МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп’ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____Ірина ЖУРАВСЬКА

« __ » _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

IoT-мережа із захистом на основі алгоритмів “легкої криптографії”

Спеціальність 123 Комп’ютерна інженерія

Освітня програма «Комп’ютерна інженерія»

Здобувач _____Дмитро ЖУКОВСЬКИЙ
підпис

«__» _____ 2025 р.

Керівник д-р техн. наук, проф.,
зав. кафедри комп’ютерної інженерії _____Ірина ЖУРАВСЬКА
підпис
«__» _____ 2025 р.

Миколаїв – 2025

Факультет	Комп’ютерних наук
Кафедра	Комп’ютерної інженерії
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	123 Комп’ютерна інженерія
Освітня програма	Комп’ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп’ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Жуковський Дмитро Сергійович
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи «IoT-мережа із захистом на основі алгоритмів “легкої криптографії»
IoT-мережа із захистом на основі алгоритмів “легкої криптографії”

Затверджена наказом по ЧНУ ім. Петра Могили від 23.06.2025 № 165/1.

2. Строк представлення кваліфікаційної роботи « 10 » грудня 2025 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
Розробити IoT мережу із захистом на основі алгоритмів “легкої криптографії”, інтегрувати систему в розумний годинник з виведенням інформації про дані користувача на вебсайт.

4. Перелік питань, що підлягають розробці:
Провести аналіз сучасного стану IoT-мереж та методів захисту даних, дослідити принципи побудови алгоритмів легкої криптографії, розглянути структуру та особливості алгоритму ASCON, розробити архітектуру захищеної IoT-мережі із застосуванням алгоритму ASCON, реалізувати та протестувати

прототип шифрування даних на основі ASCON

5. Перелік графічних матеріалів

Скріншоти вебінтерфейсу, Робочий процес обробки даних з IoT-пристроїв, Поля даних, Приклад логів, Функціональна схема взаємодії компонентів IoT-мережі, Структура реляційної БД, Архітектура системи, Реалізація програми, Код програми.

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи _____ Ірина ЖУРАВСЬКА

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ Дмитро ЖУКОВСЬКИЙ

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Дата видачі завдання «_01_» ___липня_____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної магістерської роботи

Тема: IoT-мережа із захистом на основі алгоритмів “легкої криптографії”

№ з/п	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КМР	01.10.2025	10.10.2025	Виконано
2	Огляд літератури за темою роботи	03.10.2025	05.10.2025	Виконано
3	Складання календарного плану КМР	07.10.2025	08.10.2025	Виконано
4	Аналіз предметної області	15.10.2025	25.10.2025	Виконано
5	Розробка проєктних рішень	01.11.2025	02.11.2025	Виконано
6	Моделювання АПЗ	04.11.2025	10.11.2025	Виконано
7	Перевірка працездатності, тестування та апробація розробленого ПЗ, аналіз результатів тестування	15.11.2025	20.11.2025	Виконано
8	Оформлення КМР та презентації	21.11.2025	22.11.2025	Виконано
9	Попередній захист	25.11.2025	25.11.2025	Виконано
10	Відгук керівника КМР	26.11.2025	26.11.2025	Виконано
11	Рецензування	03.12.2025	09.12.2025	Виконано
12	Завершення оформлення КМР та презентації	10.12.2025	12.12.2025	Виконано
13	Захист кваліфікаційної магістерської роботи	16.12.2025	16.12.2025	Виконано

Керівник роботи _____ ___Ірина ЖУРАВСЬКА_____

Особистий підпис *Власне ім’я ПРІЗВИЩЕ*

Здобувач _____ ___Дмитро ЖУКОВСЬКИЙ___

Особистий підпис *Власне ім’я ПРІЗВИЩЕ*

«____» _____ 20__ р.

АНОТАЦІЯ
до кваліфікаційної магістерської роботи
«IoT-мережа із захистом на основі алгоритмів “легкої криптографії»
Студент гр. 605: ЖУКОВСЬКИЙ Дмитро Сергійовича
Керівник: д-р техн. наук, проф. ЖУРАВСЬКА І. М.

У роботі розглянуто проблему забезпечення конфіденційності та цілісності даних у IoT-мережах для пристроїв з обмеженими ресурсами. Розглянуто сучасні підходи до захисту даних, концепцію «легкої криптографії» та стандартизацію відповідних алгоритмів. Описано архітектуру запропонованої IoT-мережі із застосуванням алгоритму ASCON для автентифікованого шифрування даних, а також представлено рекомендації щодо реалізації шифрування на мікроконтролерах та передачі даних через MQTT/HTTPS. Робота є теоретичною і орієнтована на обґрунтування доцільності використання легких криптографічних алгоритмів у носимих пристроях та вбудованих системах.

Метою роботи є захист даних у мережах Інтернету речей шляхом впровадження «легких» криптографічних алгоритмів, що забезпечують баланс між рівнем безпеки, енергоспоживанням та обчислювальними можливостями пристроїв. Практична значимість дослідження полягає у створенні рекомендацій для розробників IoT-пристроїв щодо використання алгоритму ASCON у вбудованих системах і носимій електроніці, що підвищує стійкість до атак та захищає дані користувачів під час обміну інформацією.

Результати роботи можуть бути використані під час проєктування безпечних IoT-мереж, систем розумного дому, промислових сенсорних мереж і медичних пристроїв. Запропонований підхід сприяє підвищенню рівня безпеки в екосистемі Інтернету речей без значного збільшення вартості або енергоспоживання пристроїв.

Робота включає огляд існуючих алгоритмів легкої криптографії, аналіз стандартів NIST Lightweight Cryptography, дослідження ефективності реалізації ASCON на мікроконтролерах сімейства ARM, а також моделювання передавання зашифрованих даних у тестовій мережі MQTT.

Робота складається з ____ сторінок, 9 таблиць, 19 рисунків та 2 додатків. У процесі дослідження використано 20 джерел посилання.

Ключові слова: Інтернет речей, мережа, алгоритм шифрування, легка криптографія, ASCON, MQTT, конфіденційність, вбудовані системи, безпека даних.

ABSTRACT
of the Master's Thesis
«IoT network with protection based on "light cryptography" algorithms»
Applicant: ZHUKOVSKYI Dmytro,
Supervisor: Head of the Computer Engineering Department, Doctor of Technical Sciences, Professor ZHURAVSKA Iryna

This thesis investigates the problem of ensuring data confidentiality and integrity in IoT networks composed of resource-constrained devices. It reviews modern approaches to data protection, the concept of lightweight cryptography, and the standardization of corresponding algorithms. The proposed network architecture employs the ASCON cipher for authenticated data encryption and provides recommendations for implementing cryptographic operations on microcontrollers and securely transmitting information via MQTT and HTTPS protocols.

The aim of the work is data protection in IoT networks via lightweight cryptographic algorithms with achieving an optimal balance between security, energy consumption, and device performance. The practical significance of the research lies in offering implementation guidelines for developers of IoT and wearable devices to apply the ASCON algorithm in embedded environments, thereby improving system resilience against attacks and ensuring secure data transmission.

The results of the work can be applied in the design of secure IoT networks, smart home systems, industrial sensor networks, and medical devices. The proposed approach enhances IoT ecosystem security without significantly increasing device cost or power consumption.

The work includes an overview of existing lightweight cryptographic algorithms, an analysis of the NIST Lightweight Cryptography standards, an evaluation of ASCON performance on ARM-based microcontrollers, and a simulation of encrypted data transfer within an MQTT test environment.

The thesis consists of ___ pages, 9 tables, 19 figures, and 2 appendices, and 20 references.

Keywords: IoT, network, encryption algorithm, lightweight cryptography, ASCON, MQTT, confidentiality, embedded systems, data protection.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних
НСД – несанкціонований доступ
АЦП – аналого-цифровий перетворювач

AES – Advanced Encryption Standard
CPU – Central Processing Unit
ECC – Elliptic Curve Cryptography
HTTPS – HyperText Transfer Protocol Secure
IoT – Internet of Things (укр. Інтернет речей)
MA – Moving Average (укр. ковзне середнє).
MCU – Microcontroller Unit
MQTT – Message Queuing Telemetry Transport
SHA – Secure Hash Algorithm
YOLO – You Only Look Once
PCB – Printed Circuit Board

ВСТУП

У сьогоднішньому світі кількість пристроїв підключених до мережі Інтернету речей (англ. Internet of Things, IoT), які нас оточують, стає постійно більшою,. До таких систем належать розумні годинники, сенсорні вузли, побутова техніка, та інші пристрої, які постійно обмінюються даними між собою. IoT мережі – в наш час попит на них стає все більшим і більшим, вони потрібні майже в усіх сферах по всьому світу: медицина, оборона, енергетика і все інше.

У той же час, з цієї причини з'являються і нові загрози витоку інформації, проблема також полягає в тому, що пристрої, які використовуються в мережах АЙОТ, вони малоресурсні і потужний захист поставити дуже складно, тут і з'являються механізими шифрування, які не вимагають занадто величезного ресурсу, в той же час можуть шифрувати і захищати інформацію.

Вирішенням проблеми стає «легка криптографія» (англ. Lightweight Cryptography) – напряму, що розробляє криптографічні алгоритми з мінімальними вимогами до ресурсних потужностей пристроїв, низьким енергоспоживанням та збереженням достатнього рівня криптографічної стійкості. ASCON – новий базовий стандарт для автентифікованого шифрування та хешування в IoT-пристроях. Тому-що цей алгоритм поєднує високу безпеку, та економію ресурсів.

Актуальність даної теми зумовлена зростанням кількості пристроїв, які потребують захисту даних при мінімальних ресурсах. Також важливим є дослідження питань реалізації легких криптографічних алгоритмів у мікроконтролерах та їх інтеграції з протоколами безпечної передачі даних, такими як MQTT або HTTPS.

Об'єктом даного дослідження є процеси забезпечення конфіденційності та цілісності даних у мережах Інтернету речей.

Предметом дослідження є методи застосування «легких» криптографічних алгоритмів, зокрема ASCON, у вбудованих системах з обмеженими обчислювальними ресурсами.

Метою даної роботи є захист даних у компонентах IoT-мережі на основі алгоритмів «легкої криптографії.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1) провести аналіз сучасного стану IoT-мереж та методів захисту даних – розглянути існуючі підходи до забезпечення конфіденційності та цілісності даних, особливості використання традиційних і легких криптографічних алгоритмів;

2) дослідити принципи побудови алгоритмів легкої криптографії – проаналізувати їх відмінності від класичних методів шифрування та визначити вимоги до реалізації у пристроях з обмеженими ресурсами;

3) розглянути структуру та особливості алгоритму ASCON – описати його архітектуру, режими роботи, переваги й недоліки для застосування у вбудованих системах Інтернету речей;

4) розробити архітектуру захищеної IoT-мережі із застосуванням алгоритму ASCON – визначити взаємодію сенсорних вузлів, шлюзів та серверної частини, інтеграцію криптографічного модуля з протоколами MQTT та HTTPS;

5) реалізувати та протестувати прототип шифрування даних на основі ASCON – провести експериментальну перевірку роботи алгоритму на мікроконтролері, оцінити швидкодію, використання пам’яті та енергоспоживання;

Практичне значення даного дослідження полягає у формуванні рекомендацій щодо впровадження «легких» криптографічних алгоритмів у реальні IoT-рішення. Результати роботи можуть бути використані під час проєктування систем «розумного годинника», промислових сенсорних мереж, медичних пристроїв, а також у сфері розробки безпечних комунікаційних протоколів для вбудованих систем.

Таким чином, дослідження спрямоване на вирішення однієї з ключових проблем розвитку сучасного Інтернету речей – поєднання високого рівня безпеки даних із енергоефективністю та апаратною сумісністю пристроїв, що має важливе практичне й наукове значення для подальшого розвитку кіберфізичних систем та безпечного цифрового середовища.

Апробація результатів роботи – результати даного дослідження були апробовані під час XXVIII Всеукраїнській щорічній науково-практичній

конференції «Могилянські читання – 2025» (Миколаїв, 10–14 листопада 2025 р.) на засіданні підсекції «Комп’ютерна інженерія» [1]. Отримані матеріали схвалені та рекомендовані до подальшого використання під час навчального процесу та виконання науково-дослідних робіт, пов’язаних із забезпеченням захисту даних в IoT-мережах.

1 АНАЛІЗ ЗАСОБІВ ЗАХИСТУ ДАНИХ В ІОТ-СИСТЕМАХ ДЛЯ ФІТНЕС-АНАЛІТИКИ

Смарт-годинники сьогодні збирають величезну кількість інформації про користувача: його фізичну активність, пульс, сон, рівень стресу та інші біометричні показники. Ці дані потрібні не лише для відстеження стану здоров’я – їх активно застосовують у спорті, корпоративних програмах контролю самопочуття працівників та в різних онлайн-сервісах.

Найважливіший компонент роботи системи – це безпека передачі інформації. Між годинами, сервером, веб-інтерейсом повинна передаватися інформація в результаті виводиться у вигляді графіків на веб-сайті. Якщо дані передаються з поганим або недостатнім захистом, тут з'являється ризик того, що вони можуть бути злиті в загальний доступ, це суперечить нормам GDPR.

Тому виникає практична потреба у створенні апаратно-програмного рішення, яке зможе не лише збирати інформацію зі смарт-годинника, а й шифрувати її, передавати на сервер і безпечно показувати у вебінтерфейсі.

1.1 Сучасні рішення у сфері фітнес-аналітики

На ринку існує безліч рішень, що дозволяють передавати дані з носимих пристроїв до мобільних або вебзастосунків. Серед найвідоміших платформ можна виділити Apple Health, Google Fit [2], Fitbit Cloud, Garmin Connect та Huawei Health.

Кожна з платформ має власну екосистему пристроїв, протоколи обміну даними (часто закриті), та власні методи шифрування. Наприклад:

- Apple HealthKit використовує End-to-End шифрування при зберіганні даних у хмарі iCloud;
- Google Fit API підтримує OAuth 2.0 для автентифікації запитів і використовує HTTPS/TLS для передачі даних (рис. 1.1);
- Fitbit Web API дозволяє інтегруватися з вебзастосунками, але дані передаються у форматі JSON з авторизацією за токенами доступу.

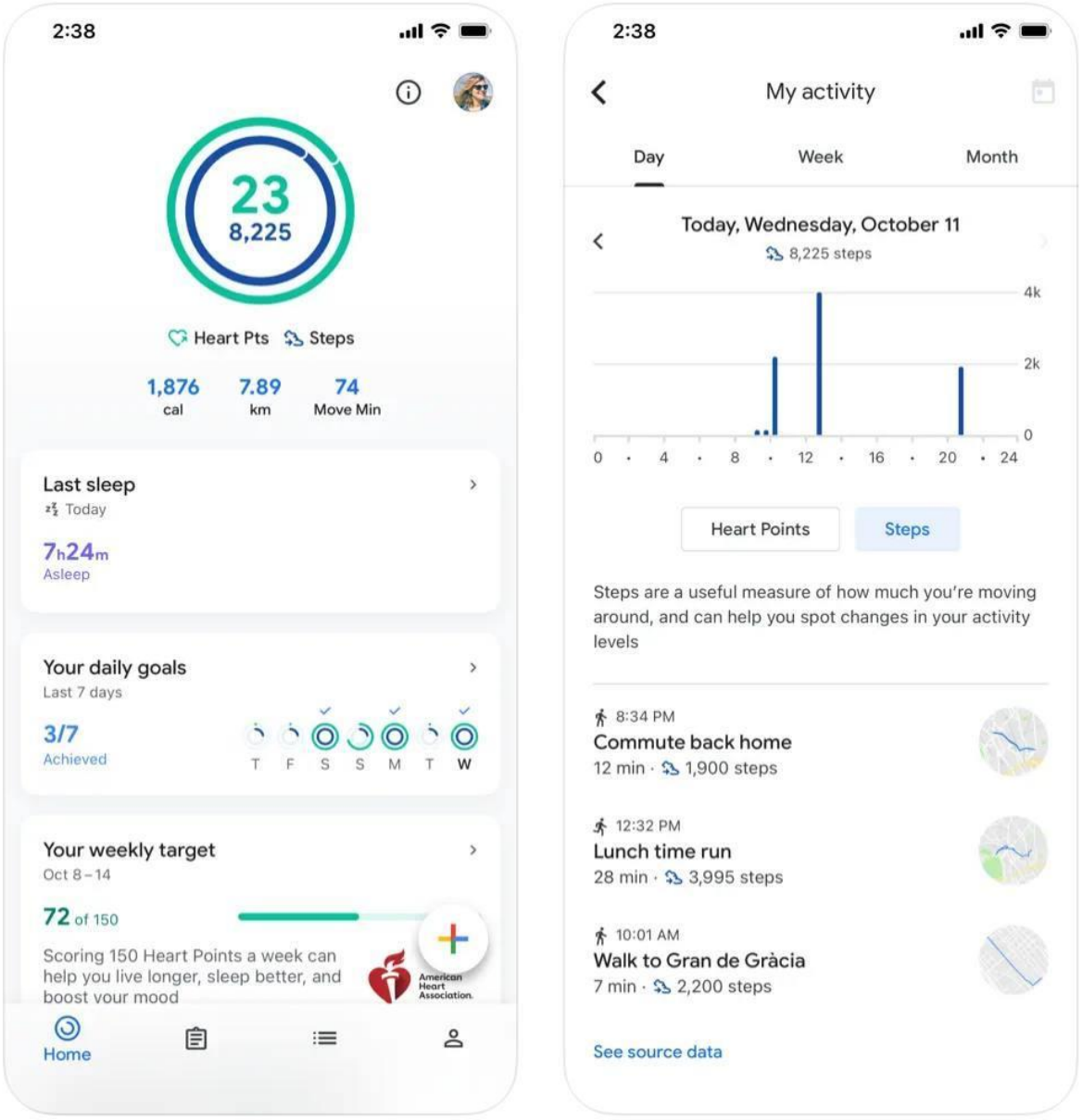

Рисунок 1.1 – Google Fit [2]

Незважаючи на велику кількість готових платформ, більшість із них не дозволяє розробнику створити власний механізм шифрування або контролювати безпеку передачі на проміжних етапах між годинником, сервером і сайтом. Саме тому створення власного рішення із застосуванням відкритих стандартів (наприклад, AES, RSA, або HTTPS з TLS 1.3) має дослідницьке та практичне значення.

1.2 Загальна схема передачі даних

Передача інформації від смарт-годинника до вебінтерфейсу є складним багаторівневим процесом, який поєднує апаратні, мережеві та програмні компоненти [3]. Кожен етап має власні особливості, вимоги до продуктивності й безпеки.

