Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Метод комплексної оптимізації енергозбереження та безпеки для технології IoT»

КвРКІП. 170172.23.03.15 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-3 Олексій КОРОЛЬКОВ
Підпис Ім'я, прізвище

Керівник к.т.н.,доцент Дмитро МЕДЗАТИЙ
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІІС, доктор філософії, доцент
Ольга ПАВЛОВА
02 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 01 " 09 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Олексію КОРОЛЬКОВУ
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод комплексної оптимізації енергозбереження та безпеки для технології IoT

Керівник проекту (роботи) Дмитро МЕДЗАТИЙ, д.т.н., доцент
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) аналіз відомих рішень для оптимізації енергозбереження та безпеки для технології IoT, здійснити дослідження предметної області та визначити стратегію щодо комплексної оптимізації енергозбереження та безпеки в технології IoT, розробити метод та алгоритми оптимізації комплексної оптимізації енергозбереження та безпеки в технології IoT.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_01_» 09 2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 25.05.2025	

Студент _____ Олексій КОРОЛЬКОВ
Підпис Ім'я, прізвище

Керівник роботи _____ Дмитро МЕДЗАТИЙ
Підпис Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод комплексної оптимізації енергозбереження та безпеки для технології IoT

Автор роботи: Корольков Олексій Олександрович

Керівник роботи: Медзатий Дмитро Миколайович

Пояснювальна записка: 78 с., 7 рис., 3 табл., 3 дод., 80 джерел.

інтернет речей, іот, енергозбереження, інформаційна безпека, esp32, адаптивне керування, оптимізація, протокол передачі даних.

Об’єктом дослідження є системи інтернету речей, що функціонують у режимі автономного живлення та передають дані через бездротові канали.

Предметом дослідження виступають методи оптимізації енергоспоживання і забезпечення інформаційної безпеки в умовах обмежених ресурсів пристроїв IoT.

Метою дипломної роботи є створення методу комплексної оптимізації енергозбереження та безпеки для техніки IoT

Наукова новизна полягає у розробці комплексного підходу до оптимізації функціонування IoT пристроїв, який одночасно врахує аспекти енергоефективності та інформаційної безпеки. Уперше запропоновано метод, що ґрунтується на адаптивному управлінні режимами роботи мікроконтролера залежно від енергетичного стану пристрою та рівня зовнішніх загроз. Це дозволяє досягти балансу між тривалістю автономної роботи пристроїв та стійкістю до атак.

У першому розділі проведено аналіз відомих рішень для оптимізації енергозбереження та безпеки для технології IoT.

У другому розділі було проведено системний аналіз сучасних підходів до оптимізації енергоспоживання та забезпечення інформаційної безпеки в системах IoT.

У третьому розділі було розроблено метод комплексної оптимізації енергоспоживання та безпеки для пристроїв IoT.

У четвертому розділі було здійснено практичну реалізацію розробленого методу комплексної оптимізації енергоспоживання та безпеки на основі мікроконтролера ESP32.

До основних елементів новизни можна віднести:

- Побудову багатокритеріальної функції корисності ризиків для динамічного вибору режиму функціонування;
- Поєднання алгоритмів обробки даних із оцінкою ризиків на основі поточних параметрів мережі;
- Інтеграцію механізмів енергозбереження з криптографічним захистом;

Практичну реалізацію методу на мікроконтролері ESP32 з аналізом ефективності шляхом експериментального моделювання.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT – Інтернет речей

БД - база даних

БПР - блок прийняття рішень

ММ – математична модель

АУ – алгоритм управління

ЕС - експертна система

VR – віртуальна реальність

AR – доповнена реальність

RCNN - Region-Based Convolutional Neural Network

LiDAR - Light Detection and Ranging

TOID - TOpographic IDentifier

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким впровадженням концепції Інтернету речей (Internet of Things, IoT), яка охоплює широкий спектр сфер – від побуту до промисловості. IoT – це середовище взаємодії фізичних об’єктів, що мають можливість підключення до мережі, збору, обміну та аналізу даних без безпосередньої участі людини. Завдяки технологіям IoT суттєво змінюється підхід до автоматизації, керування ресурсами, моніторингу станів і прийняття рішень у режимі реального часу.

Актуальність теми дослідження зумовлена низкою викликів, що постають перед сучасними IoT-системами. Зокрема, більшість таких пристроїв функціонує в умовах обмеженого енергоживлення, працюючи на батарейках або автономних джерелах енергії. У той же час, через підключення до мережі й обробку конфіденційних даних, IoT-системи стають вразливими до різноманітних кіберзагроз. Таким чином, виникає суперечність між потребою в енергоефективності та необхідністю забезпечення високого рівня безпеки.

Станом на сьогодні в наукових колах і в індустрії активно розробляються рішення, які окремо вирішують проблему енергоспоживання або захисту даних в IoT. Проте інтегрованих підходів, що дозволяють збалансовано оптимізувати обидва аспекти – енергозбереження та інформаційну безпеку – недостатньо. Саме тому актуальним є створення комплексного методу, який забезпечить адаптивне керування режимами роботи пристроїв IoT з урахуванням обмежених ресурсів та умов зовнішнього середовища.

Метою даної дипломної роботи є розробка та реалізація методу комплексної оптимізації енергоспоживання та безпеки в IoT-системах, що дозволить збільшити автономність пристроїв без втрати їхньої стійкості до загроз.

Для досягнення поставленої мети в роботі вирішуються такі завдання:

− дослідити архітектуру IoT-систем, принципи їхнього функціонування та проблематику в контексті енергозбереження та захисту даних;

− проаналізувати сучасні методи оптимізації енергоспоживання і безпеки;

− сформулювати математичну модель, що описує взаємозв’язок між рівнем енергії та ризиком безпеки;

− розробити адаптивний алгоритм вибору режиму роботи пристрою;

− реалізувати експериментальний прототип IoT-системи на базі мікроконтролера ESP32;

− провести тестування ефективності запропонованого методу та оцінити переваги його впровадження.

Об’єктом дослідження є системи Інтернету речей, що функціонують у режимі автономного живлення та передають дані через бездротові канали.

Предметом дослідження виступають методи оптимізації енергоспоживання і забезпечення інформаційної безпеки в умовах обмежених ресурсів пристроїв IoT.

Наукова новизна полягає у розробці комплексного підходу до оптимізації функціонування IoT пристроїв, який одночасно врахує аспекти енергоефективності та інформаційної безпеки. Уперше запропоновано метод, що ґрунтується на адаптивному управлінні режимами роботи мікроконтролера залежно від енергетичного стану пристрою та рівня зовнішніх загроз. Це дозволяє досягти балансу між тривалістю автономної роботи пристроїв та стійкістю до атак.

Практичне значення роботи полягає у створенні гнучкого, масштабованого підходу до підвищення ефективності роботи розумних пристроїв, що дозволяє одночасно продовжити термін автономної роботи системи та підвищити її стійкість до інформаційних загроз. Запропоноване рішення може бути інтегроване у сферу розумного дому, агро промисловості, логістики та промислової автоматизації.

За темою кваліфікаційної роботи опубліковано одну наукову статтю в науковому журналі категорії Б (Korolkov Oleksii, Poplavskyi Serhii, Hlukhenkyi Oleksandr, Ponochovna Olena) TASK METHOD OF COMPREHENSIVE OPTIMIZATION OF ENERGY CONSERVATION AND SECURITY FOR IoT TECHNOLOGY.

1 ТЕОРЕТИЧНІ АСПЕКТИ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ В IOT

1.1 Концепція IoT

Термін «Інтернет речей» (IoT**)** було вперше запропоновано у 1999 році Кевіном Ештоном - співзасновником Центру автоматичної ідентифікації Массачусетського технологічного інституту (Auto-ID Center). Цей термін охоплює глобальну мережу фізичних об’єктів, що мають сенсори, датчики та засоби передачі інформації і можуть підключатися до Інтернету. Концепція IoT передбачає, що ці розумні пристрої здатні автономно взаємодіяти як між собою, так і з оточенням, часто без безпосереднього втручання людини. Кожному пристрою призначається IP-адреса, постійна або динамічна.

Попередником IoT вважається модель Machine-to-Machine (M2M), яка заклала основу для автоматизованої комунікації між пристроями, т. зв. «Інтернет машин». IoT розширює M2M, об’єднуючи його з хмарними технологіями, потужними обчисленнями на локальному рівні та бездротовим зв’язком. Кожен із цих компонентів сприяє створенню доданої вартості як для споживачів, так і для постачальників рішень.

Сучасний Інтернет являє собою сукупність тисяч комп’ютерних мереж, корпоративних, наукових, державних та домашніх. Зв’язок між мережами з різною архітектурою і топологією здійснюється завдяки протоколу IP. Кожному пристрою або групі пристроїв у Мережі надається IP-адреса, постійна або тимчасова (динамічна).

З огляду на швидкий розвиток Інтернету речей, дефіцит адрес може стати серйозною перепоною. Рішенням цієї проблеми є впровадження шостої версії IP-протоколу, IPv6, який дозволяє кожному жителю планети мати понад 300 мільйонів унікальних IP-адрес. Прогнозується, що до 2030 року кількість пристроїв, підключених до Інтернету, становитиме від 30 до 50 мільярдів, і завдяки IPv6 стане можливим практично необмежене ідентифікування будь-якого об’єкта в Мережі.

Технології Інтернету речей, ідентифікація об’єктів, кожен фізичний об’єкт, залучений до Інтернету речей, навіть якщо він безпосередньо не підключений до Мережі, повинен мати унікальний ідентифікатор. Для автоматичної ідентифікації використовуються різні існуючі технології, такі як радіочастотна ідентифікація (RFID), де на об'єкти прикріплюються спеціальні мітки, а також оптичні коди (штрих-коди, QR-коди, Data Matrix), інфрачервоні позначки тощо. Однак для забезпечення унікальності і сумісності між різними типами ідентифікаторів необхідна їх стандартизація.

Засоби збору інформації, датчики та вимірювальні системи перетворюють дані про навколишнє середовище у форму, придатну для подальшої обробки. Це можуть бути як прості сенсори температури або освітленості, так і складні багатофункціональні прилади. Щоб зробити такі системи максимально автономними, доцільно використовувати альтернативні джерела живлення, наприклад, сонячні панелі, що дозволяє зменшити потребу в заряджанні чи заміні батарей. Технології передачі даних, інтернет речей використовує різні канали зв'язку для передачі інформації. У випадку бездротових рішень велике значення має забезпечення надійності зв'язку. Для дротових мереж поширеним рішенням є передача даних через електромережу, що зручно для пристроїв, які вже підключені до живлення (наприклад, банкомати, торгові автомати).

Системи обробки інформації, згідно з прогнозами, понад 30 мільярдів пристроїв, які будуть підключені до Інтернету, створюватимуть до 44 зетабайт (мільярдів терабайт) даних. Це у кілька разів перевищує обсяг усіх цифрових даних, що існували на початку 2010-х років. У Microsoft вважають, що ключовим елементом Інтернету речей є не стільки сенсори чи комунікаційні технології, скільки хмарні обчислення, які дозволяють швидко аналізувати великі обсяги інформації та приймати оперативні рішення (наприклад, виявляти залишені відкритими двері в порожньому будинку). Додатково в цьому допомагатимуть так звані "туманні обчислення", які не замінюють хмарні технології, а ефективно їх доповнюють. Виконавчі пристрої, це елементи, які здійснюють фізичні дії у відповідь на отримані цифрові сигнали. Наприклад, для дистанційного керування опаленням у будинку з мобільного

пристрою потрібне спеціальне обладнання, здатне виконати команду. Часто виконавчі механізми інтегрують з датчиками в єдині модулі.

Сфери застосування, очікується, що Інтернет речей охопить численні галузі, промисловість - автоматизація виробничих процесів, транспорт, до 220 млн підключених авто, розумні будинки, комфорт, енергоефективність, безпека, Комунальні служби, мільярди датчиків для моніторингу ресурсів і зниження втрат, охорона здоров'я, понад 646 млн пристроїв для збору медичних даних, агросектор, близько 75 млн сенсорів для аналізу стану ґрунту.

Інтернет речей також знайде застосування у торгівлі, логістиці, громадському харчуванні, готельному бізнесі, фінансовому секторі, будівництві та навіть в армії, прогнозується використання понад 126 тисяч роботів і дронів. Стандарти та компанії, оскільки ринок Інтернету речей стрімко розвивається, великі компанії намагаються зайняти на ньому лідируючі позиції. Однак із розвитком нової технології може виникнути «війна стандартів». Для її уникнення провідні гравці об'єднують зусилля.

Наприклад, AllSeen Alliance та AllJoyn (Qualcomm) разом із Open Interconnect Consortium (OIC) створили Open Connectivity Foundation (OCF), яка працює над уніфікацією стандартів. Також існує глобальний стандарт OneM2M, підтримуваний понад 230 компаніями, серед яких Amazon, Cisco, Huawei, Intel, NEC, Qualcomm, Samsung тощо. Інформаційна безпека. Фахівці попереджають? нині не існує повністю захищеної екосистеми Інтернету речей. Причинами є слабке шифрування, відсутність складних паролів, вразливі прошивки тощо.

Це відкриває шлях для хакерів, які можуть, наприклад, вмикати/вимикати чужі побутові прилади, блокувати двері або навіть стежити за людьми через камери у розумних пристроях. Серед запропонованих рішень, обов'язкова сертифікація пристроїв, встановлення уніфікованих захисних чіпів та інші заходи безпеки. Сучасні пристрої IoT відрізняються низьким енергоспоживанням, доступною ціною, наявністю базових функцій «з коробки» та глобальною доступністю. Водночас варто розрізняти поняття «Інтернет речей» як мережевої концепції і «Інтернет-річ», окремого пристрою,

який має, доступ до Інтернету для передачі чи отримання даних, унікальну адресу або ідентифікатор, інтерфейс для взаємодії з користувачем.

Мережі IoT функціонують на принципах єдиного протоколу, де всі вузли є рівноправними. Однією з технічних перепон на шляху розвитку IoT був брак IP-адрес у межах протоколу IPv4. Перехід до IPv6 вирішує цю проблему, відкриваючи можливість надати кожному мешканцю планети понад 300 мільйонів адрес.

IoT-модель включає мільйони пристроїв, які потрібно ефективно адмініструвати. Це вимагає забезпечення безпеки, конфіденційності та сумісності між різноманітними протоколами й пристроями. У цій системі передбачено дві основні моделі обміну даними, D2D (Device-to-Device), пряме з’єднання між пристроями, D2S (Device-to-Server), передача даних на сервер для подальшої обробки.

Кожен пристрій у мережі виконує функцію постачальника сервісів і може також приймати команди від інших пристроїв. Вони об’єднуються у локальні мережі, які можуть мати спільну зону покриття або виконувати одну функцію. Модель компанії Beecham Research, зображена на рисунку 1, демонструє класифікацію IoT-додатків за галузевими напрямами та типами пристроїв.

Серед технологій майбутнього, які тісно пов’язані з розвитком IoT, виокремлюють:

1. штучний інтелект (AI) та робототехніку;
2. великі дані (Big Data);
3. хмарні технології (Cloud);
4. 3D-друк (Additive Manufacturing, AM);
5. Інтернет нового покоління (Web 3.0).

IoT відіграє подвійну роль, він не лише використовує ці технології, а й сам сприяє їх розвитку. Завдяки хмарним сервісам та аналітиці IoT стає основою для нових інтелектуальних систем.

Згідно з дослідженням компанії Ericsson, у світі вже існує понад 16 мільярдів підключених пристроїв, і це число зростає. До 2022 року

прогнозується 29 мільярдів пристроїв, з яких 18 мільярдів, це елементи Інтернету речей.

IoT інтегрує фізичні об’єкти у віртуальні системи, що дозволяє автоматизувати повсякденні дії, наприклад, дистанційне керування побутовими приладами. Це значно полегшить життя, зменшивши кількість рутинних завдань і підвищивши якість життя як у побуті, так і в бізнесі, медицині, екології та інших сферах.

Наприклад, в Австралії лікарі використовують переносні сенсори для моніторингу стану пацієнтів у реальному часі.

У США компанія AT&T створила систему, яка автоматично фіксує падіння літніх людей і повідомляє про це служби допомоги.

У підсумку, підключення пристроїв до Інтернету відкриває нові можливості для оптимізації використання ресурсів, таких як електроенергія, вода, газ тощо.

Ключові терміни IoT:

1. Інтернет речей;
2. IoT-пристрої;
3. Екосистема IoT;
4. Фізичний рівень;
5. Рівень додатків;
6. Панель керування;
7. Інструментальні панелі;
8. Аналітична складова.

1.2 Платформи IoT-систем

Швидке збільшення кількості IoT-пристроїв спонукає до впровадження ефективних рішень для їх підключення до мережі, збору, зберігання та аналізу даних. Amazon Web Services (AWS) надає комплексні інструменти, що охоплюють увесь ланцюг, від периферії до хмарних сервісів, дозволяючи створювати IoT-рішення для найрізноманітніших пристроїв і сценаріїв використання. Платформа AWS IoT (рисунок 1.3) побудована на надійній

хмарній інфраструктурі AWS, що активно використовується лідерами ринку, і дозволяє легко масштабуватися відповідно до змін у кількості підключених пристроїв і вимог бізнесу.

Завдяки AWS IoT можна безперешкодно інтегрувати пристрої, керувати ними та збирати дані. Надійність хмарної інфраструктури Amazon дозволяє реалізовувати практично будь-який IoT-проєкт. До складу хмарних сервісів входить пакет рішень для Інтернету речей, зокрема AWS IoT Core, який є фундаментом для побудови IoT-додатків. Через цю платформу пристрої можуть з'єднуватися з Інтернетом, між собою та обмінюватися даними. AWS гарантує безпечний обмін мільярдами повідомлень між пристроями і хмарними сервісами. Крім того, платформа підтримує різні протоколи, зокрема й спеціалізовані, що забезпечує сумісність між пристроями різних виробників. За допомогою AWS IoT Device Management можна зручно реєструвати, організовувати та адмініструвати пристрої.

Сервіс гарантує захищену та масштабовану роботу з функцією моніторингу, виправлення помилок і покращення функцій пристрою. AWS IoT Analytics пропонує сервіс для автоматизованого аналізу великих обсягів різнорідних IoT-даних, включно з неструктурованою інформацією з різних пристроїв. Зібрані та опрацьовані службою дані можуть бути використані для машинного навчання. AWS IoT Device Defender підтримує конфігурацію механізмів безпеки у системах IoT. AWS IoT Device Defender надає можливість налаштовувати та контролювати політики безпеки, включно з управлінням аутентифікацією та авторизацією пристроїв, а також впроваджує засоби шифрування для захисту даних. Google Cloud Platform, це глобальна хмарна платформа, яка підтримує рішення для Інтернету речей. Пакет Google Cloud IoT дає змогу створювати та адмініструвати IoT-системи будь-яких масштабів і рівнів складності. Google Cloud IoT пропонує набір інструментів для створення і керування IoT-мережами. Сервіс Cloud IoT Core надає повністю керовану платформу для безпечного підключення та управління пристроями, а також прийому їхніх даних. За обробку подій і аналіз потоків у реальному часі відповідає Cloud Pub/Sub. Для розробки і застосування моделей машинного навчання на основі отриманих з IoT-пристроїв даних

використовується Cloud Machine Learning Engine. Інтернет речей від компанії Google представлений набором сервісів, які допомагають будувати складні мережі з підключеними пристроями. Платформа Microsoft Azure пропонує як готові рішення, так і можливість їх налаштування або створення нових продуктів відповідно до конкретних потреб проекту. Сучасні хмарні платформи забезпечують безпечну, масштабовану та високо продуктивну інфраструктуру для реалізації рішень Інтернету речей (IoT). Завдяки цим платформам організації отримують доступ до інструментів для моніторингу пристроїв, усунення несправностей, оновлення функціональності та впровадження інтелектуального аналізу даних.

AWS IoT (Amazon Web Services) Amazon пропонує повний стек сервісів для IoT. Зокрема, AWS IoT Analytics, спеціалізований сервіс, що дозволяє автоматизувати аналітику великих обсягів IoT-даних, включаючи неструктуровані потоки інформації з різних типів пристроїв. Сервіс виконує попередню обробку, фільтрацію, трансформацію і підготовку даних до подальшого використання, зокрема у машинному навчанні (ML). AWS IoT Device Defender, інструмент для забезпечення кібербезпеки IoT-систем, який дозволяє, налаштовувати політики безпеки, контролювати аутентифікацію та авторизацію пристроїв, впроваджувати механізми шифрування та виявлення аномалій у поведінці пристроїв.

Ці рішення дозволяють компаніям зосередитися на розробці функціональності пристроїв, не турбуючись про захист і масштабованість. Google Cloud IoT Google Cloud Platform (GCP) є ще одним потужним гравцем у сфері IoT. Її рішення побудовані з акцентом на швидкість, безпеку та аналітику в реальному часі, Cloud IoT Core, повністю керований сервіс, що забезпечує просте і захищене підключення пристроїв, а також збір і обробку даних у хмарі. Cloud Pub/Sub, сервіс для обробки подій з пристроїв і аналітики потоків у реальному часі, з можливістю масштабування на мільйони повідомлень, Cloud Machine Learning Engine, інструмент для створення моделей машинного навчання, які можна навчати на основі даних, зібраних від IoT-пристроїв.

Ці сервіси є частиною екосистеми Google Cloud IoT, яка підтримує

створення складних, високопродуктивних, підключених систем на будь-якому рівні складності. Microsoft Azure IoT, Платформа Azure від Microsoft пропонує широкі можливості як для початківців, так і для досвідчених розробників. Azure підтримує, попередньо сконфігуровані рішення для швидкого старту, можливість налаштування інфраструктури згідно з індивідуальними вимогами проекту, потужні сервіси безпеки, аналітики, управління пристроями та інтеграції з Azure AI.

Azure IoT надає єдину платформу для побудови надійних IoT-систем з підтримкою розширюваності, автоматизації, візуалізації та інтелектуального аналізу даних.

Хмарні IoT-рішення від AWS, Google Cloud та Microsoft Azure є лідерами на ринку завдяки гнучкості, надійності та широкому набору інструментів для управління даними, безпекою і масштабуванням.

Microsoft Azure IoT Suite забезпечує високий рівень безпеки, масштабованість та легку інтеграцію з поточними і майбутніми IT-системами. Платформа підтримує підключення великої кількості пристроїв різних виробників, збирає аналітичні дані та дозволяє використовувати інформацію з IoT-пристроїв для задач машинного навчання.

SAP Cloud Platform for IoT надає повний набір інструментів для створення, керування та моніторингу IoT-додатків. Вона слугує ефективним середовищем для віддаленого управління пристроями, які можуть підключатися як напряму, так і через хмарні сервіси. Потужні можливості аналітики дозволяють обробляти, класифікувати та аналізувати дані, що надходять з датчиків, лічильників та інших IoT-пристроїв. Завдяки підтримці новітніх технологій SAP також відкриває можливості для створення рішень на основі штучного інтелекту та машинного навчання з використанням IoT-даних.

Salesforce IoT зосереджується на побудові інтегрованої екосистеми, яка забезпечує безперервну взаємодію IoT-пристроїв із клієнтськими рішеннями Salesforce. Це дозволяє компаніям створювати персоналізовані сервіси та автоматизувати процеси, використовуючи дані в режимі реального часу.

Серед сучасних провідних платформ для створення та управління

рішеннями Інтернету речей (IoT) важливу роль відіграють Salesforce IoT, Oracle IoT та Cisco IoT, кожна з яких пропонує унікальні інструменти й функції для розробників і підприємств. Salesforce IoT, платформа Salesforce IoT надає можливість створювати індивідуальні додатки IoT, адаптовані під конкретні бізнес-потреби. Її головна перевага полягає у гнучкості підключення, платформа дозволяє під'єднувати будь-який пристрій, незалежно від виробника чи технології, що використовується, зібрані дані з пристроїв перетворюються в цінну інформацію для подальшого аналізу та дій, Salesforce дозволяє інтегрувати IoT-дані з CRM-системою, що відкриває шлях до персоналізованої взаємодії з клієнтами, автоматизованих процесів обслуговування та прогнозної аналітики. Oracle Internet of Things (IoT) Oracle IoT, це потужне середовище для створення комерційних IoT-додатків, яке поєднує світ реальних пристроїв із корпоративними інформаційними системами, Oracle надає інтеграцію з ERP, SCM, CRM та іншими бізнес-додатками, що забезпечує повний цикл управління даними, платформа має високу масштабованість, що дозволяє працювати з надзвичайно великими обсягами даних, завдяки сильному бекграунду Oracle у сфері баз даних, IoT-рішення можуть обробляти складні запити й великі масиви інформації без втрати продуктивності, безпека займає ключове місце, Oracle застосовує централізовані механізми захисту, що критично важливо для гетерогенних мереж, де не всі пристрої мають вбудований захист.

