах взаємної автентифікації та керування ключами, або на окремих аспектах зберігання та аналітики [1, 7, 11, 12, 16, 22, 24]. Часто передбачається наявність централізованих сховищ ключів та, що в реальних розподілених IoT-середовищах створює додаткові точки відмови і потенційні компрометації [11, 20, 22].

З огляду на це актуальною є задача розроблення методу збереження конфіденційності даних в IoT-системах, який би поєднував формальну модель потоків даних, політики доступу та практичні механізми шифрування й авторизації, придатні для впровадження на ресурсно обмежених пристроях та у багаторівневих архітектурах [2, 5, 10, 24, 31]. Такий підхід має забезпечувати

зниження ризику витоку конфіденційної інформації без критичного погіршення продуктивності системи.

Метою роботи є розроблення методу збереження конфіденційності в Інтернеті речей. Для досягнення поставленої мети необхідно розв’язати такі основні завдання:

1. Проаналізувати архітектурні моделі систем Інтернету речей та класи загроз конфіденційності в IoT-середовищі.
2. Узагальнити сучасні методи збереження конфіденційності та приватності в IoT-системах, виявити їх переваги й обмеження для реальних сценаріїв.
3. Побудувати формальну модель IoT-системи, яка описує множини пристроїв, користувачів, сервісів, типів даних, каналів зв’язку, ключів та політик доступу.
4. Розробити концепцію й архітектуру методу збереження конфіденційності.
5. Сформувати алгоритми реалізації методу.
6. Обрати програмно-апаратну платформу для експериментів.
7. Розробити методику експериментальної оцінки, визначити сценарії навантаження та набір метрик продуктивності.
8. Провести експериментальні дослідження.

Об’єктом дослідження є процеси збирання, передавання, зберігання та доступу до даних в багаторівневих системах Інтернету речей.

Предметом дослідження є методи та моделі забезпечення конфіденційності даних в IoT-системах.

Методи дослідження включають аналіз наукових джерел з інформаційної безпеки та IoT, формалізацію та математичне моделювання потоків даних і політик доступу, проєктування програмно-апаратних систем.

Практичне значення роботи полягає в можливості застосування розробленого методу та прототипу IoT-платформи для побудови реальних

систем моніторингу та керування, де важливими є конфіденційність телеметрії та стійкість до типових атак на трафік.

Структура та обсяг роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися на X Міжнародній студентській конференції «Теоретичне та практичне застосування результатів сучасної науки» м. Запоріжжя, Україна та V Міжнародній науковій конференції «Технології та суспільство: взаємодія, вплив, трансформація», м. Кропивницький, Україна

1 ТЕОРЕТИЧНІ ОСНОВИ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Поняття та архітектурні моделі систем Інтернету речей

Інтернет речей (Internet of Things, IoT) у сучасних дослідженнях розглядається як сукупність фізичних об’єктів, обладнаних вбудованими засобами обчислення, комунікації та ідентифікації, які здатні збирати дані з навколишнього середовища, обмінюватися ними через мережу та взаємодіяти з іншими об’єктами й сервісами без безпосередньої участі людини [5, 13, 20, 22, 25]. Такі об’єкти називають “розумними” речами або IoT-пристроями. До них відносять датчики температури, вологості, освітленості, медичні сенсори, промислові контролери, інтелектуальні лічильники, побутову техніку з модулем зв’язку, транспортні засоби, елементи міської інфраструктури тощо [5, 20, 24, 25] .

Особливістю Інтернету речей є масштабність і різнорідність середовища: в одному рішенні можуть одночасно функціонувати тисячі або мільйони пристроїв, що використовують різні апаратні платформи, операційні системи та протоколи зв’язку [13, 20, 22, 25, 26, 28]. Більшість з них є ресурсно-обмеженими: мають невеликі обсяги пам’яті, обмежену обчислювальну потужність та живляться від батарей [13, 20, 26, 29]. Саме тому архітектура IoT-систем проектується таким чином, щоб винести складні обчислювальні задачі на більш потужні вузли — шлюзи, периферійні сервери або хмарну інфраструктуру [5, 15, 24, 25].

В науковій літературі Інтернет речей описується за допомогою багаторівневих архітектурних моделей. Найпростішою і найбільш вживаною є трирівнева модель, яка включає рівень сприйняття, мережевий рівень та прикладний рівень. На рівні сприйняття розміщуються сенсори та актуатори, що безпосередньо взаємодіють з фізичним середовищем і перетворюють фізичні величини на цифрові дані. Мережевий рівень відповідає за передавання цих даних через дротові й бездротові канали зв’язку до вузлів

обробки. На прикладному рівні реалізуються сервіси, які інтерпретують отримані дані, виконують аналітику, приймають рішення і надають кінцеві функції користувачам, такі як моніторинг, керування, візуалізація, інтеграція з бізнес-процесами [5, 20, 25].

На рисунку 1.1 представлено узагальнену трирівневу архітектуру IoT-системи, де показано місце сенсорів, шлюзів, хмарних сервісів та кінцевих користувачів.

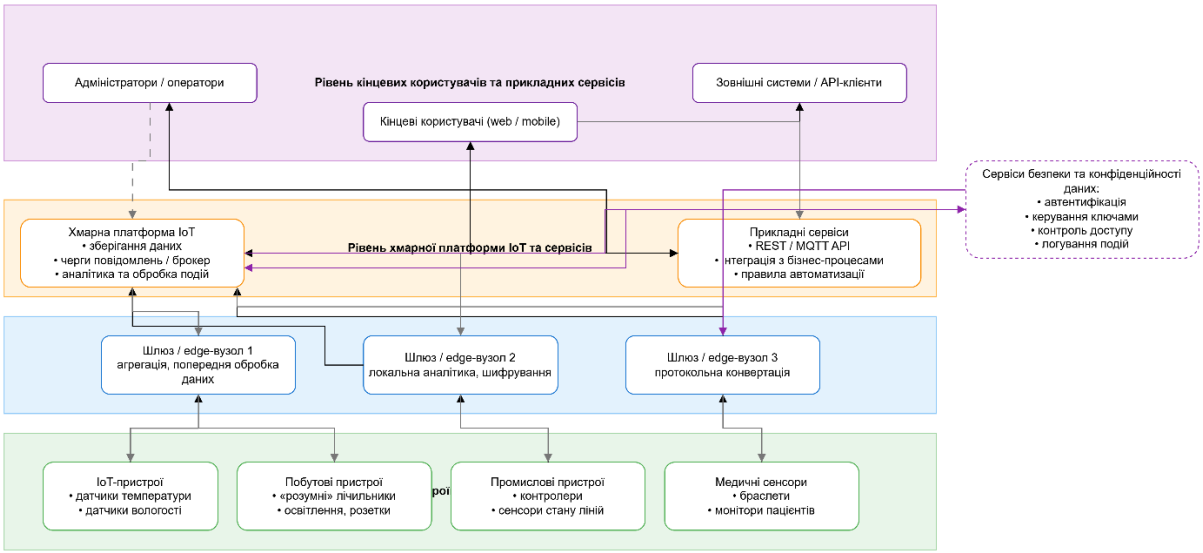


Рисунок 1.1 - Трирівнева архітектура IoT

Поряд із трирівневою моделлю в літературі також часто розглядають п’ятирівневу архітектуру, яка включає рівень бізнесу та рівень обробки, як окремі складові [5, 14, 20, 24, 25]. В такому випадку рівень сприйняття та мережевий рівень залишаються аналогічними, а рівень обробки описує системи зберігання і обробки даних, що перетворюють потік сирих сенсорних даних на більш високорівневу інформацію, потрібну для прийняття рішень. Рівень бізнесу формалізує правила, політики та бізнес-процеси, які використовують результати аналітики, наприклад, правила автоматичного керування обладнанням на виробництві або політики безпеки та конфіденційності для окремих категорій користувачів [5, 20, 24, 31].

На рисунку 1.2 представлено п’ятирівневу архітектуру системи Інтернету речей з виокремленням рівня обробки даних та бізнес-рівня.

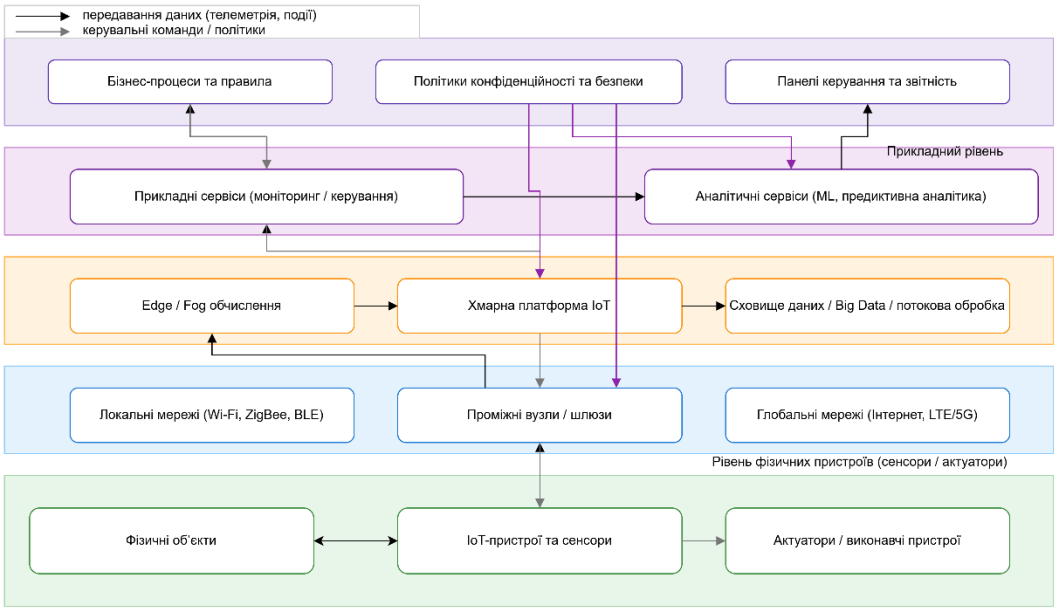


Рисунок 1.2 - П’ятирівнева архітектура системи Інтернету речей

Важливим аспектом архітектури IoT є роль шлюзів та периферійних вузлів [15, 24, 27, 31]. В класичній схемі вся інтелектуальна обробка даних відбувається в хмарі. Однак, з огляду на вимоги до затримки, обмеження пропускної здатності каналів зв’язку та потреби у збереженні конфіденційності, частину функцій все частіше переміщують ближче до джерел даних — на периферію мережі [15, 24, 31]. Edge-та fog-вузли можуть виконувати попередню фільтрацію і агрегацію даних, локальну аналітику, шифрування, мінімізацію, а також застосовувати локальні політики доступу, зменшуючи обсяг конфіденційної інформації, що потрапляє до хмарних сервісів [15, 24, 31]. Такий підхід дає змогу краще контролювати розподіл даних між різними доменами довіри та більш гнучко налаштовувати компроміс між конфіденційністю, точністю аналітики та швидкодією [2, 5, 24, 31].

В багатьох роботах, що стосуються безпеки і приватності в IoT, пропонується розширювати архітектуру додатковими компонентами —

окремими сервісами управління ідентичністю та доступом, сервісами керування ключами, модулями моніторингу безпеки та виявлення вторгнень. В архітектурних схемах такі компоненти розміщують на рівні обробки і бізнес-рівні або як горизонтальні сервіси, що перетинають усі рівні системи. Вони відповідають за автентифікацію користувачів та пристроїв, авторизацію операцій, ведення журналу доступу до чутливих даних, централізоване задавання та застосування політик конфіденційності [9, 11, 12, 16, 24].

Ще однією характерною рисою архітектури IoT є велика різноманітність комунікаційних протоколів [3, 17–19, 28]. На рівні сенсорів використовуються енергоефективні бездротові технології, такі як IEEE 802.15.4, Bluetooth Low Energy, ZigBee, різні варіанти LPWAN. Для зв'язку між шлюзами та хмарою — IP-мережі, MQTT, CoAP, HTTP(S), AMQP і спеціалізовані протоколи телеметрії [3, 17–19, 28]. В контексті конфіденційності це означає, що засоби захисту даних повинні бути адаптовані до обчислювальних можливостей конкретних пристроїв і пропускної здатності каналів, забезпечувати сумісність з уже існуючими промисловими стандартами [17–19, 22, 28].

В загальному будь-яка реальна архітектура IoT-системи є поєднанням описаних рівневих моделей з конкретизацією ролей окремих компонентів [5, 20, 25, 29]. В деяких варіантах виділяють також рівень керування, який відповідає за конфігурацію, оновлення програмного забезпечення і відстеження стану пристроїв [20, 25, 26]. В окремих сценаріях розглядають доменну декомпозицію на підсистеми різних організацій або адміністративних доменів [5, 20, 22, 29]. Для дослідження методів збереження конфіденційності важливо чітко зафіксувати, які саме компоненти і на яких рівнях відповідають за обробку чутливих даних, де вони зберігаються, які сервіси мають до них доступ та за якими протоколами здійснюється взаємодія [2, 5, 10, 24, 31].

Тобто архітектура IoT повинна розглядатися не лише з позицій функціональності, а і з точки зору моделі загроз [13, 20–22, 26, 31]. Це означає, що вже на етапі архітектурного проектування необхідно визначити довірені та недовірені домени, потенційні точки спостереження, місця можливого

перехоплення трафіку, а також вузли, на яких в принципі можливе розкриття чи спотворення даних. Саме тому в роботах, орієнтованих на збереження приватності, архітектурні схеми доповнюються елементами, які відображають шляхи розповсюдження і перетворення конфіденційної інформації, а також точки застосування механізмів її захисту [2, 5, 10, 24, 31].

1.2 Поняття конфіденційності та загрози безпеці в IoT-середовищі

Конфіденційність традиційно визначають, як властивість інформації бути доступною лише тим суб'єктам, які мають відповідні повноваження. В типових моделях інформаційної безпеки конфіденційність входить до базових цілей разом із цілісністю та доступністю, а також тісно пов'язана з автентичністю та підзвітністю. Для систем Інтернету речей це поняття має особливе значення, оскільки сенсори та вбудовані пристрої часто збирають чутливі дані, наприклад, медичні показники, параметри домашнього оточення, геолокацію, інформацію про споживання ресурсів тощо. Порушення конфіденційності в такому випадку може спричинити не лише фінансові або репутаційні втрати, а і створити прямі ризики для безпеки життя і здоров'я користувачів [5, 13, 20, 21, 26, 31].

Поняття конфіденційності також тісно пов'язане з приватністтю. Приватність у більшості підходів інтерпретується як право індивіда контролювати, коли, яким чином і в якій мірі інформація про нього збирається, використовується, передається та зберігається [2, 5, 10, 24, 25, 31]. Тобто приватність описує ступінь контролю суб'єкта над власними персональними даними протягом усього їх життєвого циклу. Конфіденційність же фокусується на технічних і організаційних механізмах, які запобігають несанкціонованому розкриттю даних третім сторонам. Можлива ситуація, коли канал зв'язку між сенсором і сервером надійно шифрується та формально забезпечує конфіденційність, однак сам факт надмірного збору даних або їх

вторинного використання без поінформованої згоди користувача є порушенням його приватності [2, 5, 10, 24, 31].

Ширшим за обидва попередні є поняття інформаційної безпеки. У класичній CIA-тріаді воно включає конфіденційність, цілісність і доступність, а в розширених моделях доповнюється автентичністю, підзвітністю, надійністю та стійкістю. Для IoT це означає, що система повинна не лише приховувати вміст даних від неавторизованих суб'єктів, але крім того гарантувати, що дані не змінені зловмисником, доступні у потрібний момент і походять від легітимних пристроїв і користувачів [5, 13, 20, 21, 26, 29]. На практиці виробники IoT-рішень часто ототожнюють безпеку з застосуванням базового шифрування каналу зв'язку і недооцінюють ризики, пов'язані з політиками збору, зберігання, агрегування та подальшого аналізу персональних даних, що безпосередньо впливають на приватність [2, 5, 10, 24, 31].

Специфіка IoT-середовища формує характерний профіль загроз конфіденційності [13, 20–22, 26, 29]. По-перше, сенсори й вбудовані пристрої зазвичай розміщені у фізично відкритих локаціях, де їх простіше атакувати фізично, модифікувати прошивку або повністю замінити на підроблений пристрій. По-друге, більшість таких пристроїв мають суттєві обмеження щодо обчислювальних ресурсів, пам'яті та енергоспоживання, що ускладнює застосування складних криптографічних алгоритмів і повноцінних протоколів керування ключами. По-третє, IoT-екосистема характеризується високою гетерогенністю, значною масштабованістю та динамікою підключень, що ускладнює реалізацію єдиної політики безпеки та централізованого моніторингу [12, 13, 20–22, 24].

Однією з базових загроз для конфіденційності є перехоплення трафіку, наприклад, через використання бездротових каналів з недостатнім рівнем захисту, зловмисник може пасивно прослуховувати радіоканал та отримувати доступ до телеметрії: показів розумних лічильників, даних про присутність у приміщенні, сигналів від датчиків безпеки чи медичних сенсорів [3, 13, 20, 21,

31]. Навіть якщо вміст пакетів зашифрований, аналіз метаданих дозволяє робити висновки щодо поведінки користувачів, їх розкладу дня, періодів відсутності вдома тощо. Такий аналіз трафіку вважається окремим класом загроз, оскільки порушує приватність користувача навіть без розкриття власне змісту даних.

Можливість прив’язати стійкий ідентифікатор пристрою чи користувача до певних типів даних створює підґрунтя для тривалого спостереження за активністю цього суб’єкта в IoT-мережі [2, 5, 10, 24, 31]. Локалізація та відстеження дозволяють будувати часові карти переміщень, визначати патерни присутності, а профілювання – виводити інтереси, стан здоров’я, соціально-економічний статус користувача на основі поєднання даних із різних сенсорів та сервісів.

На рисунку 1.3 подано узагальнену таксономію загроз конфіденційності та приватності в IoT-середовищі, яка демонструє поділ між технічними атаками на рівні каналів зв’язку та протоколів і загрозами, що виникають на рівні даних та ідентичності користувачів (ідентифікація, локалізація, відстеження, профілювання, інвентаризаційні та linkage-атаки).

Ще одним важливим класом загроз є несанкціонований доступ до пристроїв і сервісів. Його причинами можуть бути вразливості у прошивці, відкриті порти, небезпечні налаштування за замовчуванням, використання слабких паролів або повторне застосування одних і тих самих облікових даних для різних сервісів. В разі успішної атаки зловмисник отримує можливість читати або змінювати масиви телеметрії, змінювати конфігурацію пристроїв, вмикати або вимикати окремі компоненти системи.