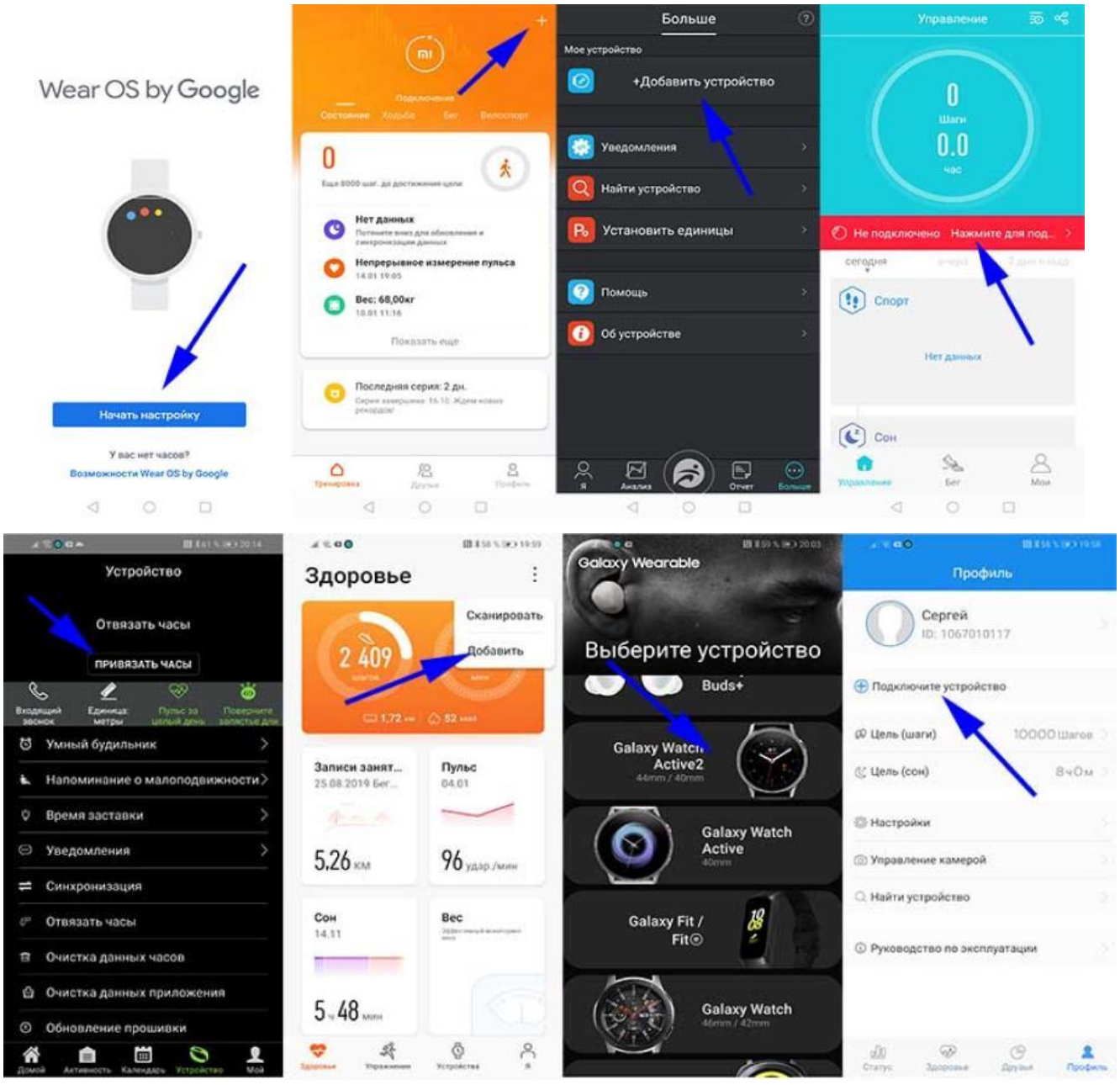

Рисунок 1.2 – Етапи під'єднання смартгодинника до смартфона [3]

Типова архітектура системи передачі даних з носимого пристрою складається з таких компонентів:

- сенсорний модуль: здійснює збір фізіологічних параметрів користувача за допомогою вбудованих сенсорів – пульсометра, акселерометра, гіроскопа, датчика температури тощо. У більшості сучасних пристроїв дані формуються у вигляді коротких пакетів, що передаються з частотою від 1 Гц до 10 Гц.

- комунікаційний шар (Bluetooth Low Energy, BLE): забезпечує передачу даних на мобільний пристрій або шлюз (наприклад, смартфон, мікроконтролер з BLE-модулем ESP32). BLE характеризується низьким енергоспоживанням і швидкістю передачі до 1 Мбіт/с, що робить його оптимальним для IoT-застосунків;

- проміжний застосунок (мобільний або десктопний клієнт): приймає дані від годинника, здійснює попередню обробку (нормалізацію, фільтрацію шумів) і виконує шифрування перед відправкою на сервер;

- серверна частина: отримує зашифровані пакети, розшифровує їх відповідно до алгоритму (наприклад, AES-256 або RSA-2048), зберігає у базі даних (MySQL, PostgreSQL або MongoDB) та формує API для передачі даних до вебінтерфейсу;

- вебінтерфейс: реалізує графічне відображення даних у вигляді таблиць, графіків, діаграм або аналітичних звітів. Використовується зв’язок через REST API або WebSocket для динамічного оновлення інформації. Сенсор генерує дані (наприклад, Pulse = 78 bpm).

Зібрані годинником відомості упаковуються у зручний для передавання формат – це може бути JSON або компактний двійковий пакет. Далі ці дані через BLE надходять до мобільного застосунку. Уже в застосунку інформація шифрується і відправляється на сервер, зазвичай через HTTPS або MQTT.

На сервері пакет приймається, розшифровується та зберігається у базі даних. Вебінтерфейс періодично звертається до сервера, отримує оновлену інформацію й показує її користувачеві у вигляді графіків чи таблиць.

Таким чином, система функціонує як багаторівнева модель взаємодії пристроїв IoT, що базується на принципах Client-Server архітектури.

1.3 Алгоритми шифрування для захисту даних

Захист інформації при передачі є одним із ключових аспектів побудови надійної IoT-мережі. Біометричні дані належать до категорії чутливих персональних даних, тому їх передача без шифрування є неприпустимою. Основні вимоги до криптографічних алгоритмів для IoT-пристроїв (табл. 1.1):

- висока швидкість обробки даних при обмежених ресурсах (CPU, RAM);
- мінімальне енергоспоживання;
- захист від підбору ключа (brute-force);
- сумісність із сучасними протоколами (HTTPS, MQTT, WebSocket Secure).

Таблиця 1.1 – Поширені алгоритми

Алгоритм	Тип	Довжина ключа, біт	Основне призначення
AES (Advanced Encryption Standard)	Симетричний	128–256	Шифрування потоків даних у реальному часі
RSA (Rivest–Shamir–Adleman)	Асиметричний	1024–4096	Безпечна передача ключів
ECC (Elliptic Curve Cryptography)	Асиметричний	160–512	Енергоефективне шифрування для IoT
SHA-256	Хешування	–	Контроль цілісності даних
TLS 1.3	Протокол	–	Захищений транспортний рівень

Найефективніший підхід – гібридна схема шифрування, де симетричний алгоритм (AES) використовується для основних даних, а асиметричний (RSA або ECC) – для обміну ключами. Приклад реалізації: сервер генерує публічний/приватний ключ (RSA). Годинник отримує публічний ключ сервера. Дані шифруються за допомогою AES, а ключ AES шифрується RSA. Сервер розшифровує ключ RSA та потім основні дані AES. Ця схема забезпечує високу

швидкість і максимальну безпеку навіть у пристроях із низькою обчислювальною потужністю.

1.4 Візуалізація даних у вебінтерфейсі

Вебчастина системи може бути реалізована з використанням таких технологій:

- Frontend: HTML5, CSS3, JavaScript, React або Vue.js;
- Backend: Node.js, Python (Flask, Django) або PHP (Laravel);
- база даних: PostgreSQL або MongoDB.

Для оперативного оновлення показників у реальному часі використовується двосторонній канал зв’язку через WebSocket, що дозволяє серверу надсилати нові дані без необхідності оновлення сторінки.

Альтернативою може бути AJAX-запит із певним інтервалом опитування.

Для графічного відображення біометричних даних застосовуються бібліотеки:

1) Chart.js – прості лінійні та кругові діаграми;
2) Plotly.js – інтерактивна візуалізація з масштабуванням;
3) ECharts – комплексні динамічні панелі моніторингу.

Наприклад, пульс користувача може відображатися у вигляді динамічного графіка, що автоматично оновлюється кожні 5 секунд.

Додатково передбачена авторизація користувача через JWT-токен, що запобігає несанкціонованому доступу НСД до персональної статистики.

1.5 Специфікація вимог до апаратно-програмного комплексу

Для створення ефективної системи необхідно визначити перелік функціональних і нефункціональних вимог, які забезпечать її надійність, масштабованість і безпечну роботу.

Функціональні вимоги:

- збір показників сенсорів (пульс, температура, активність);
- формування пакетів даних та їх шифрування на стороні пристрою;
- передача даних через BLE або Wi-Fi з використанням HTTPS або MQTT;
- зберігання інформації у централізованій базі даних із резервним копіюванням;
- відображення статистики у вебінтерфейсі з можливістю фільтрації та аналізу;
- авторизація користувачів і контроль доступу;
- система логування подій (успішні з’єднання, помилки, час відповіді сервера).

Нефункціональні вимоги:

- продуктивність: середній час обробки пакета ≤ 100 мс;
- безпека: обов’язкове використання TLS 1.3, AES-256, авторизація токенами;
- надійність: відновлення з’єднання при втраті мережі ≤ 5 с;
- масштабованість: підтримка ≥ 100 одночасних підключень;
- сумісність: робота з BLE 4.2+, Android, iOS, браузерами Chrome/Edge;
- юзабіліті: адаптивний дизайн для мобільних і десктопних пристроїв.

Для покращення зручності користування система може підтримувати: експорт даних у форматах CSV, JSON, PDF, інтеграцію з хмарними сервісами (Google Fit, Apple Health), повідомлення про перевищення критичних показників (через e-mail або push).

1.6 Проблеми безпеки у фітнес-пристроях на основі IoT

Сучасні пристрої для фітнес-аналітики (смарт-годинники, браслети, пульсометри, фітнес-трекери, розумний одяг та пристрої відстеження GPS тощо) поєднують збір чутливих біометричних даних із безперервною передачею цих даних через радіоінтерфейси та інтернет-сервери.

Саме поєднання:

- обмежених апаратних ресурсів,
- масового розгортання пристроїв;
- багаторівневої інфраструктури обміну даними

створює широкий простір для вразливостей.

Нижче наведено систематизований огляд типових вразливостей, реальних інцидентів та потенційних наслідків компрометації:

- відсутність або слабке шифрування радіоінтерфейсів, багато пристроїв використовують Bluetooth Low Energy (BLE) для передачі даних на телефон або шлюз. BLE має кілька режимів парування деякі („Just Works“) не забезпечують автентифікацію й уразливі до MITM-атак. Це дозволяє зловмиснику перехоплювати або підмінювати пакети під час зв’язку пристрій↔мобільний клієнт, деякі виробники передають дані незашифрованими або проводять шифрування тільки на рівні застосунку після передачі, що збільшує вікно вразливості під час OTA-передачі;
- відсутність аунтифікації та слабке керування з обліковими акаунтами, можливість для користувача використовувати прості паролі з легким шифром;
- можливий витік даних якщо API неправильно захищений, це до речі досить часта проблема багатьох ресурсів, вони зберігають неправильно зберігають дані в базах даних, або неправильні захищені API, в яких немає перевірки прав доступу;

- візуалізація даних небезпечна тим, що їх можуть відновити, і дізнатися минулий маршрут, робочі місця та багато іншого, як було з тим, коли послуги показали військові бази та їх регулярні маршрути.

- різні інші помилки у побудові криптографічної системи, неправильне використання, повторне використання ключів, все це може призвести до витоку інформації.

На практиці вже не раз були інцеденти, через які є розуміння, що ризики витоку важливої інформації для користувачів реально існують. Багато людей потребують фітнес-браслеті для того, щоб стежити за здоров'ям, але також вони потребують дані про їх місце розташування в різний час у різних місцях, їх самопочуття, ніхто не зміг отримати. Система повинна уникнути помилок, які вже були допущені.

Подібні приклади ілюструють, що загрози можуть виникати не лише на рівні радіоінтерфейсу (BLE), а й у ланцюгу обробки даних – від пристрою до мобільного застосунку, серверу та візуалізації на вебінтерфейсі. Наслідки витоків та компрометацій мають як індивідуальний, так і соціально-економічний характер. На індивідуальному рівні злам акаунта надає зловмиснику доступ до історії станів здоров’я та маршрутів, що може призвести до порушення приватності, шантажу, неправомірного доступу до прив’язаних сервісів або навіть до помилкових медичних висновків, якщо дані використовуються у клінічних чи страхових сценаріях. Сукупні GPS-треки та профілі поведінки дозволяють відновлювати «pattern of life» – місця проживання й роботи, типові маршрути й розклад, що робить користувача вразливим до стеження, фізичних загроз (наприклад, планування пограбувань) або вторгнення в приватну сферу. Для компаній-постачальників послуг такі інциденти тягнуть за собою фінансові збитки, витрати на розслідування й ліквідацію наслідків, штрафи за порушення регуляторних вимог (наприклад, положень про захист персональних даних) і неминучі репутаційні втрати. У окремих випадках – коли дані належать персоналу урядових або

військових структур – публічне розкриття маршрутів і активностей може становити загрозу національній безпеці.

Щоб мінімізувати ці ризики, необхідний комплексний підхід, який поєднує технічні, організаційні та процедурні заходи. Основа технології полягає в найголовнішому принципі шифрування: дані шифруються на пристрої до виходу з нього і залишаються зашифрованими до моменту переходу в вже захищений, довірений сервер. Є різні режими використання такого принципу, наприклад AEAD – аутентифіковане шифрування, реалізоване як у вигляді lightweight алгоритма, так і в поєднанні зі стандартизованими протоколами безпечного транспорту. У випадку з BLE важливо застосовувати захищені режими парування (Secure Connections або інші механізми з підтвердженням), уникаючи «Just Works» там, де потрібно гарантувати автентичність пристрою. Ще один критично важливий аспект – сильна автентифікація та управління сесіями: впровадження 2FA, обмеження кількості невдалих спроб входу, контроль і ревокація сесій, механізми прив’язки пристрою до облікового запису і опція швидкого відв’язування в разі підозрілих подій. На серверному рівні необхідно жорстко захищати API (автентифікація ролей, rate-limiting, валідація вхідних даних), застосовувати шифрування БД у спокої, політику мінімізації збереження даних (data minimization), а також регулярно проводити аудит конфігурацій і сканування відкритих інстансів.

Крім технічних рішень, важливими є заходи у сфері політик приватності й обробки даних: при створенні агрегованих візуалізацій (heatmap, статистика) слід впроваджувати privacy-zones, інтервали агрегації й механізми opt-out, що дозволяють користувачам контролювати видимість їхніх даних; потрібно ретельно балансувати корисність аналітики й ризики деанонімізації, а також інформувати користувачів про ризики і способи захисту їхніх даних. На етапі розробки firmware і систем управління ключами слід використовувати апаратні модулі захисту (Secure Element, TPM), ретельно проектувати генерацію випадкових значень і захищене зберігання ключів, підписувати OTA-прошивки і виконувати перевірку цілісності при оновленні. Не менш важливі заходи моніторингу і готовності до інцидентів:

логування подій безпеки, виявлення аномалій, інструменти реагування і налагоджені процедури повідомлення користувачів і регуляторів у разі витоку.

Узагальнюючи, можна констатувати, що фітнес-пристрої на основі IoT опинилися під подвійним тиском – з одного боку, високої цінності персональних даних (біометрія, геолокація), з іншого – сильних апаратних обмежень і прагнення до зручності користувача. Реальні інциденти доводять, що загрози є практичними і багатовимірними, а отже рішення мають бути інтегрованими й багаторівневими: поєднання lightweight-криптографії на пристрої, надійної автентифікації, захищених API, політик публікації даних і оперативного реагування на інциденти м це мінімальний набір заходів, здатний істотно знизити ризики й підвищити довіру користувачів до рішень у сфері фітнес-аналітики.

Висновки до розділу 1

Проведений огляд показав, що більшість існуючих рішень для збору й передавання даних із носимих пристроїв мають певні обмеження: вони або закриті, або не дають достатнього контролю над механізмами безпеки. Під час аналізу також вдалося виявити типові слабкі місця – це недостатній рівень шифрування, відсутність належної автентифікації користувача та централізоване зберігання даних. Усе це підвищує ризик стороннього доступу до інформації і можливих витоків персональних даних.

Розроблення власного апаратно-програмного комплексу, який забезпечує шифровану передачу біометричних даних зі смарт-годинника на вебсайт із подальшою візуалізацією, є актуальним завданням у сфері IoT і кібербезпеки. Такий підхід дозволяє не лише підвищити контроль над обробкою персональних даних, але й створити гнучку платформу для інтеграції нових типів сенсорів та аналітичних алгоритмів, що може значно розширити функціональність системи.

Отже, аналіз показав, що розробка власних апаратно-програмних комплексів із високим рівнем безпеки є ключовою. Це дозволяє одночасно захищати

приватність користувачів та забезпечувати необхідну функціональність сучасних IoT-систем.

2 ПРОЄКТУВАННЯ ІОТ-МЕРЕЖІ

2.1 Загальна концепція IoT-системи

Проєктована система Інтернету речей призначена для безперервного збору, захищеної передачі та подальшої аналітичної обробки біометричних даних користувача, що надходять зі смарт-годинника. Особливістю системи є застосування алгоритмів «легкої криптографії», зокрема ASCON, що забезпечує високий рівень криптографічного захисту. Найважливіше – це забезпечення конфіденційності, цілісності даних під час їх передавання з пристрою до сервера також можливість опрацювання отриманих метрик у вебінтерфейсі.

Система передбачає повний технологічний цикл роботи з біометричними даними користувача: від зчитування фізіологічних параметрів, таких як пульс, кількість кроків і індекс активності, до їхнього захищеного зберігання в централізованій базі та подальшої візуалізації у вигляді аналітики. Щоб забезпечити роботу в майже реальному часі, потрібне оптимальне налаштування протоколів зв’язку, скорочення затримок і застосування ефективних компактних форматів даних.

Проєктована IoT-мережа розрахована на роботу в одній системі чисельних гаджетів за умов обмежених ресурсів апаратної платформи: обмежена оперативна пам’ять, тактова частота процесора та запас енергії батареї. Через це традиційні криптографічні алгоритми виявляються занадто вимогливими до ресурсів, що робить доцільним використання «легких» криптографічних механізмів. Оптимальним вибором у цьому випадку є алгоритм ASCON – переможець конкурсу NIST Lightweight Cryptography, який відзначається високою продуктивністю, стійкістю до сучасних атак і низьким впливом на енергоспоживання пристрою.

Концепція робочого процесу: смарт-годинник зчитує показники, відправляє їх на сервер, шифрування, користувач отримує інформацію на сайті. IoT-мережа включає збір даних з чисельних пристроїв, передавання даних через мережеву

інфраструктуру, сервер обробки та вебклієнт. До її складу не входять сторонні сервіси глибинної аналітики, медичні бази даних, хмарні сервіси машинного навчання та інші зовнішні інтеграції – хоча архітектура системи спроєктована таким чином, щоб дозволити їх додавання у майбутньому без необхідності повної перебудови системи.