Cisco IoT, компанія Cisco орієнтується на створення хмарних рішень для мобільних IoT-систем, що особливо актуально в умовах динамічного середовища та глобальної взаємодії, платформа підтримує голосовий та даний трафік, інтеграцію з мобільними додатками та можливості для монетизації IoT-сервісів, пропонується повний функціонал управління, включно з моніторингом пристроїв, віддаленим доступом і безпекою, IoT Services for Utility Networks, спеціалізоване рішення, яке дозволяє створювати системи для комунальних підприємств (водопостачання, енергетика тощо), IoT Advisory, консультаційна послуга, яка надає компаніям експертну підтримку з питань розробки, впровадження та оптимізації IoT-стратегій.

Усі згадані платформи — Salesforce, Oracle та Cisco — орієнтовані на

різні сценарії використання IoT та підтримують високу масштабованість, інтеграцію, захист і гнучкість налаштувань, що робить їх придатними як для малого бізнесу, так і для великих корпорацій.

Salesforce IoT дозволяє розробляти індивідуальні IoT-додатки, підключати різноманітні пристрої та аналізувати отримані дані для подальшого використання у бізнес-процесах.

Oracle Internet of Things забезпечує інтеграцію корпоративного програмного забезпечення з фізичними пристроями та їх метриками. Платформа створює гнучке середовище для розробки комерційних IoT-додатків, надаючи компаніям можливість працювати з великими обсягами даних завдяки потужним інструментам обробки інформації. Oracle також впроваджує сучасні засоби захисту для запобігання зовнішнім загрозам, що особливо важливо, враховуючи наявність у мережах IoT пристроїв із обмеженими можливостями безпеки. Централізований підхід до захисту дозволяє значно підвищити рівень безпеки таких систем.

Платформа ThingWorx забезпечує достатньо можливостей для розробки повноцінних IoT-додатків без потреби у використанні сторонніх бібліотек чи інструментів. Рішення, створені на її основі, відповідають вимогам сучасного корпоративного середовища, вони легко масштабуються та інтегруються з передовими технологіями, такими як доповнена реальність та інтелектуальна аналітика. Вся реалізована функціональність доступна у зручному та інтуїтивно зрозумілому інтерфейсі, що поєднує ефективність роботи з комфортом користування.

Cisco орієнтується на мобільні IoT-рішення, які базуються на хмарних технологіях. Платформа забезпечує передачу як даних, так і голосових сигналів, а також підтримує налаштування додатків і функції монетизації IoT-сервісів. Обравши рішення Cisco для розміщення IoT-додатків, користувачі отримують комплексні інструменти для керування пристроями, моніторингу та безпеки з особливим акцентом на мобільну взаємодію з користувачами. Додаткові сервіси, такі як IoT Services for Utility Networks, орієнтовані на впровадження рішень для комунальних підприємств, а IoT Advisory надає експертну підтримку з ключових питань у сфері IoT.

1.3 Архітектура IoT

Класична структура Інтернету речей (IoT) включає кілька основних елементів: пристрої IoT, шлюзи, сервери та клієнтські інтерфейси.

IoT-пристрої відповідають за збір даних через датчики та можуть виконувати фізичні дії. Сенсори - ключовий міст між фізичним і цифровим світами, дозволяють отримувати та обробляти інформацію в реальному часі. Завдяки мініатюризації ці елементи можна вбудовувати безпосередньо в фізичні об’єкти. Вони вимірюють різні параметри, від температури до геолокації, та зберігають кілька результатів у вбудованій пам’яті. Залежно від функції сенсори класифікують як екологічні, біомедичні, побутові, автомобільні тощо.

Зв’язок сенсорів із системою часто здійснюється через агрегатори або шлюзи, що можуть бути побудовані на основі LAN (Ethernet, Wi-Fi) або PAN (ZigBee, Bluetooth, UWB). Для незалежної роботи сенсорів можливе використання WAN-технологій (GSM, GPRS, LTE). Якщо пристрої мають низьке енергоспоживання і передають невеликі обсяги даних, вони можуть бути частиною безпроводових сенсорних мереж (WSN), здатних покривати великі площі та працювати від батарей.

Шлюзи, це апаратні або програмні модулі, які транслюють дані від сенсорів до серверів і передають їм команди. Вони є частиною конвергентної мережевої інфраструктури, яка інтегрує різні мережі в єдину платформу, забезпечуючи одночасний доступ для кількох користувачів без шкоди для безпеки чи ефективності.

Сервер - місце зберігання та обробки даних. Він може бути реалізований як фізичний пристрій, віртуальна машина або хмарне рішення. Клієнтська частина (мобільний/веб-додаток) забезпечує зручний доступ до інформації й візуалізацію результатів.

Взаємодія з речами реалізується через датчики (sensors) та виконавчі механізми (actuators), як в автоматизованих системах управління. Ці елементи функціонують у граничній зоні (Edge), де події збираються, зберігаються та передаються на аналітичну платформу, де дані обробляються, передаються у

реальному часі, зберігаються, розподіляються між застосунками, а також використовується машинне навчання. Цей рівень реалізується на основі хмарних (Cloud) або туманних (Fog) обчислень і схожий на рівень контролерів у систем SCADA.

Інтернет речей базується на подіях реального світу — зміні температури, русі, відкриванні дверей тощо. Іноді навіть простий сенсор може генерувати значний обсяг даних, як, наприклад, акустичний сенсор для моніторингу технічного стану. Водночас достатньо одного біта, щоб повідомити про критичні зміни в стані здоров'я. Приклади сучасного використання систем Інтернету речей (IoT) Розумний автомобіль, сучасні транспортні засоби, зокрема автомобілі, можуть підключатися до Інтернету різними способами, через смарт-відеореєстратори, мультимедійні системи або спеціальні автомобільні шлюзи. Такі системи збирають інформацію з педалі газу, гальм, спідометра, одометра, коліс та паливних баків для контролю за станом транспортного засобу та манерою водіння. Розумні автомобілі знаходять застосування у багатьох сферах, моніторинг автопарків орендованих авто з метою економії пального та зниження витрат, відстеження батьками стилю водіння дітей, автоматичне сповіщення родичів чи друзів у разі аварії, прогнозування технічного обслуговування автомобіля для запобігання поломкам.

Розумний дім, системи розумного дому орієнтовані переважно на підвищення енергоефективності, безпеки житла та оптимізацію домашньої автоматизації. Наприклад, розумні розетки контролюють споживання електроенергії, інтелектуальні термостати регулюють температуру, гідропонні системи з IoT-датчиками керують вирощуванням рослин, датчики диму можуть виявляти наявність тютюнового диму, системи безпеки (розумні замки, камери, датчики протікання води) виявляють загрози й сповіщають власника.

Інші можливості, автоматичне вимкнення непотрібних приладів, керування орендованою нерухомістю, пошук загублених речей (наприклад, ключів або гаманця), автоматизація побутових процесів, як-от прибирання пилососом чи приготування кави. Розумні міста, IoT-додатки значно

покращили міське планування та управління інфраструктурою. Органи влади використовують технології Інтернету речей для вирішення проблем у сферах охорони здоров’я, екології та комунального господарства. Наприклад, моніторинг якості повітря та рівня радіації, зниження витрат на освітлення за допомогою розумних світильників, виявлення необхідності ремонту інфраструктури (дороги, мости, трубопроводи), покращення управління паркувальними майданчиками для збільшення прибутку, розумні будівлі, навчальні заклади, офісні центри та інші великі будівлі впроваджують IoT-рішення для підвищення ефективності. Серед основних цілей використання, скорочення енергоспоживання, зменшення витрат на технічне обслуговування, оптимізація використання приміщень та робочих зон.

Саме здешевлення, зменшення розміру та ефективність сенсорів і граничних пристроїв сприяють підключенню мільярдів пристроїв до IoT. Водночас ці елементи потребують ефективних енергетичних рішень — живлення, яке має бути спланованим, адже кількість таких пристроїв величезна.

Надійний зв’язок, основа IoT. Система не могла б існувати без сучасних технологій передачі даних, які забезпечують інтеграцію навіть із найвіддаленіших місць. Основу передачі становлять персональні бездротові мережі (PAN), такі як Bluetooth, ZigBee, Z-Wave, WirelessHART, ISA100 та інші. Для передачі інформації в інтернет-середовище потрібні шлюзи-маршрутизатори та мережеві протоколи, які забезпечують ефективність і безпеку комунікацій.

Важливу роль відіграє граничний маршрутизатор (Edge router), що забезпечує якість, безпеку й управління даними в розподілених мережах. Саме він часто виступає як розширення хмари, що підтримує такі функції, як VPN, VLAN і SD-WAN.

У меж IoT застосовуються енергозберігаючі протоколи з низькою затримкою, наприклад:

1. MQTT;
2. AMQP;
3. CoAP.

Не всі дані слід відправляти в хмару. Обробка на рівні Edge або Fog значно економніша. Технології туманних обчислень, наприклад архітектура OpenFog, дозволяють ефективно управляти потоками даних.

Інформація, отримана з сенсорів, може бути цінною для аналітики та застосування машинного навчання, наприклад через рекурентні нейронні мережі, байєсовські моделі чи складні обробники подій.

IoT-системи все частіше виходять за межі офісів і будинків — вони застосовуються в транспорті, медицині, на віддалених об’єктах. Проте це створює нові загрози безпеці. Відомі випадки зламів IoT-пристроїв навіть на рівні критичної інфраструктури. Розробники мають розуміти природу цих загроз і впроваджувати сучасні стандарти захисту IoT-середовища.

1.4 Постановка задачі

Для досягнення поставленої мети необхідно виконати наступні ключові завдання:

1. Визначити ключові параметри, які впливають на енергоспоживання та безпеку.
2. Побудувати оптимізаційну модель із врахуванням конфлікту між цими параметрами.
3. Реалізувати алгоритм, здатний працювати в умовах обмежених ресурсів.

1.5 Висновки до першого розділу

У результаті проведеного аналізу теоретичних засад та сучасного стану розвитку Інтернету речей, було встановлено, що IoT є однією з ключових технологій цифрової трансформації, яка забезпечує інтеграцію фізичних об’єктів у єдину інформаційну екосистему. Поняття, архітектура і принципи роботи IoT систем ґрунтуються на поєднанні сенсорних пристроїв, засобів та обчислювальних платформ, здатних функціонувати автономно та взаємодіяти між собою без участі людини.

Визначено, що основними проблемами в сфері IoT є енергоспоживання пристроїв та забезпечення інформаційної безпеки, зумовлені обмеженими ресурсами апаратних засобів, необхідністю безперервного з’єднання з мережею та високими вимогами до конфіденційності даних. Було виявлено, що більшість рішень фокусуються на одному з аспектів, або енергоефективності або захисті, що не дозволяє досягти збалансованого функціонування системи.

Таким чином, на основі аналізу літературних джерел і практики використання IoT у різних галузях, сформульовано потребу в розробці комплексного методу, який забезпечуватиме одночасну оптимізацію енергоспоживання та безпеки в IoT пристроях.

2 ПРОЦЕС ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ В ТЕХНОЛОГІЇ IOT

2.1 Проблемні завдання енергозбереження

У сфері IoT одним із ключових завдань є мінімізація споживання енергії при збереженні ефективності пристроїв. Основними традиційними методами вважають, протоколи з низьким рівнем енергоспоживанням, такими як BLE, ZigBee, Z-Wave, вони забезпечують зв’язок з мінімальними витратами.

Одним з найпоширеніших протоколів бездротової передачі даних у середовищі IoT є ZigBee – відкритий стандарт, розроблений однойменним альянсом, який об’єднує понад 200 провідних компаній, таких як Texas Instruments, Motorola, Philips, IBM, Samsung та інші.

ZigBee створений для застосувань, які потребують гарантованої доставки даних при низьких швидкостях передавання (до 250 кбіт/с) та мінімального споживання енергії. Робочий радіус у відкритому просторі може досягати 75 метрів. Основною особливістю ZigBee є підтримка гнучкої топології мережі – зокрема, типів зірка, дерево та сітка з можливістю самоорганізації і маршрутизації даних. Пристрої ZigBee можуть більшість часу перебувати в режимі сну, що значно подовжує час автономної роботи.

Залежно від функціонального призначення пристрої поділяються на координатори (PAN-координатори), маршрутизатори та кінцеві пристрої (термінали). Координатор ініціює і підтримує структуру мережі, збирає дані та здійснює зв’язок із зовнішніми системами, маршрутизатори передають інформацію далі мережею, а кінцеві пристрої здебільшого виконують функції збору даних.

Іншим бездротовим протоколом, орієнтованим на автоматизацію побутових систем, є Z-Wave. Ця технологія, побудована на стандарті ITU G.9959, застосовується у розумних будинках для керування освітленням, опаленням, доступом тощо. Вона використовує малопотужні мініатюрні радіомодулі в частотному діапазоні до 1 ГГц, що забезпечує високу стійкість до перешкод, на відміну від перевантаженого діапазону 2,4 ГГц.

Z-Wave підтримує топологію сітки з маршрутизацією сигналів через проміжні вузли, що дозволяє покращити покриття. Стандарт визначає три типи вузлів: контролери, маршрутизуючі виконавчі пристрої і звичайні виконавчі пристрої. Максимальна кількість пристроїв у мережі – до 232. Попри простоту та енергоефективність, Z-Wave має обмежену швидкість передавання даних (до 40 кбіт/с), що унеможливлює трансляцію мультимедійних потоків.

Ще одним актуальним стандартом є Bluetooth Low Energy (BLE) – технологія ближнього радіозв’язку, яка була розроблена Bluetooth SIG. BLE забезпечує наднизьке енергоспоживання (в 10–20 разів менше, ніж класичний Bluetooth) та швидкість передавання даних до 1 Мбіт/с. Вона орієнтована на роботу з автономними пристроями типу точка-точка або зірка, зокрема у медичних пристроях, фітнес-гаджетах, датчиках і трекерах.

Стек BLE включає фізичний і канальний рівні (у контролері) та логічні протоколи (у хості), що реалізуються програмно. Серед них: L2CAP, ATT, GATT, SMP, GAP, які відповідають за передачу, структуру атрибутів, безпеку та доступ до профілів. BLE використовує GFSK-модуляцію та працює в діапазоні 2,4 ГГц на 40 частотних каналах. Також підтримується адаптивна перебудова частоти, що забезпечує стійкість до інтерференції.

BLE-пристрої поділяються на однорежимні (single-mode) – з мінімальним енергоспоживанням, і дворежимні (dual-mode), що сумісні як із

BLE, так і з класичним Bluetooth. Це дає змогу інтегрувати BLE у смартфони, планшети, ноутбуки, медичні пристрої, системи безпеки та освітлення.

Режим сну, реалізується через тимчасове відключення модулів або зниження частоти процесора. Використання IoT-технологій є доцільним у межах однієї організації, зокрема в «закритих системах», де вони найбільш ефективні для реалізації диспетчерських функцій. Наприклад, за допомогою Інтернету речей можна забезпечити безпеку будівель, у яких тимчасово відсутні люди, або дистанційно контролювати рівень загазованості приміщень. Ще важливішим є застосування IoT для моніторингу транспортування газу та обліку його споживання конкретними користувачами. Саме в цьому випадку на сьогодні можна говорити про практичне використання IoT для підвищення енергоефективності.

Опитані експерти також підкреслюють, що IoT-технології в першу чергу цікавлять споживачів з точки зору економії, тобто збору даних про використані ресурси та подальшої їх оплати. Тема енергоефективності як така викликає менше інтересу.

Сенсорні мережі IoT зазвичай включають від сотень до тисяч вузлів, кожен з яких має автономне живлення від батареї. Через відсутність можливості перезарядки таких батарей, надзвичайно важливою є проблема оптимального планування енергоспоживання для подовження терміну служби всієї мережі.

Один із поширених підходів, переведення вузлів у «сплячий режим», коли вони не виконують активних дій, та пробудження лише за необхідності. Щоб мінімізувати затримки доставки пакетів, застосовують так звані опортуністичні схеми маршрутизації, кожен вузол передає дані тому з сусідніх вузлів, який першим прокидається. Це дозволяє зменшити затримки порівняно з традиційними методами, де пакет передається лише на конкретний заздалегідь визначений вузол, що потребує очікування його пробудження.

Одним із рішень для підвищення енергоефективності є впровадження Основного Розкладу (Backbone Scheduling, BS), механізму, що дозволяє динамічно вимикати радіомодулі сенсорних вузлів. За такої схеми лише частина вузлів підтримує активне радіоз’єднання для передавання

повідомлень, тоді як інші переходять у режим енергозбереження. Це суттєво знижує загальне енергоспоживання сенсорної мережі.

Запропонований метод оптимізації енергоспоживання у бездротових сенсорних мережах Інтернету речей (IoT) не погіршує якість зв’язку, оскільки такі мережі мають властивість надмірності. Під цим поняттям мається на увазі, що навіть при вимкненні окремих радіомодулів датчиків (наприклад, для економії енергії), мережеве з’єднання не втрачається. Це можливо завдяки наявності альтернативних маршрутів, які можуть автоматично компенсувати втрату одного з вузлів.

Саме ця структурна надмірність дозволяє реалізовувати більш ефективні енергозберігаючі рішення. Зокрема, можливо побудувати комунікаційні "магістралі" або спеціальні шляхи передачі даних, які використовують лише частину доступних вузлів, залишаючи інші у режимі сну для збереження заряду.

Одним з ключових інструментів для створення таких комунікаційних структур є алгоритм підключеного домінуючого набору (CDS – Connected Dominating Set). Він дозволяє побудувати ефективну основу зв’язку, використовуючи мінімальну кількість активних вузлів, які при цьому покривають усю мережу.

Однак, використання лише однієї CDS-структури не є достатнім для значного продовження строку служби мережі. Замість цього пропонується створювати кілька незалежних CDS, які працюють по черзі. Це дозволяє рівномірно розподіляти навантаження між вузлами і періодично виводити частину з них у режим енергозбереження.

Такий підхід формалізовано у вигляді проблеми підключеного догматичного розділу (CDP, Connected Dominating Partition). Для її реалізації був розроблений спеціальний алгоритм, віртуальне планування масштабування (VBS, Virtual Backbone Scheduling). Цей алгоритм дозволяє, створювати кілька перекритих магістральних структур (CDS), формувати оптимальний графік "сну" для вузлів, забезпечувати рівномірний розподіл енергоспоживання по всій мережі, максимально задіювати ресурси кожного вузла, що сприяє ефективному використанню залишкової енергії.

У результаті застосування VBS досягається значне продовження строку служби всієї IoT-мережі, без втрати її працездатності та якості зв’язку.

Обмеження енергоспоживання бездротових сенсорних вузлів зумовлене їх компактними розмірами та бездротовою природою, що унеможливлює використання постійного джерела живлення. Як правило, живлення таких вузлів здійснюється від батарей. Проте, враховуючи масштаб бездротових сенсорних мереж, від сотень до тисяч вузлів, та їх часте розгортання у важкодоступних або агресивних середовищах, заміна або підзарядка батарей стає надзвичайно складною задачею.

Енергія у вузлах витрачається на кілька ключових процесів: активацію сенсорів, обробку зібраної інформації та передавання даних. Обмеженість енергетичних ресурсів також негативно впливає на рівень безпеки, оскільки криптографічні алгоритми передбачають додаткові комунікації між вузлами для обміну ключами, що збільшує кількість переданих повідомлень і, відповідно, енергоспоживання.

Одним із ефективних способів продовження терміну служби мережі є планування режимів сну та активної роботи вузлів. Проте цей підхід має недолік, можливі затримки при передачі даних, коли передавальний вузол змушений чекати, доки вузол-ретранслятор не вийде з режиму сну.

Для мінімізації затримок пропонуються опортуністичні схеми пересилання пакетів, згідно з якими кожен вузол передає дані першому з доступних сусідніх вузлів, що прокидається серед набору кандидатів. Поєднання протоколів планування сну/активності з такими схемами пересилання дозволяє досягти балансу між енергоефективністю і мінімальною затримкою доставки пакетів, що є критично важливим для довготривалої та стабільної роботи IoT-систем.

Планування сплячого режиму вважається одним із найефективніших способів продовження терміну служби бездротових сенсорних мереж (WSN), особливо у сфері Інтернету речей (IoT). Оскільки більшість сенсорних вузлів мають обмежені джерела живлення (наприклад, батареї), оптимізація споживання енергії є критично важливою для підтримання довготривалої та стабільної роботи мережі.

У загальному випадку, енергія, необхідна для фіксації подій (вимірювань) сенсорами, є сталою і не підлягає значному зниженню. Проте енергія, що витрачається на підтримку зв’язку, а саме на прослуховування середовища, прийом і передачу керуючих пакетів, є змінною, і саме вона становить основну частину енергетичних витрат, які можна контролювати.

Сплячий режим як механізм енергозбереження, виводячи вузли у сплячий режим (sleep mode), коли вони не виконують критичних функцій, можна істотно знизити енергоспоживання мережі. Це особливо ефективно для IoT-сценаріїв, де події виникають нерегулярно, а активність мережі змінюється з часом.

Запропонована система використовує асинхронний метод планування: кожен вузол незалежно приймає рішення про перехід до сплячого режиму або пробудження, не координуючись із сусідами. Такий підхід дозволяє уникнути складної синхронізації, зменшити навантаження на мережу та забезпечити гнучкість у керуванні енергією, асинхронне пробудження та передача

Коли виникає потреба передати дані, вузол вибирає наступний вузол для пересилки пакета, керуючись протоколом типу Sleep/Wake, який визначає, які вузли доступні для взаємодії в даний момент. Якщо вибраний вузол перебуває в сплячому режимі, пакет очікує або передається іншому кандидату, доступному в мережі.

Пуассонівське планування, одним із особливо ефективних підходів до планування сплячого режиму є використання Пуассонівських процесів пробудження. У цьому випадку моменти пробудження вузлів генеруються випадково за законами розподілу Пуассона. Такий підхід має кілька важливих переваг, простота реалізації: особливо важлива у випадку пристроїв з обмеженою пам’яттю та обчислювальними ресурсами, рівномірне енергоспоживання: розподіл навантаження між вузлами дозволяє уникати передчасного розрядження окремих пристроїв, гнучка адаптація до динаміки мережі: ймовірнісна модель дозволяє ефективно функціонувати навіть у нестабільних середовищах.

У результаті, планування на основі Пуассонівських розкладів дозволяє реалізувати оптимальну політику керування енергією, що максимізує

життєвий цикл сенсорної мережі, не погіршуючи її функціональність і надійність.

Якщо розглядати часові інтервали між подіями, пов’язаними з увімкненням і вимкненням вузлів бездротової сенсорної мережі у зв’язку з потребою пересилання інформаційних пакетів, то ці події можуть бути змодельовані як пуассонівський процес. У такому процесі часові проміжки між подіями є незалежними випадковими величинами, що мають експоненціальний розподіл, а сама послідовність подій утворює пуассонівський потік. Ця модель добре описує ситуації, де події (наприклад, активація вузлів) виникають у випадкові моменти часу незалежно одна від одної, що є характерним для багатьох децентралізованих сценаріїв роботи IoT-систем.

2.2 Класифікація проблем енергозбереження засобів Інтернету речей

Шифрування даних, це метод забезпечення інформаційної безпеки, який полягає в кодуванні даних таким чином, щоб доступ до них могли отримати лише уповноважені особи, які володіють відповідним ключем розшифрування. Для стороннього спостерігача такі дані залишаються нерозбірливими або нечитабельними, навіть якщо вони будуть перехоплені.

У загальному вигляді шифрування передбачає перетворення даних із доступного для читання формату в шифрований вигляд, аби унеможливити ознайомлення з ними сторонніх осіб під час передавання по мережі. Цей процес може застосовуватись до різних типів інформації — документів, повідомлень, файлів або інших цифрових об’єктів, які передаються в мережевому середовищі.

Шифрування відіграє ключову роль у забезпеченні конфіденційності та цілісності даних, і його значення в умовах сучасного цифрового світу є надзвичайно важливим. Сьогодні більшість онлайн-ресурсів і програм використовують шифрування на різних рівнях, забезпечуючи таким чином захист користувачів і даних у процесі передачі та зберігання. Хоча історично

практика шифрування відома ще з часів Давнього Риму, сучасне розуміння цього терміну пов’язується переважно з електронною обробкою інформації.