В такому випадку порушуються як конфіденційність, так і цілісність даних, а автоматизовані рішення, що приймаються на основі підроблених показників сенсорів, можуть мати критичні наслідки.

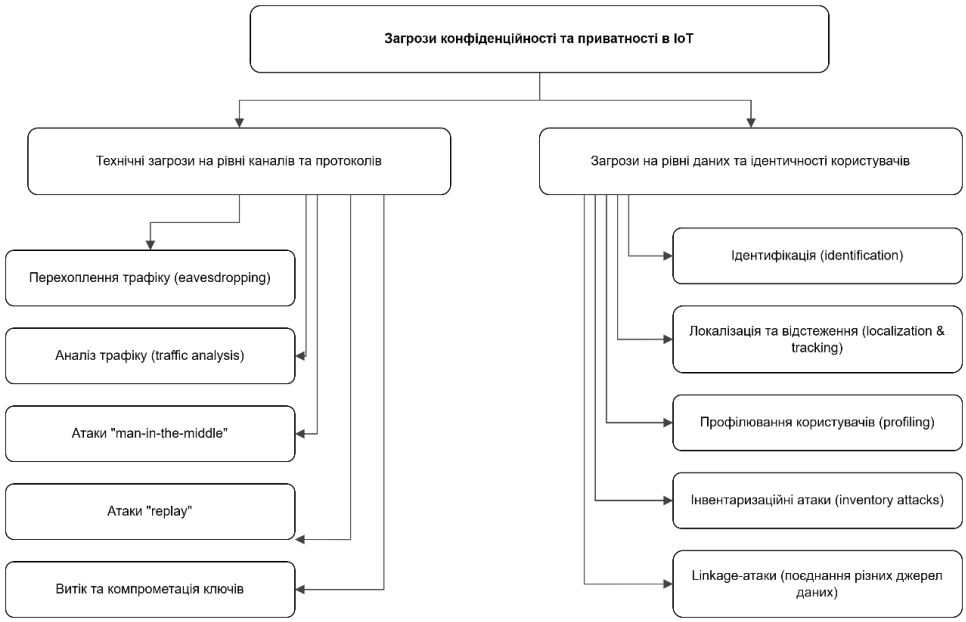


Рисунок 1.3 - Узагальнена таксономія загроз конфіденційності та приватності в IoT-середовищі

Значну небезпеку становлять атаки типу «людина посередині» та «повтор відтвореного повідомлення» [11, 13, 20, 21]. В першому випадку зловмисник прозоро вставляється між двома легітимними учасниками сеансу, перехоплює та модифікує пакети, при цьому обидві сторони вважають, що спілкуються безпосередньо одна з одною. В другому випадку зловмисник записує легітимні повідомлення, а потім повторно надсилає їх у більш пізній момент часу, наприклад для повторного відкривання «розумного» замка чи повторного виконання фінансової транзакції. Обидва види атак належать до активних, оскільки вони впливають не лише на конфіденційність, але і на цілісність обробки даних [11, 13, 20].

Окрему групу становлять загрози витоку та компрометації ключів [11, 13, 20, 22, 27]. Через обмеженість ресурсів і спрощені процеси виробництва виробники нерідко зберігають криптографічні ключі безпосередньо у прошивці або використовують однакові паролі та сертифікати для великої кількості пристроїв. Як наслідок, компрометація одного пристрою дає змогу атакувати всю лінійку. Додатково існують побічні канали, коли значення відновлюються на основі аналізу часу виконання, споживання енергії чи

електромагнітного випромінювання мікроконтролера. Усі ці сценарії прямо підривають криптографічні механізми забезпечення конфіденційності, навіть якщо з формальної точки зору використовуються стійкі алгоритми [11, 20, 22].

До загроз конфіденційності також відносять атаки, пов’язані з життєвим циклом пристроїв і даних [2, 5, 10, 24, 31]. Наприклад, lifecycle-атаки виникають, коли при зміні власника або утилізації IoT-пристрою історичні дані не видаляються належним чином і можуть бути відновлені третім суб’єктом. Інвентаризаційні атаки дають змогу з високою точністю відновити склад IoT-пристроїв в домогосподарстві чи організації на основі аналізу радіоефіру або мережевого трафіку. Linkage-атаки полягають у поєднанні кількох анонімних наборів даних, що разом утворюють значно більш детальний і чутливий профіль користувача, ніж будь-яке джерело окремо [2, 5, 10, 24].

Для узагальнення взаємозв’язку між цілями інформаційної безпеки та розглянутими класами загроз на рисунку 1.4 наведено схему відповідності, в якій для кожної цілі показано характерні для IoT-середовища типи атак, що безпосередньо її підривають [2, 5, 20, 21, 31].

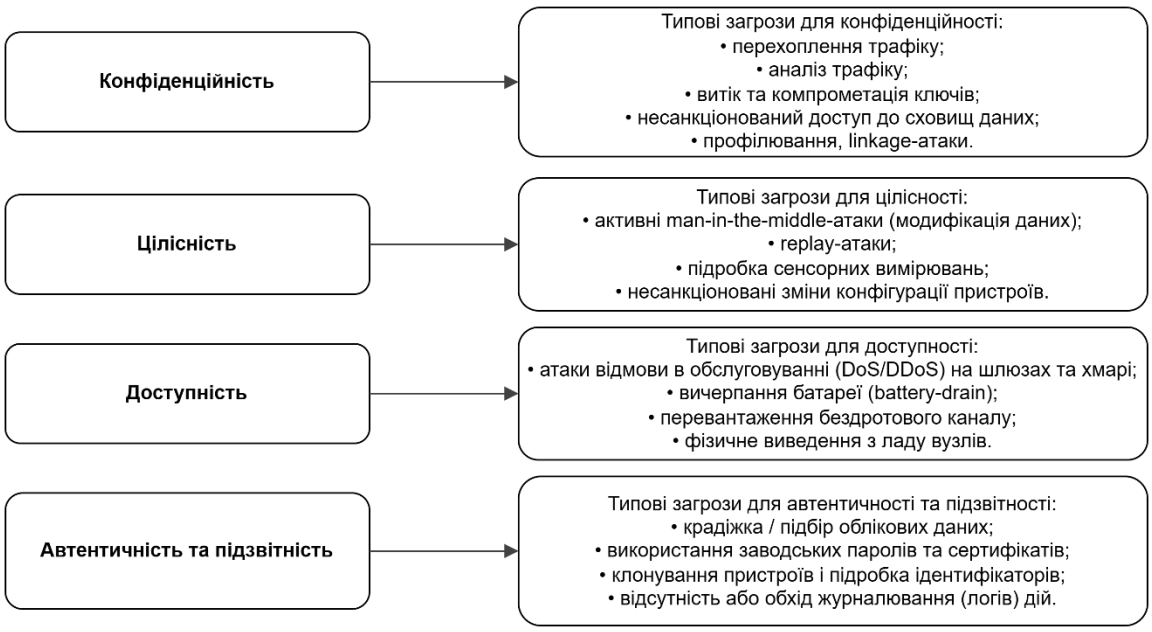


Рисунок 1.4 - Відповідність між цілями інформаційної безпеки та типовими загрозами в IoT

Підсумовуючи, конфіденційність у системах Інтернету речей не зводиться лише до питання шифрування каналів зв’язку. Це комплексна властивість, яка охоплює всі рівні IoT-архітектури – від фізичних пристроїв і мережевих протоколів до хмарних платформ, прикладних сервісів і бізнес-процесів – та весь життєвий цикл даних.

1.3 Сучасні методи збереження конфіденційності та приватності в IoT-системах

Особливість IoT-платформ полягає в поєднанні великої кількості ресурсно-обмежених пристроїв, розподілених каналів зв’язку та хмарної інфраструктури для зберігання та аналітики [2, 5, 10, 24, 25, 31]. Тому методи збереження конфіденційності мають охоплювати всі етапи життєвого циклу даних – від моменту їх збору датчиками до довготривалого архівування та використання в інтелектуальних сервісах.

В сучасній літературі пропонуються різні таксономії методів захисту приватності, які, як правило, групуються навколо кількох базових стратегій: мінімізувати обсяг зібраних та переданих даних, приховати вміст за допомогою шифрування та інших криптографічних механізмів, розподілити зберігання і контроль між різними доменами довіри та узагальнювати, агрегувати дані замість передавання сирих вимірювань [2, 5, 10, 24, 31, 32]. На основі цих стратегій формуються конкретні технічні рішення.

Узагальнено сучасні методи можна поділити на такі основні групи:

– схеми взаємної автентифікації та керування ключами;

– методи шифрування та захищеного зберігання даних;

– техніки анонімізації та псевдонімізації;

– підходи, засновані на диференційній приватності та шумуванні даних;

– розподілені моделі навчання і обробки (федеративне навчання, edge/fog-обчислення);

– блокчейн-орієнтовані та політико-семантичні моделі приватності.

На рисунку 1.5 наведено узагальнену класифікацію сучасних методів збереження конфіденційності та приватності в IoT-системах, побудовану за принципами стратегій Minimize, Hide, Separate та Aggregate, в рамках яких згруповано основні техніки захисту [2, 5, 10, 24, 31, 32].

Перший великий клас рішень зосереджений на взаємній автентифікації пристроїв та безпечному встановленні ключів доступу [7, 11, 16, 22, 24]. Типовий підхід полягає в тому, що кожен IoT-вузол має фізично неклоновану функцію – PUF, на основі якої виконується процедура реєстрації та подальшої взаємної автентифікації з іншими вузлами або шлюзами.

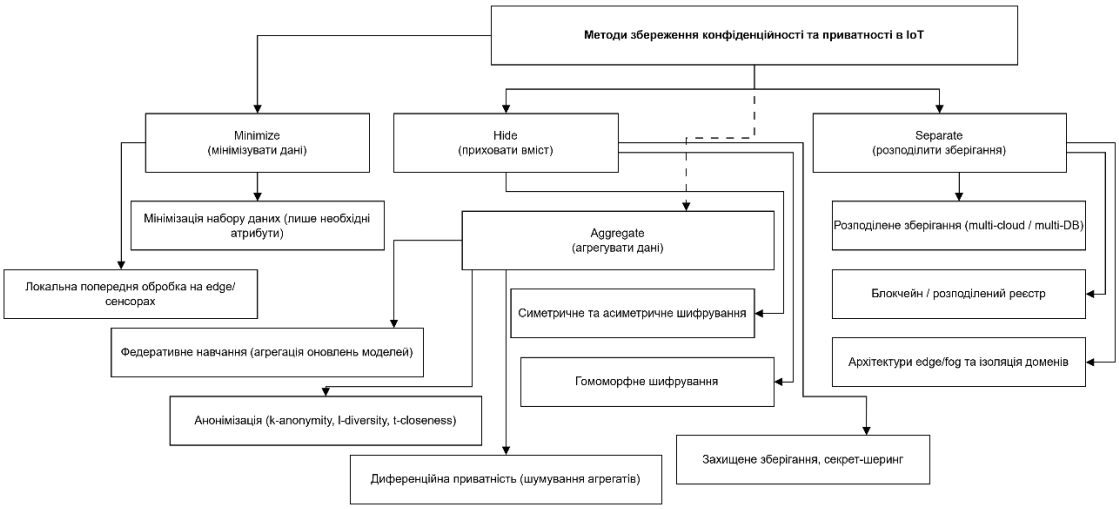


Рисунок 1.5 - Класифікація методів збереження конфіденційності та приватності

Для зменшення обчислювальних витрат такі протоколи будуються на легковагових криптографічних примітивах – хеш-функціях, операції XOR, симетричному шифруванні.

В деяких дослідженнях запропоновано ієрархічні архітектури, в яких PUF виступає апаратним коренем довіри, а сама процедура автентифікації розбита на фази ініціалізації, реєстрації, взаємної автентифікації, узгодження сеансового ключа. Для збереження приватності використовуються тимчасові ідентифікатори або псевдоніми, що дозволяє приховати реальні ID пристроїв

від сторонніх спостерігачів і ускладнити їх довготривале відстеження [7, 11, 27].

Перевагами таких схем (рисунок 1.6) є можливість автоматичної взаємної автентифікації без участі користувача, сумісність із ресурсо-обмеженими сенсорами завдяки легковаговим обчисленням, а також підтримка анонімності та невідслідковуваності пристроїв [7, 11, 24, 27]. До ключових недоліків належить те, що фокус здебільшого робиться на рівні доступу та встановлення ключів, тоді як питання конфіденційності зібраних даних під час довготривалого зберігання та аналітики часто залишаються недостатньо опрацьованими. Крім того, централізовані реєстратори або виробники пристроїв, що оперують великими наборами PUF-відповідей, стають критичними точками довіри й потенційної компрометації.

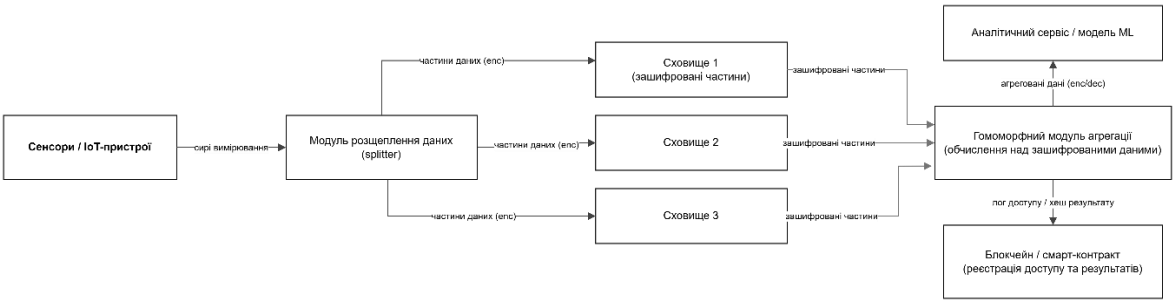


Рисунок 1.6 - Захищена схема збирання та обробки IoT-даних

1.4 Вибір перспективного шляху і постановка задачі дослідження

Аналіз архітектурних моделей Інтернету речей показав, що більшість реальних рішень є багаторівневими системами, де дані проходять шлях від сенсорних вузлів через шлюзи та периферійні сервери до хмарних сервісів та прикладних додатків. На кожному з цих рівнів дані можуть спостерігатися, копіюватися або модифікуватися, а отже, саме шлях руху даних визначає основні точки потенційного витоку конфіденційної інформації.

Огляд загроз конфіденційності показав, що просте шифрування каналу зв’язку не усуває ризиків пасивного прослуховування, аналізу трафіку, атак

повторного відправлення, профілювання користувачів та компрометації ключів.

В сучасних роботах запропоновано методи захисту, які базуються на схемах взаємної автентифікації, легких протоколах IoT, методах шифрування та федеративного навчання тощо [1, 2, 4, 7, 11, 12, 15, 16, 18, 19, 22, 24].

Більшість таких рішень фокусується на рівні доступу та встановлення ключів, або на окремих етапах зберігання та аналітики. При цьому питання комплексного контролю конфіденційності на повному шляху руху даних від пристрою до прикладних сервісів часто залишаються недостатньо опрацьованими, що є критичним для IoT-платформ [2, 5, 10, 24, 31].

Метою роботи є розроблення методу збереження конфіденційності даних в IoT-системах, який на основі формальної моделі потоків даних і множини політик доступу забезпечує багаторівневий захист інформації від моменту її збору на сенсорах до зберігання й обробки на серверній стороні, з урахуванням обмежених ресурсів вузлів Інтернету речей.

Для досягнення поставленої мети необхідно розв’язати такі основні задачі дослідження:

1. Проаналізувати архітектурні моделі та загрози конфіденційності в системах Інтернету речей.

2. Узагальнити сучасні методи збереження конфіденційності та приватності в IoT,.

3. Побудувати формальну модель IoT-системи, яка описує множини пристроїв, користувачів, сервісів, каналів зв’язку, типів даних, ключів і політик доступу.

4. Розробити концепцію та архітектуру методу збереження конфіденційності.

5. Сформувати алгоритми реалізації методу у вигляді покрокових процедур для реєстрації й автентифікації пристроїв, підготовки та передавання захищених повідомлень.

6. Вибрати програмно-апаратну платформу для експериментів та реалізувати прототип.

7. Розробити методику експериментальної оцінки запропонованого методу, визначити сценарії навантаження, набір метрик продуктивності.

8. Провести експериментальні дослідження, порівняти запропонований метод з базовими рішеннями.

Висновки до розділу 1

1. Розглянуто багаторівневі архітектури IoT. Показано, що дані проходять через велику кількість проміжних вузлів, кожен з яких може стати точкою витоку конфіденційної інформації.

2. Визначено поняття конфіденційності в контексті IoT, показано його відмінність від приватності користувача та загальної інформаційної безпеки.

3. Проаналізовано класи загроз конфіденційності в IoT-середовищі: перехоплення та аналіз трафіку, несанкціонований доступ до вузлів, профілювання користувачів, витік і компрометація ключів, побічні канали та загрози, пов’язані з життєвим циклом пристрою.

4. Проведено огляд існуючих методів збереження конфіденційності та приватності в IoT.

5. Виявлено обмеження існуючих підходів, які залежать від централізованих сховищ ключів та коренів довіри, складності інтеграції в IoT-екосистемі.

Отримані результати показують потребу розроблення методу збереження конфіденційності, який би поєднував формальну модель IoT-системи, багаторівневу обробку та захист даних.

2 МЕТОД ЗАБЕЗПЕЧЕННЯ ПРИВАТНОСТІ ДАНИХ В IoT-СИСТЕМАХ

2.1 Формальна модель IoT-системи та моделі загроз конфіденційності

Провівши аналіз попередніх робіт було запропоновано метод збереження конфіденційності, який базується не лише на інженерному описі архітектури, а і на формальній моделі системи Інтернету речей [2, 5, 10, 24, 25, 31, 32]. Формалізація дає змогу визначити об’єкти, які обмінюються даними і описати, в яких саме блоках можливе порушення конфіденційності.