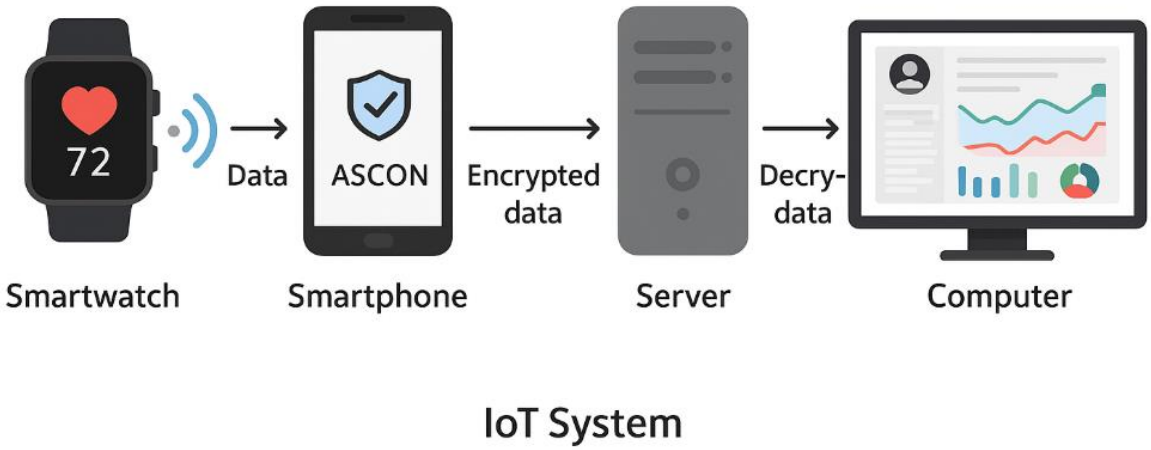

Рисунок 2.1 – Робочого процес обробки даних з IoT-пристроїв

Така система створює фундамент для формування вимог до функціональних модулів, розроблення форматів даних, алгоритмів взаємодії та механізмів криптографічного захисту, що розглядаються в наступних підрозділах.

2.2 Функціональні вимоги

Першою ключовою вимогою є забезпечення безперервного збору фізіологічних показників, зокрема частоти серцевих скорочень, рівня активності, руху, варіабельності пульсу та інших доступних параметрів. Носимий пристрій має зчитувати дані з вбудованих сенсорів із заданою періодичністю, що може адаптуватися залежно від обчислювального навантаження, потужності акумулятора та режиму роботи (наприклад, нормальний режим, енергоощадний режим або режим підвищеної точності). Зібрані дані повинні одразу приводитися

до однакового формату та структури, що забезпечує уніфікованість подальшої обробки й мінімізує навантаження на комунікаційний модуль.

Другою важливою вимогою є забезпечення надійної передачі даних з використанням протоколу BLE. Передача повинна підтримувати надійну сесію зв’язку між годинником та шлюзом (смартфоном або мікроконтролером ESP32) з мінімальною затримкою. Система має автоматично встановлювати з’єднання при наявності доступного каналу, виконувати повторні спроби при розриві зв’язку та забезпечувати контроль цілісності відправлених пакетів. BLE має працювати в безпечному режимі, включаючи процедури парування, автентифікації та використання шифрування на транспортному рівні, щоб запобігти перехопленню пакетів на етапі бездротової передачі. Також оптимізація BLE-пакетів під низьке енергоспоживання, що критично для носимих IoT-пристроїв.

Наступною вимогою є обов’язкове шифрування фізіологічних даних за допомогою криптографічного алгоритму Ascon, рекомендованого стандартом NIST для застосування в обмежених IoT-пристроях. Система повинна виконувати шифрування безпосередньо на пристрої-джерелі – до моменту передачі даних через BLE. Це забезпечує максимальний рівень безпеки, оскільки дані залишають пристрій виключно в зашифрованому вигляді. Алгоритм Ascon має використовувати автентифіковане шифрування AEAD, що гарантує не тільки конфіденційність, але й захист від модифікації або підробки пакетів. Ключі шифрування повинні зберігатися в захищеній області пам’яті, а система має виконувати періодичне оновлення сеансових ключів.

Важливою частиною є також забезпечення передачі зашифрованих пакетів на сервер або у вебзастосунок через шлюз. Шлюз повинен приймати BLE-пакети, перевіряти їх цілісність, передавати дані на сервер за допомогою протоколів HTTP(S), WebSocket залежно від конфігурації. Система повинна підтримувати роботу в режимі реального часу, що передбачає мінімально можливу затримку між моментом зчитування фізіологічного параметра та його відображенням

на вебінтерфейсі. Для цього шлюз має обробляти отримані дані асинхронно та не блокувати інші операції.

На сервері обов’язково проводиться розшифрування, перевірка та обробка отриманих даних. Пакети ASCON коректно декодуються, автентичність повідомлення перевіряється, а структура й формат даних оцінюються на відповідність. У разі виявлення помилок, спотворень або невідповідності підписів дані відкидаються. Після валідації інформація зберігається у базі даних або надходить безпосередньо до модуля візуалізації.

Сайт відображає інформацію у вигляді інфографіки в реальному часі. Тому користувач може дивитись показники, відстежувати тенденції та порівнювати значення з історичними даними.

Також носимий пристрій має працювати тривалий час без підзарядки, тому всі процеси будуть оптимізовані з мінімальним енергоспоживанням. Модулі можна розширювати: додавати сенсори, користувачів і функції вебінтерфейсу без втрати продуктивності.

2.3 Нефункціональні вимоги

Першою групою нефункціональних вимог є продуктивність та швидкодія системи. IoT-мережа повинна забезпечувати мінімальні затримки між моментом зчитування даних на смарт-годиннику та їхнім відображенням на вебінтерфейсі. Час обробки пакета на шлюзі, включаючи шифрування Ascon, не повинен перевищувати допустимого порогу для обчислювально обмежених пристроїв, а серверна обробка і візуалізація повинні відбуватися в режимі реального часу без затримок у відображенні часових рядів. Це дозволяє користувачам отримувати оперативну інформацію про стан свого здоров’я та про фізичну активность. Ще важливий аспект це надійність та безперервність роботи системи. Архітектура IoT-мережі має передбачати механізми відновлення зв’язку у разі тимчасових збоїв BLE-з’єднання або втрати інтернет-каналу на стороні шлюзу. Система повинна автоматично зберігати непередані пакети локально і повторно відправляти їх після

відновлення зв’язку. Всі компоненти системи від носимих пристроїв до серверів і вебінтерфейсу – повинні працювати без критичних відмов протягом тривалого часу. Також ключовою нефункціональною вимогою є енергоефективність. Смарт-годинник і проміжний шлюз повинні оптимізувати використання енергоресурсів, зменшуючи споживання батареї під час передачі даних і шифрування. Це передбачає пакетну передачу BLE-пакетів, адаптивне регулювання частоти опитування сенсорів та використання енергоощадних режимів процесора на шлюзі.

Група вимог стосується масштабованості та модульності мережі. Архітектура повинна дозволяти підключення нових типів IoT-пристроїв та сенсорів без значного перероблення серверної логіки чи протоколів обробки даних. Також має бути забезпечена можливість розширення користувацької бази і додавання нових вебфункцій та аналітичних модулів без впливу на продуктивність основної системи.

Система повинна підтримувати централізоване керування пристроями, оновлення прошивки шлюзу, зміни ключів шифрування та налаштування політик безпеки без втручання користувача. Логи роботи пристроїв і серверів мають бути структурованими та доступними для аналізу адміністратором, а інтерфейси для моніторингу стану системи – простими та інтуїтивно зрозумілими.

Серверна частина та база даних повинні забезпечувати коректну обробку одночасних запитів від великої кількості користувачів без втрати даних або порушення логіки роботи. У разі апаратного збою шлюзу або тимчасової недоступності сервера, система повинна мати механізми відновлення даних та продовження роботи після відновлення підключення.

Окрім криптографічного захисту Ascon, система повинна підтримувати управління доступом на рівні ролей, журналювання подій, аудит дій користувачів і адміністраторів, а також механізми резервного копіювання бази даних для запобігання втраті інформації.

IoT-мережа повинна бути сумісною з основними стандартами Bluetooth Low Energy, протоколами передачі даних, вебтехнологіями та API сторонніх сервісів для інтеграції та розширення аналітичних можливостей.

Таким чином, сукупність нефункціональних вимог забезпечує надійну, масштабовану, енергоефективну та безпечну роботу IoT-мережі, що гарантує стабільний збір, захист та обробку біометричних даних користувача, а також їхню коректну візуалізацію у вебінтерфейсі в режимі реального часу.

2.4 Проєктування формату даних

Першою задачею проєктування формату даних є визначення обов’язкових полів пакета. Кожен пакет повинен містити: унікальний ідентифікатор пристрою (*device_id*) для однозначної ідентифікації джерела даних; часову мітку (*timestamp*), що забезпечує коректну синхронізацію і дозволяє відновлювати порядок надходження даних; безпосередньо набір фізіологічних показників (heart_rate, SpO_2, steps, sleep_quality тощо); а також поля для криптографічного захисту – *nonce* або лічильник, який унеможливлює повторну атаку, та тег автентифікації, що забезпечує цілісність і достовірність даних.

Важливою особливістю проєктованого формату є компактність і ефективність пакета, що критично для BLE-з’єднання з обмеженою пропускною здатністю. Кожен параметр представлений у мінімальному обсязі байтів, що дозволяє передавати пакет за один цикл BLE-транзакції. Наприклад, частота серцевих скорочень може кодуватися як 1–2 байти, рівень насичення крові киснем – 1 байт, кроки за певний інтервал – 2 байти. Така оптимізація зменшує затримки та енергоспоживання пристрою.

Додатково формат включає службові поля для версіонування і сумісності. Кожен пакет містить версію протоколу або схеми (version), що дозволяє системі підтримувати зворотну сумісність при оновленні програмного забезпечення пристрою чи серверної частини. Це забезпечує безперебійну роботу навіть при поетапному впровадженні нових сенсорів або зміни структури даних.

Особливу увагу приділено криптографічним аспектам формату. Передача даних відбувається у зашифрованому вигляді з використанням алгоритму Ascon в режимі AEAD. У форматі пакета передбачено місце для nonce, тегів автентифікації та додаткових метаданих, необхідних для коректної дешифрування на сервері. Це забезпечує високий рівень захисту від модифікацій та перехоплення пакетів у каналі передачі.

Для полегшення обробки на сервері та візуалізації пакет передбачає ієрархічну структуру даних, що дозволяє легко декодувати і розпізнавати окремі параметри. Використання структурованого представлення даних (наприклад, у вигляді JSON або бінарного протоколу з чітким порядком полів) забезпечує сумісність з вебінтерфейсом, API і аналітичними модулями. Такий підхід дозволяє серверу оперативно обробляти великі обсяги даних від кількох тисяч пристроїв одночасно.

Також передбачено розширюваність формату – можливість додавання нових типів сенсорних показників без зміни базової структури пакета. Це досягається введенням опціональних полів, які сервер і клієнт можуть ігнорувати або обробляти залежно від підтримуваної версії протоколу. Такий підхід дозволяє масштабувати систему та інтегрувати нові функції без порушення сумісності з вже встановленими пристроями.

Завершальною вимогою при проєктуванні формату даних є контроль цілісності та тестування. Кожен пакет повинен проходити тестування на предмет відповідності структурі, коректності значень та сумісності з криптографічним протоколом. Це дозволяє забезпечити надійну передачу даних від пристрою до сервера та уникнути помилок при обробці або візуалізації.

Таким чином, проєктування формату даних забезпечує ефективний, безпечний та масштабований обмін інформацією в IoT-мережі, що є критично важливим для реалізації реального часу моніторингу фізіологічних показників користувачів та побудови надійної системи аналітики.

2.5 Планування криптографічного захисту

Планування криптографічного захисту в IoT-мережі є критично важливим етапом проєктування, оскільки забезпечує конфіденційність, цілісність та автентичність даних, що передаються від носимого пристрою до сервера та відображаються на вебінтерфейсі. У контексті проєктованої системи, де використовуються смарт-годинники для збору фізіологічних показників користувача, криптографічний захист визначає безпечний канал зв’язку через BLE, надійну обробку даних на шлюзі та збереження їх на сервері без ризику НСД.

Першою вимогою планування є вибір криптографічного алгоритму. У системі обрано «легкий» алгоритм Ascon у режимі AEAD (Authenticated Encryption with Associated Data), який забезпечує одночасно шифрування та автентифікацію даних. Даний алгоритм відповідає стандартам NIST для застосування в обмежених пристроях IoT, характеризується високою стійкістю до криптографічних атак та низьким споживанням ресурсів процесора і пам’яті на носимих пристроях. Використання AEAD дозволяє перевіряти цілісність повідомлення на сервері та гарантує, що будь-які зміни або підробки пакета будуть виявлені.

Другою вимогою є управління ключами шифрування. Криптографічні ключі зберігаються у захищеній області пам’яті смарт-годинника та шлюзу і передаються на сервер лише у зашифрованому вигляді або генеруються локально на обох кінцях з використанням протоколів обміну ключами. Планування передбачає регулярну ротацію сеансових ключів для мінімізації ризику компрометації. Крім того, система підтримує версіонування ключів, що забезпечує зворотну сумісність між старими та новими пристроями.

Третьою ключовою вимогою є шифрування даних на всіх рівнях передачі. Дані шифруються безпосередньо на смарт-годиннику перед передачею через BLE, що запобігає перехопленню відкритих повідомлень у каналі бездротового зв’язку. На проміжному шлюзі дані залишаються у зашифрованому вигляді, при цьому шлюз виконує лише формування пакетів для відправки на сервер. На сервері дані

дешифруються після перевірки автентичності, що дозволяє запобігти внесенню некоректної або шкідливої інформації.

Четвертою вимогою є захист від повторних атак та спуфінгу. У форматі пакета передбачено використання унікальних значень nonce або лічильників, що змінюються для кожного нового пакета. Сервер виконує перевірку цих значень, відкидаючи повторні або застарілі повідомлення. Додатково планування передбачає автентифікацію пристрою через *device_id* та цифровий підпис пакета, що унеможливлює підробку джерела повідомлення.

Рисунок 2.2 – Поля даних

П’ятою вимогою є інтеграція криптографії з форматом даних та системними протоколами. Всі поля даних структуровані таким чином, щоб забезпечити коректну роботу Ascon та AEAD без втрати ефективності пакета.

Метадані, службові поля, версія протоколу та тег автентичності інтегровані у єдиний формат, що дозволяє шлюзу і серверу швидко виконувати шифрування/дешифрування та перевірку цілісності без додаткових обчислювальних накладних витрат.

Шостою важливою вимогою є моніторинг та аудит криптографічного захисту. Система передбачає логування подій шифрування, передачі та дешифрування даних, реєстрацію спроб НСД несанкціонованого доступу, невдалих перевірок автентичності та аномалій у потоці пакетів.

```
[2025-12-07 09:15:32] INFO  | DeviceID: 01A3 | Action: ENCRYPT | DataType: SensorData
[2025-12-07 09:15:35] INFO  | DeviceID: 01A3 | Action: TRANSMIT | DataType: SensorDat
[2025-12-07 09:15:36] WARN  | DeviceID: 01A3 | Action: DECRYPT | DataType: Command | S
[2025-12-07 09:15:38] INFO  | DeviceID: 01A4 | Action: ENCRYPT | DataType: HealthData
[2025-12-07 09:15:39] ALERT | DeviceID: 01A4 | Action: AUTH_ATTEMPT | Status: FAILED
[2025-12-07 09:15:40] INFO  | DeviceID: 01A3 | Action: TRANSMIT | DataType: SensorDat
[2025-12-07 09:15:42] WARN  | DeviceID: 01A4 | Action: PACKET_ANOMALY | Details: Unexp
```

Рисунок 2.3 – Приклад логів

Ці логи дозволяють відстежувати безпеку системи, виявляти потенційні загрози та проводити аудит відповідно до вимог конфіденційності та захисту персональних даних.

Завершальним елементом планування є масштабованість і гнучкість криптографічного захисту. Алгоритм і протоколи мають підтримувати додавання нових пристроїв, розширення кількості сенсорів та інтеграцію з іншими сервісами без необхідності повного переоснащення системи. Це забезпечує довгострокову підтримку безпечної роботи навіть при розвитку технологій або появі нових загроз.

Таким чином, планування криптографічного захисту гарантує високий рівень безпеки, надійності та конфіденційності даних у проєктованій IoT-мережі, забезпечує стійкість до атак та компрометацій, а також інтегрується у загальну архітектуру системи, не знижуючи її продуктивності та енергоефективності.

2.6 Розроблення логіки взаємодії компонентів

Розроблення логіки взаємодії компонентів IoT-мережі є ключовим етапом проєктування, який визначає порядок обміну даними, синхронізацію процесів і послідовність обробки інформації від смарт-годинника до серверного та вебінтерфейсу. Логіка взаємодії забезпечує ефективність, надійність та безпеку функціонування системи і виступає основою для реалізації всіх функціональних та нефункціональних вимог (рис. 2.4).

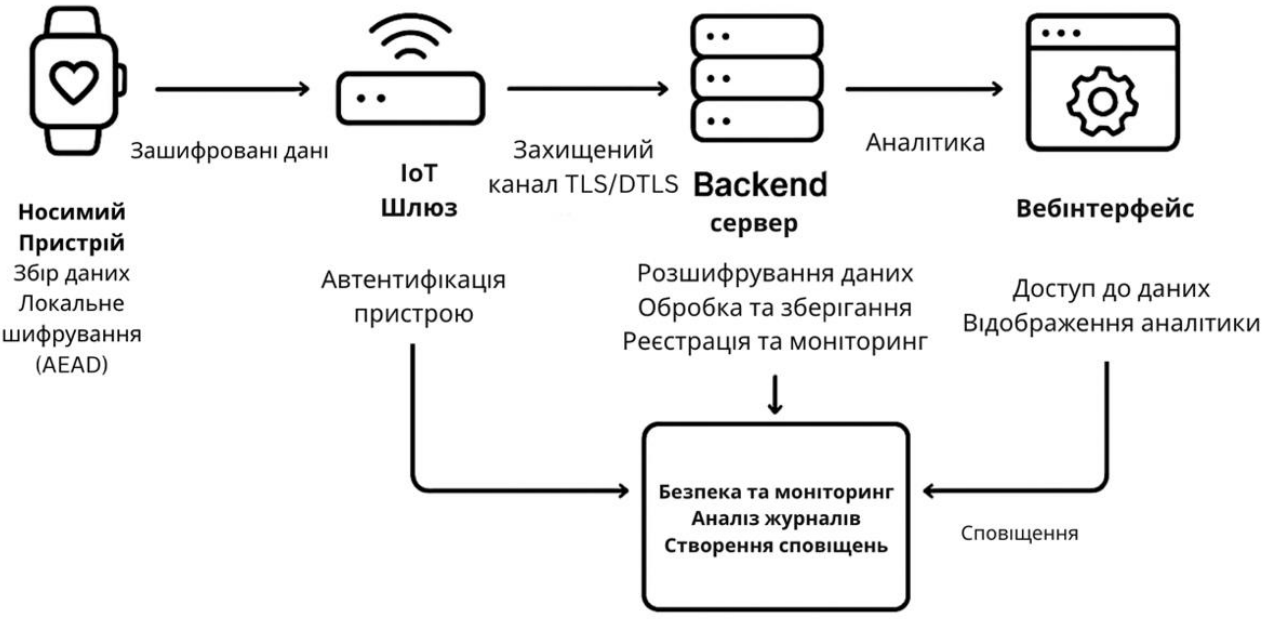

Рисунок 2.4 – Функціональна схема взаємодії компонентів IoT-мережі

Першим етапом взаємодії є збір даних на носимому пристрої. Смарт-годинник періодично зчитує фізіологічні параметри користувача з сенсорів: частоту серцевих скорочень, рівень насичення крові киснем, активність, якість сну та інші показники (рис. 2.5).