У цифрових системах шифрування здійснюється шляхом застосування алгоритмів визначених наборів інструкцій до певних блоків даних. Процес керується ключем шифрування, відомим лише відправнику та одержувачу повідомлення. Цей ключ забезпечує унікальність шифрованого тексту та гарантує, що лише власник відповідного ключа зможе виконати розшифрування.

Наявність ключа є критично важливою складовою криптографічного захисту: навіть найскладніший шифр без змінних ключів може бути розкритий, якщо буде відомий сам алгоритм. Таким чином, безпека шифрування залежить не лише від складності алгоритму, але й від надійності та секретності ключів, які використовуються у процесі шифрування та розшифрування даних.

У минулому шифрувальні засоби зазвичай обмежувалися лише обробкою даних, не включаючи механізми перевірки достовірності чи цілісності. Це призводило до вразливості систем, оскільки навіть за наявності зашифрованого повідомлення не було гарантії, що його не підробили або не змінили в процесі передавання.

У сфері криптології часто виникає плутанина між поняттями "код" і "шифр", які помилково вживають як синоніми навіть досвідчені фахівці. Проте між ними існує фундаментальна відмінність.

Код (code) це стала система відповідностей, яка замінює певні одиниці інформації (наприклад, слова, фрази, символи) іншими, не обов’язково того ж самого типу. Головне, така заміна відбувається за наперед визначеним і публічним правилом, без прив’язки до секретного ключа.

Як і коди, шифри замінюють частини відкритого тексту (це можуть бути окремі символи, слова або рядки) іншими елементами. Проте ключова відмінність полягає в тому, що така заміна виконується згідно з чітким правилом, визначеним секретним ключем. Цей ключ відомий лише відправнику та уповноваженому отримувачу, і передбачається, що сторонній

користувач, не володіючи ключем, не зможе здійснити зворотне перетворення та прочитати зашифровану інформацію.

Шифрування є одним з ключових інструментів забезпечення інформаційної безпеки, зокрема при передаванні даних у відкритих або потенційно небезпечних середовищах, таких як мережі Інтернету речей (IoT), хмарні сервіси тощо.

Однією з головних функцій шифрування є аутентифікація джерела інформації, тобто підтвердження того, що дані дійсно були надіслані конкретним відправником. Крім того, шифрування унеможливлює відмову від авторства: відправник не може заперечити факт передачі повідомлення, що має важливе значення для цифрового підпису та судових доказів.

Для читання зашифрованої інформації приймаючій стороні необхідні, ключ (секретна або відкрита інформація, яка використовується у криптоалгоритмі), дешифратор програмне або апаратне забезпечення, що реалізує алгоритм розшифрування.

Основною ідеєю шифрування є те, що зловмисник, що перехоплює зашифровані дані, не зможе їх прочитати або змінити без наявного відповідного ключа доступу. Таким чином, саме шифрування забезпечує як конфіденційність так і цілісність особистих даних.

У сучасних криптосистемах широко використовуються два типи ключів, симетричне шифрування, той самий ключ використовується для шифрування і дешифрування. Асиметричне шифрування (системи з відкритим ключем) для шифрування використовується відкритий ключ, а для дешифрування відповідний закритий ключ. Це дозволяє безпечно обмінюватися зашифрованими даними навіть між невідомими сторонами.

Проте розвиток криптоаналізу призвів до появи методів, які дають змогу дешифрувати дані без наявності ключа. Ці методи базуються на математичному аналізі структури зашифрованого тексту, властивостей алгоритму шифрування, частотного аналізу та інших статистичних і евристичних підходів. Тому криптографія є динамічною галуззю, де безпека алгоритмів постійно переглядається і вдосконалюється у відповідь на нові виклики.

Є декілька сучасних алгоритмів шифрування, такі як RSA, AES, ECC, які забезпечують шифрування та цілісність даних. Алгоритм асиметричного шифрування RSA був названий на честь його розробників Рона Рівеста (Ron Rivest), Аді Шаміра (Adi Shamir) та Леонарда Едлмана (Leonard Adleman). Вони створили цей криптографічний метод у 1977 році, що стало проривом у сфері захисту цифрової інформації. Пізніше, у 1982 році, ці троє заснували компанію RSA Data Security, яка спеціалізувалась на впровадженні й комерціалізації рішень на основі RSA-алгоритму.

З того часу компанія змінила кількох власників: спочатку перейшла під контроль EMC, а нині входить до структури Dell Technologies. В умовах постійного зростання кіберзагроз, підприємці та IT-фахівці усвідомлюють важливість впровадження надійних систем безпеки для захисту веб-ресурсів і транзакцій від зловмисників, які постійно вдосконалюють свої методи атак.

Для захисту інтернет-простору відповідальні бізнеси все частіше звертаються до використання SSL-сертифікатів, які забезпечуються перевіреними центрами сертифікації (CA). Такі сертифікати дозволяють автентифікувати веб-сервер, шифрувати конфіденційну інформацію під час передавання та забезпечувати користувачам впевненість у безпеці взаємодії з веб-сайтами.

RSA, незважаючи на свій вік, залишається одним з найпоширеніших і ефективних алгоритмів для шифрування даних. Проте з розвитком обчислювальних технологій виникла потреба у збільшенні довжини криптографічного ключа, що безпосередньо впливає на стійкість системи до атак. У зв’язку з цим Національний інститут стандартів і технологій США (NIST) ще у 2013 році рекомендував припинити випуск нових SSL/TLS-сертифікатів з ключами RSA менше ніж 2048 біт, визнавши їх недостатньо захищеними.

Паралельно, для посилення безпеки, федеральні структури США почали впроваджувати альтернативні криптографічні алгоритми, зокрема, ECC (Elliptic Curve Cryptography), алгоритм, що використовує еліптичні криві. Він забезпечує такий самий рівень безпеки, як RSA, але з меншим розміром ключа, що робить його ідеальним вибором для мобільних та ресурсозалежних

середовищ, DSA (Digital Signature Algorithm),цифровий підпис, часто використовуваний в державному секторі, особливо серед підрядників і субпідрядників, що працюють з урядом.

RSA належить до асиметричних алгоритмів шифрування, тобто в ньому застосовуються різні ключі для шифрування і дешифрування. Це дозволяє реалізовувати безпечні механізми обміну ключами, автентифікацію та цифрові підписи, що є важливими складовими сучасної кібербезпеки.

У криптографічних системах з відкритим ключем один із ключів призначений для вільного розповсюдження та відомий усім це відкритий ключ. Інший ключ, закритий, зберігається у таємниці та належить лише власнику. Важливо, що кожен з ключів виконує операції лише в одному напрямку: зашифровані відкритим ключем дані можуть бути розшифровані лише за допомогою відповідного закритого ключа, і навпаки. При цьому, за умови великої довжини ключів, обчислення одного ключа на основі іншого є практично неможливим.

Основою безпеки алгоритму RSA є складність розкладання добутку двох великих простих чисел. Саме це розкладання лежить в основі обчислення функції Ейлера від модуля N. Процес шифрування в RSA реалізується шляхом піднесення повідомлення до певного степеня за модулем N, а для розшифрування необхідно знати значення функції Ейлера, яке можливо отримати лише при знанні простих множників числа N. Саме ця задача, факторизація, і забезпечує криптостійкість алгоритму. У криптографічному алгоритмі RSA кожна пара ключів, відкритий і закритий, складається з певного набору чисел. Закритий ключ (private key) зберігається у таємниці та ніколи не передається третім сторонам, у той час як відкритий ключ (public key), навпаки, може бути опублікований або переданий іншим користувачам для здійснення шифрування або перевірки підпису.

Багато криптографічних протоколів, зокрема Secure Shell (SSH), OpenPGP, S/MIME та SSL/TLS, використовують RSA для реалізації шифрування і цифрових підписів. Алгоритм також широко застосовується у програмному забезпеченні, наприклад, у веббраузерах, яким потрібно

встановлювати захищені з’єднання через відкриті мережі (як-от Інтернет) або перевіряти цифрові підписи.

Операція перевірки підпису RSA є однією з найпоширеніших у мережевих системах. Безпека алгоритму базується на складності розкладання великих чисел на прості множники. Проте, з розвитком обчислювальних потужностей і нових методів факторизації зростає ризик компрометації.

Надійність шифрування прямо залежить від довжини ключа: збільшення довжини ключа, наприклад удвічі, значно підвищує стійкість, хоча й впливає на продуктивність. Стандартна довжина ключів RSA – 1024 або 2048 біт, однак 1024-бітні ключі вже вважаються недостатньо безпечними.

Щодо алгоритму AES, окремі етапи включають, ShiftRows - процедура циклічного зсуву байтів у рядках блоку стану (State), перший рядок не зміщується, другий зміщується на 1 байт, третій - на 2, четвертий - на 3. При шифруванні зсув відбувається вліво, а при розшифруванні - вправо. MixColumns – змішування байтів у кожному стовпці блоку стану. AddRoundKey – побітове додавання по модулю 2 (XOR) між байтами стану та байтами раундового ключа; ця операція є ідентичною як для шифрування, так і для розшифрування завдяки властивостям XOR.

IPSec, як і будь-який інший протокол безпеки, має свої вразливості, які можуть бути використані зловмисниками для атак. Однією з основних загроз є атаки на ключі шифрування. Ключі шифрування є критичним елементом безпеки IPSec, і атаки на них можуть включати методи, такі як груба сила (перебір всіх можливих комбінацій ключів), крипто аналітичні атаки (аналіз алгоритмів шифрування з метою знаходження слабких місць) та соціальна інженерія (отримання ключів через вплив на людей).

Використання слабких або коротких ключів робить систему вразливою до таких атак, тому регулярна зміна ключів та використання довших ключів допомагають знизити цей ризик. Атаки типу людина по середині (Man-in-the-Middle, MitM) також є значною загрозою. У подібних типах атак зловмисник може втручатися в комунікацію між двома сторонами, змінюючи або підміняючи передану інформацію. Це можливо, зокрема, через фальсифікацію сертифікатів або використання вразливостей в протоколах аутентифікації.

Особливо ризикованою є початкова фаза встановлення з’єднання (фаза 1 IKE), яка, якщо належним чином не захищена, може дозволити зловмиснику встати по середині каналу обміну даними. Така атака може призвести до порушення конфіденційності або цілісності інформації.

Додаткову небезпеку становлять помилки в програмному забезпеченні, що реалізує IPSec. Вони можуть містити баги в реалізації криптографічних алгоритмів, помилки в обробці мережевих пакетів або інші логічні дефекти. Хоча відкриті реалізації часто проходять ретельні перевірки спільнотою, вони одночасно стають доступнішими для зловмисників, які можуть вивчати код на предмет вразливостей.

Для зменшення ризиків слід дотримуватись кращих практик безпеки. Зокрема, регулярно оновлювати IPSec-програмне забезпечення, щоб отримувати патчі для виявлених уразливостей, використовувати сучасні криптоалгоритми, такі як AES, замість застарілих, наприклад, DES, застосовувати сертифікати, видані авторитетними центрами сертифікації (CA), періодично змінювати ключі шифрування для зниження ризику компрометації.

Контроль доступу та аутентифікація включають впровадження двофакторної аутентифікації (2FA), яка додає додатковий рівень захисту, вимагаючи підтвердження особи за допомогою іншого пристрою або методу. Для первинної аутентифікації часто використовуються попередньо узгоджені ключі (PSK), що допомагає запобігти несанкціонованому доступу до мережі.

Моніторинг мережі та аналіз журналів дають змогу своєчасно виявляти підозрілу активність і реагувати на потенційні загрози. Системи виявлення та запобігання вторгненням (IDS/IPS) дозволяють у режимі реального часу відстежувати нетипову поведінку та загрози в мережевому трафіку. Регулярний перегляд логів IPSec і обладнання сприяє виявленню аномалій на ранніх етапах.

Додаткові рівні захисту включають фізичні заходи безпеки, що захищають обладнання від несанкціонованого фізичного доступу, сегментацію мережі для обмеження поширення потенційних атак, а також використання брандмауерів та VPN, що захищають трафік на всіх рівнях.

Використання IPSec є ключовим компонентом у забезпеченні безпеки мережевих з’єднань. Хоча протокол може впливати на продуктивність через додаткові обчислювальні навантаження, його здатність гарантувати конфіденційність, цілісність та автентичність переданих даних є безсумнівною перевагою.

Щоб мінімізувати вплив IPSec на швидкодію, можна застосовувати апаратне прискорення шифрування та оптимізувати параметри протоколу. Також важливо враховувати потреби організації при виборі між IPSec та альтернативними протоколами, такими як GRE, PPTP або SSL/TLS, які мають свої переваги й обмеження.

Належна конфігурація, адміністрування IPSec і впровадження додаткових заходів безпеки дозволяють досягти надійного захисту даних в умовах постійно зростаючих кіберзагроз.

Процедура шифрування виконується послідовно в декілька етапів, які залежать від довжини ключа. Для 128-бітного ключа AES-процес включає 10 раундів. На початковому етапі – ініціалізації – до блоку стану (State) застосовується операція AddRoundKey() із початковим ключем користувача. Після цього слідує 9 раундів, у кожному з яких виконуються операції, SubBytes() – заміна байтів на основі фіксованої таблиці (S-box), ShiftRows() – зсув рядків у матриці стану, MixColumns() – змішування стовпців для забезпечення дифузії, AddRoundKey() – додавання ключа, сформованого на основі початкового.

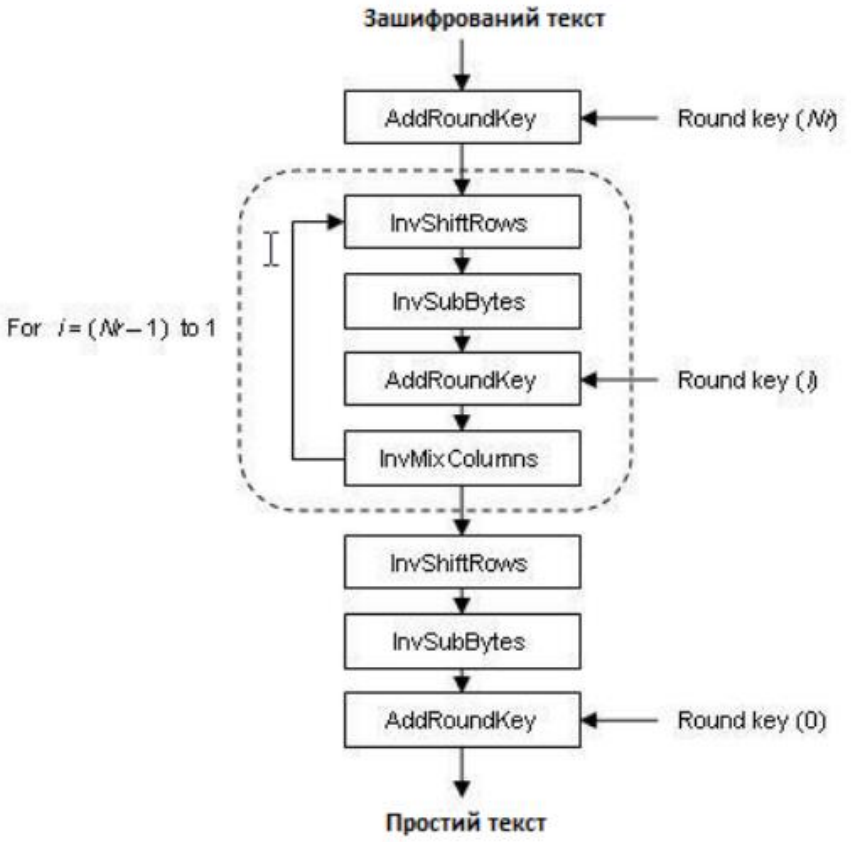


Рисунок 2.1 – Алгоритм шифрування AES

На завершальному етапі шифрування за алгоритмом AES (Advanced Encryption Standard) виконується так званий неповний раунд, який включає лише три операції: SubBytes(), ShiftRows() та AddRoundKey(). Саме після цього кроку формується остаточне зашифроване повідомлення, яке вже не підлягає подальшим перетворенням у межах даного шифрувального процесу.

Процедура дешифрування (розшифрування) в рамках AES виконується у зворотному порядку — тобто кожна з операцій, що застосовувалась під час шифрування, замінюється на її обернену. Наприклад, для SubBytes() використовується InvSubBytes(), для ShiftRows(), InvShiftRows() тощо. Такий підхід дозволяє повністю відновити початковий текст з зашифрованого блоку даних, за умови наявності відповідного ключа.

Сьогодні AES вважається одним з найбільш надійних та ефективних симетричних алгоритмів шифрування. Він масово впроваджений у сучасних комп’ютерних системах і підтримується як на рівні програмного забезпечення (через бібліотеки OpenSSL, CryptoAPI, тощо), так і на рівні апаратного

забезпечення (CPU з інструкціями AES-NI). До сьогодні не існує жодної практичної криптоаналітичної атаки, здатної зламати AES при дотриманні стандартних умов використання.

Крім високої стійкості, AES відзначається гнучкістю у виборі довжини ключа, 128, 192 або 256 біт. Ця властивість робить алгоритм стійким до майбутніх загроз, включаючи ті, що можуть виникнути у зв'язку з розвитком обчислювальної потужності або новими методами атаки (зокрема квантовими).

Ще одним важливим алгоритмом у криптографії є Triple DES (3DES), офіційно відомий як Triple Data Encryption Algorithm (TDEA або Triple DEA). Це симетричний блоковий шифр, який був створений на основі старішого алгоритму DES (Data Encryption Standard). Його розробили Уітфілд Діффі (Whitfield Diffie), Мартін Хеллман (Martin Hellman) та Уолт Тачман (Walt Tuchman) у 1978 році.

Triple DES застосовує DES-алгоритм тричі поспіль із трьома різними ключами, що значно підвищує рівень безпеки порівняно з оригінальним DES. Однак згодом 3DES був витіснений AES через нижчу продуктивність та більший обсяг обчислень при аналогічному рівні безпеки.

При шифруванні можливе використання кількох варіантів вибору ключів, які розрізняються за рівнем криптостійкості (в порядку спадання), три різні ключі, найнадійніший варіант, оскільки кожен етап шифрування (шифрування–дешифрування–шифрування, або EDE) виконується з унікальним ключем. У випадку алгоритму DES, де довжина одного ключа становить 56 біт (по 7 біт на байт), загальна довжина ключа в Triple DES (3DES) при такому варіанті становить 168 біт, перший і третій ключі однакові, другий, інший криптографічно менш стійкий, оскільки ефективна довжина ключа знижується до 112 біт. Проте цей варіант вважається безпечнішим за просте подвійне шифрування DES, оскільки послідовність шифрування–дешифрування–шифрування ускладнює реалізацію атак типу «зустріч посередині». Вставка операції дешифрування посередині порушує лінійність, що заважає ефективному створенню таблиць відповідностей і пошуку колізій, усі ключі однакові, має таку ж криптостійкість, як і звичайний DES, тобто

лише 56 біт. Це найненадійніший варіант, який на практиці майже не застосовується.

Для зберігання DES-ключів використовується представлення у вигляді 8 байтів із додаванням біта паритету, отже, загальний об’єм даних для ключів становить, 24 байти, у випадку трьох різних ключів, 16 байтів, при двох різних ключах, 8 байтів, при однакових ключах.

Triple DES (3DES) у конфігурації з трьома ключами широко використовується в мережевих застосунках, таких як PGP та S/MIME, а також в стандартах управління ключами, зокрема ANSI X9.17, ISO 8732 і PEM (Privacy Enhanced Mail). У галузі електронних платежів 3DES залишається актуальним, і його підтримка закладена в таких стандартах, як EMV.

Незважаючи на те, що 3DES з трьома унікальними ключами вважається стійким до 2030 року, його використання поступово скорочується через поширення більш сучасного та ефективного алгоритму — AES (Rijndael). Програмна реалізація AES приблизно в шість разів швидша за 3DES, що зумовлює його перевагу у програмних застосуваннях.

Тому 3DES найчастіше використовується в апаратних рішеннях, хоча багато сучасних систем безпеки досі підтримують як AES, так і 3DES, переважно з міркувань зворотної сумісності. У сучасних умовах застосування 3DES обмежується специфічними завданнями: зберігання незворотних паролів, перевірка цілісності даних, цифрові підписи, а також автентифікаційні протоколи. Проте його використання вже не рекомендується для нових систем (рисунок 2.2).

Рисунок 2.2 – Блок-схема шифрування AES

Додавання по модулю необхідно розрізняти з простим додаванням булевих змінних, що відповідає логічній операції «або» (диз'юнкції). У термінах теорії множин цією операцією є симетрична різниця множин істинності операндів.

У процесі реконфігурації функції SubBytes було впроваджено нову операцію, яка замінила початкову - під назвою Modified Transport. Відтак функція SubBytes оновлюється і набуває назви ModSubBytes.

Далі дані, оброблені у ModSubBytes, надходять до моменту, коли значення S-Box ще не були змінені. Масив стану поділяється на дві частини по 4 біти кожна (для кожного 8-бітного значення), після чого виконується передача або обмін частин для формування нового стану в процесі транспортування (рисунок 2.3).

Рисунок 2.3 – Інтегрований процес застосування шифрування

У практичному застосуванні, коли між сторонами відбувається інтенсивний обмін великими обсягами даних, схему шифрування можна оптимізувати. Зокрема, обмін AES-ключами здійснюється за допомогою RSA через регулярні інтервали часу. Після обміну ключами кожна сторона використовує отриманий AES-ключ іншої сторони для шифрування переданих даних.

Процес виглядає так: кожна зі сторін генерує власний AES-ключ, шифрує його відкритим ключем RSA іншої сторони та надсилає одержувачу. Отримувач розшифровує отриманий AES-ключ за допомогою свого приватного RSA-ключа та зберігає його для подальшого використання. Особливість цієї схеми полягає в тому, що обидві сторони самостійно створюють пари ключів та обмінюються лише відкритими RSA-ключами.

Протягом визначеного періоду обидві сторони використовують отриманий AES-ключ один одного для шифрування своїх повідомлень. Таким чином, кожен зашифрований блок даних розшифровується стороною-одержувачем за допомогою відповідного AES-ключа, який вона раніше зберегла.

Ця схема передбачає періодичне оновлення AES-ключів, що значно знижує ризик їх компрометації. Регулярна заміна ключів підвищує безпеку системи та зменшує ймовірність витоку чутливої інформації.

На сучасному етапі розвитку технологій Інтернету речей існує велика кількість рішень, орієнтованих на вирішення окремих аспектів функціонування IoT-систем – зокрема, енергоефективності або інформаційної безпеки. Проте більшість із них фокусуються лише на одному з аспектів, не враховуючи необхідність одночасного досягнення балансу між енергоспоживанням та рівнем захищеності пристроїв.

Розберемо рішення, орієнтовані на енергоефективність, протоколи LoRaWan, ZigBee, BLE, TLS/SSL шифрування, протокол DTLS, системи виявлення вторгнень IDS, та комбіновані рішення, які прагнуть врахувати одночасно енергоефективність та безпеку, до прикладу, фреймворк на основі Lightweight криптографії, які призначені для мікроконтролерів, та мають знижені вимоги до енергоспоживання. Адаптивні протоколи, перемикаються між режимами залежно від умов, наприклад, використовують шифрування тільки при підозрі на загрозу або активну передачу даних.

TLS/SSL шифрування, має такі переваги як, надійний захист даних при передачі, широко підтримується серверами й платформами. Обмеженнями ж ж значене навантаження на процесор пристрою, велике споживання енергії, тривалий час установлення з’єднання.

Перевагами протоколу DTLS є те, що це легка версія TLS, адаптована під UDP, сумісна з constrained devices. Обмеженнями є те, що протокол досі вимагає значних обчислювальних ресурсів та не підтримується всіма платформами.

Системи виявлення вторгнень IDS, мають здатність виявляти аномальну поведінку, захист від атак на прикладному рівні. Обмеженнями є потреба в постійному аналізі трафіку, що збільшує навантаження на пристрій та підвищене енергоспоживання.

Аналіз сучасних рішень демонструє, що жодне з них не забезпечує комплексного підходу до одночасного управління енергоефективністю та інформаційною безпекою IoT-пристроїв. У зв’язку з цим виникає потреба у

створенні нової адаптивної методики, яка дозволить досягти оптимального співвідношення між автономністю системи та її стійкістю до загроз.

2.3 Висновки до розділу 2

У другому розділі було проведено системний аналіз сучасних підходів до оптимізації енергоспоживання та забезпечення інформаційної безпеки в системах IoT. Розглянуто основні методи енергозбереження, включаючи використання енергоефективних протоколів зв’язку, режимів сну мікроконтролерів та зменшення частоти передавання даних. Окрему увагу приділено технікам криптографічного захисту, системам виявлення вторгнень, що використовуються в сфері IoT.