Спираючись на поширені багатошарові архітектури IoT, де дані рухаються від сенсорів через мережеву інфраструктуру до проміжного програмного шару та прикладних сервісів [5, 20, 25, 28], формально таку систему можна описати як кортеж.

$$L=(D,U,S,C,T,K,Pol) \tag{2.1}$$

Де, D представляє множину пристроїв до яких входять сенсори, актутатори, шлюзи, edge-вузли;

U –множина користувачів і організацій, що взаємодіють з системою;

S – множина сервісів до яких входять аналітичні модулі, сервіси зберігання та зовнішні хмарні сервіси;

C – множина каналів зв’язку між елементами;

T – множина типів даних;

K – множина криптографічних ключів та інших секретів;

Pol – множина політик доступу й обробки даних.

Модель пристроїв, користувачів і сервісів

Множина пристроїв D розбивається на підмножини:

D_{sens}– сенсорні вузли, що здійснюють вимірювання;

D_{act}– актуатори, які виконують керувальні команди;

D_{gw}– шлюзи та edge-вузли;

D_{aux}– допоміжні пристрої (маршрутизатори, точки доступу, проксі).

Кожен пристрій $d \in D$ описується набором атрибутів: id_d – ідентифікатор, $role_d$ – роль яку призначено вузлу - сенсор, шлюз тощо, loc_d – розташування, res_d –обчислювальні потужності, обсяг пам'яті, енергобюджет, $trust_d$ – рівень довіри для специфічних пристроїв або вузли які мають високу степень довір'я [13, 20, 25, 26].

$$attr(d) = \{id_d, role_d, loc_d, res_d, trust_d\} \quad (2.2)$$

Така деталізація важлива, тому що обмежені ресурси та наявність недовірених компонентів є ключовими чинниками вразливості до витоків даних в IoT.

Множина користувачів U включає окремих фізичних осіб, домогосподарства, операторів платформи, адміністраторів, а також сторонніх розробників, які використовують API [5, 12, 20, 24]. Для кожного користувача $u \in U$ задається підмножина ролей *roles(u)*, якими можуть бути власники даних, адміністратори або розробники тощо. Крім того в такій моделі є набір дозволених операцій над даними, визначений політикою *Pol*.

Множина сервісів *S* охоплює:

- сервіси зберігання й обробки даних у хмарі;
- сервіси аналітики (ML-моделі, статистичні модулі);
- прикладні сервіси (мобільні застосунки, веб-інтерфейси, панелі моніторингу);
- зовнішні сторонні сервіси.

Кожний сервіс $s \in S$ характеризується тим, які типи даних він приймає та генерує, а також тим, чи є він довіреною стороною щодо конфіденційних даних [2, 5, 10, 24, 31]. В останніх наукових дослідженнях та звітах IT компаній показується, що саме багатошарові архітектури з великою кількістю сервісів створюють значний простір для неконтрольованого вторинного використання даних, що потребує розробку політик та технічних гарантій конфіденційності.

Тому пропонується модель каналів зв'язку і даних, де канали зв'язку описуються множиною.

$$C \subseteq (D \cup S) \times (D \cup S) \quad (2.3)$$

де кожен канал $c = (x,y)$ має атрибути:

$$attr(c) = \{prot_c, sec_c, cap_c\} \quad (2.4)$$

де $prot_c$– представляють собою сучасні IoT протоколи, такі як MQTT, CoAP, HTTP(S), LoRaWAN;

sec_c– це набір реалізованих механізмів безпеки шифрування, цілісності та аутентифікації;

cap_c– пропускна здатність, затримка та інші параметри якості обслуговування [17–19, 22, 28].

Множина типів даних *T* поділяється на:

– T_{meas} – телеметричні дані (значення сенсорів, події);

– T_{cmd} –команди управління;

– T_{meta} – метадані: ідентифікатори пристроїв, часові мітки, топологія;

– T_{ident} – дані, що прямо ідентифікують суб'єкта куди входить ПІП, ID;

– T_{sens} – конфіденційні особисті та чутливі дані: здоров'я, поведінка, місцезнаходження.

Елемент даних формально можна подати як

$$x=(t,src,dst,ts,ctx) \quad (2.5)$$

де $t \in T$ – тип;

src та *dst* – джерело та отримувач, тобто це може бути пристрій або сервіс;

ts – час формування;

ctx –умови вимірювання, сценарій застосування, які позначаються як контекст.

Тоді можна позначити значенням функції.

$$sens(x) \in \{0,1\} \quad (2.6)$$

Далі задаються процеси, через які проходять дані в системі.

Функція збору даних:

$$collect: D_{sens} \times T_{meas} \to X, \tag{2.7}$$

де X – множина всіх елементів даних.

Для кожного сенсора $d \in D_{sens}$ можна задати інтенсивність генерації даних $\lambda_{col}(d)$, що характеризує, як часто пристрій формує нові вимірювання.

Функція передавання:

$$transmit: C \times X \to X, \tag{2.8}$$

що відображає проходження елемента даних через канал. При цьому для каналу з параметром sec_c задається ймовірність несанкціонованого розкриття $p_{leak}(c)$— чим слабша або взагалі відсутня криптографічна складова захисту, тим більшим є це значення.

Функція зберігання:

$$store: S \times X \to DB, \tag{2.9}$$

де DB – логічна множина сховищ даних.

Для кожного сховища можна ввести параметри тривалості зберігання та ступінь захищеності.

Функція доступу:

$$access: U \times DB \to 2^x, \tag{2.10}$$

яка повертає підмножину елементів даних, до яких має доступ користувач u. Ця функція обмежується політикою Pol , яку можна представити як множину правил вигляду

$$pol = (role, t, op, cond), \tag{2.11}$$

де *role* – роль суб'єкта;

$t \in T$ – тип даних;

op – операція читання, запису, видалення або агрегаці;

cond – додаткові умови до яких входять час, місце, стан пристрою.

На рисунку 2.1 представлено описану модель у вигляді діаграми потоків даних між множинами *D, C, S, DB* та *U*, а також представлено які перетворення застосовуються до елементів з *sens(x) = 1*.

Рисунок 2.1 - Формальна модель IoT-системи

Для опису загроз конфіденційності вводиться множина можливих порушників *A*. Кожен порушник $a \in A$ характеризується

$$cap(a) = (view_a, act_a, res_a) \tag{2.12}$$

де $view_a$ – множина даних і каналів, які порушник може спостерігати; act_a– набір активних дій самого порушника, куди може входити модифікація, ін'єкція, блокування повідомлень та компрометація вузлів);

res_a – доступні ресурси, можливість фізичного доступу до пристроїв та час атаки.

В сучасних наукових роботах для аналізу протоколів автентифікації та збереження конфіденційності часто використовується модель порушника типу Dolev–Yao, яка передбачає повний контроль над відкритими каналами зв'язку при неможливості порушити криптографічні примітиви [11, 24].

В рамках цієї моделі можна виділити такі базові сценарії атак:

1. Пасивне перехоплення трафіку. Порушник спостерігає множину каналів $C_{obs} \subseteq C$ та накопичує множину елементів даних $View_a \subseteq X$, не змінюючи їх. Конфіденційність порушується, якщо на основі $View_a$ порушник може з імовірністю вище за поріг θ відновити значення хоча б одного чутливого атрибуту *sens(x) = 1* для конкретного суб'єкта.

2. Активні атаки на канали, коли порушник перехоплює та змінює повідомлення, що передаються по каналу c, або повторно надсилає старі повідомлення. Формально, замість елемента x на виході функції $transmit$ з'являється елемент $x' = f_{att}(x)$, де f_{att} – довільна функція атакуючого.

3. Порушник може зкомпроментувати вузли та отримати контроль над підмножиною пристроїв $D_{comp} \subset D$ або сервісів $S_{comp} \subset S$, зокрема над їхніми локальними сховищами та ключами $K_{comp} \subseteq K$. В такому випадку він може спостерігати та модифікувати всі дані, що обробляються скомпрометованими вузлами, в тому числі до моменту застосування механізмів захисту. В багатьох дослідженнях наголошується, що саме внутрішні порушники з привілеями доступу є одними з найнебезпечніших загроз для конфіденційності в IoT [2, 5, 13, 20–22, 26, 29, 31].

4. Якщо окремі набори даних пройшли анонімізацію або агрегацію, їх об'єднання може дозволити порушнику відновити початкову інформацію. Формально, маючи кілька підмножин $X_1, X_2, \dots, X_n$, порушник обчислює їх об'єднання X_{joint} та може побудувати функцію реконструкції $g: X_{joint} \to T_{sens}$, що повертає значення чутливих атрибутів для конкретних суб'єктів.

Для реалізації методу збереження конфіденційності можна вважати, що кожен сценарій атаки задає множину можливих витоків

$$Leak_a \subseteq X \tag{2.13}$$

тобто підмножину елементів даних, до яких порушник може отримати доступ чи які може модифікувати. Завдання методу полягає в тому, щоб за

будь-якої допустимої конфігурації порушника $a \in A$ і відповідних сценаріїв атаки забезпечити виконання умови

$$\Pr\ (a \text{ коректно відновлює чутливу інформацію з } Leak_a) \leq \theta$$

де θ – максимально допустима ймовірність розкриття, визначена вимогами до безпеки системи або нормативними актами.

2.2 Концепція та архітектура запропонованого методу збереження конфіденційності в IoT-системах

В попередньому розділі було розглянуто IoT-систему у вигляді множин пристроїв D, каналів зв'язку C, сервісів S, сховищ даних DB та користувачів U, а також введено бінарну функцію чутливості $sens(x)$, що позначає, чи є конкретний елемент даних конфіденційним. На цій основі можна сформувати концепцію методу збереження конфіденційності, який забезпечує захист усіх елементів з $sens(x)$=1 на всіх етапах життєвого циклу даних – від моменту їх збору на пристрої до доступу авторизованих суб'єктів на хмарній частині системи.

Метод дотримується підходу, коли механізми збереження конфіденційності вбудовані в архітектуру IoT-рішення, а не додаються окремим зовнішнім шаром. Такий підхід узгоджується з сучасними тенденціями в IoT, де приватність повинна забезпечуватися комбінацією технологічних засобів на різних рівнях архітектури [2, 5, 10, 24, 31, 33]. Центральним об'єктом в такій системі є окремий елемент даних x який має: (1) рівень чутливості; (2) контекст що містить інформацію про джерело, тип сенсора, належність користувачу, місце збору; (3) набір дозволених операцій. Для кожного класу чутливості визначається політика, яка задає:

– які перетворення повинні бути виконані на рівні пристрою, наприклад видалення ідентифікаторів;

– які операції виконуються на шлюзі;

– які правила зберігання та доступу застосовуються в хмарі.

Тобто для будь-якої траєкторії проходження даних через послідовність $D \rightarrow C \rightarrow S \rightarrow DB \rightarrow U$ гарантується, що відкритий доступ до сирих конфіденційних значень відсутній, а всі операції з ними виконуються тільки в заздалегідь визначених, контрольованих точках.

Другою ключовою ідеєю є щільна інтеграція механізмів автентифікації та керування доступом з процедурами обробки даних [7–9, 11, 12, 16, 24, 27]. Для цього доцільно використовувати легковагові протоколи взаємної автентифікації на основі фізично неклонованих функцій (PUF) та динамічних ідентифікаторів.

В запропонованому методі автентифікація розглядається, як попередня умова будь-яких операцій з конфіденційними даними, поки пристрій, шлюз або користувач не пройшли перевірку, для них недоступні ключі шифрування, токени доступу та маршрути.

На рівні пристрою реалізується первинна класифікація і попередня обробка даних. Кожен IoT-вузол $d \in D$ має вбудований модуль політик даних, в конфігурації якого задаються типи сенсорів, належність до користувача або об’єкта і відповідний рівень чутливості.

Для всіх x із $sens(x)$=1на пристрої виконується симетричне шифрування, а також мінімізація даних – округлення, фільтрацію зайвих полів, що не потрібні для цілей сервісу. Аналогічні рекомендації щодо мінімізації та локальної обробки як ключових принципів приватності на пристроях наведені в роботах [7, 11, 27]. Крім того, при кожному сеансі взаємодії пристрій використовує динамічний псевдонім замість реального ідентифікатора.

Шлюз виступає довіреним проміжним елементом між локальною мережею пристроїв і інфраструктурою сервісів [8, 15, 24, 27, 31]. На цьому рівні застосовуються більш складні операції, які неможливо повністю реалізувати на сенсорах.

Шлюз виконує взаємну автентифікацію з пристроями на основі PUF-механізмів, які дозволяють перевірити автентичність апаратної частини та встановити унікальний сесійний ключ.

На шлюзі також здійснюється додаткова політика щодо даних з $sens(x)$=1:

– поділ конфіденційного повідомлення на кілька незалежних фрагментів;

– формування нових ключів, які будуть використовуватися вже в хмарі;

– заміна прямих ідентифікаторів користувачів, пристроїв або локацій на токени, пов’язані з відповідними записами у внутрішніх таблицях шлюзу.

В результаті навіть коли відбулась компрометація хмарної частини буде отримано тільки фрагменти зашифрованих даних з мінімальним набором метаданих, які ускладнюють відновлення повного контексту.

В хмарі реалізується логіка довгострокового зберігання даних, аналітики та інтеграції із зовнішніми сервісами. Для підтримки конфіденційності запропонований метод передбачає такі компоненти:

1. Менеджер фрагментів і ключів конфіденційних даних який розподіляє їх між кількома логічно незалежними сховищами $DB1$,$DB2$,…, Ключова інформація зберігається в окремому модулі, ізольованому від прикладних БД.

2. Доступ до операцій відновлення та дешифрування фрагментів надається лише суб’єктам, які задовольняють певним політикам, сформульованим у термінах атрибутів, наприклад, роль, організація, мета обробки, рівень довіри.

3. Модуль моніторингу слідкує щоб усі запити до чутливих даних контролювались з фіксацією суб’єкта, часу, обсягу та правової підстави доступу.

В рамках сервісів прикладного рівня аналітики, візуалізації та API доступні лише результатні узагальнені показники, що не дозволяють перейти до рівня окремих користувачів (рисунок2.2). При необхідності використання

відкритих даних метод допускає застосування диференційної приватності, коли в публічні набори додається контрольований шум.

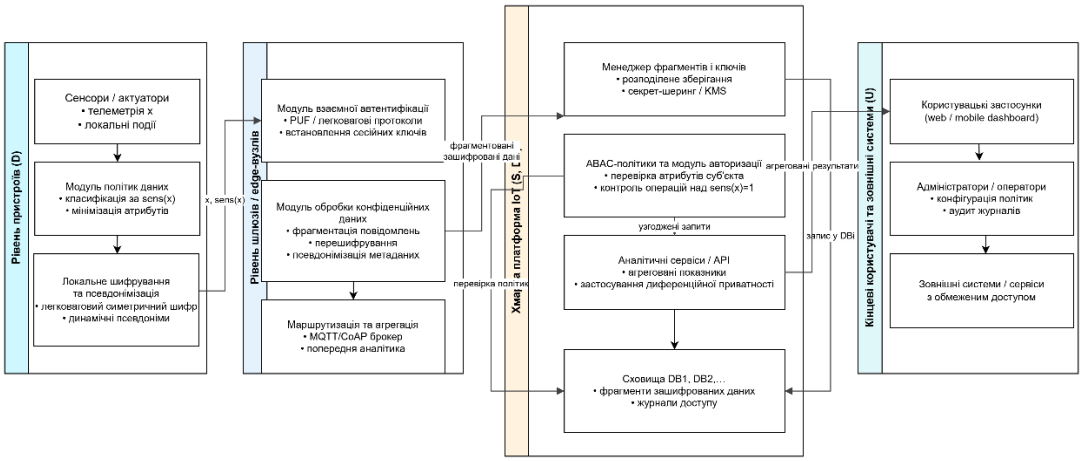


Рисунок 2.2 - Інтеграція автентифікації пристроїв і користувачів

Описані вище компоненти пов’язуються єдиною політикою ідентифікації, автентифікації та авторизації [9, 12, 16, 24, 30, 31]. На рівні пристроїв і шлюзів використовується багатофакторна схема (рисунок 2.3): апаратний відбиток (PUF), секрети, вбудовані під час виробництва або налаштування, а також динамічні параметри сеансу у вигляді часових міток [7, 8, 11, 27].

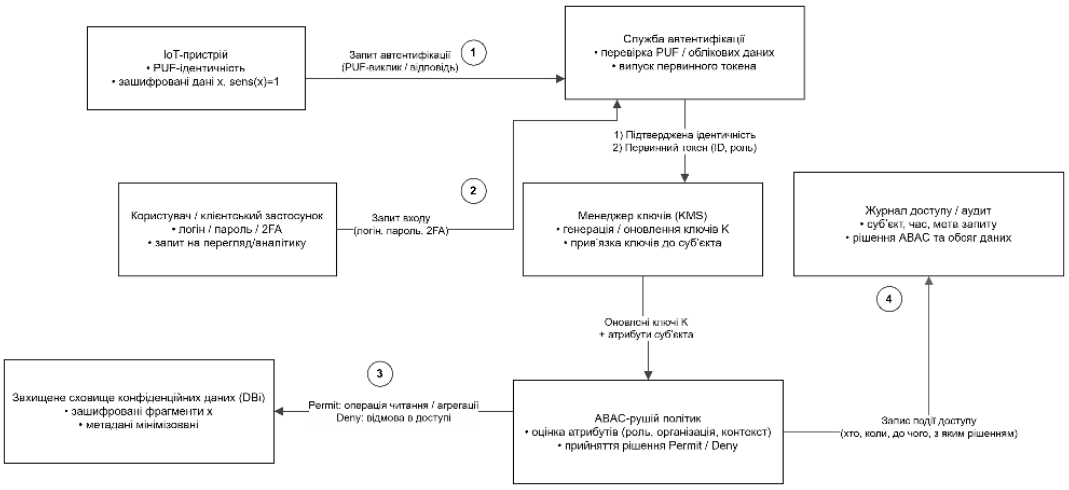


Рисунок 2.3 - Інтеграція автентифікації пристроїв і користувачів з атрибутивним керуванням доступом.

Для користувачів в хмарній частині додатково застосовуються класичні механізми, до яких входять паролі, токени та сертифікати.

Управління доступом реалізується як послідовність перевірок- спочатку апаратна автентифікація пристрою, далі перевірка права на конкретну операцію з певним рівнем чутливості даних, і потім – фактичне надання ключів дешифрування або маршрутів до фрагментів. Це забезпечує тісне поєднання механізмів конфіденційності та автентифікації.

2.3 Основні алгоритми функціонування

Описані нижче алгоритми описують повний життєвий цикл конфіденційних даних в IoT-системі [33] - від моменту підключення пристрою до системи та встановлення довіреного сеансу до зберігання фрагментованих зашифрованих даних у хмарній інфраструктурі та надання доступу до них авторизованим суб’єктам.