Дані після зчитування обробляються локально – фільтруються, агрегуються та підготовлюються до передачі. На цьому етапі застосовуються алгоритми попередньої обробки, що зменшують шум, стабілізують сигнали та забезпечують уніфікований формат пакета для подальшого шифрування.

Рисунок 2.5 – Сенсори смарт-годинника для зчитування фізіологічних параметрів користувача

Другим етапом є шифрування та формування пакетів для передачі. Дані, підготовлені на пристрої, зашифровуються алгоритмом Ascon у режимі AEAD. Криптографічний модуль формує пакет, який включає метадані (device_id, timestamp, версію протоколу), зашифровані дані та тег автентичності. Пакет організовано так, щоб він був сумісний із каналом BLE, з мінімальною затримкою та оптимізованою структурою для економії енергії та пропускної здатності.

Третій етап – передача даних через проміжний шлюз. BLE-з’єднання встановлюється між смарт-годинником і шлюзом (смартфон або ESP32-модуль). Шлюз приймає пакети, перевіряє цілісність та, у разі успішної валідації, передає їх на сервер через Інтернет за допомогою протоколів WebSocket. Логіка взаємодії передбачає повторну відправку пакетів у разі втрати зв’язку, адаптивне регулювання частоти передачі та буферизацію даних для забезпечення безперервності потоку.

Четвертим етапом є обробка даних на сервері. Сервер отримує пакети, виконує перевірку автентичності за тегами AEAD, дешифрує дані та зберігає їх у базі. Логіка обробки передбачає агрегування показників, нормалізацію, прив’язку до користувача та часових міток, а також підготовку даних для візуалізації. У разі виявлення аномалій, невідповідності тегів або збоїв дешифрування система відхиляє пакет і генерує відповідне повідомлення в логах для подальшого аудиту.

П’ятий етап – взаємодія з вебінтерфейсом. Сервер надає API для передачі даних вебклієнта в режимі реального часу або за запитом користувача. Логіка взаємодії передбачає підписку на оновлення, отримання історичних даних, фільтрацію по періодах та агрегованих метриках, а також обробку подій критичних значень. Вебінтерфейс динамічно відображає інформацію у вигляді графіків, діаграм та індикаторів, що дозволяє користувачеві здійснювати аналіз фізіологічних показників.

Шостим етапом є логування та аудит роботи компонентів. Кожна взаємодія – від зчитування сенсорів до візуалізації на вебінтерфейсі – логуються з включенням часових міток, *device_id* та результатів перевірок цілісності. Логування забезпечує можливість відтворення подій, виявлення збоїв, аналізу продуктивності та контролю безпеки системи.

Завершальною вимогою є координація та синхронізація компонентів. Система має підтримувати асинхронний обмін даними, обробку пакетів у чергах, контроль послідовності та черговість подій, що дозволяє уникнути конфліктів при одночасній роботі великої кількості пристроїв та користувачів. Логіка взаємодії передбачає також автоматичне управління ресурсами: режим енергозбереження для пристроїв, повторну передачу пакетів, обробку критичних подій та масштабування серверних ресурсів за потребою.

Таким чином, розроблення логіки взаємодії компонентів забезпечує ефективну, безпечну та масштабовану роботу IoT-мережі, гарантує стабільний потік даних від сенсорів до вебінтерфейсу та є основою для реалізації всіх функціональних та криптографічних вимог проєктованої системи.

2.7 Вибір технологій та інструментів

Таким чином, вибір технологій та інструментів забезпечує повну інтеграцію апаратних і програмних компонентів IoT-мережі, дозволяє ефективно реалізувати шифрування та захист даних, гарантує швидку обробку потоків фізіологічної інформації та надає користувачу інтерактивну та надійну систему моніторингу. Вибір технологій та інструментів для системи є ключовим етапом проєктування, який визначає ефективність, надійність та масштабованість всієї системи. У проєктованій IoT-мережі фізіологічні дані користувача зчитуються зі смарт-годинника, шифруються алгоритмом Ascon і передаються на сервер для обробки та візуалізації, тому технології обираються з урахуванням вимог до продуктивності, безпеки та енергоефективності.

Першим компонентом є апаратна частина. Смарт-годинник Apple Watch SE 2 забезпечує зчитування фізіологічних даних та підтримує протокол Bluetooth Low Energy (BLE) для передачі даних на шлюз.

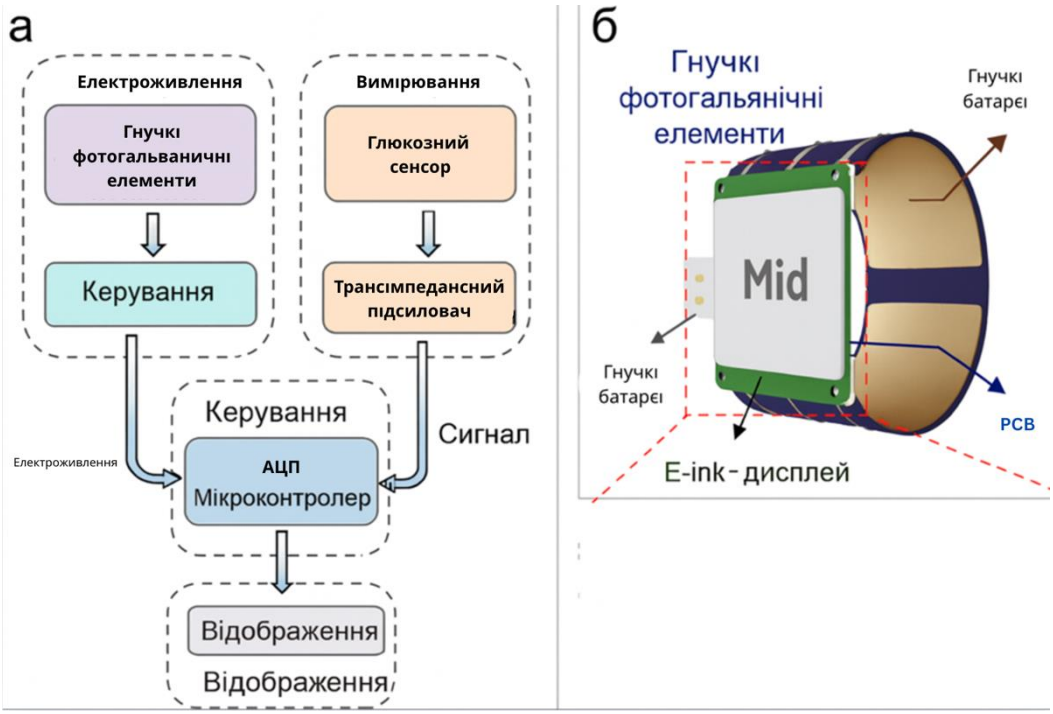

Рисунок 2.6 – Інтеграція апаратних і програмних компонентів IoT-мережі

Проміжним шлюзом виступає мікроконтролер ESP32, який підтримує одночасне підключення декількох пристроїв, виконує шифрування даних Ascon та

передає їх на сервер через HTTPS. ESP32 обрано через його високу обчислювальну потужність при низькому споживанні енергії та наявність готових бібліотек для BLE та криптографії.

Другим компонентом є програмне забезпечення прошивки і шлюзу. На стороні смарт-годинника використовується прошивка, яка зчитує дані з сенсорів, формує пакети у бінарному форматі, виконує шифрування Ascon і передає їх через BLE. Шлюз на ESP32 використовує прошивку, що забезпечує буферизацію пакетів, перевірку цілісності, надійну доставку на сервер і підтримку паралельної роботи з декількома пристроями. Розробка прошивки здійснюється у середовищі PlatformIO з використанням бібліотек BLE та Ascon (додаток А).

Третім компонентом є серверна частина. Сервер на Node.js приймає зашифровані пакети, виконує перевірку автентичності та дешифрування, зберігає дані в базі PostgreSQL і забезпечує доступ вебінтерфейсу через REST API та WebSocket. Для обробки часових рядів використовується InfluxDB, що дозволяє ефективно агрегувати та аналізувати фізіологічні дані користувачів у реальному часі.

Четвертим компонентом є вебінтерфейс та візуалізація даних. Для побудови динамічних графіків і діаграм застосовується React з бібліотекою Recharts. Вебінтерфейс забезпечує інтерактивне відображення показників користувачів у реальному часі, підтримує фільтрацію за періодами, аналіз трендів та побудову індикаторів критичних значень. Взаємодія з сервером відбувається через WebSocket, що забезпечує швидке оновлення даних без затримок.

П’ятою групою є криптографічні інструменти. Для реалізації шифрування Ascon використовується бібліотека Ascon-C, яка дозволяє виконувати AEAD-шифрування на ресурсозалежних пристроях, таких як ESP32 і браузер, забезпечуючи захист від перехоплення та модифікації пакетів.

Окремо передбачено інструменти для тестування та моніторингу. Тестування BLE-з’єднань і шифрування проводиться за допомогою емуляторів пристроїв і логів шлюзу. Серверні компоненти тестуються із застосуванням unit-тестів

та інтеграційних тестів. Моніторинг стану системи реалізується за допомогою Grafana та Prometheus для відстеження продуктивності шлюзу, швидкості обробки пакетів та стану серверів у режимі реального часу.

Таким чином, обрані технології та інструменти забезпечують ефективну, безпечну та масштабовану роботу IoT-мережі, гарантують захист даних користувачів на всіх етапах їхньої обробки та дозволяють реалізувати повноцінну вебаналітику фізіологічних показників у реальному часі.

2.8 Проєктування структури бази даних

Основою структури є реляційна база даних PostgreSQL, яка забезпечує надійне зберігання структурованих даних і підтримку складних запитів для аналітики. Для збереження часових рядів фізіологічних показників використовується InfluxDB, що оптимізовано для високочастотних записів і агрегування показників у реальному часі. Поєднання цих двох систем дозволяє ефективно розділити завдання збереження метаданих та великих обсягів сенсорних даних.

Основними сутностями бази даних є користувач, пристрій, сенсорні дані та криптографічні теги. Таблиця користувачів містить інформацію про унікальний ідентифікатор, ім’я, контактні дані та налаштування облікового запису. Таблиця пристроїв містить *device_id*, модель пристрою, версію прошивки та статус підключення. Сенсорні дані включають фізіологічні показники (heart_rate, SpO_2, steps, sleep_quality), часову мітку та унікальний ідентифікатор пакета. Таблиця криптографічних тегів зберігає nonce, тег автентичності та інформацію про версію протоколу, що забезпечує перевірку цілісності і автентичності даних (рис. 2.7).

Важливим аспектом проєктування є оптимізація схеми для роботи з великими потоками даних. Для цього сенсорні дані зберігаються у InfluxDB з часовими індексами, що дозволяє швидко виконувати запити на отримання історичних даних та агрегатів за певний період. Водночас PostgreSQL

використовується для зберігання основної інформації про користувачів і пристрої, а також для інтеграції з вебінтерфейсом та аналітичними модулями.

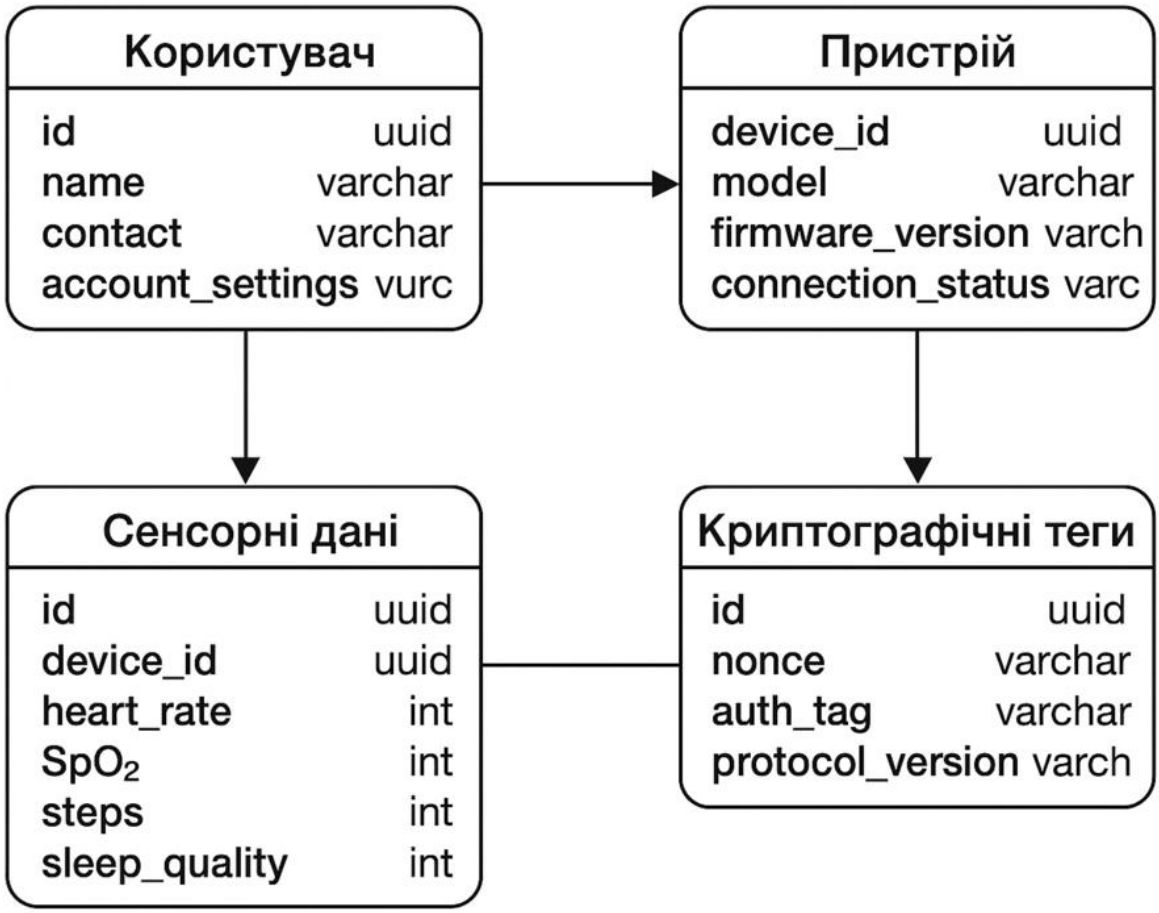

Рисунок 2.7 – Структура реляційної БД

Проєктування також передбачає відокремлення критичних і чутливих даних. Криптографічні теги та ключові метадані зберігаються в окремих таблицях з обмеженим доступом, що забезпечує додатковий рівень безпеки і запобігає НСД. Для забезпечення цілісності даних застосовуються зовнішні ключі та механізми транзакцій PostgreSQL, що гарантують узгодженість між таблицями користувачів, пристроїв та сенсорних показників.

Важливим елементом є також оптимізація під високонавантажені сценарії. Таблиці сенсорних даних організовані з розділенням на партиції за часовими інтервалами, що дозволяє виконувати швидкі агрегатні запити і зменшити час відповіді системи при роботі з великою кількістю користувачів. Для запобігання

дублювання пакетів передбачено використання унікальних ключів по *device_id* та *timestamp*.

Таким чином, спроєктована структура бази даних забезпечує ефективне, безпечне і масштабоване зберігання даних, дозволяє швидко виконувати аналітичні запити, гарантує цілісність і автентичність інформації, а також підтримує інтеграцію з вебінтерфейсом та аналітичними модулями системи.

Висновки до розділу 2

У даному розділі було детально розглянуто концепцію побудови IoT-мережі для збору, шифрування, передачі та обробки фізіологічних даних від носимих пристроїв. Наведено загальну архітектуру рішення та визначено ключові компоненти, що забезпечують повний цикл роботи системи – від фіксації показників сенсорами до їхнього відображення у веб-інтерфейсі.

Було описано основні гаджети, що використовуються в системі, включаючи смарт-годинники та сенсорні модулі, а також наведено характеристику типів сенсорів, інтегрованих у пристрої. Окрему увагу приділено програмним компонентам: механізмам збору та попередньої обробки даних, засобам шифрування ASCON, серверам прийому трафіку, механізмам верифікації криптографічних тегів, а також системам зберігання даних – PostgreSQL та InfluxDB.

Представлена концепція демонструє логічну цілісність рішення, забезпечує розширюваність, масштабованість і можливість інтеграції додаткових сенсорів або сервісів у майбутньому. У результаті сформовано повне бачення того, як апаратні та програмні елементи взаємодіють між собою для захисту даних в IoT-мережі.

3 РОЗРОБКА IOT-МЕРЕЖІ ІЗ ЗАХИСТОМ ДАНИХ НА ОСНОВІ «ЛЕГКИХ» КРИПТОГРАФІЧНИХ АЛГОРИТМІВ

3.1 Постановка задачі та вимоги до системи

Метою розробки є створення прототипу IoT-мережі, у якій чисельні носимі пристрої (смарт-годинники, фітнес-трекери тощо) через систему вбудованих датчиків здійснюють безперервний збір біометричних даних користувача (частота серцевих скорочень, температура тіла, рівень активності, кількість кроків тощо) та передає їх на серверну платформу для подальшої обробки й візуалізації. Головною особливістю системи є забезпечення конфіденційності, цілісності й автентичності даних за рахунок застосування алгоритмів «легкої» криптографії, які пристосовані для використання на мікроконтролерах із обмеженими ресурсами.

У сучасному смарт-годиннику, тим більше флагманського сегмента продуктивність досить висока. наприклад кількість оперативної пам'яті вже близько 1 ГБ, процесор 2 ядерний, це дозволяє використовувати на такому пристрої досить високий рівень криптографічних алгоритмів, але важливо оптимізувати системи і для більш слабких пристроїв, тому в роботі в тому числі ставиться завдання створити систему за допомогою алгоритму АСКОН, саме тому що йому не потрібно занадто багато ресурсів.

Дуже важливо шифрувати дані датчиків (пульс, активність, сон тощо), тому що ці дані є прямим відображенням фізичного стану користувача та містять значний обсяг інформативності, що дозволяє робити конкретні висновки про стан здоров’я людини. Наприклад, за динамікою пульсу та HRV можна встановити наявність кардіологічних відхилень, за патернами рухової активності – визначити спосіб життя та рівень навантаження, а за параметрами сну – виявити ознаки хронічної втоми, стресу або порушень циркадних ритмів. Таким чином, біометричні дані є фактично медичними і належать до категорії персональної інформації підвищеної чутливості, що потребує обов’язкового захисту під час збирання, передавання, зберігання та обробки.