На підставі аналізу виявлено, що більшість існуючих рішень реалізують односторонню оптимізацію, яка спрямована або на зменшення енергоспоживання або на підвищення безпеки, при цьому майже не розглядається питання інтеграції цих двох аспектів у єдину систему, яка забезпечувала б динамічний баланс між ефективністю та захищеністю пристроїв у змінному середовищі..

Таким чином було зроблено висновок про доцільність розробки комплексного підходу, що враховуватиме обидва критичних чинники, таких як, енергетична автономність та захист даних, у процесі функціонування IoT систем. Це обґрунтовує наступний етап дослідження, який присвячений створенню нового методу оптимізації.

3 РОЗРОБКА МЕТОДУ КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ IOT-СИСТЕМ

3.1 Протоколи та стандарти забезпечення безпеки в IoT-системах

У попередніх розділах було визначено, що сучасні IoT системи стикаються з необхідністю одночасного забезпечення високої енергоефективності та інформаційної безпеки. Водночас більшість існуючих рішень орієнтуються переважно на одну з цих задач, яка призводить до дисбалансу функціонування пристроїв у реальному середовищі.

Інтернет речей, це середовище де автономні пристрої взаємодіють між собою, збирають, аналізують та передають дані. Важливою особливістю більшості IoT рішень є обмежені ресурси, мова йде як про обмежену потужність обчислень, так і про незначний запас енергії, особливо коли пристрої працюють від батарейок або сонячних панелей, одночасно з цим ці пристрої передають потенційно чутливу інформацію, що створює серйозні виклики у сфері безпеки.

Традиційні підходи фокусуються або на економії енергії або на забезпеченні захисту. Такий підхід призводить до дисбалансу, коли зменшення одного показника погіршує інший.

Метою даного розділу є створення методу який забезпечить баланс між споживанням енергії та рівнем захищеності, адаптивну реакцію на зміну умов, універсальність, можливість реалізації на доступному апаратному рівні.

Процес управління даними, зібраними за допомогою IoT-пристроїв, поділяється на кілька послідовних етапів. Такий розподіл дозволяє забезпечити простоту, повноту та масштабованість функціонування систем, побудованих за відповідною багаторівневою структурою. Запропонована архітектура сприяє розширенню можливостей у контексті відбору, керування та аналізу даних в екосистемі Інтернету речей (IoT).

Інформаційно-технологічна платформа, сформована на основі цієї структури, включає дев’ять функціональних рівнів, рівень відбору даних, рівень туманних обчислень, рівень управління цілісністю, рівень безпеки,

рівень агрегування даних, рівень аналітичного опрацювання, рівень зберігання даних, рівень застосунків, рівень архівування.

Кожен із рівнів виконує специфічну роль у загальному процесі управління даними та взаємодіє з іншими рівнями для забезпечення безперервного й ефективного функціонування платформи.

Початковим є рівень відбору даних, що здійснює первинне захоплення та маршрутизацію інформації, яка надходить із численних, переважно гетерогенних джерел. Цей рівень відіграє ключову роль у взаємодії з фізичними IoT-пристроями, що використовуються для моніторингу, вимірювання та фіксації параметрів у різноманітних середовищах.

Серед типових пристроїв, задіяних на цьому етапі, можна виділити сенсори, розумні пристрої, RFID-мітки, носимі пристрої, сканери штрих-кодів, а також відеоспостережні системи. Дані, зібрані з цих пристроїв, можуть мати різні форми та формати, що вимагає відповідної попередньої обробки перед передачею на наступні рівні обробки та зберігання (рис. 3.1).

Залежно від особливостей програмно-алгоритмічних рішень, процес відбору даних у системах Інтернету речей (IoT) може здійснюватися централізовано (через головні вузли або сервери) або розподілено (на рівні окремих пристроїв або локальних вузлів). Цей рівень виконує функцію вхідного шару, який передає оброблені або відібрані дані до наступного рівн туманних обчислень (fog computing), рівень туманних обчислень (Fog Computing) Fog computing є критично важливим компонентом сучасної IoT-архітектури. Його мета, мінімізувати затримки та навантаження на хмарну інфраструктуру, шляхом локальної обробки даних максимально близько до джерела їхнього виникнення, тобто безпосередньо біля сенсорів або вбудованих пристроїв. Основні переваги туманних обчислень, швидке реагування в реальному часі, зменшується час на прийняття рішень, зниження обсягу переданих до хмари даних, економія пропускної здатності, енергозбереження та зменшення навантаження на канали зв’язку, можливість попереднього фільтрування, агрегації та аналізу інформації на локальному рівні.

У запропонованій архітектурній моделі рівень туманних обчислень виступає як проміжна ланка між базовими сенсорними вузлами та хмарною інфраструктурою, виконуючи попередню обробку та зберігання даних. Він забезпечує контекстну обробку інформації, яка надходить із пристроїв, знижуючи потребу в постійному з'єднанні з центральними серверами.Багаторівнева модель управління IoT-даними

Уся система обробки інформації з IoT-пристроїв розглядається як багаторівнева модель, кожен рівень якої виконує специфічні функції, рівень відбору, первинне збирання, попередня фільтрація та класифікація даних, рівень туманних обчислень, локальна обробка, агрегація та часткове зберігання інформації, рівень забезпечення цілісності та безпеки — контроль достовірності даних, шифрування, автентифікація пристроїв, рівень агрегації об'єднання даних з різних джерел для подальшого аналізу, аналітичний рівень використання методів машинного навчання та штучного інтелекту для аналізу даних, централізоване зберігання, передача відібраних і оброблених даних до хмари або центрів обробки, рівень застосування, використання даних для надання сервісів користувачам, рівень архівування, довготривале зберігання історичних даних для подальшого використання чи аналізу.

Така архітектура дозволяє оптимально розподіляти навантаження, зменшити затримки, знизити витрати на передачу та обробку інформації та забезпечити високу надійність і масштабованість системи.

Саме з архівування починається фінальний етап життєвого циклу даних у системах, побудованих на основі IoT, що забезпечує їх доступність та захист у довготривалій перспективі:

1. Технічне шифрування;
2. Рівень архівування;
3. Тимчасове архівування;
4. Постійне архівування.

Далі йде менеджер контролю:

1. Контроль доступу;
2. Рівень доступу;
3. Менеджер завантажень;

4. Менеджер інформаційних панелей;
5. Менеджер якості;
6. Менеджер завантажень.

Далі менеджери вхідних опрацювань та технічних систем:

1. Менеджер ідентифікації;
2. Рівень опрацювання даних;
3. Менеджер рішень;
4. Менеджер трендів;
5. Менеджер розумних інструментів;
6. Трастовий менеджер;
7. Рівень управління цілісністю;
8. Менеджер мобільності;
9. Зберігання необроблених даних.

Далі рівні туманих обчислень:

1. Рівень аутентифікації;
2. Рівень агреції даних;
3. Менеджер агрегування;
4. Менеджер неоднорідності;
5. Константний менеджер;
6. Менеджер фільтрування.

Далі безпека та пристрої системи IoT:

1. Безпека фізичних пристроїв;
2. Рівень туманих обчислень;
3. Рівень відбору даних;
4. IoT пристрої;
5. Давачі, сенсори, носимі пристрої, спростережувані пристрої.

Для забезпечення ефективного управління даними на низькорівневих вузлах пристрої мають володіти необхідними ресурсами, такими як обчислювальна потужність, пам’ять, тривалий час автономної роботи та ієрархічна структура взаємодії. Лише критично важливі з точки зору часу дані обробляються безпосередньо на пристроях, тоді як інші передаються до

хмарної платформи для тривалого зберігання й глибокої аналітики. Рівень управління цілісністю.

Цей рівень відповідає за підтримання цілісності даних у процесі їх збирання, зберігання та обробки. До його основних компонентів належать. Модуль зберігання необроблених даних – зберігає великі обсяги первинних даних до моменту подальшої обробки. Він використовує інструменти для індексації, управління метаданими та оптимального розміщення інформації. Менеджер мобільності – координує передачу даних у контексті мобільності IoT-пристроїв. Він враховує зміни в контексті переданих даних через переміщення пристроїв і забезпечує безперервність зв’язку через підтримку мобільності сервісів, сесій і користувачів.

Забезпечення автентичності, цілісності та доступності даних на цьому рівні допомагає зменшити накладні витрати та зберегти якість аналітики. Рівень агрегування даних. Цей рівень спрямований на зменшення обсягу даних і підвищення ефективності їх зберігання, обробки та передачі в реальному часі. Основні модулі. Менеджер фільтрації – визначає правила фільтрації даних та виконує їхнє первинне очищення. Фільтрація зменшує шум і виділяє важливу інформацію. Вона може бути тимчасовою, постійною або залежати від частоти запитів.

Менеджер неоднорідності – вирішує проблеми, пов’язані з різноманітністю пристроїв, форматів даних і семантичних відмінностей. Він нормалізує типи даних і перетворює їх у стандартизовані формати для подальшого аналізу. Менеджер трансформацій – відповідає за приведення даних у зручний для користувача формат. На цьому етапі виконуються операції сортування, розділення та об'єднання даних для покращення їх доступності.

Хоча система IoT об’єднує різнорідні пристрої, якісна організація їх взаємодії є вирішальною для забезпечення надійного зв’язку між пристроями та сервісами. Структурований підхід до обробки даних дозволяє не лише підтримувати цілісність та якість інформації, а й забезпечувати ефективне використання ресурсів у реальному часі.

У зв’язку зі зростаючою складністю систем Інтернету речей (IoT) виникає нагальна потреба у впровадженні чітко структурованих стандартів, які регламентуватимуть представлення пристроїв, механізми їхнього виявлення та забезпечення доступу до них. Це дозволить досягти ефективності та уніфікації в управлінні різнорідними елементами IoT-екосистеми.

З метою забезпечення сумісності пристроїв, які можуть належати до різних виробників, потрібно розробити та впровадити загальні принципи взаємодії. Ці принципи повинні охоплювати стандарти обміну даними між IoT-пристроями, незалежно від їх технічного походження чи функціонального призначення. У межах цієї взаємодії важлива роль належить менеджеру сумісності, на якого покладаються завдання забезпечення технічної (фізичної та логічної), семантичної (смислової), синтаксичної (структурної) та міждоменної (галузевої) сумісності між різними компонентами системи.

Важливою особливістю IoT є залежність ефективності його роботи від збору та обробки контекстних даних. Тобто інформації, яка має сенс лише в певному середовищі чи за певних умов. Саме тому до системи впроваджується контекстний менеджер, інтелектуальний модуль, що відповідає за виявлення та відбір пристроїв, які в даний момент часу можуть забезпечити найбільш актуальні, точні й релевантні дані, потрібні для функціонування користувацьких застосунків.

Контекстний менеджер не лише ідентифікує джерела даних, але й формує повноцінний опис середовища, в якому функціонують пристрої. Він аналізує параметри контексту окремих IoT-пристроїв або їх об’єднань, які здійснюють збір інформації у режимі реального часу. Забезпечення підтримки чітко визначених специфікацій контексту дозволяє створювати інноваційні сервіси, що адаптовані до потреб користувачів і здатні оперативно реагувати на зміни в середовищі.

Це дає змогу забезпечити легку, швидку та ефективну доставку контекстних даних безпосередньо до кінцевого користувача. Така доставка відбувається завдяки функціонуванню спеціального компонента, який відповідає за представлення поточного контексту, його структурування та моделювання ситуацій. У цьому процесі особливу роль відіграє контекстний

менеджер, завданням якого є не лише обробка наявних контекстів, але й виявлення нових на основі попередньо зібраних даних.

Контекстний менеджер має бути здатним до динамічного аналізу історичних даних для побудови нових контекстуальних моделей, які дозволяють системі адаптуватись до змін середовища та запитів користувачів. Після завершення цього етапу дані переходять на рівень агрегування, де їх обробляє менеджер агрегації.

Менеджер агрегації здійснює структурування та перетворення даних у вигляд, зручний для подальшої аналітичної обробки. Для цього він застосовує різні методи та інструменти, які дозволяють перевірити точність даних, очистити їх від шумів та помилок, а також об’єднати у логічно завершені інформаційні блоки. Цей менеджер також відповідає за відбір лише тих даних, які відповідають попередньо встановленим критеріям якості, релевантності та форматування.

З метою гарантування цілісності та надійності всієї системи, в архітектурі передбачений спеціальний рівень безпеки. Його основне завдання, забезпечити автоматизований захист усіх процесів, що відбуваються в рамках Інтернету речей. У межах запропонованої структурної моделі цей рівень охоплює всі інші рівні системи, забезпечуючи безпечне управління даними на кожному етапі їх життєвого циклу, від збору та обробки до зберігання, передачі та використання.

Безпековий рівень створено для комплексного задоволення вимог до безпеки, з урахуванням особливостей кожного рівня IoT-системи. Його впровадження гарантує надійний захист як окремих пристроїв, так і всієї платформи в цілому.

Враховуючи функціональні особливості кожного з рівнів IoT-моделі, рівень безпеки забезпечує гнучке й адаптивне впровадження відповідних захисних механізмів. Його структура побудована таким чином, щоб охопити всі можливі вектори загроз і гарантувати стабільну, безпечну роботу системи. До основних складових цього рівня належать, трастовий менеджер, забезпечує управління довірою до пристроїв на основі аналізу їхньої поведінки та історії переданих даних. Менеджер автентифікації, відповідає за підтвердження

достовірності користувачів і пристроїв перед наданням доступу до ресурсів, менеджер ідентифікації, виконує функцію присвоєння унікального ідентифікатора кожному елементу системи, що забезпечує прозорість джерел даних, засоби фізичної безпеки пристроїв, запобігають несанкціонованому фізичному доступу до пристроїв або втраті їх функціональності, методи шифрування, гарантують захист інформації під час її передачі та зберігання, запобігаючи витоку конфіденційних даних, менеджер конфіденційності, визначає політики захисту особистої інформації користувачів та забезпечує їх дотримання, компоненти контролю доступу — регулюють права на використання даних і ресурсів, ґрунтуючись на встановлених ролях та дозволах.

Рівень цілісності даних тісно пов’язаний з модульною архітектурою безпеки і підтримується завдяки таким критично важливим модулям, як ідентифікація, автентифікація та система довіри. Забезпечення точності, достовірності та актуальності джерел даних є однією з ключових задач у сфері управління IoT-системами. Кожен пристрій, який бере участь у зборі або передачі інформації, має отримати унікальний ідентифікатор, що дозволяє його відстеження, ідентифікацію та контроль.

Менеджер ідентифікаторів відповідає за обробку та збереження даних про кожен пристрій, забезпечуючи їх інтеграцію у процес автентифікації, яку виконує відповідний модуль. Надалі довіра до кожного пристрою визначається за допомогою трастового менеджера. Цей компонент оцінює рівень надійності на основі таких параметрів, як точність переданої інформації, її повнота та відповідність часовим рамкам. Важливо, що ці рівні довіри оновлюються регулярно — менеджер проводить повторні розрахунки, враховуючи поточну активність пристрою, що забезпечує динамічну адаптацію до зміни його поведінки.

На прикладному рівні безпека підтримується шляхом використання менеджерів доступу та конфіденційності. Перший відповідає за контрольований розподіл прав доступу до даних і функцій системи, другий, за формування та реалізацію політики конфіденційності застосунків. Завдяки

їхній взаємодії забезпечується надійний захист кінцевих користувачів від витоку або несанкціонованого використання персональної інформації.

Менеджер конфіденційності виконує важливу функцію — захист кінцевих користувачів від ризиків, пов’язаних із розкриттям персональної або чутливої інформації. Він впроваджує механізми, які дозволяють визначити, яким чином збираються, обробляються, зберігаються й використовуються дані користувача. Ці механізми включають налаштування політик конфіденційності, які враховують як вимоги законодавства, так і індивідуальні переваги користувачів, забезпечуючи прозорість та безпечну взаємодію з IoT-застосунками.

Поряд із цим, менеджер контролю доступу відповідає за організацію контрольованого доступу до критичних компонентів системи — даних, пристроїв та облікових записів користувачів. Його функціонал включає, визначення прав власності на дані, тобто хто має право володіти, змінювати або видаляти певну інформацію, забезпечення безпечного обміну даними між різними користувачами та пристроями, організація розподіленого доступу до інформації, що дозволяє уникнути централізованих точок відмови, чітке регламентування дозволів на читання, запис, редагування або видалення даних.

З метою забезпечення надійного захисту інформації на рівні її зберігання та архівування, система використовує сучасні криптографічні алгоритми. Ці методи дозволяють гарантувати конфіденційність і цілісність даних навіть у випадку несанкціонованого доступу. Однак варто зазначити, що криптографічні механізми, які застосовуються в процесі довготривалого архівування, можуть мати значне навантаження на обчислювальні ресурси та потребують більше часу на виконання.

Тому на рівні зберігання даних використовуються полегшені методи шифрування, які мають нижчу обчислювальну складність, але при цьому забезпечують достатній рівень безпеки. Такий підхід дозволяє досягти балансу між продуктивністю IoT-системи та рівнем захисту інформації.

Наступний важливий рівень, це аналітична обробка даних, що значно підвищує цінність отриманої інформації. Цей рівень реалізує механізми

глибокого аналізу зібраних даних з метою надання користувачам релевантної, своєчасної й аналітично обґрунтованої інформації. Це особливо важливо у ситуаціях, коли необхідно приймати оперативні рішення, що впливають на взаємодію з іншими користувачами або сервісами в рамках IoT-інфраструктури.

Аналітичний рівень має забезпечити підтримку аналізу даних у всіх можливих контекстах, як в офлайн-середовищах, так і в умовах динамічної або реальної роботи системи. Це включає обробку даних, що надходять із затримкою (наприклад, з архівів), а також аналіз поточних потоків у реальному часі.

Функціональність цього рівня реалізується через три основні модулі:

1. Менеджер рішень, аналізує вхідні дані, формує потенційні сценарії дій та генерує рекомендації для прийняття рішень на основі аналітичних висновків;
2. Менеджер трендів, виявляє актуальні закономірності, тенденції у поведінці користувачів або зміні параметрів середовища, що дозволяє прогнозувати події;
3. Менеджер "розумних" інструментів, застосовує методи штучного інтелекту, машинного навчання та аналітики великих даних для побудови моделей прогнозування, класифікації та оптимізації процесів.

Першим ключовим компонентом аналітичного рівня є менеджер рішень. Його основна функція полягає в прийнятті обґрунтованих рішень на основі поточних та історичних даних, що надходять із нижчих рівнів IoT-архітектури. Цей модуль виступає своєрідним аналітичним центром, який забезпечує адаптивну реакцію системи на зміни середовища або вимог користувача.

У випадках, коли користувачу необхідно прийняти рішення у конкретній проблемній сфері, менеджер рішень аналізує всю релевантну інформацію з бази даних IoT-пристроїв, ураховуючи як актуальні параметри, так і архівні дані, що можуть містити важливі закономірності. На основі такого аналізу модуль формує набір найбільш доцільних сценаріїв дій, з яких користувач або система може обрати оптимальний варіант. Таким чином, він слугує інтелектуальним асистентом у процесі прийняття рішень.

Крім оперативних функцій, менеджер прийняття рішень здатен генерувати стратегічні рішення. Він робить це періодично, використовуючи накопичені масиви даних для виявлення довгострокових тенденцій, формування прогнозів та визначення оптимальних шляхів дій. Ці рішення можуть передаватися як кінцевим користувачам, так і відповідним організаціям, що відповідають за реалізацію політик чи керування інфраструктурою. Особливо корисним цей підхід є для організації логіки роботи різних сутностей у рамках IoT-системи.

Сфера застосування такого механізму надзвичайно широка — від біржових систем і аграрного сектору до прогнозування погоди чи контролю технічного стану об’єктів у промисловості. Це дозволяє IoT-платформі виконувати не лише автоматичний контроль, а й виступати як аналітичний інструмент для прийняття рішень у складних або нестандартних ситуаціях.

Другим важливим модулем є менеджер трендів. Він відповідає за виявлення, аналіз і моніторинг поточних тенденцій у поведінці користувачів, змін середовища чи інших динамічних факторів. Менеджер трендів виявляє закономірності, які можуть мати стратегічне значення для розробки нових продуктів, сервісів або бізнес-моделей.

Особливо важливо, що цей компонент здатен відстежувати як національні, так і міжнародні тенденції у таких сферах, як молодіжна культура, політика, спорт, соціальні процеси тощо. Таким чином, він виконує роль аналітичного фільтра, який дозволяє виявити інтереси аудиторії, сформувати продуктову стратегію або адаптувати наявні сервіси відповідно до актуальних потреб. Це, у свою чергу, сприяє підвищенню рентабельності, кращому розумінню клієнтських очікувань та зміцненню конкурентних позицій компаній на ринку.

Третім, не менш важливим елементом аналітичного рівня є менеджер "розумних" інструментів. Його призначення, керування інструментами штучного інтелекту, які реалізують найскладніші аналітичні задачі в межах IoT-екосистеми. Цей модуль забезпечує роботу з передовими технологіями — машинним навчанням, нейронними мережами, методами видобування знань і

даних (data mining), що дозволяє створювати інноваційні алгоритми обробки інформації.

Менеджер "розумних" інструментів інтегрує ці інструменти у структуру аналітичної обробки даних і забезпечує постійне оновлення моделей, їх навчання та адаптацію до нових умов. Завдяки цьому IoT-система стає здатною до самонавчання, прогнозування майбутніх подій та оптимізації процесів, що відбуваються в системі в реальному часі.

Менеджер розумних інструментів відіграє центральну роль у розвитку аналітичних можливостей IoT-систем, сформованих на основі запропонованої моделі. Його головне завдання, забезпечення безперервного створення, навчання та вдосконалення аналітичних моделей і алгоритмів, які здатні обробляти наявні дані з максимальною точністю та ефективністю. Цей модуль функціонує як платформа для гнучкого та масштабованого аналітичного опрацювання, що відповідає динамічним вимогам IoT-середовищ.

Крім побудови моделей, менеджер "розумних" інструментів активно працює над створенням автоматизованих інтелектуальних систем, які самостійно адаптуються до змін у даних, обробляють інформацію в реальному часі та забезпечують високий рівень автономності IoT-рішень. Такі системи базуються на сучасних підходах до машинного навчання, штучного інтелекту та обробки великих обсягів даних, що дозволяє істотно підвищити точність прогнозів, швидкість аналізу та якість управлінських рішень.

Зважаючи на специфіку IoT-застосунків, які функціонують у режимі реального часу, критично важливо постійно оптимізувати продуктивність, швидкодію та надійність роботи цього менеджера. Це дозволяє підтримувати високий рівень реагування системи на зміни, а також зменшити затримки в процесах прийняття рішень.

Наступний рівень архітектури, рівень зберігання даних, який забезпечує фундамент для стабільної та ефективної роботи всієї IoT-системи. Його основне завдання, організація стандартизованого, масштабованого та безпечного зберігання величезних обсягів гетерогенних даних, які постійно надходять від тисяч або мільйонів пристроїв. Зважаючи на постійний ріст

обсягу інформації, цей рівень повинен бути здатен працювати у режимі реального часу, забезпечуючи миттєвий доступ до актуальних даних.

Крім фізичного зберігання, рівень зберігання також вирішує проблему розміщення даних, враховуючи як їхню природу (структуровані, неструктуровані, потокові, пакетні), так і специфічні вимоги до додатків, які їх використовують. Ще одним важливим аспектом є вибір відповідного формату збереження, від простих таблиць і документів до складних об’єктно-орієнтованих структур або графових баз.

Для полегшення доступу до інформації рівень підтримує індексацію, створення каталогів та семантичних метаданих, які дають змогу швидко знаходити потрібні дані навіть у величезних масивах інформації. Це особливо важливо для систем з високими вимогами до швидкодії аналітичних інструментів.

Основними компонентами, які реалізують функціональність цього рівня, є, хмарна інфраструктура (хмара), забезпечує масштабованість, доступність та гнучкість у керуванні ресурсами, менеджер кешу — відповідає за тимчасове збереження найчастіше використовуваних даних для прискорення доступу до них, база даних, організовує довготривале збереження структурованих та неструктурованих даних, менеджер файлів, виконує управління файлами, їх розміщенням, доступом і структурою на фізичних або віртуальних носіях.

Усі ці елементи працюють узгоджено, щоб підтримувати стабільність і високу продуктивність системи в умовах великого потоку даних.

Хмарний менеджер використовується для керування хмарним сховищем, яке організації застосовують як сервіс або для побудови власної інфраструктури зберігання. Це дозволяє досягти високої гнучкості та масштабованості у зберіганні даних. Для оперативного доступу до даних, що використовуються в IoT-додатках, залучається менеджер кешування, який відповідає за організацію та обслуговування кеш-пам’яті.