Виділено чотири ключові алгоритми:

1. Алгоритм реєстрації й автентифікації пристрою.
2. Алгоритм підготовки та передавання захищених повідомлень.
3. Алгоритм обробки та зберігання даних на серверній стороні.
4. Алгоритм надання доступу до даних авторизованим користувачам і сервісам.

Перший алгоритм визначає, як фізичний IoT-пристрій входить в систему, одержує власний ідентифікатор, реєструється в реєстрі пристроїв і потім щоразу проходить взаємну автентифікацію із шлюзом. На цьому етапі закладається довіра до апаратної частини та формується сесійний ключ, що використовується для захищеного обміну даними.

Робота алгоритму починається з моменту увімкнення або перезапуску пристрою. Пристрій завантажує внутрішню конфігурацію з параметрами PUF, початковими налаштуваннями мережі та службовою інформацією. Далі перевіряється, чи був пристрій зареєстрований у системі. Якщо в локальній

пам’яті є маркер попередньої реєстрації, пристрій переходить до етапу автентифікації. Якщо такий маркер відсутній, виконується первинна реєстрація.

Під час первинної реєстрації пристрій формує набір виклик–відповідь для PUF (CRP). Ці дані передаються у довірений компонент системи. На їх основі на стороні хмари створюється запис про новий пристрій, призначається унікальний ідентифікатор, фіксуються базові атрибути. Після успішного завершення реєстрації пристрій отримує свій DeviceID та початковий набір політик і зберігає локальний маркер, який в наступних сеансах дозволяє пропускати цей етап.

Далі в кожному сеансі зв’язку виконується взаємна автентифікація. Пристрій ініціює запит до шлюзу з використанням свого DeviceID. Шлюз, звернувшись до реєстру CRP, формує випадковий виклик, надсилає його пристрою, а пристрій обчислює відповідь PUF і повертає її. Шлюз порівнює отриману відповідь із еталонною. Якщо значення збігаються, вважається, що пристрій автентичний, і сторони переходять до встановлення сесійного ключа. У разі невідповідності фіксується помилка, збільшується лічильник помилок, і при перевищенні допустимого порогу пристрій може бути тимчасово або постійно заблокований.

Після успішної перевірки автентичності пристрою та шлюзу сторони виконують процедуру узгодження сесійного ключа. Сформований сесійний ключ зберігається в оперативній пам’яті пристрою та використовується для шифрування конфіденційних даних у межах поточного сеансу. Після завершення сеансу ключ видаляється.

Узагальнену блок-схему алгоритму реєстрації та автентифікації пристрою представлено на рисунку 2.4.

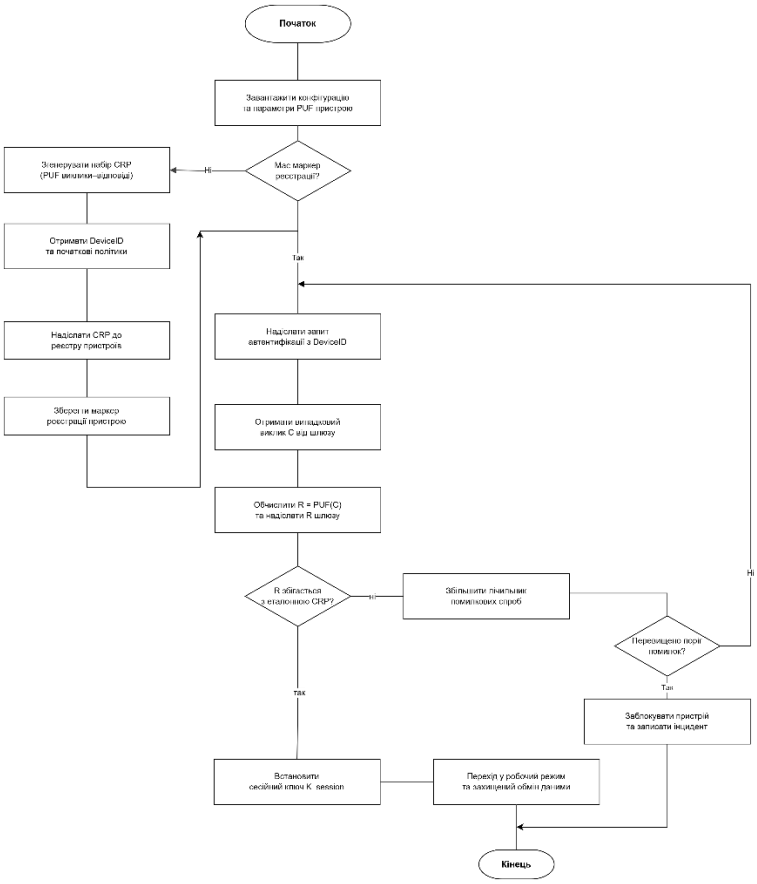


Рисунок 2.4 - Схема алгоритму реєстрації та автентифікації пристрою

Другий алгоритм описує дії, які виконує пристрій для підготовки даних до відправлення, їх перетворення відповідно до рівня чутливості та подальшої передачі через шлюз до хмарної платформи. На цьому етапі застосовується функція sens(x), що визначає конфіденційність елементів даних, а також виконується мінімізація, локальне шифрування та формування захищеного пакета. Алгоритм починається з зчитування нового вимірювання або події з сенсорів. Отримане сире значення розглядається, як x_raw. Далі модуль політик даних на пристрої визначає, чи є конкретний тип даних конфіденційним. Якщо функція sens(x_raw) повертає значення 0, дані вважаються неконфіденційними, і до них можуть застосовуватися спрощені механізми захисту. Якщо ж sens(x_raw)=1, вони відносяться до конфіденційної категорії, і алгоритм вмикає повний набір перетворень.

Для конфіденційних даних спочатку виконується мінімізація: відкидаються зайві атрибути, які не потрібні для цілей обробки. В деяких

випадках може застосовуватися агрегування або округлення значень, щоб уникнути надмірно детальної інформації. Після цього пристрій використовує сесійний ключ, отриманий під час автентифікації, для симетричного шифрування підготовлених даних. До зашифрованого корисного навантаження додаються службові метадані а також код автентичності (MAC), який дозволяє виявити спроби підміни або спотворення даних.

Сформований пакет містить мінімізований ідентифікатор пристрою, зашифровані та мінімізовані дані та метадані для контролю цілісності. Далі цей пакет передається до шлюзу. На стороні шлюзу виконується перевірка коректності формату та MAC. В разі успішної валідації шлюз за потреби виконує повторне шифрування на ключі хмарної платформи, фрагментацію і додавання власних псевдонімів та маршрутних міток. Після цього фрагменти відправляються до хмарних сервісів для подальшого зберігання та обробки.

Блок-схему алгоритму підготовки та передавання захищених повідомлень показано на рисунку 2.5.

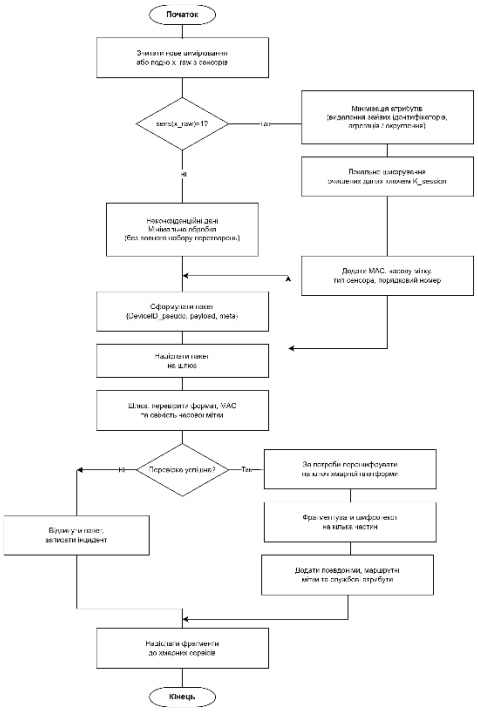


Рисунок 2.5 - Алгоритм підготовки та передавання захищених повідомлень IoT-пристрою

Третій алгоритм забезпечує приймання, валідацію, розподілене зберігання та ведення журналу фрагментів конфіденційних даних (рисунок 2.6).

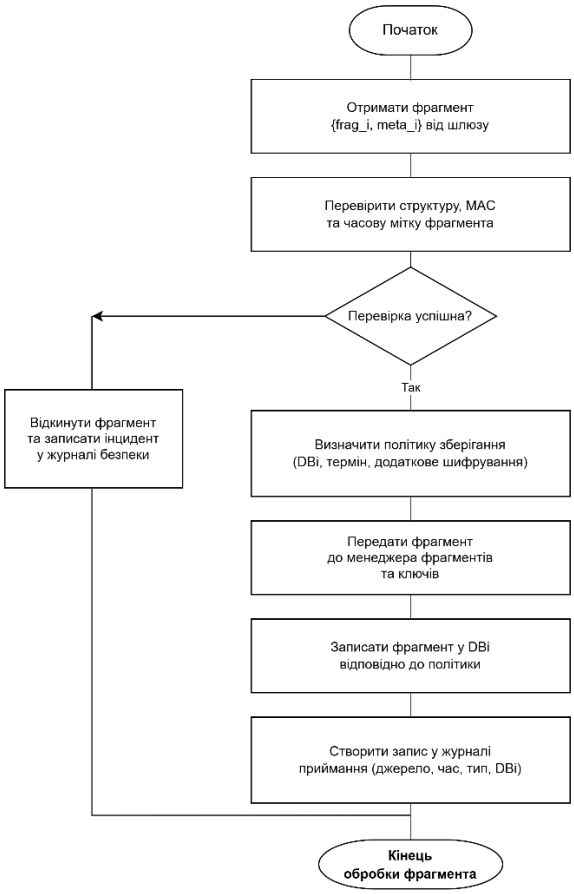


Рисунок 2.6 - Алгоритм обробки та зберігання даних на серверній стороні

Коли хмарний брокер або сервіс прийому отримує фрагмент зашифрованих даних, він спочатку перевіряє структуру повідомлення. Якщо формат не відповідає очікуваному шаблону, фрагмент відкидається і в журналі фіксується відповідна помилка. Якщо структура валідна, виконується перевірка коду автентичності або цифрового підпису та також оцінюється коректність часової мітки, що дозволяє протистояти повторному відтворенню старих повідомлень. Після успішної перевірки компонент політик визначає яке саме сховище DBi повинен бути записаний фрагмент. Це рішення приймається на основі типу даних, рівня конфіденційності, джерела та інших атрибутів. Далі фрагмент передається до менеджера фрагментів і ключів, який

відповідає за логічне групування фрагментів в записи, прив’язку їх до відповідних ключів та збереження індексів для швидкого пошуку. Фізичний запис фрагментів виконується у вибране сховище.

Паралельно створюється запис де фіксується джерело повідомлення, час, тип даних, цільове сховище, а також результат перевірок цілісності. В разі виявлення невідповідностей фрагмент не записується і формується запис про інцидент безпеки.

Четвертий алгоритм описує, як авторизований користувач або зовнішній сервіс отримує доступ до конфіденційних даних, зберігаючи при цьому всі гарантії конфіденційності. Ідея полягає в тому, що жодні чутливі дані не повертаються суб’єкту без попередньої автентифікації і успішного проходження перевірки політик ABAC.

Процес починається з формування запиту користувачем в клієнтському застосунку, де він задає тип потрібних даних, часовий інтервал, обсяг або інші параметри. Далі користувач проходить процедуру автентифікації, а служба автентифікації визначає його атрибути. На основі цих атрибутів і налаштованих політик, спеціалізований сервіс видає токен доступу, в якому зафіксовано дозволені типи операцій і обмеження за даними. Отримавши токен клієнтський застосунок формує запит до модуля політик. Потім він завантажує відповідні правила та контекстну інформацію і обчислює рішення Permit або Deny. Якщо доступ заборонено, користувач отримує відповідне повідомлення, а в журналі фіксується факт відмови. Якщо доступ дозволено, модуль політик формує запит до менеджера фрагментів про отримання необхідних фрагментів з відповідних сховищ. На наступному кроці робиться реконструкція потрібних записів. Далі виконується дешифрування або обчислення агрегованих показників. Перед формуванням відповіді застосовується додаткова мінімізація. Після цього підготовлений результат повертається клієнтському застосунку і в журналі доступу зберігається запис із зазначенням суб’єкта, часу, типу даних і характеру виконаної операції.

Блок-схему алгоритму надання доступу до даних авторизованим суб’єктам показано на рисунку 2.7.

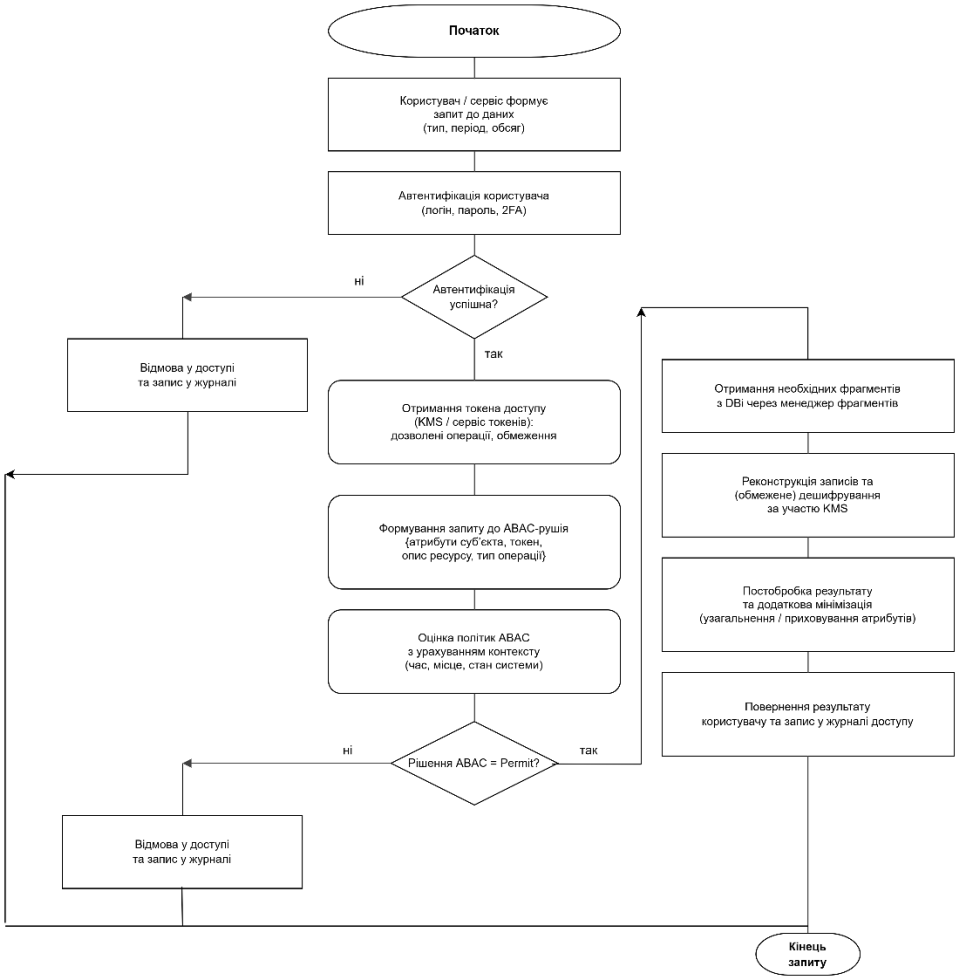


Рисунок 2.7 - Алгоритм надання доступу до конфіденційних даних авторизованим суб’єктам

Висновки до розділу 2

1. Побудовано формальну модель IoT-системи у вигляді кортежу множин пристроїв, користувачів, сервісів, каналів зв’язку, типів даних, ключів та політик доступу. Запропоновано функцію чутливості sens(x), що позначає конфіденційні елементи даних, а також описано процеси збору, передавання, зберігання та доступу до даних.

2. На основі формальної моделі запропоновано концепцію методу збереження конфіденційності, який забезпечує наскрізний захист чутливих даних на всіх етапах їх життєвого циклу.

3. Розроблено архітектуру методу, що поєднує легковагову апаратну автентифікацію пристроїв на основі PUF, динамічних псевдонімів та мінімізацію даних на IoT-вузлах.

4. Описано ключові алгоритми функціонування, до яких входять реєстрація та автентифікація пристрою, підготовка та передавання захищених повідомлень, обробка і зберігання фрагментованих даних на серверній стороні, а також надання доступу авторизованим суб’єктам.

5. Сформовано метод забезпечення приватності в IoT-системах, який поєднує формальну модель даних і загроз з архітектурними рішеннями та алгоритмами обробки. Запропонований підхід забезпечує наскрізний захист чутливих даних та зменшує ризики витоків.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА МЕТОДУ ЗБЕРЕЖЕННЯ КОНФІДЕНЦІЙНОСТІ В IoT-СИСТЕМАХ

3.1 Вибір програмно-апаратної платформи та інструментальних засобів

Обрана конфігурація повинна одночасно відображати типові умови функціонування IoT-систем та забезпечувати достатню гнучкість для реалізації криптографічних протоколів, механізмів фрагментації, мінімізації та керування доступом. Основні вимоги до експериментальної платформи можна поділити на три групи.

На рівні IoT-пристроїв необхідна підтримка бездротового зв’язку, базових криптографічних примітивів а також можливість роботи в умовах обмежених обчислювальних ресурсів, енергоспоживання та пам’яті.

На рівні шлюзу і серверної частини потрібна платформа, здатна обробляти потоки повідомлень від кількох пристроїв, реалізовувати брокер повідомлень, модуль керування ключами, сервер автентифікації та авторизації, а також компоненти ведення журналу подій, моніторингу та візуалізації.

Набір інструментальних засобів, які забезпечують реалізацію криптографічних алгоритмів, мережевих протоколів, роботи з базами даних та інтерфейсів доступу до результатів.

В ролі базового IoT-вузла можна використати платформу на основі ESP32. Дана платформа має наступні характеристики:

– інтегрований Wi-Fi та Bluetooth;

–двоядерний процесор, для виконання нескладних криптографічних операцій на стороні пристрою;

– підтримка середовища Arduino та офіційного SDK ESP-IDF;

Набір сенсорів для експериментальної установки може включати датчики температури та вологості повітря, освітленості, цифрові або аналогові сенсори руху. Конкретні типи фізичних величин в даному випадку не є

критичними, оскільки метою є перевірка того, як запропонований метод трансформує та захищає різні категорії даних залежно від їхньої конфіденційності.