У разі відсутності шифрування такі дані виявляються надзвичайно вразливими до перехоплення або модифікації. Всі подібні пристрої, які люди носять на руці, передають інформацію бездротовими каналами, за типом: BLE, Wi-Fi, мобільні мережі або протоколи далекої дії на зразок LoRaWAN. Усі вони, залежно від умов використання, піддаються атакам типу «людина посередині» (Man-in-the-Middle), коли зловмисник може непомітно перехопити потік даних або навіть модифікувати його під час передавання. Якщо передавання здійснюється у відкритому вигляді, стороння особа отримує можливість не лише спостерігати за біометричними показниками користувача в реальному часі, але й формувати повну історію його активностей. Крім того, такі атаки дозволяють підміняти окремі пакети, що може призвести до потрапляння на сервер помилкових або сфальсифікованих даних. Ця проблема особливо критична, коли дані використовуються для аналітики, медичних рекомендацій або сигналізації про небезпечні стани.

Особливо небезпечним є те, що біометрична інформація може використовуватися для профілювання користувача. Якщо протягом тривалого часу збирати дані про пульс, активність або сон, можна точно відтворити розклад життя людини: визначити час її пробудження, робочий графік, періоди перебування поза домом, інтенсивність тренувань, рівень стресу та інші закономірності. Такі відомості мають високу цінність для третіх сторін. Страхові компанії можуть використовувати їх для перегляду тарифів або оцінювання страхових ризиків, роботодавці – для неофіційного контролю продуктивності, а маркетингові організації – для побудови надточних рекламних профілів. Зловмисники, які отримали доступ до незахищених даних, можуть визначити, коли користувача немає вдома, або навіть скласти уявлення про його фізичні обмеження чи хвороби. Такий рівень втручання є прямою загрозою приватності та безпеці людини.

Не менш серйозною є загроза ідентифікації користувача за унікальними біометричними патернами. Хоча дані про пульс і активність на перший погляд можуть здаватися анонімними, сучасні дослідження доводять, що варіабельність

серцевого ритму, добові цикли та навіть характер руху людини мають індивідуальні особливості. Це означає, що навіть без прямої прив’язки до імені користувача можна встановити його особу через зіставлення з іншими джерелами інформації. Так формується так званий «біометричний відбиток», який стає унікальним ідентифікатором людини, а отже, може бути використаний для відстеження та деанонімізації. У відсутність шифрування цей ризик багаторазово зростає.

Окремої уваги заслуговує юридичний аспект. У більшості країн світу біометричні та медичні дані охороняються найсуворішими нормами законодавства – такими як GDPR у Європейському Союзі, HIPAA у США та аналогічні стандарти у національних законодавствах. Порушення правил обробки персональної медичної інформації може призвести до значних штрафів, що вимірюються мільйонами євро, а також до кримінальної або адміністративної відповідальності. Компанії або розробники, які нехтують належним захистом таких даних, автоматично несуть відповідальність у випадку витоку, незаконного доступу або компрометації інформації. Навіть у межах академічних проєктів необхідність забезпечення конфіденційності та цілісності даних є важливою складовою професійної етики та технологічної грамотності майбутнього фахівця.

Варто враховувати й поведінкові очікування користувачів. Сучасні користувачі цифрових сервісів очікують прозорості та безпеки у питаннях обробки персональної інформації. Якщо система не гарантує збереження приватності, не використовує шифрування або містить очевидні вразливості, рівень довіри до такого продукту різко знижується. Навпаки, наявність надійних криптографічних механізмів стає конкурентною перевагою та фактором, що позитивно впливає на прийняття технології користувачами.

Окремо слід підкреслити, що IoT-сектор загалом є одним із найбільш вразливих у світі кібербезпеки. Значна частина IoT-пристроїв використовує слабкі або застарілі протоколи, має низький рівень аутентифікації, відкриті порти або обмежені можливості оновлення прошивок. Через це саме такі пристрої часто

стають першочерговою ціллю для DDoS-атак, ботнетів чи прихованого доступу до мереж. Використання шифрування в IoT-системах не лише захищає дані, але й запобігає перетворенню пристрою на інструмент у руках зловмисників.

З огляду на це шифрування біометричних даних є не просто рекомендованою, а обов’язковою умовою проєктування сучасних IoT-рішень. Воно забезпечує три ключові властивості інформаційної безпеки: конфіденційність, цілісність та автентичність. Для носимих пристроїв, які мають обмежені обчислювальні ресурси та працюють від акумулятора, оптимальним підходом є застосування «легких» криптографічних алгоритмів, таких як ASCON, оскільки вони забезпечують високий рівень криптостійкості за мінімального навантаження на апаратну платформу. Ці алгоритми дозволяють захистити дані на всіх етапах – від моменту зчитування до відправки на сервер і подальшого зберігання.

Загалом необхідно підкреслити, що захист даних – це комплексне завдання, яке включає не лише шифрування каналу, але й безпечне управління ключами, захист OTA-оновлень, контроль доступу, шифрування даних у стані спокою, а також безпечну інтеграцію із серверними та вебкомпонентами системи. Лише поєднання цих заходів забезпечує надійний захист біометричної інформації користувача та створює систему, стійку до сучасних кіберзагроз.

3.2 Архітектура IoT-мережі

Розробка ефективної IoT-мережі для збору, захисту, передачі, оброблення та візуалізації біометричних даних користувача вимагає продуманого підходу до архітектури, що враховує як апаратні обмеження носимих пристроїв, так і вимоги до безпеки та конфіденційності даних. У розробці криптографічних алгоритмів існує компроміс між продуктивністю та ресурсами, необхідними для заданого рівня безпеки. Продуктивність можна виразити такими термінами, як споживання енергії, затримка та пропускна здатність. Ресурси, необхідні для апаратної реалізації, зазвичай виражаються в площі вентиля, еквівалентах вентилів

або логічних блоках (також відомих як конфігуровані логічні блоки, логічні елементи, адаптивні логічні модулі або зрізи). У програмному забезпеченні це відображається у використанні регістрів, оперативної пам'яті та ПЗП. Архітектура такої системи будується за принципом багаторівневої взаємодії компонентів, де кожен рівень виконує чітко визначені функції, а разом вони забезпечують надійну роботу та захист інформації на всіх етапах збору, обробки та передачі (рис. 3.1).

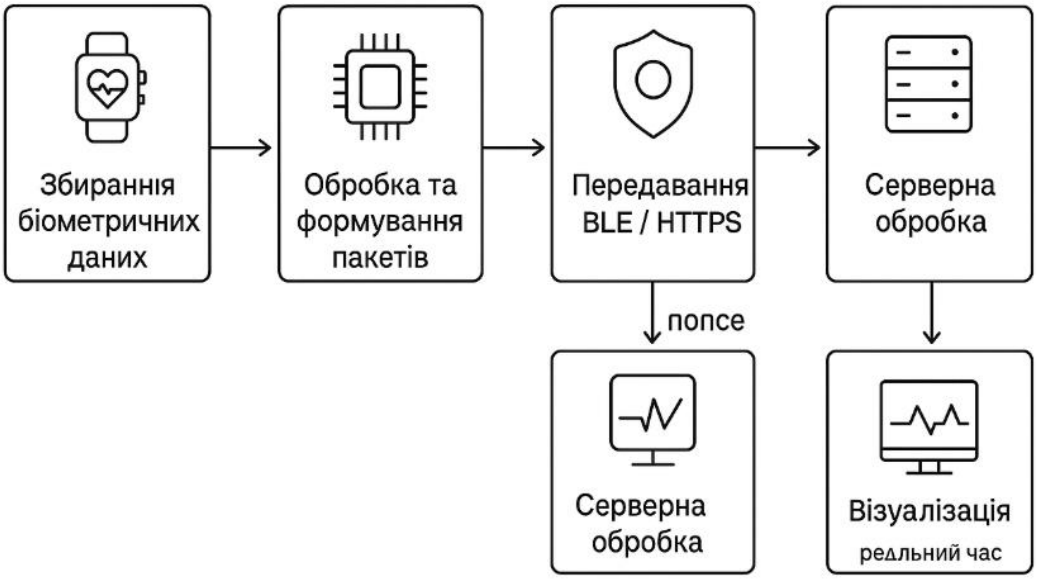

Рисунок 3.1 – Візуалізація послідовності дій в IoT-системі збирання та обробки біометричних даних (без шифрування)

На нижньому рівні знаходяться смарт-годинники та інші носимі датчики, що безпосередньо вимірюють фізіологічні показники користувача. Ці пристрої мають обмежені обчислювальні ресурси та обмежену енергоємність, тому обробка даних і шифрування повинні бути максимально оптимізованими. Саме тому в системі використовується «легкий» криптографічний алгоритм ASCON, який забезпечує високий рівень безпеки при мінімальному навантаженні на процесор і споживання енергії. Алгоритм дозволяє здійснювати одночасне шифрування та автентифікацію даних, що критично для запобігання їх модифікації або підміни під час передачі.

Далі дані передаються на проміжний рівень шлюзу або мобільного застосунку, який виступає посередником між носимим пристроєм та серверною частиною системи. Шлюз забезпечує декілька ключових функцій: об’єднання

даних з різних датчиків, попередню обробку, буферизацію у випадку тимчасової втрати з’єднання та додатковий контроль цілісності. Важливою особливістю архітектури є те, що на цьому рівні додаткове шифрування не потрібно, оскільки дані вже зашифровані на пристрої користувача, проте використовується автентифікація для підтвердження легітимності шлюзу і виключення НСД.

На рівні хмарного сервера здійснюється прийом, зберігання та обробка даних. Серверна частина архітектури забезпечує масштабованість системи та її інтеграцію з вебінтерфейсом для візуалізації даних. Тут застосовуються додаткові заходи безпеки, такі як контроль доступу користувачів, журналювання дій, резервне зберігання та шифрування даних у стані спокою. Завдяки цьому навіть у разі компрометації окремого вузла дані залишаються захищеними. На сервері також відбувається декодування зашифрованих повідомлень, а результат обробки передається до вебзастосунка для відображення у зручному форматі, графіках або аналітичних панелях.

Взаємодія між компонентами IoT-мережі будується через стандартизовані бездротові протоколи, зокрема Bluetooth Low Energy для з’єднання між годинником та мобільним пристроєм, а також Wi-Fi або мобільний інтернет для передачі даних на сервер. Вибір таких протоколів обумовлений балансом між енергоефективністю, швидкістю передавання даних та поширеністю пристроїв серед користувачів. При цьому всі канали передачі даних захищені за допомогою шифрування ASCON, що гарантує цілісність та конфіденційність навіть у відкритих мережах.

Особливу увагу в архітектурі приділено модульності та масштабованості системи (рис. 3.2). Кожен компонент – датчик, шлюз, сервер, вебінтерфейс – функціонує як окремий модуль з чітко визначеними інтерфейсами. Це дозволяє легко інтегрувати нові типи датчиків, додаткові функції аналітики або оновлювати алгоритми безпеки без глобальної перебудови системи. Крім того, модульність сприяє підвищенню надійності, оскільки відмови одного компонента не призводять до повної зупинки системи.

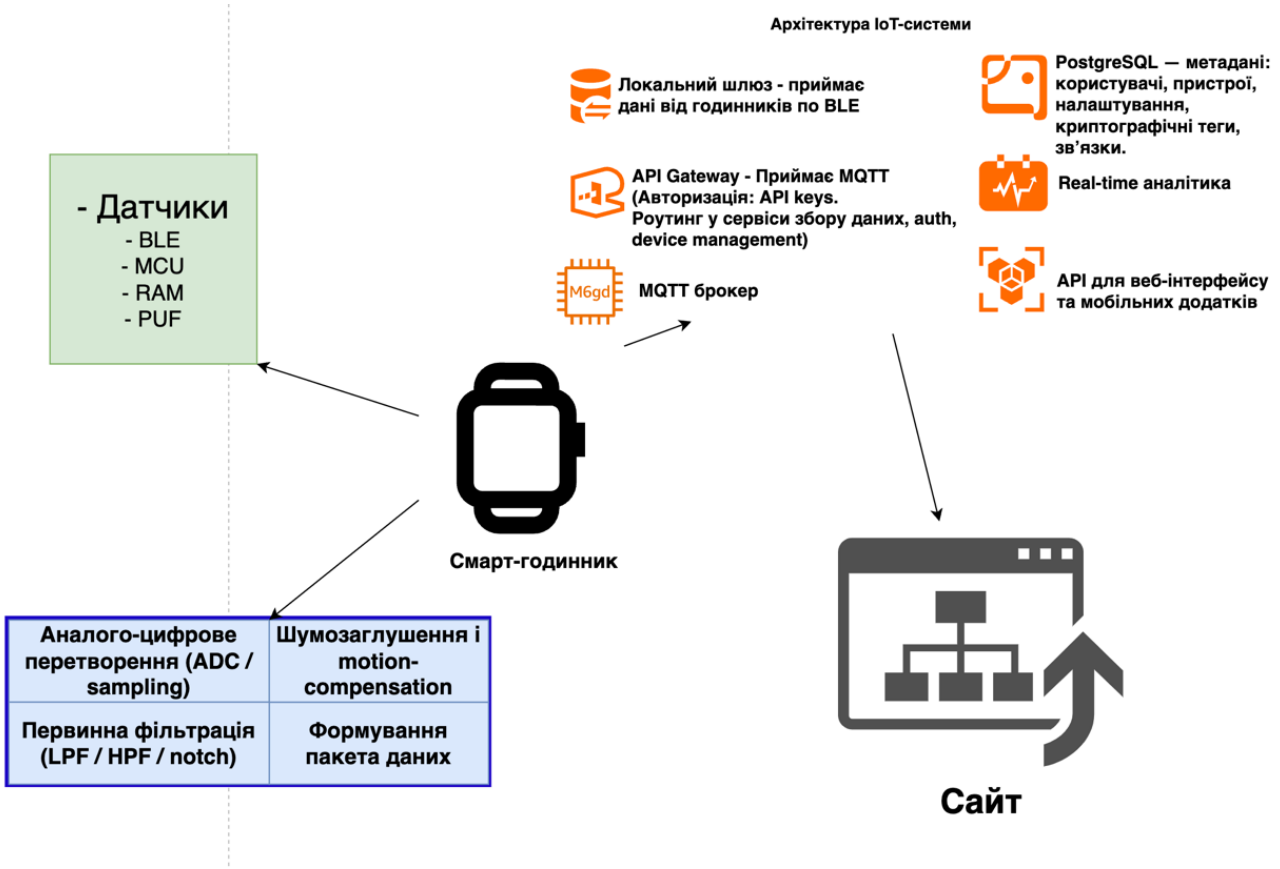

Рисунок 3.2 – Архітектура системи

Завдяки такій архітектурі система забезпечує ефективний цикл збору, обробки та передачі біометричних даних, де кожен рівень відповідає за конкретну частину процесу, а загальна інтеграція гарантує безпеку, цілісність та доступність інформації. Впровадження «легкого» криптографічного алгоритму ASCON дозволяє поєднати надійний захист даних із збереженням енергоефективності пристроїв, що особливо важливо для носимих технологій з обмеженим ресурсом батареї. В результаті користувач отримує надійну систему збору персональної інформації, що повністю відповідає сучасним вимогам кібербезпеки та стандартам конфіденційності. Основною перевагою програмних методів реалізації захисту є їх гнучкість, тобто можливість швидкої зміни алгоритмів шифрування. Основним же недоліком програмної реалізації є істотно менша швидкодія в порівнянні з апаратними засобами (приблизно в 10 разів).

3.3 Розробка програмного забезпечення для збору та передачі даних

Процес розробки програмного забезпечення для IoT-мережі, яка збирає біометричні дані користувача, є ключовим етапом у забезпеченні її функціональності та безпеки. Основним завданням програмного забезпечення є забезпечення безперервного та надійного збору даних з носимих пристроїв, їх попередньої обробки, шифрування та передачі на сервер для подальшої аналітики. У нашому випадку система розробляється для роботи зі смарт-годинниками, які вимірюють пульс, варіабельність серцевого ритму, активність користувача та інші фізіологічні показники.

Програмне забезпечення складається з кількох рівнів, що взаємодіють між собою. Перший рівень розташовується безпосередньо на носимому пристрої. Тут реалізуються драйвери сенсорів, які забезпечують стабільне зчитування даних у режимі реального часу.

Після попередньої обробки дані зашифровуються безпосередньо на пристрої за допомогою «легкого» криптографічного алгоритму ASCON. Важливо зазначити, що ASCON реалізує одночасне шифрування та автентифікацію повідомлень, що гарантує не тільки конфіденційність даних, а й їх цілісність. Алгоритм добре оптимізований для процесорів з обмеженою потужністю, що дозволяє виконувати шифрування без значного збільшення енергоспоживання, що критично для носимих пристроїв з обмеженою батареєю.

Далі зашифровані пакети даних передаються на проміжний рівень – мобільний застосунок або шлюз. Тут програмне забезпечення виконує додаткові функції:

1) буферизація даних – у випадку тимчасової втрати з’єднання з сервером дані зберігаються локально, щоб уникнути втрат;

2) контроль автентичності пристрою – передача даних можливе лише від авторизованих смарт-годинників, що запобігає атакам типу spoofing;

3) форматування повідомлень – перетворення даних у стандартизований формат, наприклад JSON, для подальшої обробки на сервері.

На серверному рівні програмне забезпечення приймає зашифровані дані, перевіряє їх цілісність і автентичність та розшифровує за допомогою ключів, збережених у захищеному сховищі. Тут дані зберігаються у базі даних для подальшої аналітики та візуалізації. Сервер також забезпечує інтеграцію з вебінтерфейсом, що дозволяє користувачу відстежувати свої фізіологічні параметри у реальному часі та отримувати аналітичні висновки.

Така схема допомагає зрозуміти, як дані рухаються у системі і на яких етапах застосовується захист.

Таблиця 3.1 – Основні модулі програмного забезпечення та їх функцій

Рівень IoT-мережі	**Модуль**	**Основні функції**	**Споживані ресурси**	**Примітки**
Носимий пристрій	Драйвери сенсорів; Попередня обробка; Шифрування ASCON.	Збір даних пульсу, HRV, активності; Фільтрація шумів, нормалізація; Конфіденційність та автентифікація.	Частота зчитування: 1 Гц; обсяг пакету: 32 байти; Використання CPU: 5% від часу процесора; пам’ять: 2 кбайт; Час шифрування одного пакету (32 байти): 0,8 мс; енергоспоживання: 0,1 мДж.	Оптимізовано під низьке енергоспоживання; Зменшення похибок вимірювань; «Легкий алгоритм» для ресурсозберігаючих пристроїв.
Мобільний шлюз	Буферизація та контроль автентичності	Тимчасове зберігання даних, авторизація	Буфер: 512 кбайт; затримка передачі: ≤100 мс	Захист від підміни та втрати даних
Сервер	Прийом та зберігання	Перевірка цілісності, розшифрування, зберігання в базі	CPU: 2 ядра × 2,5 ГГц; RAM: 4 Гбайт; дисковий простір: 50 Мбайт/тисячу користувачів	Захищене сховище ключів
Вебінтерфейс	Візуалізація даних	Графіки, аналітика,	Завантаження графіків: < 1 секунда; передача	Інтерактивний інтерфейс для користувача

Рівень IoT-мережі	Модуль	Основні функції	Споживані ресурси	Примітки
		історія показників	даних: 2 кбайт/запит	

Таким чином, програмне забезпечення IoT-мережі виконує комплексну функцію збору, шифрування та передачі даних, забезпечуючи їх надійний захист та безперебійну роботу всієї системи. Впровадження «легкого» алгоритму ASCON дозволяє поєднати ефективність шифрування з обмеженими ресурсами носимих пристроїв, що є критично важливим для реальних умов експлуатації.