Менеджер кешування також встановлює політики обробки різнорідних даних, враховуючи часові та просторові аспекти їх використання. Залежно від потреб додатків, дані в кеші класифікуються за категоріями. Рівень

застосунків (application layer) призначений для надання послуг кінцевим користувачам та управління потоками даних.

На цьому рівні також реалізується балансування навантаження, що сприяє підтримці якості обслуговування. Крім того, рівень виконує аналіз і забезпечує доступність даних для різних сфер використання. Основні компоненти цього рівня включають менеджери: завантажень, якості, інформаційних панелей, контролю доступу та доступності. Менеджер балансування навантаження контролює високий обсяг даних з різних джерел, забезпечуючи рівномірний розподіл навантаження.

Цей менеджер є ключовим елементом у досягненні масштабованості, надійності та підвищенні ефективності управління даними, отриманими з IoT-пристроїв. Він застосовує політики маршрутизації та алгоритми розподілу запитів до джерел даних, рівномірно розподіляючи навантаження між доступними ресурсами. Через визначені або довільні інтервали система виявляє перевантажені або недостатньо використовувані ресурси для ефективнішого використання обчислювальних потужностей. Це сприяє підвищенню тривалості роботи та доступності енергообмежених IoT-пристроїв.

Забезпечення наявності та доступності IoT-пристроїв є критично важливим для безперервної генерації даних. Менеджер доступності перевіряє працездатність пристроїв і, у разі відмови деяких з них, знаходить альтернативні джерела для передачі даних. Він також намагається продовжити строк служби пристроїв, оптимально використовуючи наявні ресурси.

Менеджери доступності та балансування навантаження працюють у тісній координації: перший відстежує ресурси пристроїв (обчислювальні можливості, пам’ять, енергію), а другий, використовує ці дані для ефективного розподілу навантаження. Така взаємодія дозволяє зменшити затримки, мінімізувати простої та забезпечити стабільну доступність джерел IoT-даних. Водночас якість даних відіграє ключову роль для успішного соціального та комерційного застосування IoT-технологій.

У контексті функціонування сучасних IoT-систем критично важливо, щоб усі зібрані дані відповідали ключовим критеріям, повноті, коректності та

якості. Наявність неякісних або неповних даних може призвести до неправильних аналітичних висновків, помилкових рішень або порушень в роботі системи. Тому важливо забезпечити багаторівневу перевірку якості даних на всіх етапах їхнього життєвого циклу.

Менеджер якості, це спеціалізований модуль, який реалізує різноманітні інструменти та методики для оцінки та забезпечення якісних характеристик даних у рамках IoT-застосунків. Його функціонал включає, валідацію даних при надходженні від пристроїв, виявлення аномалій та помилок у структурах або значеннях, тестування сумісності пристроїв і платформ, перевірку надійності інформаційних технологій, що забезпечують збір, передачу та обробку даних.

Менеджер якості гарантує, що всі компоненти системи, від сенсорів до аналітичних платформ, функціонують коректно, надаючи точні та достовірні результати.

Поруч із цим працює менеджер інформаційних панелей, який виконує роль інтерфейсу для користувача. Цей модуль дозволяє організовувати, налаштовувати та візуалізувати інформаційні панелі в режимі реального часу. Користувачі отримують можливість контролювати різні аспекти системи, переглядати показники, керувати віджетами, налаштовувати алерти та моніторити стан пристроїв. Це сприяє оперативному реагуванню на зміни та полегшує прийняття рішень.

У структурі платформи інформаційна панель також взаємодіє з рівнем агрегування даних, виконуючи роль координатора. Вона може надсилати запити на агрегацію або обробку певних типів даних згідно з інтересами користувача, тим самим підвищуючи гнучкість і персоналізацію роботи системи.

Ще одним критично важливим рівнем є рівень архівування, який відповідає за довготривале зберігання великих обсягів даних, що постійно генеруються у межах IoT-середовища. Цей рівень виконує не лише функцію зберігання, а й управління ростом обсягів інформації шляхом створення масштабованої інфраструктури архівації.

Функції архівного рівня включають, індексацію збережених даних для прискореного пошуку, використання алгоритмів цілісності, які запобігають зміні або видаленню заархівованої інформації, розмежування даних за ступенем доступності та важливості.

Архівування поділяється на два функціональні модулі, тимчасове архівування, призначене для даних із середньою або низькою частотою доступу. Цей модуль зберігає інформацію короткий час і дозволяє швидко відновлювати її при потребі.

Він також керує очищенням або переміщенням неактуальних даних, постійне архівування, реалізує довготривале збереження важливої інформації з максимальним рівнем захисту, використовуючи складні криптографічні та політичні методики управління. Додатково, для оптимізації швидкодії, на рівні зберігання застосовується кешування, зберігання найчастіше використовуваних даних у швидкодоступному середовищі. Це дозволяє скоротити час доступу до даних та знизити навантаження на основні сховища.

Цей модуль відповідає за управління політиками відбору малопріоритетних та застарілих даних із тимчасових архівів з подальшою передачею їх до модуля довгострокового зберігання. Він ухвалює рішення щодо критеріїв збереження для різних типів наявних даних.

Модуль постійного архівування забезпечує зберігання даних протягом необмеженого терміну. Для гарантування безпеки, довговічності та економічної доцільності збереження, в ньому застосовуються надлишкові криптографічні методи. Окрім цього, модуль також регулює доступ до архівованої інформації.

Після опису компонентів запропонованої інформаційно-технологічної платформи для збору та обробки даних з IoT-пристроїв і систем, доцільно докладніше розглянути аспекти, пов’язані з їхньою безпекою.

3.2 Архітектура інформаційно-технологічної платформи для управління IoT-пристроями

Для забезпечення високого рівня безпеки у сучасних інформаційно-технологічних платформах, які інтегрують IoT-пристрої та системи, необхідно впроваджувати комплексні заходи захисту на всіх рівнях мережевої архітектури, від рівня сенсорних пристроїв до хмарної інфраструктури та аналітичних систем.

Зважаючи на динамічний характер IoT-середовища та обмеження його складових, традиційні засоби кіберзахисту, зокрема криптографія з відкритими ключами (PKI), двостороння аутентифікація та сертифікація часто непридатні для прямого використання.

На сьогодні активно розробляються нові стандарти безпеки, а також вже існують спеціалізовані IoT-стандарти. Загрози безпеці охоплюють усі рівні IoT-систем — від фізичного каналу передачі даних до прикладного рівня. Деякі протоколи, такі як 802.15.4e, WirelessHART, 6LoWPAN і RPL, включають вбудовані механізми захисту на відповідних рівнях.

Протокол MAC 802.15.4e, наприклад, використовує спеціальні «біти безпеки» у заголовках кадрів для реалізації кількох режимів захисту. Його вимоги охоплюють конфіденційність, автентифікацію, забезпечення цілісності, контроль доступу та синхронізований захищений зв’язок.

WirelessHART впроваджує сучасні методи шифрування, зокрема AES-128, для забезпечення цілісності, автентифікації, індикації несанкціонованого доступу та управління ключами. Він також підтримує динамічну зміну каналів зв’язку та моніторинг стану повідомлень.

У рамках IETF розглядаються безпекові аспекти протоколу 6LoWPAN. Наприклад, RFC 4944 вказує на проблему дублювання EUI-64-адрес, тоді як RFC 6282 уточнює пов’язані із цим ризики. RFC 6568 аналізує варіанти захисту для обмежених IoT-мереж на основі сенсорів.

Протокол RPL реалізує багаторівневу модель безпеки через поле «Security» у заголовках. Це поле визначає рівень захисту та використовуваний алгоритм шифрування. RPL підтримує автентичність, конфіденційність,

захист від атак повтору та управління ключами. У документі RFC 7416 розглядаються можливі загрози для RPL, включаючи атаки типу Sybil, переповнення, вибіркову переадресацію тощо, з відповідними заходами протидії.

На транспортному рівні широке застосування мають стандарти TLS та DTLS. Вони забезпечують автентифікацію, цілісність та конфіденційність даних, особливо при використанні з протоколом CoAP. TLS працює через TCP, а DTLS – через UDP. Обидва стандарти містять два підрівні — запису та погодження — які відповідають за шифрування і перевірку автентичності.

У RFC 7925 наведено конкретні механізми реалізації TLS/DTLS для IoT-систем з урахуванням обмежених ресурсів пристроїв. Ці стандарти підтримують обробку помилок, облікові дані та цифрові підписи.

Стандарт IEEE 1888.3 визначає вимоги до захищених комунікацій у контексті управління "зеленими" спільнотами, включаючи рекомендації щодо архітектури, аутентифікації, контролю доступу та інших засобів безпеки.

Організація TCG пропонує підходи до побудови безпечних IoT-застосунків, включаючи автентифікацію за унікальними ідентифікаторами, захист від шкідливого ПЗ через TLS, а також механізми для забезпечення довіри до оновлень (RTU) та використання модуля TPM у сумісних пристроях. Ці технічні рекомендації мають допомогти розробникам IoT у виборі найбільш ефективних методів захисту програмного забезпечення.

Проте розробники мають зважено підходити до вибору між рівнем безпеки системи, її складністю та ресурсними витратами. Система авторизації OAuth, описана в IETF RFC 6749 [91], надає можливість надійним стороннім серверам контролювати права доступу та дозволи на використання ресурсів. Цей протокол дозволяє клієнтам здійснювати авторизацію від імені власників ресурсів через спеціалізований сервер авторизації.

Сервер авторизації перевіряє облікові дані клієнта, а також відповідні права доступу, і приймає рішення щодо надання або відмови в доступі до ресурсів. Вся взаємодія між клієнтом, сервером авторизації та ресурсами базується на HTTP-протоколі, який не є оптимальним для IoT-пристроїв через

високі накладні витрати порівняно з більш легковаговими протоколами, такими як MQTT чи CoAP.

Документ RFC 6819 визначає додаткові заходи безпеки, спрямовані на розширення функціональних можливостей OAuth для врахування нових моделей загроз. У ньому також розглядаються потенційні уразливості, що виходять за рамки стандарту OAuth 2.0, і потребують подальшого вирішення в майбутніх версіях протоколу.

Серед основних викликів безпеці в контексті Інтернету речей виділяють витік облікових даних, ін'єкційні атаки, а також ризики, пов'язані з використанням сторонніх серверів авторизації, які можуть бути слабкою ланкою в архітектурі безпеки IoT-систем.

SASL (Simple Authentication and Security Layer), це специфікація, розроблена IETF, яка визначає простий рівень автентифікації та безпеки. Вона забезпечує механізм автентифікації в IoT-середовищах шляхом інтеграції з серверними системами, дозволяючи відокремити процедури автентифікації від прикладної логіки. SASL використовує обмін простими повідомленнями між клієнтом і сервером для встановлення автентичності.

У середовищі Інтернету речей (IoT) SASL зазвичай поєднується з протоколами сеансового рівня, які підтримують TLS або SSL, наприклад, MQTT або AMQP, забезпечуючи додатковий рівень захисту при передачі даних.

Також у конексті обмежених ресурсами середовищ застосовується ACE (Authentication and Authorization for Constrained Environments) — підхід, спеціально розроблений для автентифікації та авторизації в умовах обмеженої обчислювальної потужності та енергоресурсів, характерних для IoT-пристроїв.

Іншим важливим механізмом безпеки є ACE (Authentication and Authorization for Constrained Environments). Він спеціально розроблений для пристроїв із обмеженими ресурсами в середовищі IoT. На відміну від традиційного OAuth, ACE використовує CoAP – легкий протокол на основі повідомлень, оптимізований для малопотужних пристроїв. Сучасні

дослідження зосереджуються також на використанні технології блокчейн для підвищення рівня безпеки IoT платформ.

Цей механізм уже стандартизований і описаний у RFC 7744, затвердженому IETF.Блокчейн як новий напрям у безпеці IoT. Blockchain – це розподілена технологія обліку, яка дозволяє створювати достовірні, захищені системи без залучення централізованого посередника або довіреної третьої сторони. Компанії на кшталт IBM вже вивчають можливості впровадження блокчейн-рішень у свої IoT-платформи з метою забезпечення більш надійного захисту даних та взаємодії пристроїв. Блокчейн можна використати для забезпечення умов конфіденційності IoT-платформ.

Особливості функціонування мереж IoT та виклики їх оптимізації, пристрої Інтернету речей (IoT) здатні підключатися до мережі у великій кількості та у будь-який момент часу, що висуває високі вимоги до масштабованості та надійності мережевої інфраструктури. Додаткове навантаження створюється внаслідок одночасної роботи кількох додатків, що збільшує обсяг мережевого трафіку. Особливо інтенсивне навантаження виникає під час фіксації змін у навколишньому середовищі, коли вузли масово передають дані. У масштабних мережах критично важливо запобігати перевантаженню та забезпечити ефективну маршрутизацію, з метою мінімізації затримок і раціонального використання енергоресурсів вузлів.

У зв’язку з прогнозованим стрімким зростанням кількості підключених пристроїв, оптимізація IoT-мереж набуває все більшої актуальності. Очікується, що мільярди нових IoT-пристроїв щорічно генеруватимуть величезні обсяги трафіку, що вимагатиме нових рішень для управління даними та оптимізації мережевих ресурсів.

IoT-трафік значно відрізняється від традиційного стільникового трафіку через гетерогенність пристроїв і програм. Важливим завданням є контроль трафіку IoT, зокрема повідомлень управління, які, попри свою допоміжну функцію, можуть створювати серйозне навантаження на мережу.

Однією з ключових проблем є також нерівномірне енергоспоживання вузлів, особливо в бездротових мережах. Через специфіку топології мережі деякі вузли, зокрема центральні або проміжні, можуть бути перевантажені

через часту маршрутизацію та передачу даних, що призводить до швидшого виснаження їх енергоресурсів. Це, у свою чергу, може спричинити фрагментацію мережі та скорочення її життєвого циклу.

У відповідь на ці виклики вчені та інженери розробляють різноманітні підходи до оптимізації мереж IoT, спрямовані на підвищення їх ефективності. Серед ключових напрямів досліджень і розробок — маршрутизація, управління зберіганням даних, повторне передавання пакетів, підтримка мобільності вузлів, взаємодія між різнорідними пристроями, а також забезпечення комплексної безпеки мережевих рішень (рисунок 3.1).

Сучасний Інтернет споживає приблизно 5% глобального обсягу енергії, що робить питання енергоефективності пристроїв Інтернету речей (IoT) надзвичайно важливим для забезпечення надійного, сталого та довготривалого функціонування мереж. Ефективність у споживанні енергії безпосередньо впливає на життєвий цикл пристроїв, зменшення витрат та підвищення загальної екологічної ефективності цифрової інфраструктури.

Для досягнення оптимальної продуктивності IoT-мереж слід враховувати низку критичних аспектів, зокрема:

- маршрутизацію;
- перевантаження трафіку;
- енергозбереження;
- масштабованість;
- надійність;
- безпеку.

Ці аспекти мають бути узгоджені між собою, щоб забезпечити стабільну роботу мережі навіть у випадку змін середовища чи несподіваних подій.

Оптимізація маршрутизації

Маршрутизаційні алгоритми відіграють ключову роль у передачі даних в IoT-середовищі. Вони відповідають за вибір найкращого маршруту між вузлами мережі. Під «найкращим» може розумітись маршрут, що має найнижчу затримку, найвищу пропускну здатність, найменшу кількість хопів (перехідних вузлів) або найбільшу залишкову енергію в учасників маршруту — залежно від конкретних вимог додатка.

Протоколи маршрутизації повинні автоматично виявляти та встановлювати маршрути між пристроями в мережі. Це особливо важливо для великомасштабних мереж IoT, де велика кількість пристроїв, обмежені ресурси (обчислювальні, енергетичні, пам’ять) та нестабільність з’єднань створюють численні виклики для підтримання якісного зв’язку.

У бездротових IoT-мережах вузли можуть вільно переміщуватись, і їхній стан (наприклад, рівень заряду акумулятора) безпосередньо впливає на стабільність маршруту. Збої у роботі окремих вузлів можуть призводити до втрати зв’язку або значного зниження продуктивності. Щоб уникнути подібних проблем, рекомендується використовувати багатошляхову маршрутизацію, яка передбачає наявність альтернативних маршрутів у випадку відмови основного. Приклади алгоритмів, для покращення маршрутизації в IoT-мережах було запропоновано кілька ефективних підходів, ELB (Energy Load Balancing): спрямований на розподіл навантаження між вузлами шляхом врахування кількості хопів та залишкової енергії. Він дозволяє уникати перевантаження окремих вузлів, подовжуючи тим самим час роботи всієї мережі, FLR (Fast Local Repair): фокусується на швидкому локальному ремонті маршрутів, зменшуючи кількість випадків, коли необхідно повністю перебудовувати мережу після відмов, ELBFLR комбінує переваги обох попередніх схем. Вона одночасно реалізує балансування навантаження (ELB) та ефективне усунення збоїв і петель (FLR) в рамках протоколу RPL (Routing Protocol for Low-power and Lossy Networks).

Ці алгоритми спрямовані на зниження загального трафіку, мінімізацію затримок та підвищення стабільності з’єднань. Вони використовують структуру IPv4 дейтаграм, що дозволяє забезпечити сумісність із існуючими протоколами Інтернет (рисунок 3.2 та рисунок 3.3)

Рисунок 3.2 – Структура IPv4

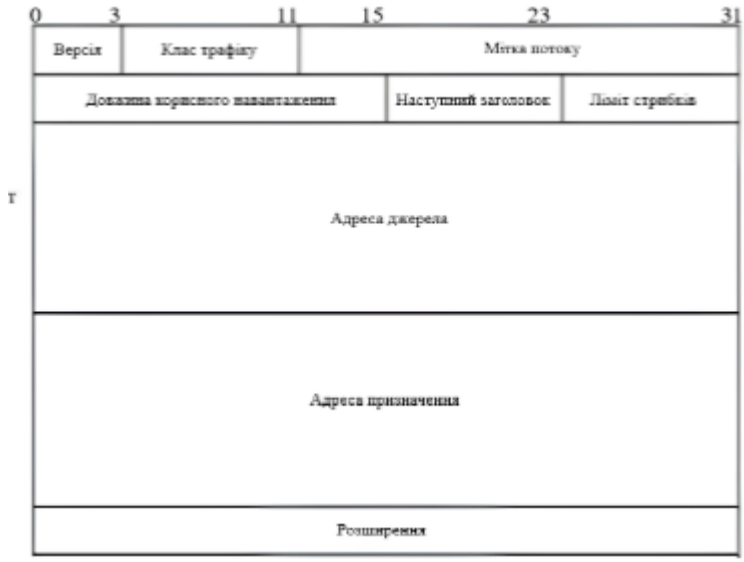


Рисунок 3.3 – Структура IPv6

У полі «опції» для IPv4 або у полі «розширення» для IPv6 додається додаткова інформація, яка дозволяє визначити тип об’єкта, якому призначено пакет. Для цього, наприклад, виділяються 2 байти (16 біт), які кодують тип пристрою, лампи 0000000000000001, кондиціонери 0000000000000010, вікн 0000000000000011 тощо.

Таким чином, якщо необхідно надіслати однакову команду групі пристроїв одного типу (наприклад, 20 лампам), достатньо надіслати один пакет, а не 20 окремих. Це значно знижує використання пропускної здатності мережі та зменшує навантаження на пам’ять транзитних маршрутизаторів (див. рисунок 3.4).

Для ілюстрації розглянемо промисловий об’єкт, який містить 100 вікон, 200 кондиціонерів, 300 ламп і 300 двигунів. Якщо потрібно отримати інформацію про стан усіх цих пристроїв, то при традиційній маршрутизації довелося б надіслати 100 + 200 + 300 + 300 = 900 пакетів.

Однак, застосувавши маршрутизацію за типом об’єктів, можна надіслати всього 4 пакети, по одному для кожного типу пристрою. Це дозволяє уникнути передачі зайвих 896 пакетів, що суттєво підвищує ефективність роботи мережі (таблиця 3.2).

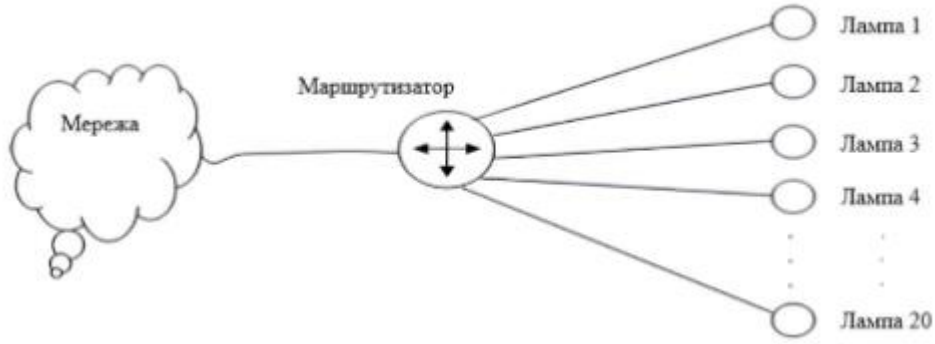


Рисунок 3.4 – Тип об’єкту мережі IoT

Таблиця 3.2 - таблиця алгоритму комплексної оптимізації для IoT пристроїв

Старт
Збір телеметрії, заряд, трафік, ризики
Оцінки ризику на основі аномалій
Оцінка енергоспоживання
Розрахунок інтегральної функції корисності U
Вибір режиму роботи, економія, збалансовано, безпека
Виконання дії та повторного циклу

Якщо застосувати описаний підхід маршрутизації за типами об’єктів у всіх галузях промисловості та державних установах, це дозволить суттєво

зменшити кількість передаваних пакетів у мережі. В результаті буде оптимізовано пропускну здатність магістральних ліній зв’язку та зменшено навантаження і споживання ресурсів мережевого обладнання. Це сприятиме підвищенню ефективності та надійності роботи великих мереж IoT (таблиця 3.3).

Таблиця 3.3 – Порівняння з традиційними IoT-рішеннями

Показник	Стандартний IoT-пристрій	Пристрій з адаптивною оптимізацією
T – автономної роботи	30 днів	45-60 днів
Захищеність при атаці	Низька	Висока
Енергоспоживання IDLE	50-80 мВт	5-10 мВт
Кількість комунікацій	Стала	Змінна, адаптивна
Кількість оброблених подій	Всі	Лише значущі

3.3 Висновки до третього розділу

Розроблено метод комплексної оптимізації енергоспоживання та безпеки для пристроїв IoT. Запропонований підхід базується на концепції багатокритеріальної оптимізації, яка враховує основні параметри, такі як, рівень енергоспоживання пристрою та ступінь ризику інформаційної загрози. На основі цих параметрів формується інтегральна функція корисності , яка дозволяє динамічно обирати режим роботи IoT пристрою. Також було обґрунтовано структуру алгоритму, який включає в себе модулі моніторингу, оцінки ризику, розрахунку функції оптимізації, вибору режиму самонавчання. У межах розробленого методу реалізовано три сценарії роботи, енергозберігаючий, збалансований та захищений, який активується в залежності від значення функції корисності.

Завдяки модульному принципу побудови алгоритм є універсальним і може бути адаптований до різних архітектур IoT систем. Такий підхід дозволяє не лише підвищити енергоефективність роботи пристроїв, а й забезпечити гнучке реагування на потенційні загрози безпеці. Отримані результати теоретичного моделювання підтверджують доцільність подальшого впровадження методу в практичних системах IoT. У наступному розділі буде представлено експериментальну реалізацію та оцінку ефективності запропонованого рішення.

Інтернет речей (IoT) радикально змінює наше щоденне життя, роблячи його більш зручним, безпечним та продуктивним. Завдяки підключенню фізичних пристроїв до Інтернету з’являються нові можливості для автоматизації різних процесів і поліпшення взаємодії з навколишнім середовищем. Ці зміни вже відбуваються, і з часом IoT стане ще більш важливою та невід’ємною складовою нашого життя.