Програмування ESP32 здійснюється мовою C/C++ у середовищі Arduino IDE. Такий вибір обумовлений наявністю великої кількості готових бібліотек для роботи з сенсорами, мережевими стеками.

Як проміжну ланку між ресурсно-обмеженими IoT-вузлами та хмарною інфраструктурою можна використати Raspberry Pi. Ця плата має достатню обчислювальну потужність, підтримує повноцінну операційну систему на базі Linux і забезпечує стабільний мережевий зв’язок через Ethernet та Wi-Fi.

Raspberry Pi виконує роль IoT-шлюзу та компактного серверного вузла, на якому розгортаються:

– брокер повідомлень для приймання MQTT-повідомлень від ESP32 і передачі їх далі до серверних сервісів;

– веб-сервіс або REST API, що реалізує інтерфейси для модулів керування ключами, авторизації та доступу до даних;

– база даних для зберігання журналів подій, метаданих про пристрої, користувачів та політики доступу;

– допоміжні сервіси моніторингу та візуалізації.

Для організації обміну даними між ESP32 та шлюзом використовується протокол MQTT як транспортний протокол для передачі вже захищених повідомлень. Це дозволяє чітко розділити відповідальність:

– на рівні застосунку де реалізується криптографічна логіка та перетворення даних згідно із запропонованим методом;

– на мережевому рівні MQTT забезпечує маршрутизацію повідомлень, буферизацію та базовий надійний обмін.

Узагальнену схему мережевої взаємодії між ESP32, брокером MQTT, серверними компонентами та клієнтом адміністратора наведено на рисунку 3.1.

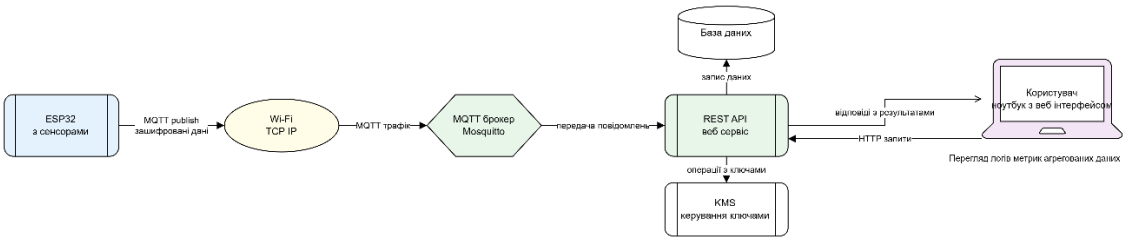


Рисунок 3.1 – Схема мережевої взаємодії компонентів експериментального стенду

На шлюзі розгортається брокер Eclipse Mosquitto, який приймає публікації від ESP32 та в залежності від варіанту архітектури, або зберігає їх локально, або пересилає на інші сервіси.

Серверні компоненти, що реалізують логіку методу збереження конфіденційності будуть реалізуватись на Python. Цей вибір обумовлений:

– наявністю веб-фреймворків, що дозволяють швидко створювати REST-сервіси;

– широким набором криптографічних бібліотек для реалізації симетричного шифрування, електронного підпису, HMAC, генерації ключів;

– підтримкою драйверів для SQLite, що спрощує експерименти з різними схемами зберігання метаданих і журналів;

– можливістю інтеграції з інструментами моніторингу та збору метрик.

В ролі баз даних икористано SQLite для локальних експериментів та PostgreSQL, для сценаріїв з більшою кількістю записів та складнішими транзакціями.

Для реалізації протоколу авторизації та політик ABAC використані спеціалізовані бібліотеки, що працюють із таблицями політик в СУБД..

Для моніторингу та ведення журналу можна застосувати стандартні засоби логування Python з виведенням ключових подій в базу даних. За потреби ці журнали можуть бути інтегровані з зовнішніми засобами візуалізації.

Обрана програмно-апаратна платформа (рисунок 3.2) поєднує в собі ESP32 як IoT-вузол, Raspberry Pi як шлюзу або серверного вузла, MQTT як

транспортного протоколу, Python-сервісів з використанням криптографічних та мережевих бібліотек і реляційної бази даних – забезпечує необхідний баланс між наближеністю до реальних IoT-систем і можливістю гнучкого налаштування для дослідження різних конфігурацій запропонованого методу збереження конфіденційності.

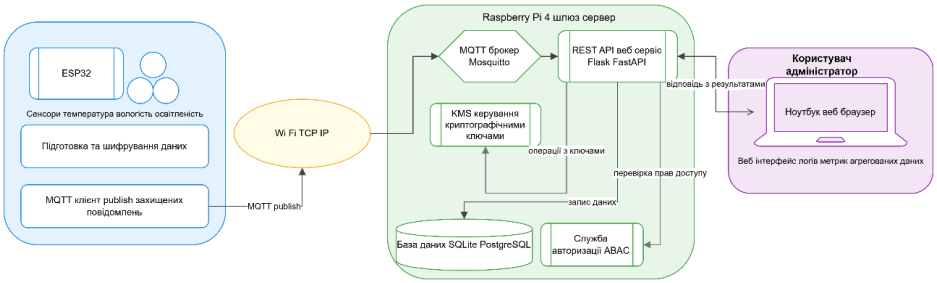


Рисунок 3.2 – Загальна архітектура експериментального стенду

Узагальнено обрану конфігурацію можна розглядати, як багаторівневий програмно-апаратний стек, що включає апаратний рівень ESP32 та Raspberry Pi, системний і мережевий рівні з підтримкою Wi-Fi, TCP/IP та MQTT, прикладний рівень IoT- та безпекових сервісів, а також рівень управління і моніторингу з веб-інтерфейсом для адміністратора, як представлено на рисунку 3.3.

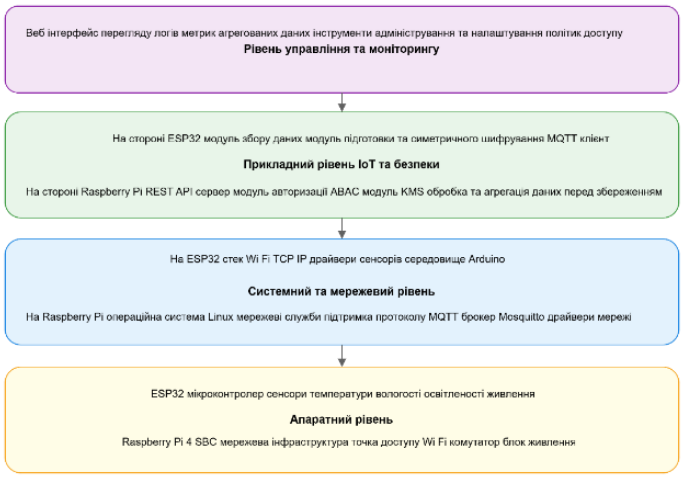


Рисунок 3.3 – Рівнева схема програмно-апаратного стеку експериментального стенду

3.2 Реалізація програмних модулів

Реалізація програмних модулів базується на архітектурних рішеннях, сформованих в попередніх розділах, та враховує обмежені обчислювальні ресурси IoT-пристроїв, необхідність мінімізації накладних витрат і забезпечення відтворюваності експериментів.

Модуль шифрування та підготовки даних (рисунок 3.4) розгорнуто безпосередньо на вузлі ESP32, який виконує роль типового IoT-пристрою з підключеними сенсорами температури, вологості та освітленості. Програмна частина реалізована мовою C/C++ у середовищі Arduino, що забезпечує доступ до бібліотек роботи з периферією, таймерами та мережевим стеком Wi-Fi.

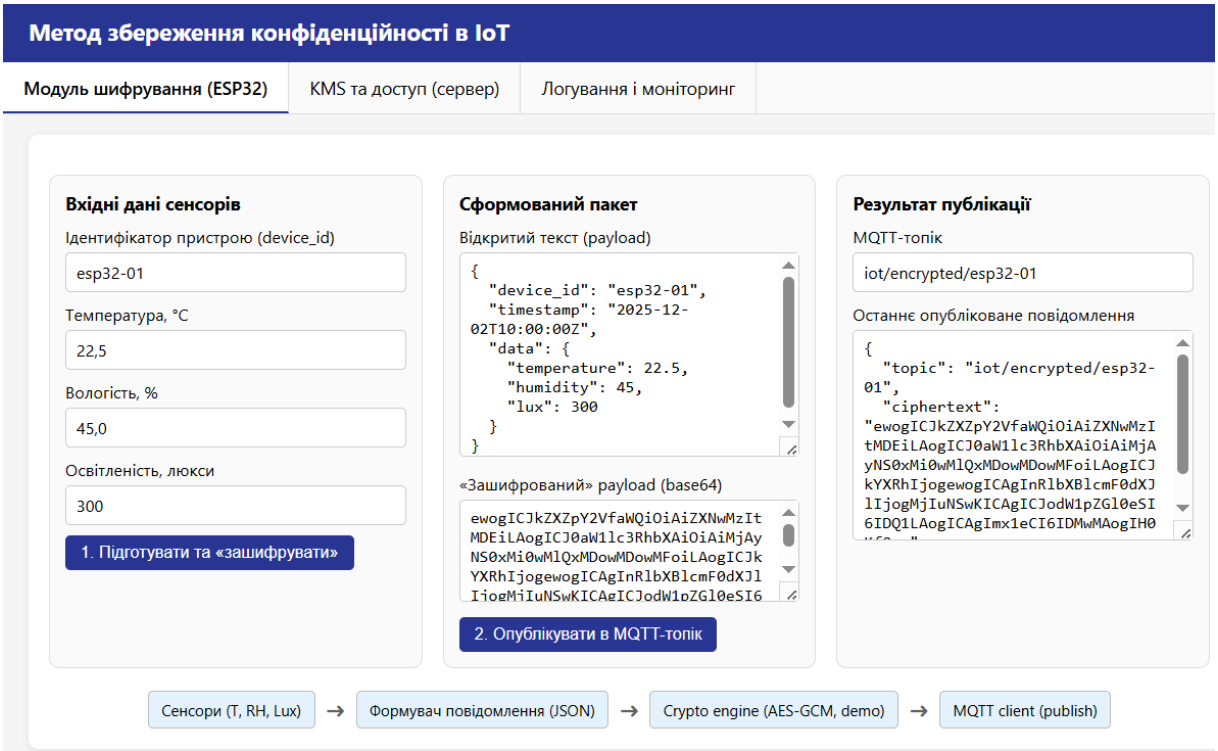


Рисунок 3.4 – Модуль шифрування та підготовки даних

Логіка роботи модуля відповідає алгоритму підготовки та передавання захищених повідомлень, сформульованому вище. На кожній ітерації циклу опитування модуль:

– зчитує поточні значення сенсорів;

– нормалізує їх до узгодженого формату;

– формує структурований об’єкт даних, що включає device_id, часову мітку, номер сеансу і лічильник повідомлень, а також значення вимірювань.

Для представлення корисного навантаження використовується компактний JSON-подібний формат або бінарна структура, що полегшує подальший парсинг на сервері. На основі цієї структури формується буфер відкритого тексту, який передається у криптографічний підмодуль. З врахуванням обмежень ESP32.

Криптографічні ключі та допоміжні параметри зберігаються у захищеній ділянці пам’яті контролера і завантажуються під час ініціалізації. Для кожного повідомлення генерується унікальний вектор ініціалізації (nonce), який включається до заголовка, що потім дозволяє серверу правильно розшифрувати пакет та виявляти повторну відправку (replay).

Після виконання шифрування формується MQTT-повідомлення, що містить:

– заголовок із device_id, nonce, номером повідомлення;

– зашифроване корисне навантаження;

– тег автентифікації.

Сформований пакет публікується в заздалегідь визначений MQTT-топік, рівень QoS обирається з врахуванням компромісу між надійністю доставлення та затримками. На шлюзі або проміжному пристрої (Raspberry Pi) може бути реалізована функція повторного шифрування чи тунелювання даних в разі потреби каскадного захисту, однак базова реалізація зосереджується на шифруванні на кінцевому вузлі.

Додатково модуль підтримує процедуру ініціалізації та перевірки конфігурації. При першому запуску або після скидання він виконує коротку взаємодію з сервером для підтвердження свого статусу та отримання ідентифікатора політики, термін дії ключа.

Серверна частина методу реалізована на Raspberry Pi 4, що виконує роль шлюзу й основного серверного вузла. Архітектурно вона поділяється на веб-/REST-сервіс і модуль керування ключами та доступом (KMS + авторизація).

Модуль KMS відповідає за (рисунок 3.5):

– генерацію криптографічних ключів для пристроїв (на основі надійного генератора випадкових чисел операційної системи);

– їх зберігання у зашифрованому вигляді в базі даних (наприклад, SQLite або PostgreSQL з шифруванням на рівні файлу чи таблиці);

– прив’язку ключів до device_id, ролей, політик доступу та термінів дії;

– обслуговування запитів на ротацію ключів та перевірку їх актуальності.

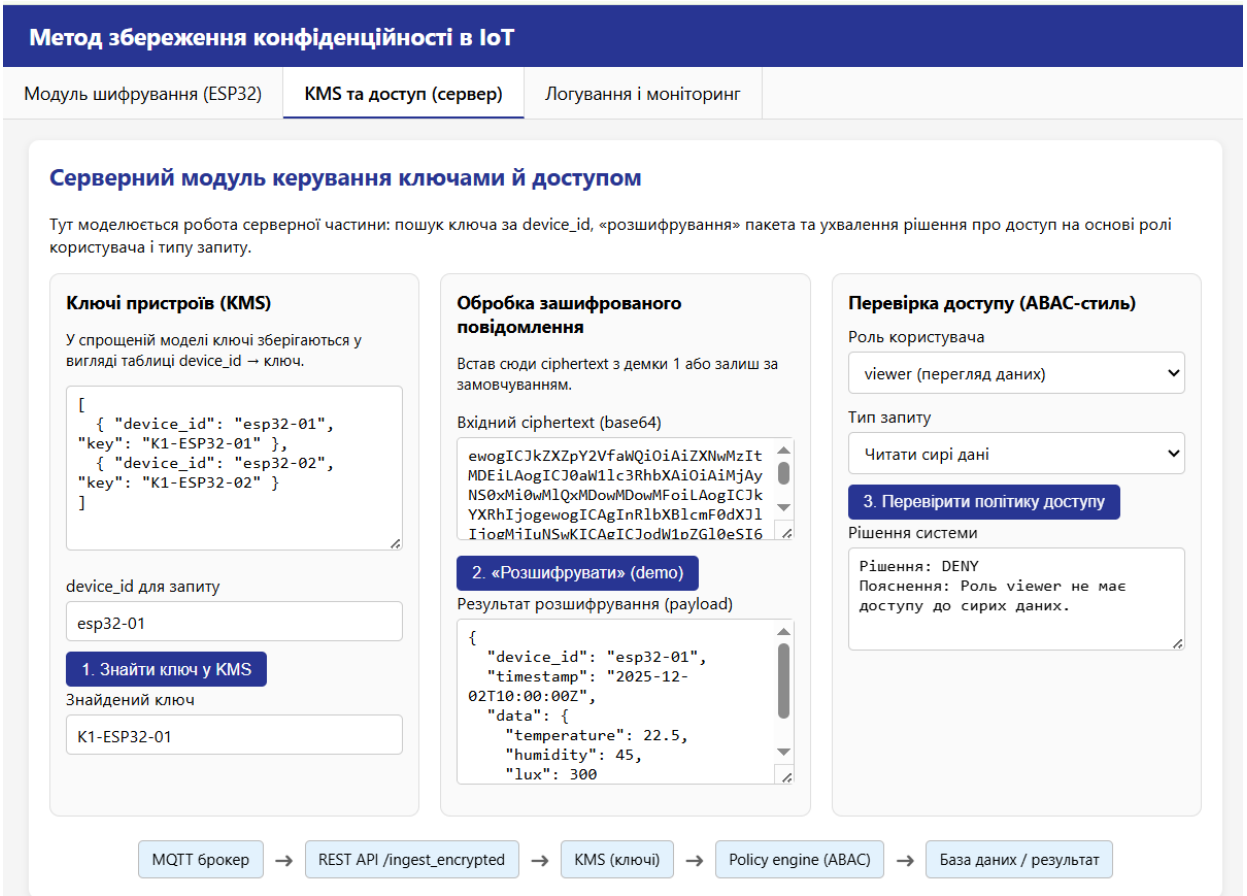


Рисунок 3.5 – Модуль керування ключами та доступом в IoT-системі

REST-сервіс надає набір кінцевих точок, що реалізують алгоритми, сформульовані в розділі 2. Зокрема:

/register_device – приймання запитів на реєстрацію нових пристроїв, верифікація параметрів, запис відповідних записів у БД та ініціалізація ключів;

/ingest_encrypted – приймання зашифрованих MQTT-повідомлень, витягнення заголовка, пошук ключа для device_id, розшифрування та перевірка тегу автентифікації;

/data – надання доступу до агрегованих або первинних даних авторизованим користувачам згідно з політиками доступу.

Механізм контролю доступу реалізовано у вигляді атрибутно-орієнтованої моделі (ABAC), де рішення про надання доступу ухвалюється на підставі набору атрибутів: ролі користувача, категорії даних, часових обмежень, рівня конфіденційності тощо. В реалізації це набір правил або політик, оформлених у вигляді конфігураційних записів у БД або окремих JSON-файлах. REST-сервіс перевіряє авторизаційні токени, витягує атрибути суб’єкта та об’єкта доступу, після чого звертається до модуля ухвалення рішень, який повертає результат “дозволити/заборонити/обмежити”.

Всі операції з ключами логуються у захищеному журналі. Доступ до інтерфейсів керування ключами обмежено окремою роллю адміністратора безпеки.

Модуль логування і моніторингу реалізований як окремий сервіс, що централізовано збирає, структурує та зберігає записи про події.

Кожен лог-запис має уніфікований формат, наприклад:

– час події у локальному часі з часовим поясом;

– джерело IoT-пристрій, REST-сервіс, KMS, модуль авторизації;

– тип події що включає інформацію про аутентифікацію, відмову в доступі, помилку розшифрування, порушення цілісності, підозру на повторне повідомлення, помилки ротації ключа, тощо;

– критичність (info, warning, error, security);

– додаткові атрибути такі як device_id, IP-адреса, фрагмент повідомлення.