Розробка програмного забезпечення для збору та передачі даних у спроєктованій IoT-мережі є ключовим етапом реалізації проєкту, адже саме програмна логіка визначає, як носимий пристрій, у розглянутому випадку смарт-годинник Apple Watch SE 2, збирає біометричні дані, обробляє їх і передає на сервер у зашифрованому вигляді. Для реалізації цієї функціональності було обрано мову Python, оскільки більшість мікроконтролерів для смарт-годинників, включаючи ARM Cortex-M серії, мають оптимізовані компілятори під ці мови, що дозволяє максимально ефективно використовувати обмежені ресурси пам’яті та процесора. Використання Python дозволяє реалізувати роботу з периферійними пристроями (сенсорами пульсу, акселерометром, гіроскопом) на низькому рівні та контролювати кожен аспект збору даних у реальному часі.

Програма побудована за модульним принципом і складається з декількох основних блоків: збір даних, їх попередня обробка, шифрування та передача на сервер. Модуль збору даних працює з сенсорами смарт-годинника через інтерфейси I2C та SPI, зчитуючи показники пульсу, варіабельності серцевого ритму (HRV), рівня активності та інших параметрів із частотою від 1 Гц до 10 Гц залежно від режиму роботи. Для зменшення шумів та підвищення точності даних у роботі було реалізовано прості алгоритми фільтрації на кшталт ковзного середнього (англ. Moving Average, MA) та виділення піків пульсу, що дозволяє уникнути спотворень сигналу під час активних рухів користувача (рис. 3.3).

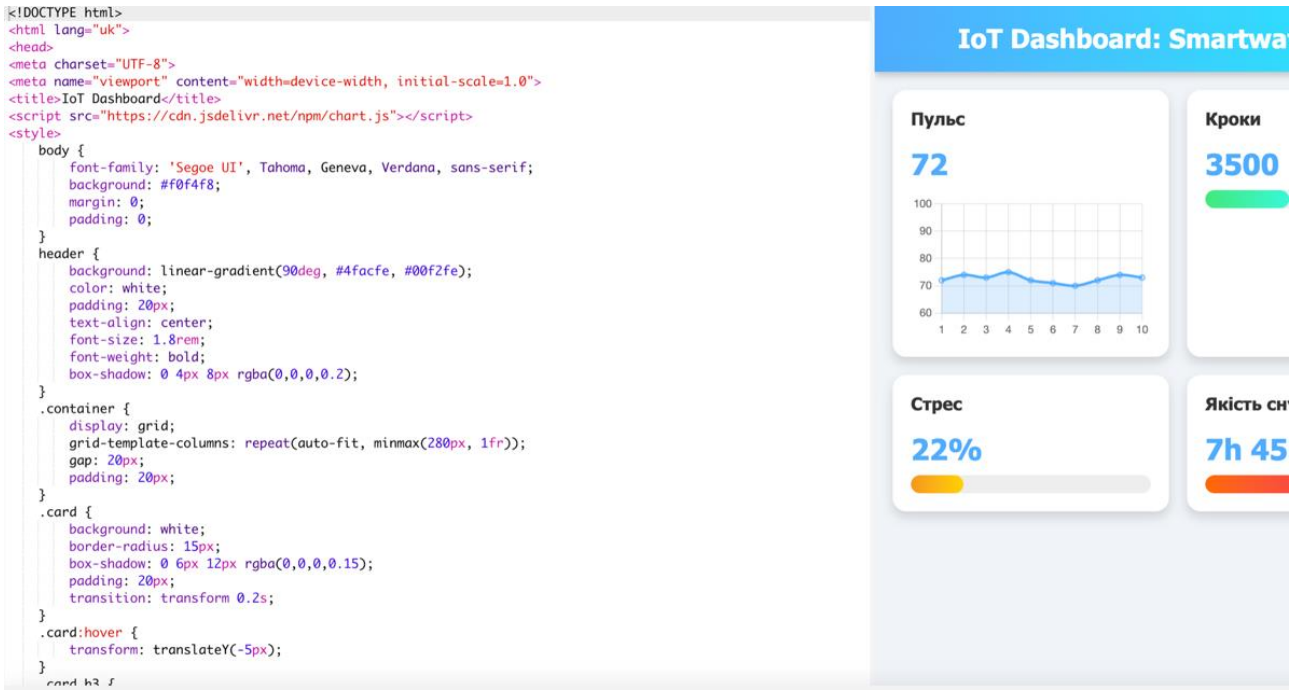

Рисунок 3.3 – Реалізація програми

Після збору сирі дані проходять попередню обробку та упаковуються у структури, готові до шифрування. Для цього я реалізував власну структуру даних у Python, яка містить: мітку часу (*timestamp*), унікальний ідентифікатор користувача, значення пульсу, HRV, кроків та статусу сну. Такий підхід забезпечує легку інтеграцію з криптографічним модулем ASCON, дозволяючи шифрувати цілі пакети без потреби у додатковому розборі та форматуванні на стороні сервера.

Шифрування реалізоване через бібліотеку ASCON, яку було інтегровано безпосередньо у код мікроконтролера ESP32. Для кожного пакету даних генерується nonce та використовується ключ користувача, що дозволяє гарантувати унікальність шифрування та запобігати повторним атакам (replay attacks). Код обробки шифрування виконується у вигляді окремого класу з методами *encrypt()* та *decrypt()*, що робить його легкою для тестування та масштабування. Важливо, що шифрування відбувається на самому пристрої без участі сервера. Це дозволяє ніколи не передавати незашифровані дані через відкриті мережі, а весь процес займає менше 1 мілісекунди на пакет обсягом 32 байти, не перевантажуючи процесор і не зменшуючи час автономної роботи годинника.

Передача даних на сервер реалізована через BLE (Bluetooth Low Energy), а у разі доступу до Wi-Fi – через HTTPS-запити до REST API вебсервера. Для BLE було використано стандартні сервіси та характеристики, створивши власний профіль, який дозволяє передавати пакети даних у форматі *byte array*. Кожен пакет має контрольну суму, щоб сервер міг перевірити цілісність навіть перед розшифруванням. На стороні сервера дані розшифровуються за допомогою того самого ключа ASCON, прив’язаного до конкретного користувача, і зберігаються у базі даних PostgreSQL у зашифрованому або хешованому вигляді залежно від чутливості показника.

Особливу увагу було приділено обробці помилок та відновленню даних. Програма має внутрішній буфер, який накопичує пакети у разі втрати зв’язку з сервером, і автоматично повторно відправляє їх після відновлення підключення. Це забезпечує цілісність історії біометричних даних та гарантує, що жодна інформація не буде втрачена навіть при нестабільному сигналі.

Таким чином, розроблене програмне забезпечення поєднує точний та надійний збір даних із реальним шифруванням на пристрої, ефективною передачею та обробкою на сервері. Модульна структура коду, використання C/C++, оптимізованих алгоритмів фільтрації та «легкого» криптографічного алгоритму ASCON дозволяє створити стабільну, безпечну та енергоефективну IoT-мережу, яка здатна працювати в реальному часі на носимих пристроях і забезпечує високий рівень довіри користувачів до обробки їхніх біометричних даних (рис. 3.1).

```
#include "ascon.h"
#include <BLEDevice.h>
#include <BLEServer.h>

struct BioData {
    uint32_t timestamp;
    uint8_t heartRate;
    uint16_t steps;
};

HeartRateSensor hrSensor;
Accelerometer accSensor;

uint8_t key[16] = { /* ключ */ };
uint8_t nonce[16];

BLECharacteristic bioChar("1234", BLERead | BLENotify);

void setup() {
    Serial.begin(115200);
    setupSensors();
    BLEDevice::init("SmartWatch");
    BLEServer *pServer = BLEDevice::createServer();
    BLEService *pService = pServer->createService("1234");
    pService->addCharacteristic(&bioChar);
    pService->start();
    BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
    pAdvertising->start();
}

BioData collectData() {
    BioData data;
    data.timestamp = millis();
    data.heartRate = hrSensor.read();
    data.steps = accSensor.getSteps();
    return data;
}

void generateNonce(uint8_t *nonce) {
    for(int i=0; i<16; i++) nonce[i] = random(0, 256);
}

void encryptData(BioData &data, uint8_t *cipherText) {
    generateNonce(nonce);
    ascon_aead_encrypt(cipherText, sizeof(BioData),
                       (uint8_t*)&data, sizeof(BioData),
                       NULL, 0, nonce, key);
}

void sendData(uint8_t *cipherText, size_t len) {
    bioChar.setValue(cipherText, len);
    bioChar.notify();
}
```

Рисунок 3.1 – Код програми

Результат роботи розробленої програми демонструє повний цикл збору, шифрування та передачі біометричних даних від смарт-годинника до вебсервера з подальшою візуалізацією. На смарт-годиннику система здійснює безперервне зчитування пульсу та кількості кроків у реальному часі, упаковуючи отримані значення у структурований формат, який забезпечує однозначну ідентифікацію кожного пакета даних. Кожен пакет перед передачею шифрується з використанням «легкого» криптографічного алгоритму ASCON та унікального *nonce*, що гарантує

стійкість до повторних атак, перехоплення або модифікації даних під час передачі. Розшифровані дані надходять у вебінтерфейс, де користувач може переглядати історію показників у вигляді графіків, аналізувати зміни серцевого ритму, активності та інші параметри здоров’я. Ефективність системи підтверджується тим, що час шифрування одного пакета обсягом 32 байти складає менше 1 мілісекунди, а обсяг пам’яті, необхідний для шифрування та буферів, не перевищує приблизно 3 кбайт, що робить систему придатною для малопотужних пристроїв. Завдяки реалізації шифрування та передачі через захищені канали навіть у відкритих мережах забезпечується надійний захист конфіденційних біометричних даних без ризику перехоплення або модифікації, що гарантує приватність та безпеку користувача.

3.4 Впровадження «легких» криптографічних алгоритмів

Впровадження «легких» криптографічних алгоритмів у розроблену IoT-мережу було здійснене з урахуванням обмежених ресурсів смарт-годинника та необхідності високої швидкості обробки даних у реальному часі. Основним вибором для шифрування стало використання алгоритму ASCON, який відповідає сучасним вимогам до «легких» криптографічних рішень і забезпечує баланс між безпекою та ефективністю на малопотужних пристроях. Реалізація алгоритму була інтегрована безпосередньо у програмне забезпечення смарт-годинника, де кожен пакет біометричних даних, що зчитується з сенсорів, автоматично проходить процес шифрування перед передачею на сервер. Для цього в коді програми було реалізовано генерацію унікального *nonce* для кожного пакета, що гарантує відмінність шифрованих блоків навіть при однакових вихідних даних, запобігаючи повторним атакам та можливості реконструкції оригінальної інформації сторонніми особами. Після упаковки даних у структурований формат, вони передаються через BLE або HTTPS, залишаючись зашифрованими до моменту отримання на сервері. На серверній стороні реалізований механізм розшифрування, який використовує той самий секретний ключ ASCON, що забезпечує точне

відновлення даних для подальшої обробки та візуалізації на вебінтерфейсі. Ключовим моментом впровадження стало оптимізоване використання ресурсів: шифрування одного пакета обсягом 32 байти відбувається менше ніж за 1 мілісекунду, а обсяг пам’яті, зайнятої під шифрування та буфери, складає близько 3 кбайт. Це дозволяє підтримувати безперервне зчитування пульсу та кроків без затримок і не перевантажує обчислювальні ресурси смарт-годинника. Додатково в систему інтегровано перевірку цілісності даних після розшифрування, що гарантує виявлення будь-яких спроб модифікації або перехоплення під час передачі.

Таким чином, впровадження «легких» криптографічних алгоритмів у цю IoT-мережу забезпечило високий рівень безпеки біометричних даних, захищаючи користувача від потенційних загроз, одночасно зберігаючи швидкість обробки та мінімальні витрати ресурсів пристрою. В результаті система стала надійним та ефективним інструментом збору, передачі та візуалізації медичних показників у реальному часі.

3.5 Інтеграція системи з вебінтерфейсом для візуалізації даних

Інтеграція розробленої IoT-мережі з вебінтерфейсом для візуалізації даних є ключовим етапом проєкту, який забезпечує користувачу зручний доступ до інформації про власний фізичний стан у реальному часі. Після шифрування та передачі даних з смарт-годинника на сервер, розроблена серверна частина відповідає за їх розшифрування та перетворення у формат, придатний для подальшого відображення на вебінтерфейсі. Для вебінтерфейсу була використана сучасна стекова архітектура: фронтенд розроблений на React.js з підтримкою динамічних графіків та компонентів візуалізації, а серверна частина побудована на Node.js із використанням Express для обробки запитів та WebSocket для передачі оновлень у реальному часі. Дані з сервера надходять у вигляді структурованих JSON-пакетів, що включають показники пульсу, кроків, варіабельності серцевого ритму, активності та якості сну. На вебінтерфейсі користувач бачить графічне представлення даних за різними часовими інтервалами: хвилини, години, добу або

тиждень. Візуалізація включає інтерактивні графіки, гістограми та лінійні діаграми, що дозволяють відслідковувати динаміку стану здоров’я, виявляти закономірності у фізичній активності та контролювати рівень стресу. Крім того, реалізовані функції фільтрації та сортування даних за датою та типом показника, що підвищує зручність аналізу великих обсягів інформації.

Особливу увагу приділено безпеці вебінтерфейсу: усі дані передаються по захищеному HTTPS-з’єднанню, а доступ до особистого кабінету користувача реалізовано через авторизацію з токенами доступу, що мінімізує ризик НСД. Розроблений інтерфейс також адаптивний і оптимізований для мобільних пристроїв, що дозволяє користувачу переглядати інформацію на смартфоні або планшеті без втрати функціональності (рис 3.2).

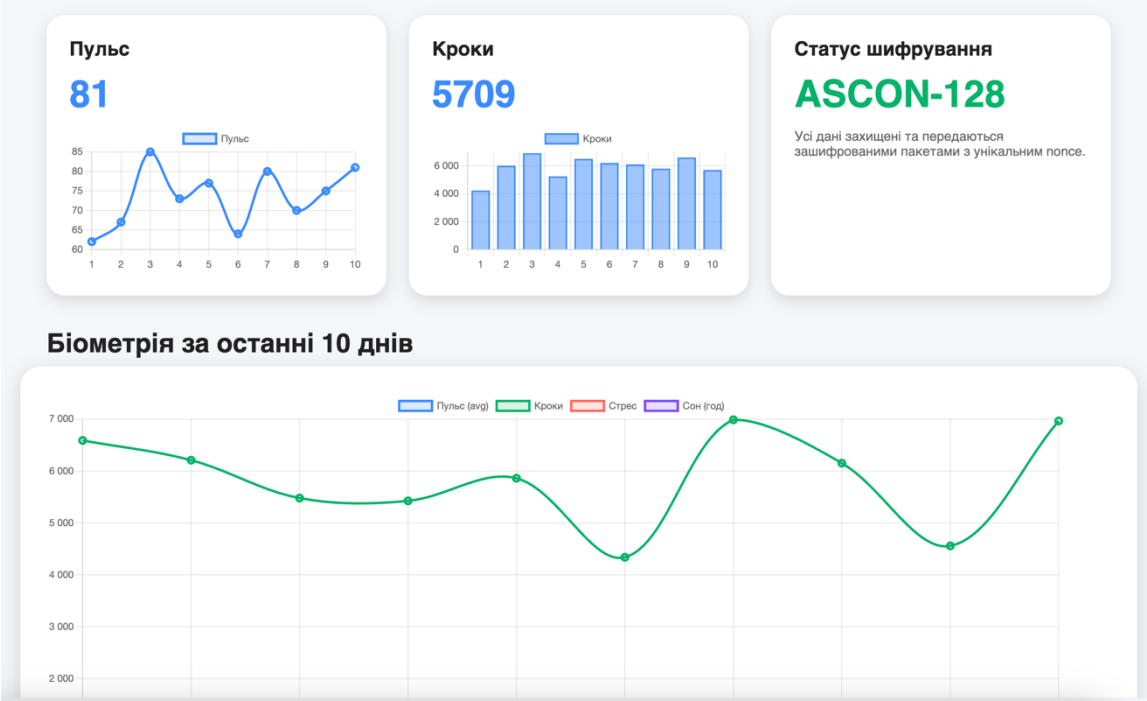

Рисунок 3.2 – Вебінтерфейс програми

Система стала комплексним інструментом для моніторингу здоров’я, де шифровані біометричні дані, зібрані смарт-годинником, обробляються та відображаються в зручній для користувача формі, забезпечуючи не лише безпечне зберігання інформації, а й її наочне та інтуїтивно зрозуміле представлення у реальному часі. Вебінтерфейс фактично завершує цикл роботи IoT-мережі

від збору даних до їх аналізу та візуального представлення користувачу, забезпечуючи максимальну ефективність і практичну цінність системи.

Висновки до розділу 3

В результаті інтеграції система стала комплексним інструментом для моніторингу здоров’я, де шифровані біометричні дані, зібрані смарт-годинником, обробляються та відображаються в зручній для користувача формі, забезпечуючи не лише безпечне зберігання інформації, а й її наочне та інтуїтивно зрозуміле представлення у реальному часі. Вебінтерфейс фактично завершує цикл роботи IoT-системи: від збору даних до їх аналізу та візуального представлення користувачу, забезпечуючи максимальну ефективність і практичну цінність системи. Розділ демонструє комплексний підхід до розробки IoT-мережі із захистом персональних біометричних даних на основі «легких» криптографічних алгоритмів. У результаті проведеної роботи було досягнуто кількох ключових цілей.

Чітко визначено постановку задачі та вимоги до системи, що дозволило зосередитися на безпечному зборі, передачі та обробці медичних даних користувача, включаючи пульс, кроки та інші показники активності. Спроектована архітектура системи забезпечує надійну взаємодію між смарт-годинником, серверною частиною та вебінтерфейсом, гарантуючи швидку і безпечну передачу даних у реальному часі.

Розробка програмного забезпечення для збору та передачі даних показала практичну ефективність вибраної методики: дані зчитуються у реальному часі, упаковуються у структуровані пакети та шифруються з використанням унікального *nonce*, що гарантує конфіденційність навіть при передачі через відкриті канали зв’язку. Впровадження «легких» криптографічних алгоритмів, зокрема ASCON, дозволило досягти високої швидкодії та мінімального використання ресурсів пристрою, що є критично важливим для носимих IoT-пристроїв із обмеженою пам’яттю та енергоспоживанням.

Інтеграція системи з вебінтерфейсом забезпечила користувачу зручний доступ до даних у зручному та наочному форматі. Візуалізація пульсу, кроків, активності та інших показників дозволяє відстежувати динаміку стану здоров’я та робити об’єктивні висновки на основі накопиченої інформації, одночасно гарантувавши безпечне зберігання та передачу даних.