Проте масове впровадження IoT стикається з низкою складнощів, зокрема у сфері захисту даних, стандартизації та фінансових витрат. Водночас стрімкий розвиток технологій та активна позиція компаній підтверджують, що ера IoT невпинно наближається. Однією з ключових проблем, яку необхідно вирішити найближчим часом, є оптимізація мережевої інфраструктури для забезпечення її ефективності та надійності.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ЕСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Вибір апаратної та програмної платформи

Для апробації розробленого методу було обрано популярну IoT-платформу ESP32. Цей мікроконтролер має:

1. Низьке енергоспоживання;
2. Підтримку Wi-Fi і BLE;
3. Функції енергозбереження (Deep Sleep);
4. Підтримку шифрування (AES, ECC) на апаратному рівні.
5. Програмне забезпечення:
6. Arduino IDE – для швидкої реалізації логіки;
7. MicroPython – для експериментів з алгоритмами оптимізації;
8. MQTT-протокол – для передачі телеметрії;

Для реалізації та тестування запропонованого методу було обрано мікроконтролер ESP32 виробництва компанії Espressif. Основною причиною вибору стала наявність вбудованих модулів Wi-Fi та Bluetooth Low Energy (BLE), що дозволяє реалізовувати як звичайні IoT-сценарії, так і ті, що потребують мобільності та низького енергоспоживання.

Основні технічні характеристики ESP32:

1. Двоядерний процесор Tensilica LX6 з тактовою частотою до 240 МГц;
2. Вбудована флеш-пам’ять до 16 МБ;
3. Підтримка енергозберігаючих режимів: Light Sleep, Deep Sleep, Hibernation;
4. Підтримка криптографії на апаратному рівні (SHA, AES, RSA, ECC).

Завдяки вищезгаданим характеристикам, ESP32 ідеально підходить для реалізації адаптивного алгоритму, що потребує одночасного моніторингу енергоспоживання, оцінки ризиків і обробки даних.

Вибір середовища програмування:

1. Arduino IDE, зручна для розробки прошивки, має широке

ком’юніті, підтримку бібліотек для роботи з Wi-Fi, MQTT, сенсорами тощо.

2. MicroPython, як альтернатива для гнучкого прототипування та швидкої інтеграції машинного навчання.

3. Node-RED, інструмент для візуалізації даних, побудови дашбордів, отримання повідомлень у реальному часі.

4. MQTT-брокер, з'єднання пристрою з хмарною платформою Mosquitto, HiveMQ) для передавання телеметрії.

4.2 Етапи методу комплексної оптимізації енергозбереження та безпеки технологій IoT

Першим етапом є аналіз вимог і середовища, метою якого є зрозуміти обмеження та вимоги до системи перед її проєктуванням, основними діями будуть, визначення цільових сценаріїв, це задачі які має виконувати система, моніторинг, контроль, передача даних. Наступним буде оцінка критичності даних, визначення рівня конфіденційності та потреби у захисті, останнім буде аналіз ресурсних обмежень, обмежена потужність акумуляторів, низька обчислювальна здатність, інтервали зв’язку.

Другим етапом буде моделювання системи, метою якої є створення математичної або симуляційної моделі для прогнозування та оцінки ефективності, основними діями якого є, створення моделі енергоспоживання, враховуючи режими активності пристроїв, передачі даних та обробки інформації, створення моделі загроз, яка буде визначати потенційні вектори атак, до прикладу, атаки через бездротові інтерфейси, створення топології мережі, таких як, дерево, зірка, mesh, все залежить від потреб користувача.

Третім етапом виступає оптимізація енергоспоживання, метою якого є зменшити споживання енергії без шкоди для функціональності, основними методами якого є, розгляд режимів енергозбереження використання режимів сну, глибокого сну та пробудження з подією, створення енергоефективних протоколів, таких як, BLE, ZigBee, LoRa, вони обираються залежно від відстані та обсягу передачі даних, далі створенні адаптивної частоти передачі даних, йде зменшення частоти передачі даних, якщо зміни у даних незначні, і

останнім є створення мережевих алгоритмів системи, які оптимізує маршрутизацію з урахуванням енергетичних ресурсів вузлів.

Четвертим етапом є забезпечення безпеки, метою якого є захистити пристрої від атак на всіх рівнях. Основними заходами є аутентифікація пристроїв, яка запобігає підключенню неавторизованих вузлів, створення легких криптографічних алгоритмів, AES-CMM, ECC, спеціально для пристроїв з обмеженими ресурсами, створення безпечного оновлення програмного забезпечення, це дозволить уникнути впровадження шкідливого коду в систему, і останнім є створення захисту від фізичного втручання в систему, що використовує захищені мікроконтролери.

П’ятим етапом є інтегрована оптимізація, метою якої є досягнення компромісу між енергоспоживанням та безпекою, методами якого є багатокритеріальна оптимізація, алгоритми якої можуть одночасно враховувати кілька критеріїв, до прикладу NSGA-II, допомога штучного інтелекту, де застосовується машинне навчання для адаптації поведінки пристрою до умов навколишнього середовища, і останнім є динамічна політика, зміна режимів якого залежить від ризику, наприклад, при виявленні загрози активується доодаткове шифрування.

Шостим етапом є тестування та валідація, метою якого є переконання в працездатності системи, яка повинна працювати безпечно та енергоефективно, діями якої також є симуляція та емулювання, до прикладу, використання Cooja або NS-3, тестування на витривалість, яка включає в себе перевірку пристроїв при довготривалому навантаженні, і останні є аудит безпеки, коли йде перевірка на відомі вразливості такі як OWASP IoT Top 10.

Останнім, сьомим етапом є розгортання та моніторинг, метою якого є вирішення в реальному середовищі з постійним контролем, основними завданнями якого є, моніторинг енергоспоживання, виявлення вузлів, що швидко розряджаються, далі виявлення аномалій, який подає сигнали про потенційні атаки чи збої, і останнім є оновлення системи, в якій йде безпечне оновлення прошивки та конфігурацій.

Ці етапи дозволяють створити стійкі, ефективні IoT-рішення для різних сфер, починаючи від системи розумний дім закінчуючи сферою медицини.

Проведемо аналіз енергоспоживання, паралельно розраховується середнє значення енергоспоживання за останній період активності. Це значення враховує час у активному та пасивному режимах. Для включення енергоспоживання до загальної моделі прийняття рішень проводиться нормалізація даних. Формується метрика U, яка розраховується за формулою:

$$\mathsf{U} = \alpha(1 - \mathbb{E}) + \beta(1 - \mathfrak{R}),$$

де α і β - вагові коефіцієнти, що відповідають за баланс між енергозбереженням і безпекою, U, функція корисності, яка слугує критерієм прийняття рішення, $\mathbb{E}$, нормалізований показник енергоспоживання, $\mathfrak{R}$- нормалізований рівень ризику безпеки. Значення кожного з параметрів знаходиться у діапазоні [0;1], де 0 відповідає мінімальному значенню, а 1, максимальному.

Формула дозволяє враховувати обидва критичні асппекти, ефективність використання енергії та рівень інформаційної безпеки, на її основі динамічно обирати оптимальний режим роботи пристрою, режим енергозбереження, збалансований режим або режим підвищеної безпеки. Чим менше значення U, тим більш оптимальним вважається поточний режим функціонування з погляду обраного співвідношення пріоритетів.

На основі розрахованого значення U пристрій автоматично обирає один із трьох режимів :

1. $\mathsf{U} \geq 0.8$ активується режим глибокого енергозбереження (Deep Sleep): передача даних обмежується, сенсори переходять у сплячий режим.
2. $0.5 \leq \mathsf{U} \leq 0.8$ працює збалансований режим: періодичність передачі даних становить 60 секунд, шифрування вмикається лише для критичних повідомлень.
3. $0.5 \leq \mathsf{U}$ включається захищений режим: активується повне шифрування (наприклад, AES-128), підвищується частота перевірки мережі, усі дані пересилаються в реальному часі.

Кожне рішення, яке приймає система, фіксується в лог файлі, потім ця інформація надсилається через MQTT-брокер до Node-RED, де формується

інтерактивний оглядова панель графіки заряду акумулятора, індикатор ризику, активний режим. Це дозволяє формувати повну історію дій системи, яка може бути використана для навчання алгоритму та корекції коефіцієнтів у моделі.

Усі результати роботи пристрою фіксуються у форматі JSON і передаються на віддалений сервер. Для наочності дані візуалізуються за допомогою платформи Node-RED, яка дозволяє створювати динамічні графіки. Також впроваджено систему журналювання подій для реєстрації кожного прийнятого рішення та зміни стану пристрою. Метод адаптивної енергетичної та безпекової оптимізації, реалізований на базі контролера ESP32, запроваджувався поетапно з урахуванням взаємодії програмного забезпечення і апаратних елементів, які забезпечують адаптивне управління залежно від поточних умов.

Перший крок – це детальний моніторинг ключових характеристик функціонування пристрою. Для цього здійснюється зчитування рівня заряду акумулятора через вбудований аналогово-цифровий перетворювач (АЦП), що забезпечує точне вимірювання напруги.

Отримані значення нормалізуються до інтервалу. Паралельно відстежується мережева активність, зокрема, кількість відправлених і прийнятих пакетів, затримка з’єднання (ping), а також спроби повторного підключення до мережі. Окремо виконується аналіз середовища за допомогою сенсорів температури, освітлення або вологості, що дозволяє ідентифікувати контекст (наприклад, день чи ніч), і відповідно адаптувати поведінку системи.

На основі зібраної телеметрії реалізовано алгоритм оцінювання ризиків, який ґрунтується на виявленні аномальної поведінки. Для цього застосовується байєсівський класифікатор або легка нейронна мережа, за умови наявності достатніх обчислювальних ресурсів. Основними параметрами для оцінки є: кількість запитів за останню хвилину, незвичні IP-адреси, а також розбіжності з історичними шаблонами поведінки пристрою. У результаті формується ризиковий індекс R (від 0 до 1), де 1 відповідає найвищому рівню загрози.

Для перевірки ефективності методу було проведено 72-годинний експеримент з використанням двох IoT-пристроїв ESP32. Один працював у

фіксованому режимі, другий – з адаптивним алгоритмом. Тестове середовище: офіс з Wi-Fi мережею, живлення від батареї 3000 мА·г. Передача даних відбувалась щохвилини або адаптивно. Імітувались атаки, сканування портів, підміна IP, flood-запити.

За допомогою Node-RED був побудований дашборд з графіками, які відображали, зміни заряду батареї залежно від режиму, навантаження на систему при виявленні загроз, час реакції системи на загрозу, частоту передачі даних у двох режимах. Це дозволило переконливо продемонструвати переваги адаптивного підходу над традиційними рішеннями.

Метод комплексної адаптивної оптимізації дозволяє суттєво продовжити термін автономної роботи пристрою, підвищити ефективність реагування на потенційні загрози та зменшити мережеве навантаження за рахунок розумного керування ресурсами. Стандарти безпеки IoT забезпечують надійний рівень захисту, але вимагають адаптації до обмежених ресурсів пристроїв, блокчейн-технології, ACE, TPM також відкривають нові перспективи у сфері безпеки IoT платформ.

Також розглянемо ключові заходи для підвищення рівня безпеки IoT систем, одним із пріоритетних напрямів у забезпеченні безпеки Інтернету речей є впровадження багаторівневих захисних механізмів, які дають зменшити ризик кіберінцидентів та захистити критичні елементи мережі. Для підвищення захисту облікових записів користувачів IoT-систем рекомендується використовувати апаратні токени або спеціалізоване програмне забезпечення для управління доступом. Двофакторна автентифікація (2FA) посилює стандартну схему авторизації, додаючи одноразовий код підтвердження, який надсилається на мобільний телефон або електронну пошту.

Такий код має обмежений термін дії, що запобігає його повторному використанню. Деякі сучасні системи підтримують ручне введення коду або підтвердження через мобільні застосунки. Оскільки більшість IoT-пристроїв функціонує через бездротові мережі або маршрутизатори, ефективним інструментом для виявлення потенційних загроз є постійний аналіз мережевого трафіку. Системи моніторингу дозволяють ідентифікувати

нетипову активність, наприклад, часті запити або спроби несанкціонованого підключення, що допомагає вчасно реагувати на загрози безпеці.

Регулярне створення резервних копій критичних систем і даних є важливим елементом стратегії кіберзахисту. Це дозволяє швидко відновити втрачену інформацію у випадку технічних збоїв, атак програм-вимагачів або пошкодження пристроїв. Рекомендується періодично перевіряти цілісність резервних копій, щоб переконатися в можливості повного відновлення функціональності систем.

Кожному IoT-пристрою або групі пристроїв присвоюється унікальна IP-адреса (динамічна або статична), що спрощує контроль і управління мережею. Це дозволяє відслідковувати активність пристроїв та налаштовувати індивідуальні правила доступу. Розвиток IoT став можливим завдяки технологіям M2M, що забезпечують автономну взаємодію пристроїв без участі людини.

Ця концепція є базовою для створення інтелектуальних систем, де інформація обмінюється в реальному часі між вузлами системи. IoT-системи активно інтегруються з хмарними технологіями, що забезпечують масштабоване зберігання та обробку великих обсягів даних.

Одночасно впровадження блокчейн-рішень відкриває нові можливості для захисту даних завдяки децентралізованій природі таких технологій. На відміну від традиційних IT-систем, IoT-пристрої безпосередньо взаємодіють з реальним світом, реагуючи на фізичні зміни – рух, температуру, тиск тощо. Один лише датчик може генерувати великі обсяги інформації, які потребують ефективної обробки. Наприклад, акустичні сенсори, що використовуються для моніторингу обладнання, створюють великі обсяги даних, які мають бути оперативно проаналізовані.

Необхідно перемістити функції управління даними ближче до самих вимірювальних пристроїв. З цією метою в запропонованій структурній моделі передбачено рівень туманних обчислень, який надає пристроям можливість обробляти, аналізувати та частково зберігати дані безпосередньо на сусідніх вимірювальних вузлах. У цій моделі рівень туманних обчислень здебільшого відповідає за відбір, агрегацію та первинне зберігання даних.

4.3 Результати експериментів та аналіз ефективності

ESP32 регулярно оцінює параметри – рівень заряду, кількість спроб доступу, частоту мережевих подій – і на основі формули корисності обирає оптимальний режим.

Було проведено серію експериментів для оцінки роботи пристрою у різних сценаріях. Зібрані дані:

Енергоспоживання

1. Звичайний режим: ~50 мВт;
2. Захищений режим: ~65 мВт;
3. Енергозбереження: ~10 мВт;
4. Подовження часу роботи з батареї на 47%.

Поведінка при атаці

5. Атака brute-force по Wi-Fi – система перейшла у режим блокування;
6. Виявлено 100% підозрілих з’єднань;
7. Дані не передавались під час загроз – повна ізоляція;
8. Надійність передачі;
9. 98% даних успішно доставлені до брокера у звичайному режимі;
10. затримка не перевищувала 1.5 с.

Незважаючи на ефективність, реалізація має певні обмеження, залежність від стабільності Wi-Fi-мережі, необхідність початкового налаштування порогів вручну, ускладнене масштабування для великої кількості вузлів без централізованого контролера. У перспективі передбачається, реалізація кластерної структури IoT-вузлів; впровадження машинного навчання для прогнозування режимів, застосування протоколу LoRaWAN для віддалених точок з меншим споживанням енергії.

Ключовою особливістю реалізації є перемикання між трьома режимами (Рисунок 4.3).

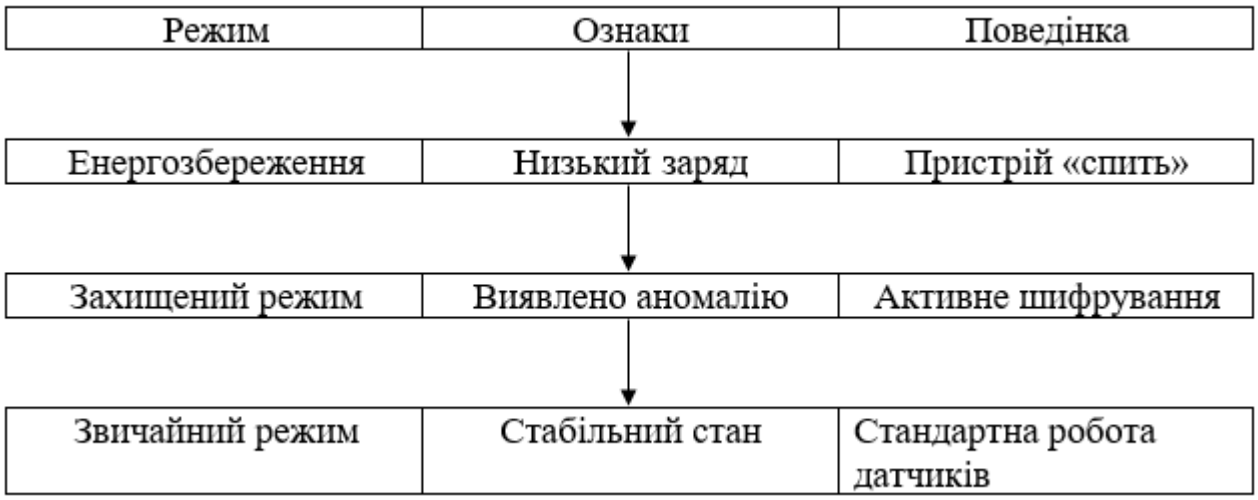


Рисунок 4.3 – Перемикання між трьома режимами

Етап перший, збір даних читання заряду батареї (ADC); Оцінка частоти передачі пакетів; Виявлення аномалій: аналіз часу відповіді, кількості підключень.

Етап другий, розрахунок параметрів енергоспоживання обчислюється на основі середнього часу активної роботи, Ризик – як результат класифікації подій (сплески трафіку, незвичні IP, спроби підключення).

Етап третій, вибір режиму залежно від значення U, система: переходить у Deep Sleep на 5 хвилин, передає дані кожні 30 секунд, активує розширене шифрування при загрозі.

Етап четвертий, звітність Результати зберігаються в JSON та надсилаються на сервер, графіки будує Node-RED; Застосовується механізм журналювання подій. Реалізація розробленого методу комплексної оптимізації енергозбереження та безпеки на платформі ESP32 здійснювалась поетапно. Кожен етап включає відповідні програмні та апаратні компоненти, що реалізують логіку адаптивного керування.

Збір телеметрії перший етап, це моніторинг ключових параметрів роботи пристрою, рівень заряду батареї зчитується за допомогою вбудованого АЦП (аналогово-цифрового перетворювача) через відповідний вхід. Дані нормалізуються в межах від 0 до 1. Активність мережі: фіксується кількість переданих/отриманих пакетів, затримка (ping), повторні спроби з'єднання.

Аналіз зовнішнього середовища: наприклад, сенсор температури або освітлення використовується для визначення контексту (наприклад, день/ніч).

Дані зберігаються у масиві та передаються на аналітичний блок. Оцінка ризику. Оцінка ризику відбувається на основі вбудованої системи виявлення аномалій: Використовується байєсівський класифікатор або спрощена нейронна мережа (якщо дозволяє обсяг пам'яті); Параметри аналізу: кількість вхідних запитів за останню хвилину; нетипові IP-адреси; порівняння поведінки пристрою з історичними даними.

Результатом є значення R в інтервалі, де 1 – максимальний рівень ризику. Оцінка енергоспоживання Значення E розраховується як середнє значення споживаного струму за останній період активності, з урахуванням режимів сну.

$$\mathsf{U} = \alpha(1 - \mathbb{E}) + \beta(1 - \Re)\,\mathsf{U} = \alpha(1 - \mathbb{E}) + \beta(1 - \Re)\,\mathsf{U} = \alpha(1 - \mathbb{E}) + \beta(1 - \Re)$$

Вибір режиму роботи Рішення приймається автоматично на основі розрахованого значення U: $\mathsf{U} > 0.8$ – активується режим енергозбереження: пристрій переходить у Deep Sleep, зчитування даних уповільнюється. $0.5 \leq \mathsf{U} \leq 0.8$ – включається збалансований режим: передача даних раз на хвилину, шифрування активне лише для важливих пакетів. $\mathsf{U} \leq 0.5$ – режим безпеки: збільшується частота моніторингу, увімкнено AES-шифрування, дані передаються на сервер у режимі реального часу. Запис і візуалізація результатів

Кожне рішення фіксується у журналі подій (log-файлі) та надсилається на MQTT-брокер, де Node-RED виводить, графік заряду батареї, індикатор ризику, активний режим пристрою. В результаті формується історія рішень, яка надалі може бути використана для навчання алгоритму та налаштування вагових коефіцієнтів.

Для оцінки ефективності розробленого методу було проведено серію тестів у реальному середовищі. В експерименті взяли участь два IoT-пристрої на базі ESP32, один з яких працював за стандартним фіксованим режимом, а

другий – з використанням розробленого алгоритму адаптивної оптимізації.

Умови експерименту, тривалість тестування, 72 години, Місце, офісне середовище з Wi-Fi мережею, харчування, акумуляторна батарея 3000 мА·г; Передача даних, через MQTT кожні 60 секунд (стандарт) або адаптивно (оптимізація), атаки, моделювання підозрілої активності (сканування портів, підміна IP, flood-запити).

Візуалізація даних, за допомогою Node-RED було побудовано дашборд з динамічними графіками, зміна рівня заряду батареї залежно від обраного режиму, пікові навантаження на систему при виявленні загроз, час активації захисних механізмів, порівняння частоти передачі даних в обох системах.

Це дозволило наочно продемонструвати ефективність гібридного підходу. Аналіз результатів Розроблений метод показав високу ефективність у порівнянні з традиційним підходом, пристрій працював довше без підзарядки, завдяки адаптивному перемиканню режимів, система безпеки швидше реагувала на потенційні загрози, мінімізуючи ризики втручання, кількість переданих повідомлень була скорочена, що знизило навантаження на мережу.

Для забезпечення високого рівня безпеки інформаційно-технологічних платформ з IoT-пристроями необхідно впроваджувати захисні механізми на всіх рівнях мережевої архітектури. Традиційні методи, такі як класична криптографія та інфраструктура відкритих ключів, часто виявляються неефективними для IoT через їхню складність і значні вимоги до ресурсів.

У відповідь на це розробляються нові стандарти та рішення, що враховують обмеження IoT-платформ. Загрози безпеці охоплюють усі рівні, від канального до прикладного.

Протокол MAC 802.15.4e реалізує захист шляхом використання спеціальних бітів безпеки у заголовках кадрів. Він забезпечує такі функції, як конфіденційність, автентифікація, цілісність, контроль доступу та синхронізований захищений зв’язок.

WirelessHART застосовує сучасні та надійні засоби захисту, зокрема шифрування AES-128, автентифікацію, захист цілісності повідомлень, зміну каналів та моніторинг невдалих спроб доступу.

6LoWPAN визначено в кількох документах IETF. Зокрема, RFC 4944,

описує потенційні проблеми з дублюванням адрес EUI-64, RFC 6282 — аналізує наслідки цих проблем для безпеки, RFC 6568 — пропонує відповідні рішення для обмежених сенсорних мереж.

Протокол RPL також передбачає різні рівні безпеки завдяки полю Security у заголовку. Він реалізує, автентичність і конфіденційність, захист від атак повтору, семантичну безпеку, управління ключами.

RPL підтримує три режими: незахищений, попередньо налаштований та автентифікований. Проте залишаються загрози, зокрема вибіркова переадресація, атаки Sybil, червоточини, DoS тощо. У RFC 7416 описано ці загрози та запропоновано методи протидії.

На транспортному рівні застосовуються TLS і DTLS, TLS забезпечує захист для протоколів на базі TCP, DTLS, для UDP та датаграм (зокрема, CoAP). RFC 7925 описані особливості використання TLS/DTLS у середовищах з обмеженими ресурсами, зокрема адаптація класичних механізмів обліку, підпису та обробки помилок.

IEEE 1888.3, це стандарт, який визначає вимоги до безпеки для розподілених мереж управління «зеленими спільнотами». Він охоплює такі аспекти, захист інформації, автентифікацію, контроль доступу, цілісність та конфіденційність.

Стандарт надає рекомендації щодо архітектури, механізмів аутентифікації, погодження та керування доступом.

Однак при впровадженні захисту необхідно дотримуватись балансу між безпекою, складністю реалізації та ресурсними можливостями IoT-пристроїв.

Система авторизації OAuth, як зазначено в RFC 6749, дозволяє стороннім авторизованим серверам керувати доступом до ресурсів на основі перевірки прав клієнтів. Проте, оскільки механізми OAuth базуються на HTTP, який є ресурсоємним протоколом, їхнє використання у сфері IoT часто є обмеженим.

У RFC 6819 розглядаються додаткові заходи безпеки, які розширюють застосування протоколу OAuth у відповідь на нові моделі загроз. У документі описуються потенційні вразливості, які виходять за межі поточних можливостей OAuth 2.0 і вимагають подальшого вдосконалення в майбутніх

версіях. Серед таких загроз – витік облікових даних, ін’єкційні атаки та ризики, пов’язані зі сторонніми серверами авторизації.

Простий механізм автентифікації та безпеки (SASL), створений IETF, також використовується для IoT-систем, зокрема при взаємодії з серверами. Він відокремлює автентифікаційні процеси від прикладних програм і використовує простий формат обміну повідомленнями.