На сервері використовується стандартний логувальний механізм з конфігурацією ротації файлів і, за потреби, дублюванням записів у окрему таблицю БД. Для зручності аналізу подій передбачено простий веб-інтерфейс (рисунок 3.6), який дозволяє:

– фільтрувати події за типом, критичністю, device_id або часовим інтервалом;

– переглядати статистичні агрегати;

– візуалізувати динаміку подій у часі.

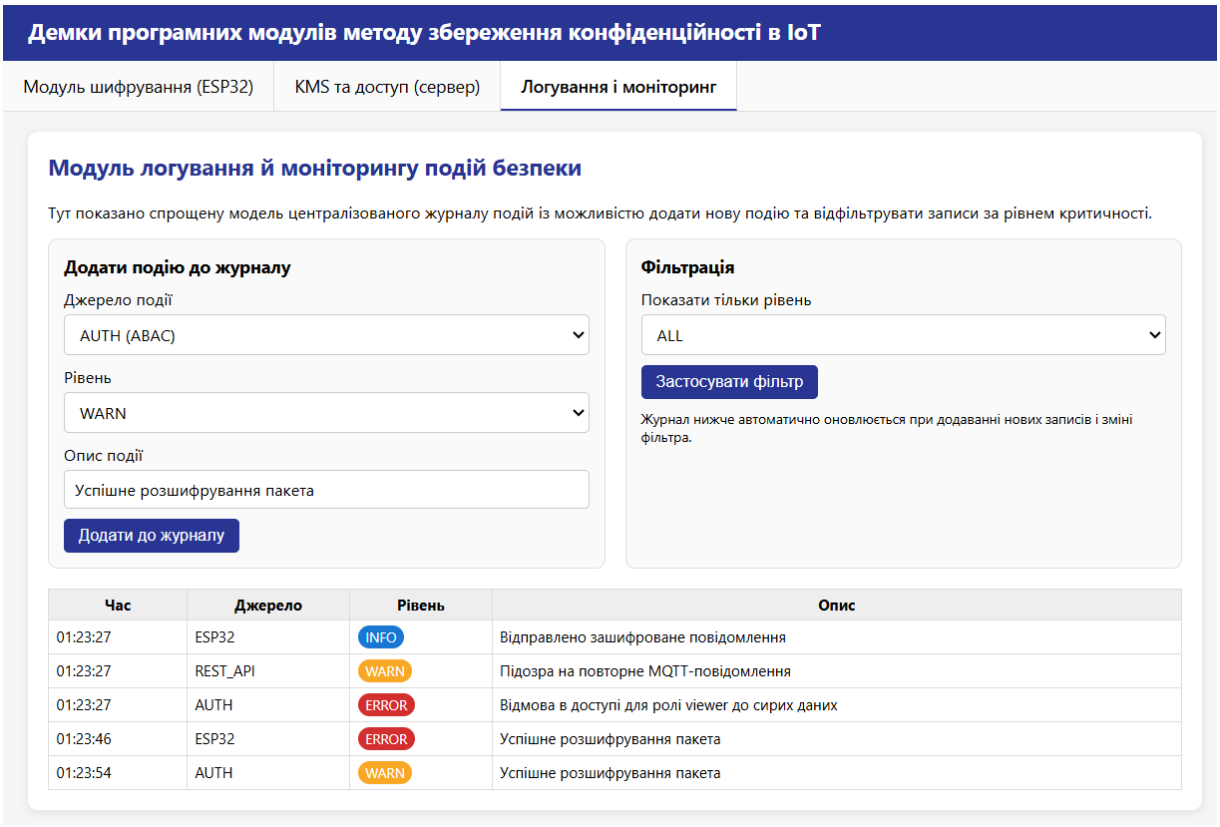


Рисунок 3.6 – Веб-інтерфейс аналізу подій

Завдяки поєднанню описаних трьох модулів – шифрування та підготовки даних на IoT-пристрої, серверного керування ключами та доступом, а також логування і моніторингу – реалізується повний цикл роботи запропонованого методу збереження конфіденційності, який придатний як для демонстрації, так і для кількісної оцінки впливу механізмів захисту на продуктивність і безпеку IoT-системи в цілому.

3.3 Сценарії оцінки методу

Експериментальна оцінка запропонованого методу збереження конфіденційності в IoT-системах спрямована на дві категорії:

– аналіз впливу механізмів захисту на продуктивність системи;

– оцінювання того, наскільки ці механізми реально ускладнюють або роблять неможливими типові атаки на конфіденційність даних.

Для цього формуються групи сценаріїв роботи системи, задаються кількісні метрики та обираються базові рішення, з якими порівнюється запропонований метод.

Так як базовою платформою для експериментів є стенд, який складається з декількох вузлів ESP32 з сенсорами, шлюзу на Raspberry Pi з брокером MQTT, REST-сервісом, модулем керування ключами та авторизаціцією, а також модулем логування. На цьому стенді задаються такі основні сценарії:

1) В роботі бере участь один вузол ESP32, який надсилає захищені вимірювання з фіксованим інтервалом 5–10 секунд. Такий сценарій моделює типову домашню або лабораторну IoT-систему з невеликим навантаженням.

2) До системи додаються кілька вузлів, 5–10 ESP32, що надсилають повідомлення з інтервалом 1–5 секунд. Це дозволяє змоделювати невелику мережу датчиків.

3) Передавання повідомлень максимізується шляхом зменшення інтервалу до часток секунди або емулювання більшої кількості пристроїв програмно.

4) На вузлах ESP32 генеруються повідомлення різної довжини Це дозволяє оцінити, як шифрування й передавання впливають на затримку та пропускну здатність для різних обсягів даних.

5) В середині окремої групи експериментів моделюються типові загрози конфіденційності, штучно генеруються повторні або модифіковані повідомлення з підробленими заголовками;

Далі для кількісного аналізу експериментів застосовується набір метрик, що характеризують як продуктивність системи, так і рівень захищеності.

Вимірюється час між моментом формування повідомлення на ESP32 та його успішним занесенням у базу даних на сервері. На основі цих даних обчислюється середня затримка, медіана та, за потреби, розподіл затримок для різних сценаріїв навантаження. Залежність середньої затримки від кількості пристроїв для базового рішення та запропонованого методу наведено на рисунку 3.7.

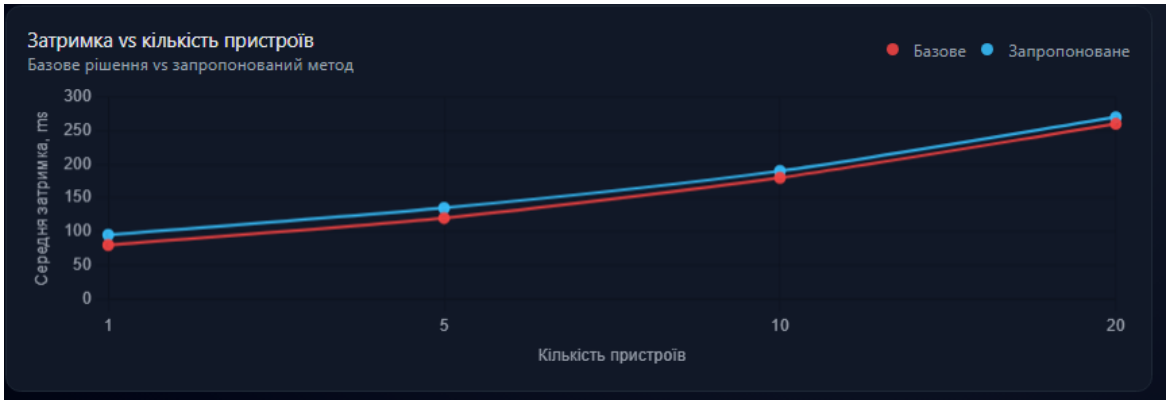


Рисунок 3.7 – Залежність середньої затримки від кількості пристроїв для базового рішення та запропонованого методу

Також оцінюється кількість успішно оброблених зашифрованих повідомлень за одиницю часу. Для кожного сценарію фіксується максимальна стійка пропускна здатність, при якій система не починає втрачати повідомлення або різко збільшувати затримку.

Вплив масштабування кількості IoT-пристроїв на пропускну здатність системи для базового рішення та запропонованого методу показано на рисунку 3.8.

Рисунок 3.8 – Зміна пропускної здатності при масштабуванні кількості IoT-пристроїв

На стороні серверного вузла збираються показники завантаження CPU та використання оперативної пам’яті під час різних сценаріїв. Це здійснюється за допомогою стандартних інструментів Linux, таких як htop та через програмний збір метрик psutil.

Крім того для IoT-пристроїв також необхідно, щоб додаткові обчислення не приводили до скорочення часу автономної роботи. За наявності відповідного обладнання енергоспоживання можна вимірювати за струмом споживання в різних режимах роботи або на основі режимів енергоспоживання, наведених у документації на ESP32, та тривалості активних фаз обробки.

Для атаки повторного відправлення можна ввести показник успішності як частку підроблених повідомлень, які система прийняла, від загальної кількості спроб. Для аналізу трафіку застосовується точність класифікації, з якою умовний порушник, маючи лише зашифрований трафік і часові мітки, намагається розрізнити різні типи подій, наприклад звичайні вимірювання та особливі події. Чим ближчою до випадкового вгадування є така точність, тим вищим є рівень конфіденційності. Узагальнене порівняння ймовірності успішної атаки на конфіденційність для базового рішення та запропонованого методу подано на рисунку 3.9.

Рисунок 3.9 – Порівняння ймовірності успішної атаки на конфіденційність для базового рішення та запропонованого методу

Щоб результати експериментів були інтерпретованими, запропонований метод порівнюється з кількома базовими варіантами конфігурації системи.

При базовому випадку ESP32 надсилає незашифровані повідомлення MQTT, а серверна частина приймає їх без додаткової аутентифікації та криптографічної обробки. Такий сценарій дозволяє оцінити мінімально можливі затримки та пропускну здатність.

В іншому варіанті коли використовується TLS-з’єднання між брокером MQTT і сервером, але повідомлення не шифруються на рівні застосунку.

Крім того є спрощений варіант схеми взаємної автентифікації та захисту конфіденційності для IoT де порівняння проводиться за продуктивністю і за функціональними можливостями.

Варіанти порівняння трьох базових рішень та запропонованого методу за середньою затримкою і ймовірністю успішної атаки наведено на рисунку 3.10.

Рисунок 3.10 – Порівняння базових рішень та запропонованого методу за показниками затримки та ймовірності успішної атаки

На основі результатів, отриманих за всіма сценаріями та метриками, можна проводити аналіз компромісу між рівнем конфіденційності,

продуктивністю системи та складністю реалізації запропонованого методу в реальних IoT-середовищах.

Висновки до розділу 3

1. Реалізовано повноцінний експериментальний стенд на базі платформи ESP32 та шлюзу Raspberry Pi з використанням протоколу MQTT, REST-сервісів. Така конфігурація відтворює типові умови функціонування IoT-систем і забезпечує гнучкість налаштування.

2. Реалізовано метод збереження конфіденційності у вигляді трьох взаємопов’язаних модулів куди входять модуль шифрування і підготовки даних, серверного модуля керування ключами та модуля централізованого логування і моніторингу подій.

3. Побудовано набір сценаріїв експериментальної оцінки, які покривають різні режими навантаження, що дозволило дослідити поведінку системи як у нормальному режимі, так і за наявності порушника.

4. Застосовано систему кількісних метрик, яка включає середню затримку, пропускну здатність, завантаження процесора і пам’яті.

5. Проведено порівняння запропонованого методу з базовими рішеннями які включають незахищений MQTT-трафік, використання лише транспортного TLS-захисту без прикладного шифрування та з спрощеною схемою автентифікації.

ВИСНОВКИ

1. Проведено аналіз архітектур систем Інтернету речей та загроз конфіденційності даних. Показано, що дані проходять через багато проміжних вузлів, кожен з яких може стати точкою витоку, а наявні підходи зосереджуються переважно на автентифікації і встановленні ключів, не забезпечуючи наскрізного контролю конфіденційності на всьому шляху руху даних.

2. Побудовано формальну модель IoT-системи у вигляді кортежу множин пристроїв, користувачів, сервісів, каналів зв'язку, типів даних, ключів і політик доступу, а також введено модель витоків.

3. Розроблено концепцію та архітектуру методу збереження конфіденційності, який забезпечує багаторівневий захист даних від моменту збору на сенсорах до зберігання й аналітики на серверній стороні. Метод реалізовано у вигляді трьох взаємопов'язаних програмних модулів шифрування та підготовки даних на IoT-пристрої, серверного керування ключами та політиками доступу, а також цлогування і моніторингу подій.

4. Створено експериментальний стенд на базі вузлів ESP32 із сенсорами та шлюзу Raspberry Pi з брокером MQTT і REST-сервісами, на якому реалізовано прототип запропонованого методу.

5. Розроблено методику експериментальної оцінки, що включає набір сценаріїв навантаження і атак, систему кількісних метрик та три базові варіанти конфігурації системи.

6. Експериментальні дослідження показали, що запропонований метод в порівнянні з базовими рішеннями знижує ймовірність успішної атаки на конфіденційність, у тому числі при повторному надсиланні та аналізі трафіку, при цьому зберігаючи прийнятні значення середньої затримки та пропускної здатності для типових IoT-навантажень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sharma P. et al. A Survey of Methods Guaranteeing User Privacy Based on Blockchain in Internet-of-Things. IEEE Access. 2019.

2. Safaei Yaraziz, M., Jalili, A., Gheisari, M., & Liu, Y. (2023). Recent trends towards privacy-preservation in Internet of Things, its challenges and future directions. IET circuits, devices & systems, 17(2), 53-61.

3. Hindy H. et al. An MQTT Case Study (MQTT-IoT-IDS2020 Dataset). arXiv preprint arXiv:2006.15340. 2020.

4. Alotaibi B., Alotaibi M. A Stacked Deep Learning Approach for IoT Cyberattack Detection. Journal of Sensors. 2020.

5. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer networks, 76, 146-164.

6. Rejeb, A., Rejeb, K., Simske, S. J., & Keogh, J. G. (2022). Blockchain technology in the smart city: A bibliometric review. Quality & quantity, 56(5), 2875-2906.

7. Gope, P., & Sikdar, B. (2018). Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet of Things Journal, 6(1), 580-589.

8. Yuhala, P., Ménétrey, J., Felber, P., Pasin, M., & Schiavoni, V. (2024, April). Fortress: Securing IoT Peripherals with Trusted Execution Environments. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (pp. 243-250).

9. Ameer, S., Benson, J., & Sandhu, R. (2022). An attribute-based approach toward a secured smart-home IoT access control and a comparison with a role-based approach. Information, 13(2), 60.

10. Wakili, A., & Bakkali, S. (2025). Privacy-preserving security of IoT networks: A comparative analysis of methods and applications. Cyber Security and Applications, 3, 100084.

11. Ferrag, M. A., Maglaras, L. A., Janicke, H., Jiang, J., & Shu, L. (2017). Authentication protocols for internet of things: a comprehensive survey. Security and Communication Networks, 2017(1), 6562953.

12. Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S., & Fang, B. (2020). A survey on access control in the age of internet of things. IEEE Internet of Things Journal, 7(6), 4682-4696.

13. Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. IEEE Communications Surveys & Tutorials, 21(3), 2702-2733.

14. Chakrabarty, S., & Engels, D. W. (2020, December). Secure smart cities framework using IoT and AI. In 2020 IEEE global conference on artificial intelligence and Internet of Things (GCAIoT) (pp. 1-6). IEEE..

15. Fantacci, R., Nizzi, F., Pecorella, T., Pierucci, L., & Roveri, M. (2019). False data detection for fog and internet of things networks. Sensors, 19(19), 4235.

16. Rasori M., La Manna M., Perazzo P., Dini G. A survey on attribute-based encryption schemes suitable for the Internet of Things. IEEE Internet of Things Journal. 2022. Vol. 9, No. 11. Pp. 8269–8290. DOI: 10.1109/JIOT.2022.3154039.

17. Bhattacharjya, A., Zhong, X., Wang, J., & Li, X. (2019). CoAP—application layer connection-less lightweight protocol for the Internet of Things (IoT) and CoAP-IPSEC Security with DTLS Supporting CoAP. In Digital twin technologies and smart cities (pp. 151-175). Cham: Springer International Publishing.

18. Spina, M. G., De Rango, F., & Marotta, G. M. (2021, September). Lightweight dynamic topic-centric end-to-end security mechanism for MQTT. In 2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT) (pp. 1-7). IEEE.

19. Chien, H. Y., & Ciou, P. P. (2023). Design and implementation of efficient IoT authentication schemes for MQTT 5.0. Journal of Internet Technology, 24(3), 665-674.

20. Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2017). Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of things journal, 5(4), 2483-2495.

21. Kowta, A. S. L., Harida, P. K., Venkatraman, S. V., Das, S., & Priya, V. (2022, February). Cyber security and the Internet of Things: vulnerabilities, threats, intruders, and attacks. In Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2021 (pp. 387-401). Singapore: Springer Nature Singapore.

22. Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future generation computer systems, 82, 395-411.

23. Aldawira, C. R., Putra, H. W., Hanafiah, N., Surjarwo, S., & Wibisurya, A. (2019). Door security system for home monitoring based on ESp32. Procedia Computer Science, 157, 673-682.

24. Ataullah, M., & Chauhan, N. (2024). Exploring security and privacy enhancement technologies in the Internet of Things: A comprehensive review. Security and Privacy, 7(6), e448.

25. Hu, F. (2016). Security and privacy in Internet of things (IoTs): Models, Algorithms, and Implementations. CRC Press.

26. Кузнєцов, Д. І., & Рябчина, Л. С. (2019). Інформаційна безпека систем інтернету речей. Вісник Криворізького національного університету, (49), 80-84.

27. Журило, О., Ляшенко, О., & Аветісова, К. (2023). Огляд рішень з апаратної безпеки кінцевих пристроїв туманних обчислень у Інтернеті речей. Сучасний стан наукових досліджень та технологій в промисловості, (1 (23)), 57-71.

28. Малохвій Е. Е., Молчанов Г. І. Дослідження протоколів передачі даних в умовах Інтернету речей, *Системи управління, навігації та зв'язку* : зб. наук. пр. 2022. № 1(67). С. 66–74.

29. Сотник, І. М., Сотник, И. Н., Завражний, К. Ю., & Завражный, К. Ю. (2017). Підходи до забезпечення інформаційної безпеки промислового Інтернету речей на підприємстві.

30. Чікін, Д. М., Науменко, С. В., & Розломій, І. О. (2025). Захист персональних даних в iot-пристроях із застосуванням штучного інтелекту. Тези доповідей, 12.