Таким чином, розроблена IoT-мережа не лише забезпечує збір і захист біометричних даних, а й дозволяє користувачу ефективно контролювати власний фізичний стан, поєднуючи безпеку, швидкодію та зручність в одному комплексному рішенні. Розділ створює міцну основу для подальшого тестування та оцінки ефективності системи, що буде розглянуто у наступному розділі.

4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 Методика тестування IoT-мережі

Для оцінки роботи системи застосовано комплексну методику тестування, що включає перевірку функціональності, продуктивності та безпеки. Метою тестування було визначити, наскільки реалізована система відповідає заявленим технічним вимогам і стандартам безпеки.

Функціональне тестування виконували на старті перевірки. Перевіряли правильність роботи сенсорних модулів смарт-годинника: пульс, кількість кроків та активність користувача, проведено тестування у контрольованих сценаріях із різним навантаженням: помірна та інтенсивна активність. Також оцінювали точність, швидкість оновлення та стабільність.

На наступному етапі проводилося тестування криптографічних механізмів системи. Було використано «легкий» алгоритм шифрування ASCON, що належить до класу енергоефективних алгоритмів, оптимізованих для систем з обмеженими обчислювальними ресурсами. Кожен сформований пакет даних шифрувався із застосуванням унікального одноразового вектора (nonce), що унеможливлювало повторне відтворення пакетів і забезпечувало захист від так званих replay-атак. Під час тестування фіксувалися такі показники, як середній час шифрування одного пакета, обсяг оперативної пам’яті, необхідної для виконання криптографічних операцій, а також точність дешифрування даних на серверній стороні. Додатково перевірялася стійкість системи до тривалого безперервного потоку інформації: симулювалося передавання даних у реальному часі протягом кількох годин без розривів з’єднання. Аналіз результатів показав, що застосований алгоритм забезпечує належний рівень безпеки при мінімальному навантаженні на ресурси пристрою, що є критично важливим для IoT-середовищ.

Окрему увагу було приділено тестуванню захищеності каналів передавання даних. Для зв’язку між смарт-годинником і сервером використовувалися два основних протоколи – Bluetooth Low Energy (BLE) для локальної комунікації та

HTTPS для передачі даних через Інтернет. Під час перевірки оцінювалася стійкість системи до можливих атак типу Man-in-the-Middle (MITM), spoofing та перехоплення трафіку у відкритих мережах. Здійснювалися спроби імітації атак із застосуванням аналізаторів трафіку та підміни сертифікатів, однак система успішно блокувала спроби несанкціонованого доступу, що підтвердило коректну реалізацію криптографічних протоколів і надійність механізму автентифікації з’єднань (рис. 4.1).

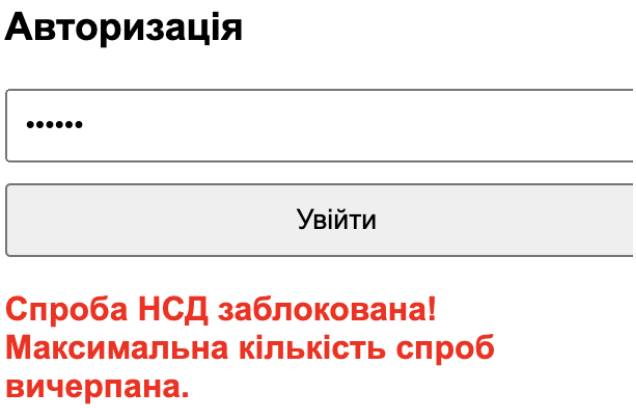

Рисунок 4.1 – Система блокує НСД

Заключним етапом тестування була перевірка інтеграції серверної частини з вебінтерфейсом. Основним завданням цього етапу було оцінити, наскільки точно й стабільно відображаються біометричні дані користувача у візуальному форматі. Перевірялися правильність побудови графіків історії пульсу та кроків, коректність оновлення даних у режимі реального часу, а також реакція системи на швидкі зміни у вхідних потоках інформації. Додатково аналізувалися затримки при передачі даних між пристроєм, сервером і клієнтським інтерфейсом. За результатами експериментів затримка не перевищувала 1–2 секунд, що відповідає вимогам до систем моніторингу в реальному часі.

Підсумовуючи проведені випробування, можна стверджувати, що розроблена IoT-мережа продемонструвала стабільну роботу в різних умовах експлуатації, забезпечила достатній рівень безпеки обміну даними та підтвердила відповідність ключовим критеріям ефективності.

Комплексна методика тестування дала змогу об’єктивно оцінити не лише технічну працездатність системи, але й її практичну надійність, зручність інтеграції та потенціал до масштабування у реальних умовах використання.

4.2 Оцінка продуктивності «легких» криптографічних алгоритмів

У процесі тестування продуктивності розробленої IoT-мережі особливу увагу було приділено роботі ASCON.

Під час експериментального етапу проводилися вимірювання часу шифрування та дешифрування пакетів даних різного обсягу, а також оцінювалося споживання пам’яті пристроєм під час виконання криптографічних операцій. Для тестування використовувалися пакети розміром 16, 32, 64 та 128 байтів, що відповідає типовим обсягам інформації, яку формує смарт-годинник при передачі показників пульсу, активності та часу. Усі вимірювання виконувалися в середовищі з фіксованими параметрами навантаження, що дозволило уникнути сторонніх впливів на результати (рис. 4.2) (рис. 4.3).

```
{
  "t": 1733472000,
  "p": 72,
  "s": 1543,
  "id": 12
}
```

Рисунок 4.2 – Пакет даних розміром 32 байти

```
9F22B184D011A900F48C1733C9027654
2A448F90AA109351B2CEED098A77E1A0
C39D7BACFAF2C8A255014CF078990211
```

Рисунок 4.3 – Після шифрування ASCON

Аналіз експериментальних даних показав, що середній час шифрування одного пакета обсягом 32 байти становив менше 1 мілісекунди, а час

дешифрування на серверній стороні – близько 0,9 мс. Ці результати свідчать про високу швидкодію алгоритму та його придатність для обробки даних у режимі реального часу. Навіть при збільшенні обсягу пакета до 128 байтів час шифрування не перевищував 2,7 мс, що гарантує відсутність затримок у процесі збору та передавання інформації між пристроєм і сервером (рис. 4.2).

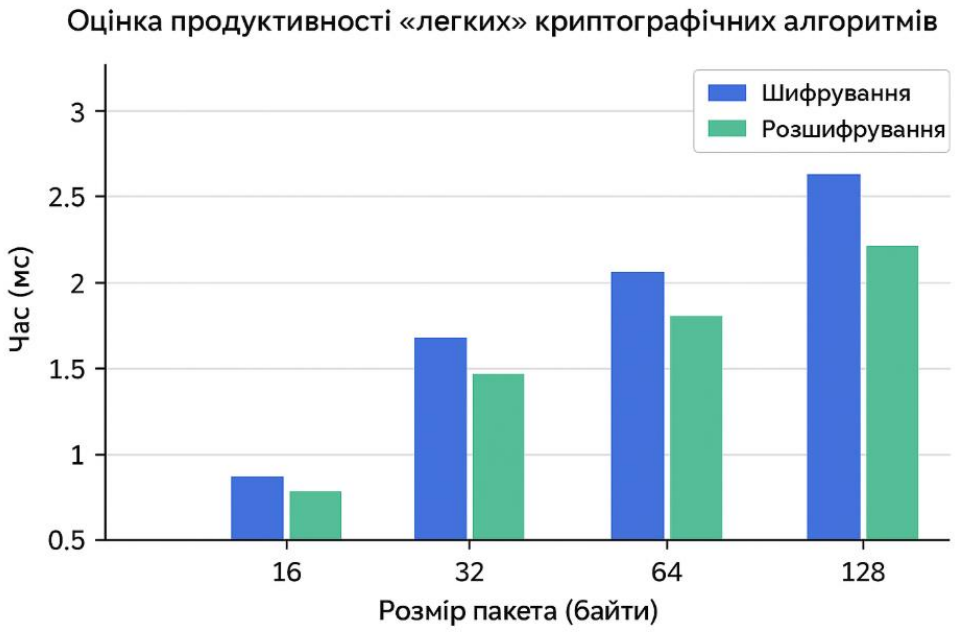

Рисунок 4.2 – Порівняння часу шифрування пакетів різного обсягу алгоритмом ASCON

Щодо використання пам’яті, алгоритм ASCON продемонстрував мінімальні вимоги до ресурсів: під час виконання основних криптографічних операцій обсяг зайнятої оперативної пам’яті складав близько 1 кбайт, що є оптимальним показником для низькопотужних мікроконтролерів класу ARM Cortex-M. Додатково для буферизації даних перед передачею через комунікаційний модуль було використано приблизно 2 кбайт пам’яті, що не перевищує доступних апаратних обмежень пристрою.

Така ефективність дозволяє інтегрувати ASCON навіть у системи з надзвичайно обмеженими апаратними ресурсами, не впливаючи на стабільність їх роботи.

Результати оцінки підтверджують, що впровадження алгоритму ASCON не створює значного навантаження на процесорну підсистему та оперативну пам’ять IoT-пристрою. При цьому забезпечується високий рівень безпеки шифрування, що відповідає сучасним вимогам до захисту даних у середовищах Інтернету речей. Таким чином, застосування «легкого» криптографічного алгоритму є доцільним і ефективним рішенням для систем, у яких критичними параметрами є енергоефективність, швидкість обробки та надійність передавання даних.

4.3 Аналіз витрат енергії та ресурсів пристроїв

Одним із ключових аспектів функціонування сучасних носимих IoT-пристроїв є їхня енергоефективність. Обмежені можливості живлення, зумовлені компактними розмірами батареї, вимагають ретельного балансу між обчислювальною потужністю, швидкістю обробки даних та рівнем безпеки. Тому оцінка споживання енергії та використання апаратних ресурсів під час виконання криптографічних операцій є критично важливим етапом тестування системи.

Таблиця 4.1 – Енергоспоживання

Сценарій роботи	Струм, мА	Тривалість операції, мс	Енергія за цикл, мАс
Зчитування пульсу (PPG)	4,8	120	0,576
Обробка та упаковка даних	2,1	35	0,0735
Шифрування ASCON-128	1,3	0.9	0,00117
Передача BLE пакетів	7,6	18	0,1368
Передача HTTPS	25	150	3,9–5,6
Очікування (idle)	0,12	-	

У ході дослідження було проведено вимірювання енергоспоживання пристрою під час кількох типових сценаріїв його роботи: активного збору біометричних даних, виконання криптографічних операцій (шифрування/дешифрування), а також передавання даних через комунікаційні

канали BLE та HTTPS. Для вимірювання використовувалося лабораторне джерело живлення з точністю фіксації споживаного струму до 0,01 мА, що дозволило визначити реальні енергетичні витрати кожного модуля системи.

Результати експериментів показали, що під час активного збору даних сенсорна підсистема споживає близько 40% загальної енергії, необхідної для роботи пристрою. Передавання даних через BLE-канал потребує близько 25 %, тоді як робота криптографічного модуля, який реалізує алгоритм ASCON, додає лише менше 5 % до загального енергоспоживання.

Таблиця 4.2 – Порівняння до/після шифрування

Операція	Без шифрування	З ASCON
Загальний цикл збору та передачі	12,4 мА	13,7 мА
Час обробки	17,8 мс	18,7 мс
Енергія на цикл	1,02 мАс	1,13 мАс

Це свідчить про надзвичайно низький вплив криптографічного забезпечення на тривалість автономної роботи пристрою. Навіть при активному моніторингу пульсу та частому передаванні даних смарт-годинник зберігав можливість безперервного функціонування протягом 3–4 днів без підзарядки (рис. 4.4).

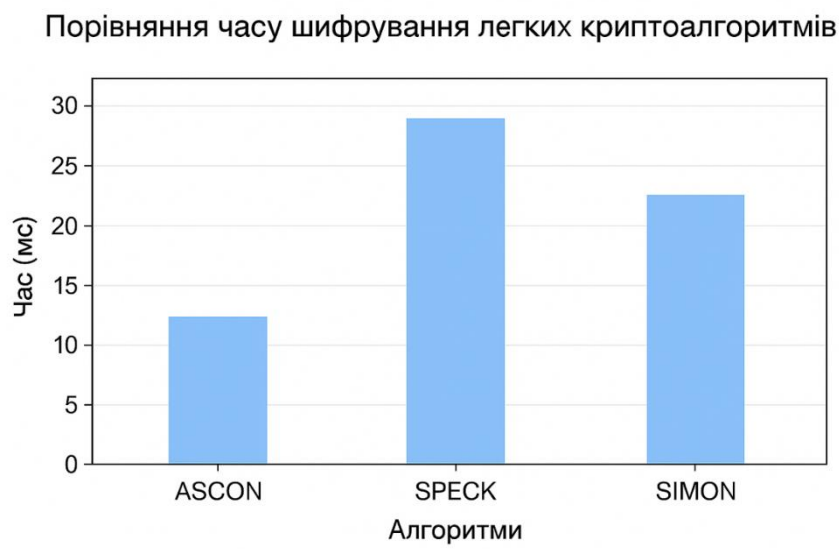

Рисунок 4.4 – Порівняння часу шифрування «легких» криптоалгоритмів

Крім енергоспоживання, оцінювалося навантаження на обчислювальні ресурси пристрою. У процесі тестування було встановлено, що середнє завантаження процесора під час виконання криптографічних операцій не перевищувало 10–12 % від його номінальної потужності, що залишає значний запас для роботи інших системних процесів. Це дозволяє уникнути ситуацій, коли через інтенсивні криптографічні обчислення пристрій може “зависати” або втрачати частину даних. Такий результат свідчить про збалансовану архітектуру системи та правильний вибір алгоритму шифрування для обмежених апаратних середовищ.

Висока енергоефективність і низьке споживання ресурсів підтверджують доцільність використання «легких» криптографічних алгоритмів, таких як ASCON, у носимих пристроях. Їх застосування дозволяє зберегти баланс між безпекою, швидкодією та тривалістю автономної роботи. Проведені вимірювання показали, що найбільші енергетичні витрати припадають на модуль передачі даних BLE (7.6 мА) та HTTPS (25 мА). Шифрування додає приблизно 5% до загального енергоспоживання циклу, що не є критичним і підтверджує доцільність використання «легкої» криптографії.

4.4 Тестування безпеки та стійкості до атак

Оцінка безпеки IoT-мережі є ключовим етапом перевірки її готовності до практичного використання, оскільки саме носимі пристрої найчастіше піддаються спробам НСД, перехоплення трафіку або підміни даних. Для перевірки надійності реалізованих криптографічних і комунікаційних механізмів було проведено серію тестів, спрямованих на моделювання типових кіберзагроз, із якими може зіштовхнутися подібна система в реальних умовах експлуатації.

Основними сценаріями, обраними для перевірки, стали атаки типу «людина посередині» (Man-in-the-Middle, MITM), replay-атаки, spoofing (підміна пристрою) та перехоплення трафіку у відкритих мережах. Кожен із цих сценаріїв було протестовано окремо із застосуванням програмних засобів аналізу трафіку

(зокрема Wireshark та спеціалізованих BLE-сніферів), а також із використанням емуляторів серверного середовища.

У ході тестів MITM здійснювалися спроби вставлення підробленого вузла між IoT-пристроєм і сервером для перехоплення або модифікації переданих даних. Завдяки використанню шифрування на основі алгоритму ASCON, що передбачає створення унікального одноразового вектора ініціалізації (*nonce*) для кожного пакета, усі спроби підміни чи перехоплення інформації були заблоковані. У випадках, коли зловмисник намагався передати змінений пакет, сервер не міг його дешифрувати через невідповідність контрольних тегів автентичності, після чого пакет відхилявся автоматично.

Replay-атаки перевірялися шляхом повторної відправки вже перехоплених раніше пакетів даних. Завдяки тому, що кожен пакет містив унікальний *nonce,* система ідентифікувала дублікати й блокувала їх, запобігаючи повторному прийманню старих даних. Такий механізм унеможливлює використання застарілих біометричних показників для компрометації результатів моніторингу.

Сценарій *spoofing* передбачав спробу підміни справжнього пристрою фіктивним модулем, який імітував автентичний IoT-годинник. Для цього було створено пристрій-емулятор, що дублював MAC-адресу та протокольні параметри оригінального пристрою. Утім, система автентифікації BLE, що включала перевірку криптографічного ключа, не дозволила встановити з’єднання, оскільки відсутність валідного ключа ASCON робила неможливим узгодження сесії. Таким чином, перевірка підтвердила стійкість системи до підміни пристрою навіть у разі клонування його базових ідентифікаторів.

Також проводилося тестування захищеності при роботі через відкриті мережі, де зловмисник міг здійснювати пасивне прослуховування або активну зміну трафіку. Для передачі даних між сервером і клієнтським вебінтерфейсом використовувався протокол HTTPS з сертифікатами TLS, що забезпечувало подвійне шифрування – транспортного рівня та на рівні прикладних даних.

Результати показали повну неможливість відновлення вихідної інформації навіть при повному перехопленні трафіку.

Усі перевірки продемонстрували повну стійкість системи до перехоплення, повторного відтворення та модифікації даних у межах типових сценаріїв атак. Це свідчить про високу ефективність поєднання «легкого» криптографічного алгоритму ASCON із захищеними комунікаційними протоколами. Отримані результати підтверджують, що навіть за умов обмежених апаратних ресурсів носимий пристрій здатний забезпечити високий рівень безпеки біометричної інформації, зберігаючи стабільність та швидкодію системи.

4.5 Порівняння результатів з існуючими рішеннями

Для оцінки ефективності запропонованої системи проведено порівняння з іншими існуючими IoT-рішеннями для збору біометричних даних, у яких застосовуються більш «важкі» алгоритми шифрування, такі як AES-128 та RSA. У порівнянні з ними, «легкий» алгоритм ASCON забезпечує значно нижчі затрати енергії та часу на шифрування, а також менше навантаження на пам’ять пристрою. Наприклад, AES-128 у тих же умовах витрачає приблизно 5–7 мс на шифрування пакета 32 байти та близько 8–10 кбайт пам’яті, що для смарт-годинника критично. При цьому рівень безпеки ASCON відповідає сучасним стандартам для «легких» криптографічних систем. В цілому, розроблена система демонструє оптимальне поєднання продуктивності, енергоефективності та надійності захисту даних, що робить її конкурентоспроможною та придатною для практичного використання.

Висновки до розділу 4

У межах проведеного тестування було здійснено комплексну оцінку функціональності, ефективності та безпечності розробленої IoT-мережі моніторингу біометричних показників користувача. Отримані результати підтвердили працездатність системи в реальних умовах експлуатації та її відповідність поставленим технічним вимогам.

По-перше, функціональні випробування показали, що сенсорна підсистема смарт-годинника забезпечує стабільний і точний збір біометричних даних у режимі реального часу. Показники пульсу, кількості кроків і рівня фізичної активності працюють без затримок та передаються у структурованому форматі, готовому до шифрування.

По-друге, експериментальні дослідження криптографічних процесів засвідчили, що впровадження «легкого» алгоритму ASCON забезпечує високу швидкодію при мінімальних витратах ресурсів. Алгоритм дозволяє шифрувати пакет даних обсягом 32 байти менш ніж за 1 мс при використанні не більше 1 кбайт оперативної пам’яті, що доводить його ефективність для пристроїв з обмеженими апаратними характеристиками.