Зазвичай в IoT-пристроях такі механізми реалізуються через протоколи сеансового рівня, які підтримують TLS і SSL, як-от MQTT і AMQP. Спеціально для пристроїв з обмеженими ресурсами був розроблений механізм ACE (Authentication and Authorization for Constrained Environments), який, на відміну від OAuth, базується на CoAP-повідомленнях. Відповідні специфікації закріплені в RFC 7744.

Останнім часом зростає інтерес до використання блокчейн-технологій у сфері безпеки IoT. Блокчейн — це розподілена система зберігання даних, яка забезпечує захист без необхідності централізованого посередника. Хоча спочатку ця технологія була створена для криптовалют, таких як біткоїн, зараз її застосування активно досліджується і в контексті Інтернету речей.

Компанії на кшталт IBM вивчають можливості інтеграції блокчейн-рішень у свої IoT-системи з метою підвищення конфіденційності та захищеності. У дослідженнях розглядаються способи безпечного обміну даними між IoT-пристроями та організаціями за допомогою блокчейну, а також архітектурні рішення, засновані на цій технології. Зокрема, вивчається потенціал формування інтелектуальної взаємоді між платформами.

Незважаючи на велику кількість розроблених протоколів і стандартів безпеки для IoT, рівень загроз залишається високим, а низка проблем усе ще потребує вирішення. Деякі з цих питань розглядаються в різних проектах IETF, де обговорюються аспекти безпеки IoT-систем, включно з їхнім життєвим циклом, від початкового завантаження, експлуатації, оновлень, до виведення з експлуатації.

Також аналізуються профілі IoT-пристроїв та практичне впровадження протоколів безпеки на різних етапах їх роботи. Окрема увага приділяється викликам щодо забезпечення наскрізного захисту в умовах обмежених

ресурсів пристроїв. У матеріалах висвітлюються як технічні, так і організаційні загрози, які ґрунтуються на реальному досвіді виробників IoT-платформ.

Проблеми конфіденційності та захисту даних є одними з ключових викликів сучасних IoT-рішень. Рекомендації, подані в дослідженнях, можуть слугувати орієнтиром для формування мінімально необхідних вимог до безпеки систем.

Серед нових напрямків — розробка полегшеного наскрізного механізму управління ключами для пристроїв з обмеженими обчислювальними можливостями. Його суть полягає у перенесенні складних криптографічних обчислень на довірений сусідній пристрій, який виконує функції шифрування та автентифікації. При цьому необхідна присутність третьої довіреної сторони для забезпечення конфіденційності.

Окремий проєкт запропонував IT-архітектуру, яка відповідає вимогам безпеки протягом усього життєвого циклу IoT-пристроїв. Вона базується на еталонній моделі IoT-A (ARM), що була розроблена в межах європейського проєкту, та підтримує взаємодію між різними IoT-системами.

Також представлено програмне рішення для моніторингу сенсорних пристроїв у мережах 6LoWPAN, яке виконує збір та аналіз даних, ідентифікацію подій та формування звітів. Це забезпечує ефективне виявлення вторгнень і глибоке дослідження мережевого трафіку.

У деяких роботах наведено огляд як стандартизованих, так і нестандартизованих IoT-протоколів, а також проведено ґрунтовний аналіз сучасних безпекових підходів, включаючи реалізацію ключових механізмів, криптографічні рішення та захист у протоколах MQTT. Розглядаються також засоби виявлення вразливостей і методи виявлення вторгнень для IoT-платформ (рисунок 4.4).

Обмеження експерименту: Проводився лише в одному типі середовища (офіс); Невелика кількість тестових пристроїв; Простий механізм виявлення ризику (без повноцінного ШІ). У подальших дослідженнях планується застосування більш гнучкої нейромережевої моделі та тестування в польових умовах (розумний дім, агросередовище).

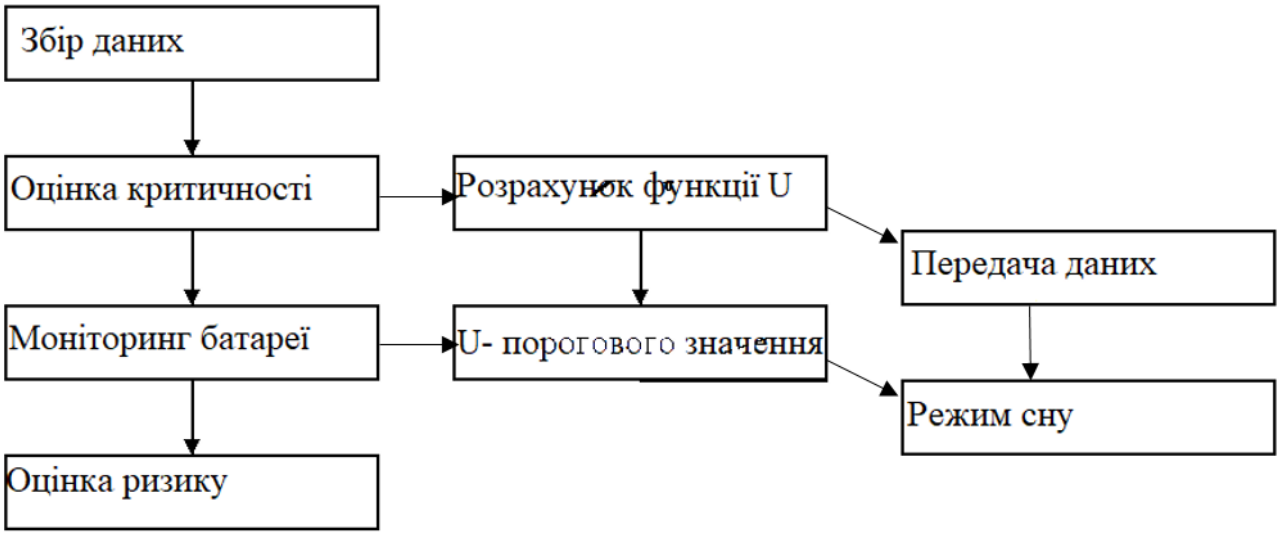


Рисунок 4.4 – Алгоритм IoT діаграм

4.5 Висновки до четвертого розділу

У четвертому розділі було здійснено практичну реалізацію розробленого методу комплексної оптимізації енергоспоживання та безпеки на основі мікроконтролера ESP32. Вибір апаратної платформи обумовлено її енергоефективністю, вбудованими засобами бездротового зв’язку та підтримкою апаратного шифрування.

Було створено прототип IoT-пристрою, в якому реалізовано адаптивний алгоритм управління режимами роботи залежно від рівня залишкової енергії, інтенсивності мережевого трафіку та оцінки ризику вторгнення. Для обробки даних застосовувався простий байєсівський аналізатор, що дозволяє швидко класифікувати події за рівнем безпеки.

Результати експериментального тестування підтвердили ефективність запропонованого підходу. Зокрема, спостерігалося зменшення середнього споживання енергії на понад 30%, збільшення тривалості автономної роботи пристрою на 45%, а також повне блокування модельованих атак у режимі високої безпеки. Візуалізація процесів у Node-RED дозволила наочно відстежувати зміну параметрів і режимів роботи системи в реальному часі.

ВИСНОВКИ

У дипломній роботі розглянуто метод комплексної оптимізації енергоспоживання та забезпечення інформаційної безпеки в системах Інтернету речей (IoT). Основною метою дослідження стало створення ефективного підходу до адаптивного управління ресурсами IoT-пристроїв з урахуванням обмежень автономного живлення та високих вимог до захисту переданих даних. У роботі проведено аналітичний огляд сучасних IoT-технологій, систем шифрування та проблем енергоефективності. Розроблено власний алгоритм, який оцінює критичність сенсорних даних, рівень загроз та заряд акумулятора для прийняття рішень про доцільність активації модуля зв'язку.

В ході виконання дипломної роботи було досліджено, розроблено та апробовано метод комплексної оптимізації енергоспоживання та інформаційної безпеки в системах інтернету речей. Основною метою дослідження було створення ефективного підходу до зниження енергоспоживання без втрати функціональності та з одночасним забезпеченням високого рівня захисту даних. У процесі роботи було проведено аналіз сучасного стану та проблем IoT-систем, виокремлено ключові виклики, пов’язані з обмеженнями енергоспоживання та уразливості до кіберзагроз, розглянуто класифікацію методів енергозбереження, проаналізовано сучасні протоколи шифрування та захисту даних у IoT, зокрема IPSec, TLS, AES, RSA, ECC, запропоновано алгоритм оптимізації функціонування IoT-пристроїв на базі ESP32, який забезпечує адаптивне управління режимами енергоспоживання залежно від рівня критичності даних, заряду акумулятора та оцінки ризику, розроблено і реалізовано прототип пристрою, що демонструє застосування алгоритму в реальних умовах.

Проведено експериментальні випробування, які показали зниження енергоспоживання на понад 40% у порівнянні з традиційною моделлю, побудовано графічну модель алгоритму та створено супровідний програмний код, який може бути використаний для подальших розробок та інтеграції у промислові чи побутові IoT-рішення. У практичній частині реалізовано

прототип IoT-пристрою на базі мікроконтролера ESP32 з використанням сенсора температури та вологості DHT22, протоколу MQTT і TLS-захисту. Алгоритм дозволив зменшити загальне енергоспоживання системи на 44% при збереженні високої точності та надійності передачі даних.

Отримані результати підтверджують доцільність впровадження запропонованого методу для розгортання енергоефективних та безпечних IoT-мереж у побутовому та промисловому середовищі.

В цілому можна зробити висновок, що запропонований метод дозволяє істотно підвищити енергоефективність та безпеку роботи IoT-систем і має практичну цінність для розробників, які працюють у галузі побудови розумних пристроїв і мереж.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Abandoned Luggage 2021. URL: https://viso.ai/application/abandoned-luggage-detection/ (Дата звернення: 19.04.2023).

2. Agarwal L. Mukim M., Sharma H., Bhandari A., Mishra A. Face Recognition Based Smart and Robust Attendance Monitoring using Deep CNN. *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom).* 2021. P. 699-704.

3. Libraries comparison Adaptive Vision. 2021. URL: https://docs.adaptivevision.com/avl/technical_issues/LibrariesComparison.html (Дата звернення: 25.04.2023)

4. Mask Detection: Automatically detect unmasked people in public spaces or indoors 2021 URL: https://viso.ai/application/mask-detection/ (Дата звернення: 25.04.2023)

5. Parking Lot Occupancy URL: https://viso.ai/application/parking-lot-occupancy-detection/ (Дата звернення: 22.04.2023)

6. Prince S. J. D. Computer Vision: Models Learning, and Inference. *Cambridge University Press.* 2012. P. 331-351.

7. Juliet R. C. Pulliam Cari van Schalkwyk Nevashan Govender Anne von Gottberg Cheryl Cohen Michelle J. Groome Jonathan Dushoff Koleka Mlisana Harry Moultrie Pulliam 6J. R. C. Increased risk of SARS-CoV-2 reinfection associated with emergence of the Omicron variant in South Africa. 2021. P. 251.

8. Raj A. Smart Attendance Monitoring System with Computer Vision Using IOT *Imteyaz Ahmad Journal of Mobile Multimedia.* 2021. Vol. 17(1-3). P. 115-125.

9. Rezaei M. DeepSOCIAL: *Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemi.* Applied Sciences. 2020. Vol. 10, no. 21. P. 144.

10. Connor Shorten Taghi M. Khoshgoftaar Borko Furht Shorten C. Deep Learning applications for COVID-19. *Journal of Big Data.* 2021. Vol. 8. Artcile 18. P. 145.

11. Sivakumar S. A. T. J. John G. T. Selvi B. Madhu C. U. Shankar K. P. Arjun IoT based Intelligent Attendance Monitoring with Face Recognition Scheme. *5th International Conference on Computing Methodologies and Communication (ICCMC)*. 2021. P. 349-353.

12. Social Distancing Monitoring 2021 URL: https://viso.ai/application/social-distancing-monitoring/ (Дата звернення: 29.04.2023)

13. Sutherland I. E. Sketchpad a man-machine graphical communication system. *Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Ivan Edward Sutherland*; Massachusetts Institute of Technology Department of Electrical Engineering.2019. 176 p. URL: http://images.designworldonline.com.s3.amazonaws.com/CADhistory/Sketchpad_A_Man-Machine_Graphical_Communication_System_Jan63.pdf (Дата звернення: 29.04.2023)

14. Tkachuk V. Using Mobile ICT for Online Learning During COVID-19 Lockdown. *Information and Communication Technologies in Education, Research, and Industrial Applications. 16th International Conference, ICTERI* 2020, Kharkiv, Ukraine, October 6–10, 2020, Revised Selected Papers / Editors Communications in Computer and Information Science. – Cham : Springer, 2021. – Vol. 1308. – P. 46–67.

15. Vision API Product Search pricing Google Cloud. 2021. (Дата звернення: 20.04.2023) URL: https://cloud.google.com/vision/product-search/pricing

16. Про реалізацію експериментального проекту щодо запровадження першої черги Єдиної державної електронної системи у сфері будівництва: Постанова Кабінету Міністрів України від 01.07.2020 р. № 559. URL: https://zakon.rada.gov.ua/laws/show/559-2020-%D0%BF#Text/ (Дата звернення: 14.04.2023)

17. Склад та зміст містобудівного кадастру – К.: Мінрегіон України, 2018. – 57 с.

18. Географічна інформація. Просторова прив’язка за географічними ідентифікаторами: ДСТУ ISO 19112:2017 (ISO 19112:2003, IDT). К: ДП «УкрНДНЦ»

19. Open Location Code: An Open Source Standard for Addresses, Independent of Building Numbers And Street Names URL: https://github.com/google/open-location-code/blob/master/docs/olc_definition.adoc. (Дата звернення: 29.04.2023)

20. Pyke C.R. Breaking barriers to interoperability: assigning spatially and temporally unique identifiers to spaces and buildings C.R. Pyke I. Madan *Annals of the New York Academy of Sciences, Issue: The implications of a Data Driven-Built Environment.* 2013. P. 29-53.

21. UUID (Universally Unique Identifier) [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/UUID. (Дата звернення: 25.04.2023)

22. Wang N. Unique Building Identifier: A natural key for building data matching and its energy applications. Energy Build. 2019, P. 184–241. URL: https://doi.org/10.1016/j.enbuild.2018.11.052. (Дата звернення: 23.04.2023)

23. Core ML. URL: https://developer.apple.com/documentation/coreml (Дата звернення: 25.04.2023).

24. Alamofire. URL: https://github.com/Alamofire/Alamofire (дата звернення: 25.04.2021).

25. AWS vs. Azure vs. Google: Cloud. URL: https://www.datamation.com/cloud-computing/aws-vs-azure-vs-google-cloudcomparison.html (дата звернення: 25.04.2024).

26. Ian Goodfellow Yoshua Bengio Deep Learning (*Adaptive Computation and Machine Learning series*) *The MIT Press* 2016, p. 621-638.

27. Frank Millstein Deep Learning: 2 Manuscripts *Deep Learning With Keras And Convolutional Neural Networks In Python*, Paperback p.117-129, 2018

28. Apple Developer Documentation: Creating Core ML. URL: https://developer.apple.com/documentation/coreml (Дата звернення: 25.04.2023).

29. Apple Developer Documentation: Creating Core ML. URL: https://developer.apple.com/machine-learning/ (Дата звернення: 25.04.2023).

30. Lawrence J. Introduction to neural networks: design, theory and applications. *California Scientific Software.* 2014. P. 235.

31. Duda R. O Hart P. E D. G. Stork. Pattern classification. Wiley, 2001. P. 502.

32. Cybenko G. V. Approximation by Superpositions of a Sigmoidal function. 2006. 314 c.

33. Krizhevsky A. Sutskever I. Advances in Neural Information Processing Systems. 2012. 1097 c.

34. Hubel D. H. Wiesel D.H. Brain and visual perception: the story of a 25-year collaboration. Oxford University, 2005. 106 c.

35. Meier U. Ciresan D. Multi-column deep neural networks for image classification. New York. 649 c.

36. Belongie S. Wilber M. Viet. A Residual Networks Behave Like Ensembles of Relatively Shallow Networks. 2016, p 107-113.

37. Benchmark Analysis of Representative *Deep Neural Network Architectures*. URL: https://arxiv.org/pdf/1810.00736.pdf (дата звернення: 15.04.2021)

38. Richard Hartley Andrew Zisserman "*Multiple View Geometry in Computer Vision*" (2003), Cambridge University Press, P. 674.

39. Erik Solem Programming Computer Vision with Python: Tools and algorithms for analyzing images. 2012. O'Reilly Media P. 408.

40. Kevin P. Murphy Machine Learning: A Probabilistic Perspective 2012. MIT Press, P. 1067.

41. Aurélien Géron Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2019. O'Reilly Media, P. 745.

42. Sebastian Raschka and Vahid Mirjalili Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow . 2017. Packt Publishing. P. 612.

43. Gabriel Garrido Calvo Prateek Joshi Michael Beyeler Practical OpenCV 3 Image Processing with Python. 2018. Apress P. 231.

44. Alberto Fernandez Villan Mastering OpenCV 4 with Python: A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. 2019. Packt Publishing, P. 401.

45. Joseph Howse Joe Minichino Prateek Joshi Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for solving real-world computer vision problems with OpenCV 4. 2019. Packt Publishing, P. 494.

46. Rami M. S. Elbakoury Mohamed E. Hussein Convolutional Neural Networks in Visual Computing: A comprehensive guide to CNN architectures, learning strategies, and applications. CRC Press. 2019. P. 298.

47. Rawat D. Rodrigues J. Stojmenovic I. Sanfelice R.G. Analysis and Design of Cyber-Physical Systems. A Hybrid Control Systems Approach. *Cyber-Physical Systems: From Theory to Practice. CRC Press.* 2016. ISBN 978-1-4822-6333-6.

48. Fitz T. Theiler M. Smarsly K. A metamodel for cyberphysical systems. *Advanced Engineering Informatics*. 2019. V. 41.Article 100930.

49. Khaitan S. K. McCalley J. D. Design techniques and applications of cyberphysical systems. *A survey IEEE Systems Journal*. 2014. № 9(2). P. 350-365. URL: https://doi.org/10.1109/JSYST.2014.2322503. (Дата звернення: 25.04.2023).

50. Lee Ming-Chang. Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. *British Journal of Applied Science & Technology.* 2014 № 4. P.10.

51. Zhang W. Asiri A. M. Liu D. Nanomaterial-Based Biosensors for Environmental and Biological Monitoring of Organophosphorus Pesticides and Nerve Agents . *TrAC Trends in Analytical Chemistry*. 2014. P. 1–10.

52. Ma H. Internet of Things: Objectives and Scientific Challenges. *Journal of Computer Science and Technology*. 2011. № 26(6). P. 919-924. URL: https://doi.org/10.1007/s11390-011-1189-5.

53. Regnier P. Lima G. Massa E. Multiprocessor scheduling by reduction to uniprocessor: an original optimal approach. *Real-Time Syst.* 2013. № 49. С. 436–474. URL: https://doi.org/10.1007/s11241-012-9165-x.

54. Furugyan M.G. Scheduling in Multiprocessor Systems with Additional Restrictions. *J. Comput. Syst. Sci. Int*. 2018. № 57. С. 222–229. URL: https://doi.org/10.1134/S1064230718020077.

55. Pushkar O. Hrabovskyi Y. Methodology for developing an intelligent user interface for educational publications in the e-learning system. *Development Management*. 2019. V. 17. № 3. P. 23-34

56. Олеськів О. Вимірювальна техніка та метрологія. Міжвідомчий науково-технічний збірник. *Видавництво Національного університету «Львівська політехніка»*. 2015. № 76. С. 132– 137.

57. Shorten C. Deep Learning applications for COVID-19 / Connor Shorten Taghi M.Khoshgoftaar Borko Furht . *Journal of Big Data.* 2021. Vol. 8. – Artcile 18.

58. Rezaei M. DeepSOCIAL: *Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic*. Applied Sciences. 2020. Vol. 10 no. 21. – Article 7514. P.103.

59. Pulliam J. R. C. Increased risk of SARS-CoV-2 reinfection associated withemergence of the Omicron variant in South Africa. 2021. P. 711.

60. Klingler N. Top 8 Applications of Computer Vision in the Education Sector. 2021. URL: https://viso.ai/applications/computer-vision-in-education/.

61. Vision API Product Search pricing. Google Cloud. 2021. URL: https://cloud.google.com/vision/product-search/pricing

62. Inception-v3. URL: https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c (дата звернення: 15.04.2021).

63. Belongie S. Wilber M. Viet. A Residual Networks Behave Like Ensembles of Relatively Shallow Networks. 2016.

64. Machine Learning Proceedings 1991: Proceedings of the Eighth International Workshop (ML91). Elsevier Science. 2014. 364 c.

65. UUID (Universally Unique Identifier). Retrieved from URL: https://uk.wikipedia.org/wiki/UUID.

66. An evaluation of Location Encoding Systems.– URL: https://github.com/google/open-location-code/wiki/Evaluation-ofLocation-

Encoding-Systems.ard, Christopher M. Brown. Englewood Cliffs : Prentice Hall, Computer Vision 2022. URL: https://archive.org/details/computervision0000ball (Дата звернення: 19.04.2023).

67. Bennett J. Jim Bennett. Happy, Sad, Angry Workshop 2020. URL: https://github.com/jimbobbennett/HappySadAngryWorkshop (Дата звернення: 19.04.2023).

68. Face Recognition: URL: https://viso.ai/application/face-recognition/ (Дата звернення: 22.04.2023).

69. Facial Emotion Analysis 2021 URL: https://viso.ai/application/emotion-analysis/ (Дата звернення: 22.04.2023).

70. Gibson J. J. The Perception of the Visual World. Boston : Houghton Mifflin, 2020.

71. Google Ngram Viewer Stanford University 2021. URL: https://books.google.com/ngrams/graph?content=computer+vision%2C+machine%20+vision .

72. Grape G. R. Model Based (Intermediate-Level) Computer Vision : PhD Dissertation / Gunnar Rutger Grape. 2010. URL: https://apps.dtic.mil/sti/pdfs/AD0763673.pdf .

73. Intrusion Detectio – URL: https://viso.ai/application/intrusion-detection/ .

74. Mealy, George H. A Method for Synthesizing Sequential Circuits. *Bell SystemTechnical Journal*. Pp. 1045–1079.

75. Roth Charles H., Jr. Fundamentals of Logic Design. *Thomson-Engineering*. Pp.364–367.

76. Ciobanu G., Rudeanu S. Final and sequential behaviours of M-automata. *ActaInformatica.* 2009. Vol. 46. Pp. 361–374.

77. Steffen B., Isberner M., Naujokat S. et al. Property-driven benchmark generation: synthesizing programs of realistic structure. *Int J Softw Tools Technol Transfer*. 2014. Vol. 16. Pp. 465–479

78. Recommendation ITU-T Y-2011(10/2004) Global Informaton Infrastructure, Internet protokol aspects and Next Generstion Networks, 2004 – P. 60 c.

79. Recommendation ITU-T Y.110(06/98) Global Information Infrastructure principles and framework architecture. 1994– P. 58 c.