31. Підлісний, Ю. І. Шляхи підвищення конфіденційності в мережах інтернету речей / Ю. І. Підлісний // Центральноукраїнський науковий вісник. Технічні науки : наук. зб. - Кропивницький : ЦНТУ, 2025. - Вип. 11(42). - Ч. 1. - С. 46-55.

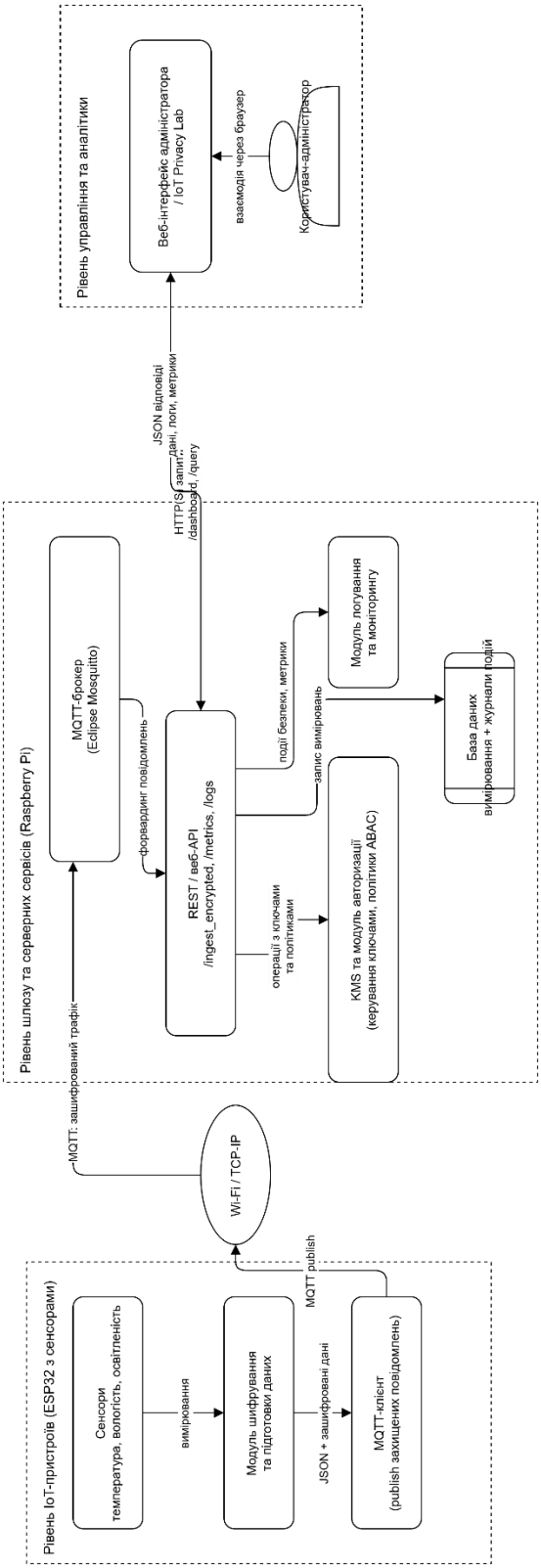
32. Гончаров Ю.В., Осолінський О. Р., Метод забезпечення приватності даних в IoT-системах, X Міжнародної мультидисциплінарної студентської наукової конференції «Теоретичне та практичне застосування результатів сучасної науки», яка відбулася 12 грудня 2025 року у місті Запоріжжя, Україна., С.515-517.

33. Гончаров Ю.В., Осолінський О. Р., Алгоритм збереження конфіденційності в інтернеті речей, V Міжнародної наукової конференції «Технології та суспільство: взаємодія, вплив, трансформація», яка відбулася 12 грудня 2025 року у місті Кропивницький., С. 241-244.

34. Комар М.П., Саченко А.О., Васильків Н.М., Загородня Д.І. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки» спеціальності 122 «Комп’ютерні науки» за другим (магістерським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2024. – 32 с.

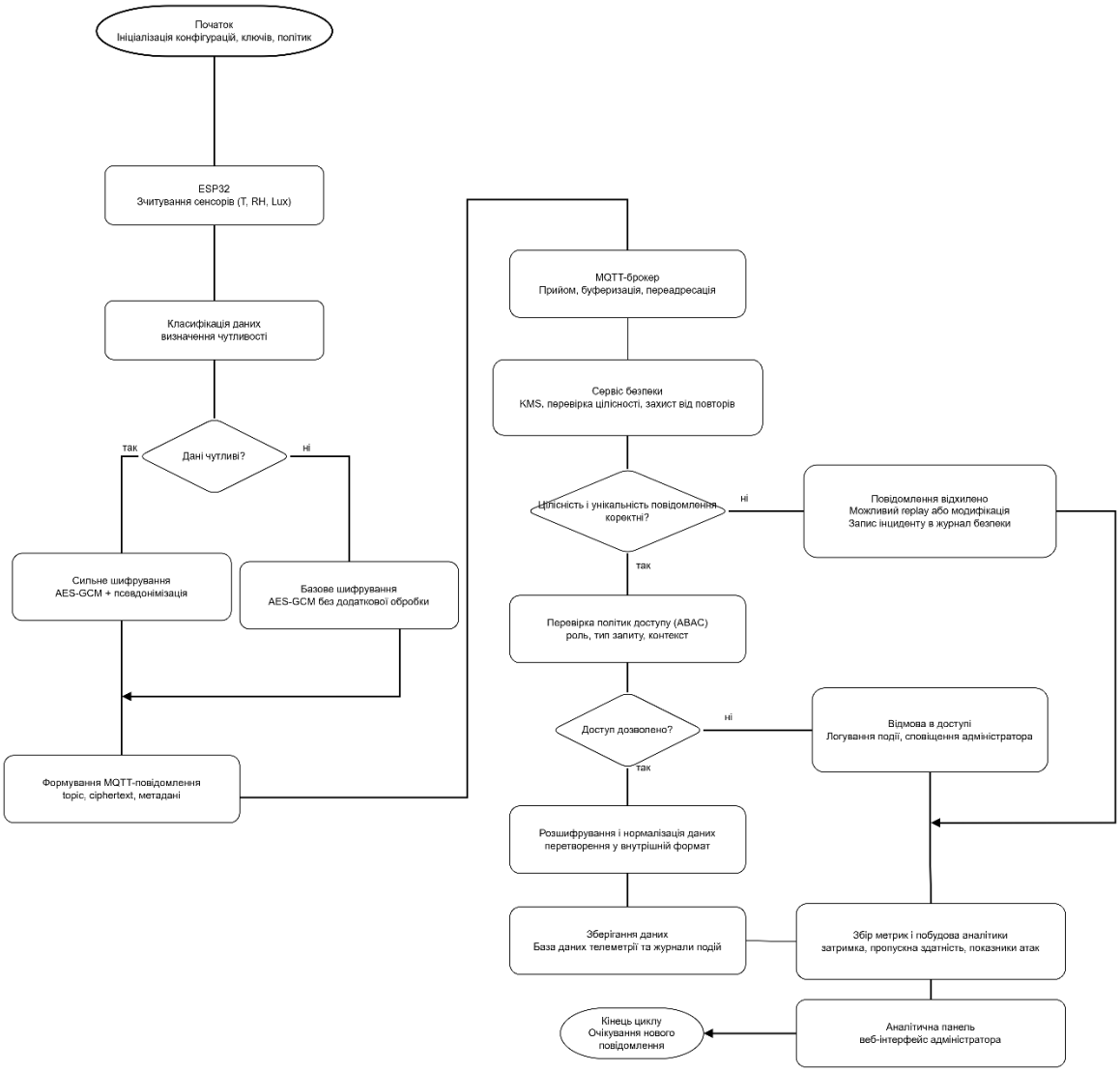
Додаток А

Загальна архітектури системи



Додаток Б

Загальний алгоритм роботи всіх модулів



Додаток В

Код програми модуля підготовки та передавання захищених повідомлень IoT-пристрою

```python
import time
import json
import hmac
import hashlib
from dataclasses import dataclass, asdict
from typing import Any, Dict
import paho.mqtt.client as mqtt
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

SENSITIVITY_PUBLIC = 0
SENSITIVITY_CONFIDENTIAL = 1
DATA_POLICY = {
    "temperature": SENSITIVITY_CONFIDENTIAL,
    "humidity": SENSITIVITY_CONFIDENTIAL,
    "device_status": SENSITIVITY_PUBLIC,
}
DEVICE_ID = "device-001"
SESSION_KEY = get_random_bytes(32)  # ключ AES-256 на поточний сеанс
MAC_KEY = get_random_bytes(32)      # ключ для HMAC

MQTT_BROKER = "broker.example.com"
MQTT_PORT = 1883
MQTT_TOPIC = "iot/protected"
@dataclass
class Measurement:
    """Структура сирого вимірювання (x_raw)."""
    data_type: str          # тип даних: temperature, humidity тощо
    value: float            # числове значення
    timestamp: float        # час вимірювання
    device_id: str          # ідентифікатор пристрою
    location: str           # локація (може бути чутлива)

def sens(meas: Measurement) -> int:
    """
    Функція чутливості sens(x).
    Повертає 1, якщо дані конфіденційні, і 0 - якщо публічні.
    """
    sensitivity = DATA_POLICY.get(meas.data_type, SENSITIVITY_CONFIDENTIAL)
    return sensitivity

def minimize_measurement(meas: Measurement) -> Dict[str, Any]:
    rounded_ts = int(meas.timestamp // 60 * 60)

    minimized = {
        "type": meas.data_type,
        "value": round(meas.value, 2),
        "ts": rounded_ts,
```

```python
        # device_id буде замінений на псевдонім на шлюзі
    }
    return minimized

def encrypt_payload(payload: bytes, session_key: bytes) -> Dict[str, Any]:

    cipher = AES.new(session_key, AES.MODE_GCM)
    ciphertext, tag = cipher.encrypt_and_digest(payload)
    return {
        "nonce": cipher.nonce.hex(),
        "ciphertext": ciphertext.hex(),
        "tag": tag.hex(),
    }


def compute_mac(message: bytes, mac_key: bytes) -> str:
    """Обчислення коду автентичності повідомлення (MAC) на базі HMAC-SHA256."""
    mac = hmac.new(mac_key, message, hashlib.sha256).hexdigest()
    return mac

def build_packet(meas: Measurement) -> Dict[str, Any]:

    sensitivity = sens(meas)

    minimized = minimize_measurement(meas)

    header = {
        "device_id": DEVICE_ID,
        "sensitivity": sensitivity,
        "timestamp": int(time.time()),
    }

    if sensitivity == SENSITIVITY_CONFIDENTIAL:

        payload_bytes = json.dumps(minimized).encode("utf-8")
        enc = encrypt_payload(payload_bytes, SESSION_KEY)

        packet = {
            "header": header,
            "encrypted_payload": enc,
            "meta": {
                "algo": "AES-256-GCM",
                "format": "json",
            }
        }
```

```
    else:
        packet = {
            "header": header,
            "payload": minimized,
            "meta": {
                "algo": "PLAINTEXT",
                "format": "json",
            }
        }

    # MAC для всього пакета
    packet_bytes = json.dumps(packet, sort_keys=True).encode("utf-8")
    mac = compute_mac(packet_bytes, MAC_KEY)
    packet["mac"] = mac

    return packet


def send_packet(packet: Dict[str, Any]) -> None:

    client = mqtt.Client(client_id=DEVICE_ID)
    client.connect(MQTT_BROKER, MQTT_PORT, keepalive=60)
    payload_str = json.dumps(packet)
    client.publish(MQTT_TOPIC, payload_str)
    client.disconnect()


# ===================== Основний цикл роботи пристрою =====================

def read_sensor_temperature() -> float:

    return 23.56


def main_loop():

    while True:
        temp = read_sensor_temperature()

        x_raw = Measurement(
            data_type="temperature",
            value=temp,
            timestamp=time.time(),
            device_id=DEVICE_ID,
            location="room-101"
        )

        packet = build_packet(x_raw)
        print("DEBUG packet:", json.dumps(packet, indent=2, ensure_ascii=False))

        send_packet(packet)
        time.sleep(10)


if __name__ == "__main__":
    main_loop()
```

Додаток Г

Апробація отриманих результатів

МОЛОДІЖНА НАУКОВА ЛІГА

МАТЕРІАЛИ X МІЖНАРОДНОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ

КОНФЕРЕНЦІЇ

ТЕОРЕТИЧНЕ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ СУЧАСНОЇ НАУКИ

м. Запоріжжя, Україна
12 грудня 2025 рік

Вінниця, Україна
«UKRLOGOS Group»
2025

УДК 082:001
Т 11

Голова оргкомітету: Коренюк І.О.
Верстка: Білоус Т.В.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 49 від 11.12.2025 року.

Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 460 від 10.06.2025).

Т 11 **Теоретичне та практичне застосування результатів сучасної науки**: матеріали X Міжнародної студентської наукової конференції, м. Запоріжжя, 12 грудня, 2025 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2025. — 766 с.

ISBN 978-617-8582-08-1
DOI 10.62732/liga-inter-12.12.2025

Викладено матеріали учасників X Міжнародної мультидисциплінарної студентської наукової конференції «Теоретичне та практичне застосування результатів сучасної науки», яка відбулася 12 грудня 2025 року у місті Запоріжжя, Україна.

УДК 082:001

ISBN 978-617-8582-08-1

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В УПРАВЛІННІ БІЗНЕС-ПРОЦЕСАМИ
Дехтяренко В.Д., ***Науковий керівник: Слюсаренко О.К.*** **505**

ЕТИЧНІ АСПЕКТИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ТА ДАНИХ: ГЛОБАЛЬНІ СТАНДАРТИ І РЕГУЛЮВАННЯ
Байрак В.Д., ***Науковий керівник: Савченко І.Є.*** **506**

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПІДТРИМКИ ОНЛАЙН-ПРОДАЖІВ У INSTAGRAM НА ОСНОВІ LLM-АСИСТЕНТА
Підкович В.А., ***Науковий керівник: Белз О.Г.*** **509**

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ВИЗНАЧЕННЯ РЕЗУЛЬТАТИВНОСТІ СПІВПРАЦІ З КОНТРАГЕНТАМИ
Черепаха М.О., ***Науковий керівник: Морозова А.І.*** **512**

МЕТОД ЗАБЕЗПЕЧЕННЯ ПРИВАТНОСТІ ДАНИХ В IoT-СИСТЕМАХ
Гончаров Ю.В., ***Науковий керівник: Осолінський О.Р.*** **515**

МЕТОДИ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ ТА АНОМАЛІЙ У КОМП'ЮТЕРНИХ МЕРЕЖАХ
Савченко К.О., ***Науковий керівник: Слюсаренко О.К.*** **518**

МОДЕЛЬ ТА ЗАСОБИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ФАРМАЦЕВТИЧНИХ СЕРВІСІВ
Децик К.В., ***Науковий керівник: Опотяк Ю.В.*** **520**

ОПТИМІЗАЦІЯ АЛГОРИТМІВ ІНДЕКСАЦІЇ ТА ПОШУКУ ТЕКСТОВИХ ДАНИХ ЗАСОБАМИ ЕВОЛЮЦІЙНИХ МЕТОДІВ ПРОГРАМУВАННЯ
Білокінь Є.Т., ***Науковий керівник: Панченко Т.В.*** **523**

РОЗРОБКА ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АПРОКСИМАЦІЇ ПОКАЗНИКІВ В ЗАДАЧАХ ВИСВІЧУВАННЯ
Костюк О.А., ***Науковий керівник: Маслюк В.Т.*** **526**

СЕКЦІЯ 21.
ТРАНСПОРТ ТА ТРАНСПОРТНІ ТЕХНОЛОГІЇ

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ВПЛИВУ ГБО НА ЕКОЛОГІЧНІ ПОКАЗНИКИ ДВЗ RENAULT LOGAN
Бражиненко Є., ***Науковий керівник: Дмитрів І.*** **528**

СЕКЦІЯ 22.
ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

ЗАСТОСУВАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ У ВІЙСЬКОВІЙ СПРАВІ
Копач В.-І.І., ***Науковий керівник: Нагорний М.С.*** **530**

МАТЕМАТИЧНІ ОСНОВИ СУЧАСНОЇ АРТИЛЕРІЇ
Шкурат С.О., ***Науковий керівник: Гузик Н.М.*** **533**

СЕКЦІЯ 23.
СОЦІОЛОГІЯ ТА СТАТИСТИКА

ГЕНДЕР У КОЛЕКТИВІ ЯК ФАКТОР ФОРМУВАННЯ МІЖОСОБИСТІСНИХ ВІДНОСИН ТА ПСИХОЛОГІЧНОГО КЛІМАТУ
Голобородько Д.В., ***Науковий керівник: Шинкаренко І.О.*** **536**

Гончаров Юрій Вікторович, магістр спеціальності "Комп'ютерні науки"
Західноукраїнський національний університет, Україна

Науковий керівник: Осолінський Олександр Романович, канд.техн.наук, доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

МЕТОД ЗАБЕЗПЕЧЕННЯ ПРИВАТНОСТІ ДАНИХ В IoT-СИСТЕМАХ

Вступ

Інтернет речей (IoT) перетворився на ключову інфраструктуру для збору, обробки та аналізу різнорідних даних — від промислових сенсорів до пристроїв розумного дому. Особливістю IoT є поєднання великої кількості ресурсно-обмежених вузлів, розподілених каналів зв'язку та хмарних сервісів для зберігання й аналітики даних [2]. В таких умовах загрози конфіденційності даних виникають на всіх етапах життєвого циклу систем і під час збору, передавання, проміжної обробки, довготривалого зберігання та використання в інтелектуальних сервісах [1].

Значна частина рішень зосереджується або на окремих протоколах автентифікації, або на хмарних механізмах шифрування, не забезпечуючи узгоджений, формально обгрунтований захист на всіх рівнях IoT-екосистеми [3].

Концепція методу збереження конфіденційності в IoT-системах

Метод дотримується підходу, коли механізми збереження конфіденційності вбудовані в архітектуру IoT-рішення, а не додаються окремим зовнішнім шаром [6]. Такий підхід узгоджується з сучасними тенденціями в IoT, де приватність повинна забезпечуватися комбінацією технологічних засобів на різних рівнях архітектури. Центральним об'єктом в такій системі є окремий елемент даних x який має: (1) рівень чутливості; (2) контекст що містить інформацію про джерело, тип сенсора, належність користувачу, місце збору; (3) набір дозволених операцій. Для кожного класу чутливості визначається політика, яка задає:

– які перетворення повинні бути виконані на рівні пристрою, наприклад видалення ідентифікаторів;

– які операції виконуються на шлюзі;

– які правила зберігання та доступу застосовуються в хмарі.