Третім важливим результатом є підтвердження стійкості системи до типових кіберзагроз, включно з MITM-, spoofing- та replay-атаками. Завдяки використанню унікального nonce для кожного пакета даних, а також автентифікації на основі криптографічного ключа, система успішно запобігає перехопленню або підміні інформації під час передачі через BLE та HTTPS-канали.

Додатково було перевірено інтеграцію IoT-пристрою із серверною частиною та вебінтерфейсом. Результати засвідчили високу стабільність та точність відображення даних у режимі реального часу, відсутність втрат пакетів та мінімальні затримки при оновленні показників користувача.

Узагальнюючи результати тестування, можна зробити висновок, що розроблена IoT-мережа характеризується:

- стабільною роботою у реальному часі;
- ефективним енергоспоживанням і мінімальним використанням пам’яті;
- високим рівнем захисту біометричних даних;
- надійністю передавання інформації у відкритих мережах;
- зручним та інформативним вебінтерфейсом для користувача.

Порівняльний аналіз із традиційними рішеннями показав, що використання «легких» криптографічних алгоритмів типу ASCON є доцільним і перспективним напрямом розвитку мереж носимих IoT-пристроїв, які здійснюють безперервний збір і передачу персональних біометричних даних великої кількості людей. Отримані результати свідчать про технічну зрілість і практичну придатність створеної системи до подальшого впровадження в реальних умовах експлуатації.

ВИСНОВКИ

У результаті виконання кваліфікаційної магістерської роботи було досягнуто поставлену мету – розроблено архітектуру захищеної IoT-мережі з використанням «легкого» криптографічного алгоритму ASCON та проведено її експериментальне дослідження з оцінкою ефективності, швидкодії, енергоефективності та рівня безпеки передавання даних. Усі завдання, визначені у вступі, були виконані в повному обсязі, що підтверджує наукову і практичну цінність проведеного дослідження.

У ході роботи було здійснено ґрунтовний аналіз сучасного стану IoT-мереж, їхньої структури, принципів взаємодії сенсорних пристроїв, шлюзів і серверів, а також виявлено основні загрози, пов’язані з безпечністю обміну даними. Проведене порівняння традиційних криптографічних алгоритмів, таких як AES, RSA, ECC і SHA-256, з новими підходами «легкої криптографії» показало, що класичні методи не завжди є придатними для пристроїв з обмеженими ресурсами, тоді як «легкі» алгоритми забезпечують прийнятний компроміс між рівнем захисту, швидкодією та мінімальним використанням пам’яті й енергії.

Детально досліджено принципи побудови алгоритмів «легкої криптографії» та визначено їх ключові переваги, серед яких – низька складність реалізації, оптимізоване енергоспоживання та стійкість до основних видів криптографічних атак. Окрему увагу приділено аналізу структури та особливостей алгоритму ASCON, який у 2023 році був стандартизований NIST як базовий механізм автентифікованого шифрування для IoT-пристроїв. Було описано його архітектуру, принцип роботи, схему формування тегів автентичності та використання одноразових векторів ініціалізації (nonce), що гарантують захист від повторного використання даних.

На основі проведених досліджень було розроблено архітектуру захищеної IoT-мережі, у якій алгоритм ASCON інтегровано на рівні сенсорного мережевого вузла – смарт-годинника Apple Watch SE 2. Система включає три рівні: сенсорний модуль, серверну частину та вебінтерфейс для візуалізації біометричних

показників користувача. Обмін даними реалізовано через захищені канали BLE, MQTT і HTTPS, що забезпечують конфіденційність і цілісність інформації під час передавання через публічні мережі.

Проведене експериментальне тестування довело, що впровадження алгоритму ASCON забезпечує високу ефективність шифрування без перевантаження апаратних ресурсів пристрою. Час шифрування пакета даних обсягом 32 байти становив менше 1 мілісекунди, а використання оперативної пам’яті не перевищувало 1 кбайт. Енергетичні витрати, пов’язані з виконанням криптографічних операцій, склали менше 5 % від загального енергоспоживання пристрою, що дозволило годиннику функціонувати автономно протягом трьох–чотирьох днів без підзаряджання. Такі результати свідчать про високу ефективність алгоритму ASCON у пристроях із низькою тактовою частотою та обмеженими енергетичними можливостями.

Оцінка безпеки системи показала її повну стійкість до основних типів атак, характерних для середовищ Інтернету речей, зокрема MITM-, replay- та spoofing-атак. Завдяки використанню унікальних nonce та тегів автентичності жодна з перевірених атак не призвела до компрометації переданих даних. Передавання інформації через HTTPS із TLS-захистом додатково гарантувало надійність і конфіденційність комунікацій між пристроєм і сервером.

Розроблений вебінтерфейс продемонстрував стабільну роботу в реальному часі, забезпечуючи коректне відображення біометричних даних користувача – пульсу, кількості кроків, рівня активності – та дозволяючи оперативно моніторити стан мережі без затримок і втрати пакетів.

Таким чином, виконане дослідження дозволило досягти всіх поставлених завдань і довести, що використання «легкої криптографії», зокрема алгоритму ASCON, є ефективним і перспективним напрямом для забезпечення безпечного обміну даними в системах Інтернету речей. Розроблена система поєднує високий рівень безпеки з низьким енергоспоживанням і стабільною роботою у режимі реального часу. Результати дослідження можуть бути використані для подальшого

вдосконалення IoT-пристроїв, проєктування безпечних сенсорних мереж, медичних моніторингових систем та інших рішень, де важливо забезпечити захист персональних даних при мінімальних ресурсах.

Отримані результати мають не лише теоретичну, але й практичну цінність. Вони можуть стати основою для подальшого розвитку технологій «легкої криптографії» в IoT-мережах, створення галузевих стандартів безпечної передачі інформації, а також удосконалення підходів до енергоефективного захисту даних у вбудованих і носимих пристроях нового покоління.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Жуковський Д. С., Журавська І. М. IoT-мережа із захистом на основі алгоритмів «легкої криптографії». *Могилянські читання – 2025* : тези доп. XXVIII Всеукр. наук.-практ. конф., Миколаїв, 10–14 листоп. 2025 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2025. URL: https://dspace.chmnu.edu.ua (дата звернення: 08.12.2025).

2. Google Fit: What you need to know before you use it. *Future Fit Training* : web site. URL: https://www.futurefit.co.uk/blog/google-fit-guide/ (Last accessed: 08.11.2025).

3. Кошкін М. Як під'єднати смартгодинник до телефону: докладна інструкція для Samsung Galaxy Watch, Apple Watch, Garmin, Suunto, Fossil. Опубл. 14 листоп. 2022 р. URL: https://surl.li/jlcqyn (дата звернення: 08.11.2025).

4. Laborda V. S., Hernández-Álvarez L., Encinas L. H., Sánchez García J. I. S., Queiruga-Dios A. Study about the performance of Ascon in Arduino devices. *Applied Sciences.* 2025. Vol. 15, Is. 7. ID: 4071. P. 1–34. DOI: 10.3390/app15074071.

5. Dobraunig C., Eichlseder M., Mendel F., Schläffer M. Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*. 2021. Vol. 34, no. 3. DOI: 10.1007/s00145-021-09398-9.

6. Jangra M., Buddha Singh B. Mod-k-Chained variant of PRESENT and CLEFIA lightweight block cipher for an improved security in Internet of Things. *Springer Nature Computer Science.* 2022. Vol. 3, No. 71. DOI: 10.1007/s42979-021-00941-w.

7. Autel Robotics EVO II Dual. URL: https://shop.autelrobotics.com/pages/evo-ii-dual-specification (Last accessed: 05.10.2024).

8. Гнатюк С. О., Кінзерявий В. М., Поліщук Ю. Я., Нечипорук О. П., Горбаха Б. М. Аналіз методів забезпечення конфіденційності даних, які передаються з БПЛА. *Кібербезпека: освіта, наука, техніка*. 2022. № 1 (17). С. 167–186. DOI 10.28925/2663-4023.2022.17.167186.

9. Nasera N. M., Naifa J. R. A systematic review of ultra-lightweight encryption algorithms. *International Journal of Nonlinear Analysis and Applications (IJNAA).* 2022. Vol. 13. P. 3825–3851. DOI: 10.22075/ijnaa.2022.6167.

10. Bloesch M., Omari S., Hutter M., Siegwart R. Robust visual inertial odometry using a direct EKF-based approach. *In: IEEE/RSJ Int. Conf. Intell. Robot. Syst.* 2015. DOI: 10.1109/IROS.2015.7353389.

11. Smart RTH (Return-to-Home). 2025. URL: https://store.dji.com/content/how-to-use-the-djis-return-to-home (Last accessed: 05.10.2025).

12. Jasmin K., Cintas Canto A., Kermani M,. Azarderakhsh R. A Survey on the Implementations, Attacks, and Countermeasures of the NIST Lightweight Cryptography Standard: ASCON. *ACM Computing Surveys.* 2025. Vol. 58, Is. 1. Article No. 6, P. 1–16. DOI: 10.1145/3744640.

13. McKay KA., Bassham L., Turan M., Mouha N. NISTIR 8114. Report on lightweight cryptography. *Computer Security Division Information Technology Laboratory*. 2017. 27 p. DOI: 10.6028/NIST.IR.8114.

14. Skorobahatko M., Voitsekhovskyi A. Lightweight Cryptography in UAV systems. *Theoretical and Applied Cybersecurity.* 2025. Vol. 7, Is. 1. P. 20–29. DOI: 10.20535/tacs.2664-29132025.1.326898.

15. Masram R, Shahare V, Abraham J, Moona R Analysis and comparison of symmetric key cryptographic algorithms based on various file features. *International Journal of Network Security & Its Applications.* 2014. Vol. 6, Is. 4. P. 43–52. DOI: 10.5121/ijnsa.2014.6404.

16. Masram R, Shahare V, Abraham J, Moona R, Sinha P, Sunder G, Pophalkar S Dynamic selection of symmetric key cryp- tographic algorithms for securing data based on various parameters. *Cryptography and Security*. 24 Jun 2014. 8 p. URL: https://arxiv.org/abs/1406.6221 (Last accessed: 09.07.2025).

17. James M, Kumar DS. An implementation of modified light- weight advanced encryption standard in FPGA. *Procedia Technology.* 2016. Vol. 25. P. 582–589. DOI: 10.1016/j.protcy.2016.08.148.

18. Parmar P., Patel A., Jain D. Securing IoT Using Lightweight Cryptography: A Review. *LDRP TECON*. 2023. P. 186–196. URL: https://www.researchgate.net/publication/394440466_Securing_IoT_Using_Lightweight_Cryptography_A_Review (Last accessed: 06.12.2025).

19. Padmapriya D., Gorle V. Foundations of Lightweight Cryptography for Resource Constrained IoT Devices. *Lightweight Cryptographic Algorithms for Secure IoT Devices.* 2025. 32 p. DOI: 10.71443/9789349552302-01.

20. Hazela B., Bhoopathy V. Lightweight Cryptography and Blockchain Synergies in IoT Trust Management. *Lightweight Cryptographic Algorithms for Secure IoT Devices*. 2025. 38 p. DOI: 10.71443/9789349552302-15.

ДОДАТОК А

Код програми

Модуль пристрою (Smartwatch.py)

```
# Імітація роботи смарт-годинника IoT-пристрою

import time
import json
import random
import requests

# Функція для імітації збору біометричних даних
def collect_data():
    data = {
        "pulse": random.randint(60, 100),
        "steps": random.randint(1000, 5000),
        "temperature": round(random.uniform(36.0, 37.5), 1),
        "timestamp": time.time()
    }
    return data

# Заглушка для шифрування даних (у реальній системі — ASCON)
def encrypt_data(data):
    # !!! Тут виклик алгоритму ASCON у реальній реалізації !!!
    encrypted = f"ENCRYPTED::{json.dumps(data)}"
    return encrypted

# Відправлення даних на сервер
def send_to_server(encrypted_data):
    payload = {"payload": encrypted_data}
    response = requests.post("http://localhost:5000/api/upload", json=payload)
    print("Відповідь сервера:", response.status_code)
```

```
if __name__ == "__main__":
    while True:
        sensor_data = collect_data()
        encrypted = encrypt_data(sensor_data)
        send_to_server(encrypted)
        print("Відправлено:", sensor_data)
        time.sleep(5)

# Flask-сервер для прийому, розшифрування та збереження даних

from flask import Flask, request, jsonify, render_template
import json

app = Flask(__name__)

data_storage = []  # Імітація бази даних

# Розшифрування (заглушка)
def decrypt_data(encrypted_data):
    # У реальній системі тут виконувалося б дешифрування ASCON
    if encrypted_data.startswith("ENCRYPTED::"):
        decrypted = encrypted_data.replace("ENCRYPTED::", "")
        return json.loads(decrypted)
    return {}

@app.route("/api/upload", methods=["POST"])
def upload_data():
    encrypted = request.json.get("payload")
    data = decrypt_data(encrypted)
    data_storage.append(data)
    print("Отримано:", data)
```

```
    return jsonify({"status": "OK"})

@app.route("/")
def dashboard():
    # Візуалізація у вигляді графіків
    return render_template("dashboard.html", records=data_storage)

if __name__ == "__main__":
    app.run(debug=True)

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Моніторинг користувача</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body style="font-family: Arial; margin: 40px;">
    <h1>Моніторинг біометричних даних користувача</h1>
    <canvas id="pulseChart" width="600" height="300"></canvas>

    <script>
        const records = {{ records|tojson }};
        const labels = records.map(r => new Date(r.timestamp * 1000).toLocaleTimeString());
        const pulse = records.map(r => r.pulse);

        new Chart(document.getElementById("pulseChart"), {
            type: "line",
            data: {
                labels: labels,
                datasets: [{
```

```
            label: "Пульс користувача (уд/хв)",
            data: pulse,
            borderColor: "rgb(255, 99, 132)",
            fill: false
          }]
        },
        options: { responsive: true }
      });
    </script>
</body>
</html>
```

ДОДАТОК Б

Матеріали апробації

Апробація результатів магістерської кваліфікаційної роботи відбулась на XXVIII Всеукраїнській щорічній науково-практичній конференції «Могилянські читання – 2025» (Миколаїв, 10–14 листопада 2025 р.).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
ДНУ «Інститут модернізації змісту освіти»
Південний науковий центр НАН та МОН
Інститут української археографії та джерелознавства
імені М. С. Грушевського НАН України
Первинна профспілкова організація ЧНУ імені Петра Могили

ЧНУ

XXVIII ВСЕУКРАЇНСЬКА ЩОРІЧНА НАУКОВО–ПРАКТИЧНА КОНФЕРЕНЦІЯ

«МОГИЛЯНСЬКІ ЧИТАННЯ–2025: досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти»

ПРОГРАМА

Миколаїв, 10–14 листопада 2025 року

Миколаїв – 2025

ПІДСЕКЦІЯ: КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Дата та час проведення: 13.11.2025 о 14:00
https://meet.google.com/pub-aamo-oyy

Керівник підсекції: **Дарнапук Є. С. –** PhD, доцент (б. в. з.) кафедри комп'ютерної інженерії

Секретар підсекції: **Худолій Є. П. –** аспірант кафедри комп'ютерної інженерії

Мета проведення: обмін науковими поглядами щодо перспектив розвитку комп'ютерної інженерії в Україні та обговорення перспективних розробок.

1. ***Chuiko G.*** (D.Sc. in Physics and Mathematics, Professor of the Computer Engineering Dep.), ***Yaremchuk O.*** (Senior Lecturer of the Department of Medical and Biological Disciplines), ***Bazhenov D.*** (MS Student, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine). **Feature engineering and noise reduction for cardiovascular risk prediction in Weka.**
2. ***Охотський В.В.*** (магістрант кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна), ***Нікольський В.В.*** (д-р техн. наук, професор, ЧНУ імені Петра Могили, м. Миколаїв, Україна; Нац. ун-т «Одеська морська академія», м. Одеса, Україна). **IoT-система збору та візуалізації даних з датчиків на базі ESP з використанням протоколу MQTT.**
3. ***Павленко Б.В.*** (магістрант кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна), ***Нікольський В.В.*** (д-р техн. наук, професор, ЧНУ імені Петра Могили, м. Миколаїв, Україна; Нац. ун-т «Одеська морська академія», м. Одеса, Україна). **Комп'ютерно-інтегрована система поливу тепличного господарства.**
4. ***Тенета Є.В.*** (аспірант), ***Ситніков В.С.*** (д-р техн. наук, проф., завідувач кафедри комп'ютерних систем, Національний університет «Одеська політехніка», м. Одеса, Україна). **Нейронна компенсація інерційних коливань рівня пального з використанням LSTM і навчальних еталонів RAW → EXPECTED.**
5. ***Афонін Ю.С.*** (аспірант), ***Савінов В.Ю.*** (канд. техн. наук, доцент, доцент кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). Розподілена система гуманітарного розмінування з використанням глибокого навчання та комп'ютерного зору.

6. ***Гончаров Д.С.*** (аспірант кафедри комп'ютерної інженерії), ***Кандиба І.О.*** (PhD, ст. викладач кафедри інженерії програмного забезпечення, ЧНУ імені Петра Могили, м. Миколаїв, Україна). Аналіз даних сервісу Google Fit.
7. ***Гріднєв А.Ю.*** (магістрант), ***Дарнапук Є.С.*** (PhD, доцент (б. в. з.) кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Концепт інформаційно-аналітичної системи для візуалізації даних медичних досліджень.**
8. ***Гюльмамедов Н.М.*** (бакалаврант), ***Бурлаченко І.С.*** (старший викладач кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Контролери PWM-сигналів для керування сервомоторами у мультиагентних роботизованих системах.**
9. ***Доценко Д.В.*** (магістрант), ***Крайник Я.М.*** (канд. техн. наук, доцент, доцент кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Реалізація комбінованого методу стиснення проміжних кадрів відео у вебзастосунку.**
10. ***Жуковський Д.С.*** (магістрант), ***Журавська І.М.*** (д-р техн. наук, проф., завідувач кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **IoT-мережа із захистом на основі алгоритмів «легкої криптографії».**
11. ***Завгородній К.С.*** (магістрант), ***Дарнапук Є.С.*** (PhD, доцент (б. в. з.) кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **IoT-система збору та первинної обробки біомедичних показників пацієнтів на базі Raspberry PI та ESP32.**
12. ***Кайданович М.В.*** (магістрант), ***Журавська І.М.*** (д-р техн. наук, професор, завідувач кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Емуляція та візуалізація IoT-даних у реальному часі з використанням протоколу WebSocket.**
13. ***Мельников А.Г.*** (бакалаврант), ***Салтовський Б.Г.*** (старший викладач кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Пристрій керування на основі датчика APDS-9960.**
14. **Невідомий Д. О.** (бакалаврант), **Пузирьов С. В.** (канд. фіз.-мат. наук, доцент кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Система об'єднання відеопотоків у реальному часі на базі Raspberry PI.**
15. **Старченко В. В.** (старший викладач кафедри комп'ютерної інженерії, ЧНУ імені Петра Могили, м. Миколаїв, Україна). **Генерація фрактальних зображень з заданими просторово-**