Тобто для будь-якої траєкторії проходження даних через послідовність пристроїв→ каналів зв'язку → сервісів → сховищ даних → користувачів гарантується, що відкритий доступ до сирих конфіденційних значень відсутній, а всі операції з ними виконуються тільки в заздалегідь визначених, контрольованих точках.

Другою ключовою ідеєю є щільна інтеграція механізмів автентифікації та керування доступом з процедурами обробки даних [4]. Для цього доцільно використовувати легковагові протоколи взаємної автентифікації на основі фізично неклонованих функцій (PUF) та динамічних ідентифікаторів.

В запропонованому методі автентифікація розглядається, як попередня умова будь-яких операцій з конфіденційними даними, поки пристрій, шлюз або користувач не пройшли перевірку, для них недоступні ключі шифрування, токени доступу та маршрути.

На рівні пристрою реалізується первинна класифікація і попередня обробка даних. Кожен IoT-вузол має вбудований модуль політик даних, в конфігурації якого задаються типи сенсорів, належність до користувача і відповідний рівень чутливості [6].

Для всіх пристроїв виконується симетричне шифрування, а також мінімізація даних. Крім того, при кожному сеансі взаємодії пристрій використовує динамічний псевдонім замість реального ідентифікатора.

Шлюз виступає довіреним проміжним елементом між локальною мережею пристроїв і інфраструктурою сервісів. На цьому рівні застосовуються більш складні операції, які неможливо повністю реалізувати на сенсорах до яких входять взаємна автентифікація з пристроями на основі PUF-механізмів. На шлюзі також здійснюється додаткова політика щодо даних:

– поділ конфіденційного повідомлення на кілька незалежних фрагментів;

– формування нових ключів, які будуть використовуватися вже в хмарі;

– заміна прямих ідентифікаторів користувачів, пристроїв або локацій на токени, пов'язані з відповідними записами у внутрішніх таблицях шлюзу.

В результаті навіть коли відбулась компрометація хмарної частини буде отримано тільки фрагменти зашифрованих даних з мінімальним набором метаданих.

В хмарі реалізується логіка довгострокового зберігання даних, аналітики та інтеграції із зовнішніми сервісами [6]. Для підтримки конфіденційності запропонований метод передбачає такі компоненти:

1. Менеджер фрагментів і ключів конфіденційних даних розподіляє їх між кількома логічно незалежними сховищами. Ключова інформація зберігається в окремому модулі, ізольованому від прикладних БД [3].

2. Доступ до операцій відновлення та дешифрування фрагментів надається лише суб'єктам, які задовольняють певним політикам, сформульованим у термінах атрибутів.

3. Модуль моніторингу слідкує щоб усі запити до чутливих даних контролювались з фіксацією суб'єкта, часу, обсягу та правової підстави доступу.

В рамках сервісів прикладного рівня аналітики, візуалізації та API доступні лише результатні узагальнені показники, що не дозволяють перейти до рівня окремих користувачів (рисунок 1). При необхідності використання відкритих даних метод допускає застосування диференційної приватності, коли в публічні набори додається контрольований шум.

Описані компоненти пов'язуються єдиною політикою ідентифікації, автентифікації та авторизації. На рівні пристроїв і шлюзів використовується багатофакторна схема: апаратний відбиток (PUF), секрети, вбудовані під час виробництва або налаштування, а також динамічні параметри сеансу у вигляді часових міток.

Для користувачів в хмарній частині додатково застосовуються класичні механізми, до яких входять паролі, токени та сертифікати.

Управління доступом реалізується як послідовність перевірок- спочатку апаратна автентифікація пристрою, далі перевірка права на конкретну операцію з певним рівнем чутливості даних, і потім – фактичне надання ключів дешифрування або маршрутів до фрагментів. Це забезпечує тісне поєднання механізмів конфіденційності та автентифікації.

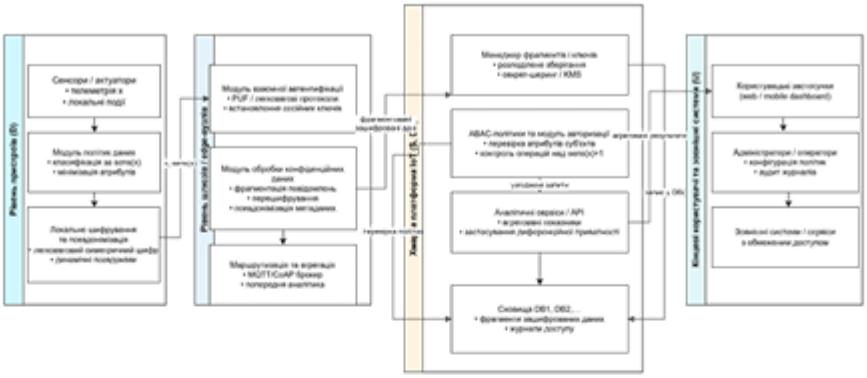


Рис. 1. **Інтеграція автентифікації пристроїв і користувачів**

Висновки. Було запропоновано метод забезпечення приватності даних в IoT-системах, який поєднує формальну модель IoT-екосистеми, модель порушника та багаторівневу архітектуру захисту. На відміну від підходів, орієнтованих лише на окремий протокол або компонент, метод:

– розглядає конфіденційність на рівні окремого елемента даних з врахуванням його чутливості й контексту;

– забезпечує безперервний захист даних на всіх етапах життєвого циклу – від збору на пристрої до зберігання та доступу в хмарній інфраструктурі;

– інтегрує автентифікацію, керування доступом, шифрування та фрагментацію в єдину архітектуру;

– використовує механізми PUF та динамічні ідентифікатори, що робить його придатним для ресурсно-обмежених IoT-пристроїв.

Список використаних джерел:

1. Safaei Yaraziz M., Jalili A., Gheisari M., Liu Y. Recent trends towards privacy-preservation in Internet of Things, its challenges and future directions. *IET Circuits, Devices & Systems*. 2023. Vol. 17, No. 2. P. 53–61.
2. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*. 2015. Vol. 76. P. 146–164.
3. Wakili A., Bakkali S. Privacy-preserving security of IoT networks: A comparative analysis of methods and applications. *Cyber Security and Applications*. 2025. Vol. 3. Art. 100084.
4. Ferrag M. A., Maglaras L. A., Janicke H., Jiang J., Shu L. Authentication protocols for Internet of Things: A comprehensive survey. *Security and Communication Networks*. 2017. Vol. 2017. Art. 6562953.
5. Ataullah M., Chauhan N. Exploring security and privacy enhancement technologies in the Internet of Things: A comprehensive review. *Security and Privacy*. 2024. Vol. 7, No. 6. Art. e448.
6. Hu F. Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations. Boca Raton : *CRC Press*, 2016.
7. Підлісний Ю. І. Шляхи підвищення конфіденційності в мережах інтернету речей. *Центральноукраїнський науковий вісник. Технічні науки*. Кропивницький : ЦНТУ, 2025. Вип. 11(42), ч. 1. С. 46–55.

ЗБІРНИК НАУКОВИХ ПРАЦЬ

З МАТЕРІАЛАМИ V МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ

12 ГРУДНЯ 2025 РІК

М. КРОПИВНИЦЬКИЙ, УКРАЇНА

**«ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО:
ВЗАЄМОДІЯ, ВПЛИВ, ТРАНСФОРМАЦІЯ»**

ЗБІРНИК НАУКОВИХ
ПРАЦЬ З МАТЕРІАЛАМИ
V МІЖНАРОДНОЇ
НАУКОВОЇ КОНФЕРЕНЦІЇ

ТЕХНОЛОГІЇ ТА СУСПІЛЬСТВО: ВЗАЄМОДІЯ, ВПЛИВ, ТРАНСФОРМАЦІЯ

| 12 грудня 2025 рік
м. Кропивницький, Україна

Вінниця, Україна
«UKRLOGOS Group»
2025

УДК 082:001
Т 38

Організація, від імені якої випущено видання:
ГО «Міжнародний центр наукових досліджень»
Номер запису організації в Єдиному реєстрі громадських об'єднань: 1499141

Голова оргкомітету: Сотник С.Г.

Верстка: Білоус Т.В.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 49 від 11.12.2025 року.

*Конференцію зареєстровано Державною науковою установою у сфері управління Міністерства освіти і науки «Український інститут науково-технічної експертизи та інформації» в базі даних науково-технічних заходів України на поточний рік та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (**Посвідчення № 501 від 10.06.2025**).*

*Збірник наукових праць з матеріалами конференції видано офіційно суб'єктом видавничої справи зі **Свідоцтвом ДК № 7860 від 22.06.2023**.*

Т 38 **Технології та суспільство: взаємодія, вплив, трансформація**: збірник наукових праць з матеріалами V Міжнародної наукової конференції, м. Кропивницький, 12 грудня, 2025 р. / Міжнародний центр наукових досліджень. — Вінниця: ТОВ «УКРЛОГОС Груп, 2025. — 452 с.

ISBN 978-617-8582-07-4
DOI 10.62731/mcnd-12.12.2025

Викладено матеріали учасників V Міжнародної наукової конференції «Технології та суспільство: взаємодія, вплив, трансформація», яка відбулася 12 грудня 2025 року у місті Кропивницький.

УДК 082:001

ISBN 978-617-8582-07-4

ОСОБЛИВОСТІ СТРУКТУРИ БАЗИ ДАНИХ JSON-ФОРМАТУ ОПИСУ ФІЗИЧНИХ ВПРАВ ДЛЯ ЇХ КОМП'ЮТЕРНОЇ ІГРОВІЗАЦІЇ
Безугла А.І. **213**

ПІДВИЩЕННЯ ТОЧНОСТІ СИСТЕМИ РОЗПІЗНАВАННЯ ЖЕСТІВ НА ОСНОВІ MEDIAPIPE ЗА ДОПОМОГОЮ НЕЙРОННОЇ КОРЕКЦІЇ ОСВІТЛЕННЯ
Панько М.І. **218**

СИСТЕМНЕ ПРОЄКТУВАННЯ ВЕБРЕСУРСУ ДЛЯ ОНЛАЙН-ЗАМОВЛЕННЯ ЇЖІ З ІНТЕЛЕКТУАЛЬНИМИ РЕКОМЕНДАЦІЯМИ В КОНТЕКСТІ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ СФЕРИ ПОСЛУГ
Сочесло О.Ю. **225**

СТІЙКІСТЬ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ КІБЕРЗАХИСТУ ДО АТАК ADVERSARIAL MACHINE LEARNING ЯК ЧИННИК ПІДВИЩЕННЯ БЕЗПЕКИ ІНФОРМАЦІЙНИХ СИСТЕМ
Лозовський Р.Л. **231**

СЕКЦІЯ XV.
СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ

СУЧАСНІ МЕТОДИ ОПТИМІЗАЦІЇ ТА ЇХ ЗАСТОСУВАННЯ
Фінько В.В. **236**

СЕКЦІЯ XVI.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

АЛГОРИТМ ЗБЕРЕЖЕННЯ КОНФІДЕНЦІЙНОСТІ В ІНТЕРНЕТІ РЕЧЕЙ
Гончаров Ю.В. **241**

АНАЛІЗ ВРАЗЛИВОСТЕЙ МОБІЛЬНИХ ДОДАТКІВ ІЗ ВИКОРИСТАННЯМ ЗАСОБІВ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ
Любченко О.О. **246**

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ RAG-СИСТЕМ ДЛЯ РОБОТИ З ТЕХНІЧНОЮ ДОКУМЕНТАЦІЄЮ
Рудіченко М.О. **248**

ЗАСТОСУВАННЯ КОГНІТИВНОГО МОДЕЛЮВАННЯ У СУЧАСНИХ ПРОГРАМНИХ СИСТЕМАХ. НЕЙРО-СИМВОЛЬНИЙ ШТУЧНИЙ ІНТЕЛЕКТ
Крюков А.В. **251**

МЕТОДИ ДЕТЕКТУВАННЯ ТА РОЗПІЗНАВАННЯ ЖЕСТІВ З УКРАЇНСЬКОЇ ЖЕСТОВОЇ МОВИ
Чуєв К.Д., Машталір С.В. **256**

СЕКЦІЯ XVI.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

АЛГОРИТМ ЗБЕРЕЖЕННЯ КОНФІДЕНЦІЙНОСТІ В ІНТЕРНЕТІ РЕЧЕЙ

Гончаров Юрій Вікторович
Магістр спеціальності "Комп'ютерні науки"
Західноукраїнський національний університет, Україна

Науковий керівник: Осолінський Олександр Романович
ORCID ID: 0000-0002-0136-395X
канд.техн.наук, доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

Вступ

Абстрактна парадигма Інтернету речей (Internet of Things, IoT) в даний час трансформує повсякденну діяльність людини, спрямовуючи її в бік глибшої автоматизації процесів та підвищення якості життя [1]. Більшість фізичних об'єктів («речей») в інфраструктурі IoT характеризуються обмеженими енергетичними ресурсами, обсягами пам'яті та обчислювальними можливостями. У зв'язку з цим основні операції із збирання, обробки та зберігання даних переносяться до хмарних обчислювальних середовищ [2]. Перевага у вигляді доступу до даних «будь-коли та будь-де» водночас породжує суттєві проблеми у сфері безпеки та конфіденційності, що може призводити до розкриття персональної та конфіденційної інформації користувачів і, як наслідок, до втрати довіри між усіма задіяними сторонами [3].

З операційної точки зору однією з ключових проблем для IoT є саме забезпечення конфіденційності даних [4]. Більшість дослідників розглядають конфіденційність та безпеку, як найскладніші аспекти функціонування Інтернету речей, оскільки значна частина сервісів IoT потенційно становить ризик для персональної інформації користувачів. Загрози конфіденційності можуть виникати на різних етапах життєвого циклу даних: під час їх зберігання на пристроях, підключених до мережі

Інтернет, під час передавання каналами зв'язку між пристроями, а також під час зберігання та обробки у хмарній інфраструктурі [5].

Пропоновані алгоритми для збереження конфіденційності в інтернеті речей

Враховуючи що конфіденційність в IoT це прохід від сенсора до хмари, потрібно забезпечити повний життєвий цикл конфіденційних даних в IoT-системі - від моменту підключення пристрою до системи та встановлення довіреного сеансу до зберігання фрагментованих зашифрованих даних у хмарній інфраструктурі та надання доступу до них авторизованим суб'єктам [6].

Виділено чотири ключові алгоритми:

1. Алгоритм реєстрації й автентифікації пристрою.
2. Алгоритм підготовки та передавання захищених повідомлень.
3. Алгоритм обробки та зберігання даних на серверній стороні.
4. Алгоритм надання доступу до даних авторизованим користувачам і сервісам.

Перший алгоритм визначає, як фізичний IoT-пристрій входить в систему, одержує власний ідентифікатор, реєструється в реєстрі пристроїв і потім щоразу проходить взаємну автентифікацію із шлюзом. На цьому етапі закладається довіра до апаратної частини та формується сесійний ключ, що використовується для захищеного обміну даними. Далі в кожному сеансі зв'язку виконується взаємна автентифікація. Якщо значення збігаються, вважається, що пристрій автентичний, і сторони переходять до встановлення сесійного ключа. У разі невідповідності фіксується помилка, збільшується лічильник помилок, і при перевищенні допустимого порогу пристрій може бути тимчасово або постійно заблокований.

Узагальнену блок-схему алгоритму реєстрації та автентифікації пристрою представлено на рисунку 1А.

Другий алгоритм описує дії, які виконує пристрій для підготовки даних до відправлення, їх перетворення відповідно до рівня чутливості та подальшої передачі через шлюз до хмарної платформи.

На цьому етапі визначається конфіденційність елементів даних, а також виконується мінімізація, локальне шифрування та формування захищеного пакета [7]. Блок-схему алгоритму підготовки та передавання захищених повідомлень показано на рисунку 1Б.

А) Б)

Рис. 1. **Блок-схема алгоритму№1 та№2**

А) Б)

Рис. 2. **Блок-схема алгоритму№3 та №4**

Третій алгоритм забезпечує приймання, валідацію, розподілене зберігання та ведення журналу фрагментів конфіденційних даних (рисунок 2А).

Четвертий алгоритм описує, як авторизований користувач або зовнішній сервіс отримує доступ до конфіденційних даних, зберігаючи при цьому всі гарантії конфіденційності. Ідея полягає в тому, що жодні чутливі дані не повертаються суб'єкту без попередньої автентифікації і успішного проходження перевірки політик ABAC.

Блок-схему алгоритму надання доступу до даних авторизованим суб'єктам показано на рисунку 2Б.

Процес починається з формування запиту користувачем в клієнтському застосунку, де він задає тип потрібних даних, часовий інтервал, обсяг або інші параметри. Далі користувач проходить процедуру автентифікації, а служба автентифікації визначає його атрибути. На основі цих атрибутів і налаштованих політик, спеціалізований сервіс видає токен доступу, в якому зафіксовано дозволені типи операцій і обмеження за даними.

Висновки.

Було запропоновано узгоджений підхід до збереження конфіденційності даних в інфраструктурі Інтернету речей, заснований на чотирьох взаємопов'язаних алгоритмах, які охоплюють повний життєвий цикл обробки конфіденційних даних — від моменту підключення пристрою до системи до надання доступу до фрагментованих зашифрованих записів авторизованим користувачам та сервісам. На відміну від рішень, що фокусуються лише на окремих протоколах або компонентах, запропонований підхід інтегрує реєстрацію та автентифікацію пристроїв, підготовку захищених повідомлень, розподілене зберігання та атрибутивне керування доступом у єдиний алгоритмічний ланцюг.

Список використаних джерел:

1. Kaur R., Rodrigues T., Kadir N., Kashef R. A Survey on Privacy Preservation Techniques in IoT Systems. *Sensors*. 2025. Vol. 25, No. 22. Art. 6967.
2. Ogonji M. M., Okeyo G., Wafula J. M. A survey on privacy and security of Internet of Things. *Computer Science Review*. 2020. Vol. 38. Art. 100312.
3. Sun P., Shen S., Wan Y., Wu Z., Fang Z., Gao X. Z. A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends. *IEEE Internet of Things Journal*. 2024. Vol. 11. P. 34567–34591.