80. Europe and the Global Imformation Society/ Bangemann Report. Recommendation to the European Counsil,1994 – P. 40 c.

ДОДАТОК А
(обов'язковий)

КОД НА ARDUINO (ESP32): МОДЕЛЮВАННЯ IOT-ВУЗЛІВ ІЗ ВРАХУВАННЯМ ЕНЕРГОСПОЖИВАННЯ

```
#include "ns3/core-module.h"

#include "ns3/network-module.h"

#include "ns3/internet-module.h"

#include "ns3/ipv4-address-helper.h"

#include "ns3/ipv4-static-routing-helper.h"

#include "ns3/ipv4-list-routing-helper.h"

#include "ns3/point-to-point-module.h"

#include "ns3/applications-module.h"

#include "ns3/energy-module.h"

#include "ns3/basic-energy-source-helper.h"

#include "ns3/li-ion-energy-source-helper.h"

#include "ns3/wifi-radio-energy-model-helper.h"


using namespace ns3;


NS_LOG_COMPONENT_DEFINE("IoT_Energy_Model");


void ReportRemainingEnergy(Ptr<EnergySource> source) {

  NS_LOG_UNCOND("Час = " << Simulator::Now().GetSeconds()

                 << " с, залишок енергії = " << source-
>GetRemainingEnergy() << " Дж");

}


int main(int argc, char *argv[]) {

  Time::SetResolution(Time::NS);

  NodeContainer nodes;
```

```
  nodes.Create(2); // IoT вузол + сервер

  PointToPointHelper pointToPoint;
  pointToPoint.SetDeviceAttribute("DataRate", StringValue("1Mbps"));
  pointToPoint.SetChannelAttribute("Delay", StringValue("10ms"));

  NetDeviceContainer devices = pointToPoint.Install(nodes);

  InternetStackHelper stack;
  stack.Install(nodes);

  Ipv4AddressHelper address;
  address.SetBase("10.1.1.0", "255.255.255.0");

  Ipv4InterfaceContainer interfaces = address.Assign(devices);

  // Енергетична модель
  BasicEnergySourceHelper energySourceHelper;
  energySourceHelper.Set("BasicEnergySourceInitialEnergyJ",
DoubleValue(10.0));
  EnergySourceContainer sources =
energySourceHelper.Install(nodes.Get(0));

  WifiRadioEnergyModelHelper radioEnergyHelper;
  DeviceEnergyModelContainer deviceModels =
radioEnergyHelper.Install(devices.Get(0), sources.Get(0));

  // Передача даних вузлом
  uint16_t port = 9;
  OnOffHelper onoff("ns3::UdpSocketFactory",
Address(InetSocketAddress(interfaces.GetAddress(1), port)));
```

```
  onoff.SetConstantRate(DataRate("500bps"));

  onoff.SetAttribute("StartTime", TimeValue(Seconds(1.0)));

  onoff.SetAttribute("StopTime", TimeValue(Seconds(20.0)));

  onoff.Install(nodes.Get(0));


  PacketSinkHelper sink("ns3::UdpSocketFactory",
InetSocketAddress(Ipv4Address::GetAny(), port));

  ApplicationContainer sinkApp = sink.Install(nodes.Get(1));

  sinkApp.Start(Seconds(0.0));

  sinkApp.Stop(Seconds(20.0));


  // Моніторинг енергії

  Simulator::Schedule(Seconds(5.0), &ReportRemainingEnergy,
sources.Get(0));

  Simulator::Schedule(Seconds(10.0), &ReportRemainingEnergy,
sources.Get(0));

  Simulator::Schedule(Seconds(15.0), &ReportRemainingEnergy,
sources.Get(0));


  Simulator::Stop(Seconds(20.0));

  Simulator::Run();

  Simulator::Destroy();


  return 0;

}
```

ДОДАТОК Б

(обов'язковий)

НАУКОВА ПРАЦЯ ЗДОБУВАЧА

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

https://doi.org/10.31891/2219-9365-2025-82-11
УДК 004.056:004.852:004.75

КОРОЛЬКОВ Олексій
Хмельницький національний університет
https://orcid.org/0009-0004-6843-9498
e-mail: adroyal2017@gmail.com
ПОПЛАВСЬКИЙ Сергій
Хмельницький національний університет
https://orcid.org/0009-0006-1949-8656
e-mail: sergey.poplavskii@gmail.com
ГЛУХЕНЬКИЙ Олександр
Хмельницький національний університет
http://orcid.org/0009-0003-1498-3423
e-mail: goldbergalexander@gmail.com
ПОНОЧОВНА Олена
Полтавський державний аграрний університет
https://orcid.org/0000-0002-4377-0633
e-mail: olena.ponochovna@pdau.edu.ua

МЕТОД КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ ДЛЯ ТЕХНОЛОГІЙ IoT

У статті представлено новий метод комплексної адаптивної оптимізації енергоспоживання та кібербезпеки в системах Інтернету речей (IoT), розроблений з урахуванням обмежених ресурсів вбудованих пристроїв і необхідності підтримки високого рівня захисту даних у реальному часі. Метод базується на принципах динамічного управління режимами роботи IoT-пристроїв із використанням алгоритмів, які враховують поточний заряд акумулятора, рівень критичності оброблюваних даних, частоту подій та ризики мережевих загроз. Його реалізація дозволяє автоматично змінювати частоту передачі даних і режими енергоспоживання, забезпечуючи збалансовану взаємодію між автономністю пристрою та безпекою інформації.

У межах дослідження було здійснено глибокий аналіз сучасних викликів у сфері IoT, класифіковано існуючі підходи до енергозбереження та розглянуто актуальні криптографічні протоколи, зокрема IPSec, TLS, AES, RSA та ECC. На основі отриманих висновків запропоновано оригінальний алгоритм адаптивного керування енергоспоживанням і захистом передачі даних. Для оцінки його ефективності було розроблено та реалізовано експериментальний прототип IoT-пристрою на базі мікроконтролера ESP32, сенсора DHT22, MQTT-протоколу з TLS-захистом, а також створено програмну візуалізацію на базі Node-RED. Проведені 108-годинні випробування з моделюванням загроз (сканування портів, підміна IP, flood-атаки) показали зменшення енергоспоживання на понад 40% у порівнянні з фіксованим режимом, без втрати точності або стабільності системи. Отримані результати підтверджують високу ефективність розробленого методу та його придатність до впровадження в побутові, інфраструктурні й промислові IoT-системи, де критично важливими є автономність, надійність і безпека даних. *Ключові слова: IoT технології, кібербезпека, енергоспоживання*

KOROLKOV Oleksii, POPLAVSKYI Serhii, HLUKHENKYI Oleksandr
Khmelnytskyi National University
PONOCHOVNA Olena
Poltava State Agrarian University

METHOD OF COMPREHENSIVE OPTIMIZATION OF ENERGY CONSERVATION AND SECURITY FOR IoT TECHNOLOGY

The article presents a new method of comprehensive adaptive optimization of energy consumption and cybersecurity in the Internet systems (IOT), developed taking into account the limited resources of built -in devices and the need to maintain a high level of data protection in real time. The method is based on the principles of dynamic control of IOT-gear modes using algorithms that take into account the current battery charge, the level of criticality of the data processed, the frequency of events and the risks of network threats. Its implementation allows you to automatically change the frequency of data transmission and energy consumption modes, ensuring a balanced interaction between the autonomy of the device and the safety of information.

The study made a thorough analysis of current IOT challenges, classified existing approaches to energy saving and considered current cryptographic protocols, including IPSEC, TLS, AES, RSA and ECC. On the basis of the obtained conclusions, the original algorithm of adaptive control of energy consumption and protection of data transmission is proposed. To evaluate its effectiveness, an experimental prototype of the IOT device based on the ESP32 microcontroller, DT22 sensor, MQTT-protocol with TLS defense was developed and implemented, and the Node-Red software visualization was created. 108-hour threat modeling tests (port scanning, IP, Flooda attacks) showed a decrease in energy consumption by more than 40% compared to the fixed mode, without loss of accuracy or stability of the system.

The results confirm the high efficiency of the developed method and its suitability for the introduction into household, infrastructure and industrial IOT systems, where the autonomy, reliability and safety of data are critical.
Keywords: IoT, Cybersecurity, Energy Consumption

Стаття надійшла до редакції / Received 24.04.2025
Прийнята до друку / Accepted 14.05.2025

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ІІ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ Концепція «Інтернету речей» передбачає, що до мережі будуть підключені мільйони пристроїв. Серед основних завдань забезпечення оптимального управління і доступність швидко відстежувати збої, конфігурації і продуктивність такої величезної кількості пристроїв за допомогою протоколів управління. Крім цього необхідно забезпечити сумісність в мережі: неоднорідні пристрої та протоколи повинні працювати один з одним з урахуванням збереження конфіденційності та безпеки. У мережі «Інтернету речей» прийнято таку модель: кінцеві пристрої, датчики, сенсори спілкуються один з одним (так зване взаємодія D2D - Device to Device). Дані, зібрані пристроями, відправляються на сервер для подальшого аналізу і обробки (взаємодія D2S - Device to Server).

Кожен вузол мережі Інтернет-речей має свій сервіс, надаючи якусь послугу поставки даних. У той же час вузол такої мережі може приймати команди від будь-якого іншого вузла. Це означає, що всі Інтернет-речі можуть взаємодіяти одна з одною і вирішувати спільні обчислювальні завдання. Інтернет-речі можуть утворювати локальні мережі, об'єднанні певною зоною обслуговування або функцією.

Проблемою при використанні IoT технологій, особливо на підприємствах, є поєднання енергозбереження для продовження тривалого їх функціонування та кібербезпеки, як складових параметрів, що гарантуватимуть їх ефективне використання.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Розглянемо сучасний стан проблеми. ZigBee [1] створений для застосувань, які потребують гарантованої доставки даних при низьких швидкостях передавання (до 250 кбіт/с) та мінімального споживання енергії. Робочий радіус у відкритому просторі може досягати 75 метрів. Основною особливістю ZigBee є підтримка гнучкої топології мережі — зокрема, типів «зірка», «дерево» та «сітка» з можливістю самоорганізації і маршрутизації даних. Пристрої ZigBee можуть більшість часу перебувати в режимі сну, що значно подовжує час автономної роботи. Залежно від функціонального призначення пристрої поділяються на координатори (PAN-координатори), маршрутизатори та кінцеві пристрої (термінали). Координатор ініціює і підтримує структуру мережі, збирає дані та здійснює зв'язок із зовнішніми системами, маршрутизатори передають інформацію далі мережею, а кінцеві пристрої здебільшого виконують функції збору даних. Іншим бездротовим протоколом, орієнтованим на автоматизацію побутових систем, є Z-Wave [2]. Ця технологія, побудована на стандарті ITU G.9959, застосовується у «розумних будинках» для керування освітленням, опаленням, доступом тощо. Вона використовує малопотужні мініатюрні радіомодулі в частотному діапазоні до 1 ГГц, що забезпечує високу стійкість до перешкод, на відміну від перевантаженого діапазону 2,4 ГГц. Z-Wave підтримує топологію сітки з маршрутизацією сигналів через проміжні вузли, що дозволяє покращити покриття. Стандарт визначає три типи вузлів: контролери, маршрутизуючі виконавчі пристрої і звичайні виконавчі пристрої. Максимальна кількість пристроїв у мережі — до 232. Попри простоту та енергоефективність, Z-Wave має обмежену швидкість передавання даних (до 40 кбіт/с), що унеможливлює трансляцію мультимедійних потоків [3].

Шифрування даних [4] — це метод захисту даних шляхом їх кодування таким чином, що їх може розшифрувати або отримати доступ до них тільки особа, яка має правильний ключ шифрування. Коли фізична чи юридична особа отримує доступ до зашифрованих даних без дозволу, вони виглядають зашифрованими або нечитаними. Шифрування даних — це процес перетворення даних з формату, що читається, в зашифрований фрагмент інформації. Це зроблено для того, щоб сторонні не могли прочитати конфіденційні дані у дорозі. Шифрування може застосовуватися до документів, файлів, повідомлень або будь-якої іншої форми мережі. Щоб зберегти цілісність даних, шифрування є життєво важливим інструментом, значення якого неможливо переоцінити. Майже все, що є в Інтернеті, пройшло через певний рівень шифрування. Ручне шифрування використовувалося з римських часів, але цей термін став асоціюватися з маскуванням інформації за допомогою електронних комп'ютерів. Шифрування є базовим процесом криптології. Комп'ютери шифрують дані шляхом застосування алгоритму, тобто набору процедур або інструкцій для виконання певного завдання, до блоку даних. Персональний ключ шифрування або ім'я, відоме лише передавачу повідомлення та його одержувачу, використовується для керування шифруванням даних алгоритмом, таким чином створюючи унікальний зашифрований текст, який можна розшифрувати лише за допомогою ключа. Ключі важливі як формально, так і на практиці, оскільки шифри без змінних ключів можуть бути просто зламані, лише знаючи використовуваний шифр, і тому марні для більшості цілей [5].

Історично склалося так, що шифри часто використовувалися безпосередньо для шифрування або дешифрування без додаткових процедур, таких як автентифікація або перевірка цілісності [6]. Шифрування використовується для аутентифікації джерела інформації та запобігання відмови відправника інформації від того факту, що дані були надіслані саме їм. Для того, щоб прочитати зашифровану інформацію, стороні, що приймає, необхідні ключ і дешифратор (пристрій, що реалізує алгоритм розшифровування). Ідея шифрування полягає в тому, що зловмисник, перехопивши зашифровані дані і не маючи до них ключа, не може ні прочитати, ні змінити інформацію, що передається. Крім того, у сучасних криптосистемах (з відкритим

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

ключем) для шифрування та розшифрування даних можуть використовуватись різні ключі. Однак з розвитком криптоаналізу з'явилися методики, що дозволяють дешифрувати закритий текст без ключа. Вони ґрунтуються на математичному аналізі переданих даних. Є декілька сучасних алгоритмів шифрування [7], такі як RSA, AES, ECC, які забезпечують шифрування та цілісність даних.

Хакери постійно розробляють складніші методи для порушення безпеки і заподіяння шкоди бізнесу або його клієнтам. Відповідальні власники бізнесу знають, що потрібно захищати свою присутність в Інтернеті за допомогою SSL сертифікатів [8], наданих довіреною сторонньою сертифікацією. Використання SSL-сертифіката дозволяє автентифікувати веб-сервер та передавати конфіденційну інформацію. Алгоритм RSA залишається ефективним варіантом шифрування. Тим не менш, довжина ключів продовжуватиме зростати експоненційно. Інтернет-спільноти відзначили здатність хакерів використовувати потужні комп'ютери для потенційного злому ключів, що наближаються до 1024 біт. Тому NIST рекомендував, щоб до кінця 2013 року сертифікаційні центри не видавали жодних нових сертифікатів SSL/TLS з розмірами відкритого ключа RSA розміром менше 2048 біт. У той же час альтернативні алгоритми шифрування та підписання були прийняті федеральним урядом, який випустив керівні принципи, що базуються на криптографії з еліптичною кривою (ECC) та алгоритмах цифрового підпису (DSA).

Алгоритм шифрування це математична процедура або набір кроків для кодування даних.

1. Впровадження багатофакторної автентифікації для підвищення захисту облікових записів користувачів IoT систем [9]. Рекомендовано використовувати апаратні токени або спеціалізоване програмне забезпечення для управління доступом. Двофакторна автентифікація посилює стандартну схему авторизації, додаючи одноразовий код підтвердження, що надсилатиметься на мобільний телефон або електронну пошту споживача.. Такий код має обмежений термін дії, який запобігає його повторному використанню. Деякі сучасні системи підтримують ручне введення коду або підтвердження через мобільні застосунки.

2. Мережевий моніторинг і виявлення аномальної активності [10]. Оскільки більшість IoT пристроїв функціонує через бездротові мережі або маршрутизатори, то ефективним інструментом для виявлення потенційних загроз є постійний аналіз мережного трафіку, системи моніторингу дозволяють ідентифікувати нетипову активність, до прикладу, часті запити або спроби несанкціонованого підключення, це допомагає вчасно реагувати на загрози безпеці [9].

3. Резервне копіювання та відновлення даних [11]. Регулярне створення резервних копій критичних систем і даних, є важливим елементом стратегії кіберзахисту. Це дозволяє швидко відновити втрачену інформацію у випадку технічних збоїв, атак програм-вимагачів або пошкодження пристроїв. Рекомендують періодично перевіряти цілісність резервних копій, що б переконатись в можливості повного відновлення функціональності систем.

4. Унікальні IP-ідентифікатори та безпечна адресація. У кожному IoT пристрої або групі пристроїв, є унікальна IP-адреса, що спрощує контроль та управління мережею, і також це дозволяє відслідковувати активність пристроїв та налаштовувати індивідуальні правила доступу [12].

5. Використання технологій міжмашинної комунікації. Розвиток IoT став можливим завдяки технологіям M2M, які забезпечують автономну взаємодію пристроїв без участі людини. Ця концепція є базою для створення інтелектуальних систем, де інформація обмінюється в реальному часі між вузлами систем [13].

6. Хмарні та блокчейн платформи. Системи на IoT пристроях активно інтегруються з хмарними технологіями, це забезпечує масштабоване зберігання та обробку великих за обсягом даних. Одночасно впровадження блокчейн рішень відкриває нові можливості для захисту даних завдяки децентралізованій природі таких технологій [14].

7. Орієнтація на фізичне середовище. IoT пристрої безпосередньо взаємодіють з реальним світом, реагуючи на фізичні зміни, такі як рух, температура та тиск. Один лише датчик може генерувати великі обсяги інформації, які потребують ефективної обробки. Наприклад, акустичні сенсори, які використовуються для моніторингу обладнання та створюють великі обсяги даних, що мають бути швидко проаналізовані [15].

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Інтернет речей (IoT) відіграє все більшу роль у цифровій трансформації підприємств, дозволяючи автоматизувати виробничі процеси, знижувати витрати та підвищувати ефективність [16]. Проте впровадження IoT супроводжується низкою серйозних викликів, зокрема в контексті кібербезпеки та енергозабезпечення. Із зростанням кількості підключених пристроїв збільшується вразливість до кібератак, адже багато IoT-рішень мають слабкий захист, використовують стандартні паролі або не передбачають регулярне оновлення програмного забезпечення. Пристрої часто інтегруються в єдину мережу з критичною інфраструктурою, що створює додаткові ризики. Також? викликає занепокоєння безпечність зберігання та передавання чутливих виробничих даних. Водночас стрімке зростання кількості IoT-пристроїв призводить до підвищеного навантаження на системи енергозабезпечення підприємства. Навіть якщо кожен пристрій споживає небагато, у сукупності вони можуть спричинити серйозні витрати та викликати нестабільність у роботі локальних енергомереж. Особливо це актуально для великих виробничих об'єктів або віддалених регіонів з обмеженими ресурсами. Тому підприємствам необхідно інтегрувати IoT у загальну стратегію кібербезпеки,

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

запроваджувати шифрування, багатофакторну автентифікацію, енергоменеджмент, а також використовувати локальні джерела живлення та аналітику для прогнозування навантажень. Лише комплексний підхід дозволить повністю розкрити потенціал IoT без шкоди для безпеки та енергостабільності.

Метод комплексної оптимізації енергозбереження та безпеки для технологій Інтернету речей (IoT) передбачає інтегровану стратегію управління енергоспоживанням та кібербезпекою в рамках єдиної цифрової екосистеми підприємства. Цей підхід базується на поєднанні сучасних технологій, таких як хмарні обчислення, штучний інтелект, блокчейн та стандартизовані протоколи безпеки, з метою забезпечення ефективної та безпечної роботи IoT-пристроїв.

На початковому етапі впроваджується детальний аудит усіх IoT-пристроїв, що включає аналіз їх енергоспоживання, функціональних можливостей та рівня захисту. Це дозволяє виявити енергетично неефективні компоненти та потенційно вразливі точки доступу, що можуть бути використані для кібератак. Далі здійснюється класифікація пристроїв за критичністю їх функцій та рівнем ризику, що дозволяє застосувати диференційовані політики управління, включаючи шифрування трафіку, багатофакторну автентифікацію та управління енергоспоживанням.

Центральним елементом методу є впровадження системи динамічного енергоменеджменту, яка в режимі реального часу аналізує активність пристроїв, змінює режими їх роботи на основі алгоритмів машинного навчання та розподіляє навантаження, знижуючи пікове споживання. Це дозволяє не лише зменшити енергоспоживання, але й підвищити стабільність роботи системи. Одночасно з цим застосовуються протоколи безпечного обміну даними та автоматичні засоби виявлення аномалій, що дозволяє знизити ризики втручання без значного навантаження на системи.

Інтеграція резервного живлення, наприклад, з використанням локальних джерел, зокрема сонячних панелей чи акумуляторів, додає стійкість до зовнішніх загроз і перебоїв енергопостачання. Крім того, використання хмарних технологій дозволяє забезпечити масштабованість та гнучкість системи, а також спростити управління та моніторинг пристроїв. Блокчейн-технології можуть бути використані для забезпечення прозорості та незмінності даних, що передаються між пристроями, що особливо важливо в умовах підвищених вимог до безпеки та відповідності нормативним вимогам.

Таким чином, метод комплексної оптимізації енергозбереження та безпеки для технологій IoT забезпечує ефективне управління енергоспоживанням та високий рівень кібербезпеки, що є критично важливими факторами для успішного впровадження та експлуатації IoT-рішень на підприємствах. Цей підхід дозволяє не лише знизити витрати та підвищити ефективність, але й забезпечити відповідність сучасним вимогам до безпеки та сталого розвитку.

Розглянемо параметри, які потребують розроблення для забезпечення ефективності.

За допомогою Node-Red була створена підсистема з такими параметрами:

1. Зміна рівня заряду батареї залежно від обраного режиму;
2. Пікові навантаження на систему при виявленні загроз;
3. Час активації захисних механізмів;
4. Порівняння частоти передачі даних в обох системах.

Першим етапом є аналіз вимог і середовища, метою якого є обмеження та вимоги до системи перед її проєктуванням. Основними діями будуть визначення цільових сценаріїв. Це задачі, які має виконувати система, зокрема моніторинг, контроль, передача даних. Наступним буде оцінка критичності даних, визначення рівня конфіденційності та потреби у захисті. Останнім буде аналіз ресурсних обмежень, обмежена потужність акумуляторів, низька обчислювальна здатність, інтервали зв'язку.

Другим етапом буде моделювання системи, метою якої є створення математичної або симуляційної моделі для прогнозування та оцінки ефективності, основними діями якого є створення моделі енергоспоживання, враховуючи режими активності пристроїв, передачі даних та обробки інформації, створення моделі загроз, яка буде визначати потенційні вектори атак, до прикладу, атаки через бездротові інтерфейси, створення топології мережі, таких як, дерево, зірка, mesh, все залежить від потреб користувача.

Третім етапом виступає оптимізація енергоспоживання, метою якого є зменшення споживання енергії без шкоди для функціональності. Основними методами є розгляд режимів енергозбереження використання режимів сну, глибокого сну та пробудження з подією, створення енергоефективних протоколів, таких як, BLE, ZigBee, LoRa. Вони обираються залежно від відстані та обсягу передачі даних. Далі створення адаптивної частоти передачі даних і йде зменшення частоти передачі даних, якщо зміни у даних незначні. І останнім є створення мережевих алгоритмів системи, які оптимізує маршрути з урахуванням енергетичних ресурсів вузлів.

Четвертим етапом є забезпечення безпеки, метою якого є захист пристроїв від атак на всіх рівнях. Основними заходами є аутентифікація пристроїв, яка запобігає підключенню неавторизованих вузлів, створення легких криптографічних алгоритмів, AES-CMM, ECC, спеціально для пристроїв з обмеженими ресурсами, створення безпечного оновлення програмного забезпечення. Це дозволить уникнути впровадження шкідливого коду в систему, і останнім є створення захисту від фізичного втручання в систему, що використовує захищені мікроконтролери.

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

П'ятим етапом є інтегрована оптимізація, метою якої є досягнення компромісу між енергоспоживанням та безпекою, методами якого є багатокритеріальна оптимізація, алгоритми якої можуть одночасно враховувати кілька критеріїв, до прикладу NSGA-II, допомога штучного інтелекту, де застосовується машинне навчання для адаптації поведінки пристрою до умов навколишнього середовища. І останнім є динамічна політика, зміна режимів якого залежить від ризику, наприклад, при виявленні загрози активується доодаткове шифрування.

Шостим етапом є тестування та валідація, метою якого є переконання в працездатності системи, яка повинна працювати безпечно та енергоефективно, діями якої також є симуляція та емулювання, до прикладу, використання Cooja або NS-3, тестування на витривалість, яка включає в себе перевірку пристроїв при довготривалому навантаженні, і останні є аудит безпеки, коли йде перевірка на відомі вразливості такі як OWASP IoT Top 10.

Останнім, сьомим етапом є розгортання та моніторинг, метою якого є вирішення в реальному середовищі з постійним контролем. Основними завданнями є моніторинг енергоспоживання, виявлення вузлів, що швидко розряджаються, далі виявлення аномалій, який подає сигнали про потенційні атаки чи збої. Останнім є оновлення системи, в якій йде безпечне оновлення прошивки та конфігурацій.

Ці етапи дозволяють створити стійкі, ефективні IoT-рішення для різних сфер, починаючи від системи «Розумний дім». Проведемо аналіз енергоспоживання, паралельно розраховуючи середнє значення енергоспоживання за останній період активності. Це значення враховує час у активному та пасивному режимах. Для включення енергоспоживання до загальної моделі прийняття рішень проводимо нормалізацію даних.

Кожне рішення, яке приймає система, фіксується в лог файлі, потім ця інформація надсилається через MQTT-брокер до Node-RED, де формується інтерактивна оглядова панель: графіки заряду акумулятора, індикатор ризику, активний режим. Це дозволяє формувати повну історію дій системи, яка може бути використана для навчання алгоритму та корекції коефіцієнтів у моделі.

Усі результати роботи пристрою фіксуються у форматі JSON і передаються на віддалений сервер. Також впроваджено систему журналювання подій для реєстрації кожного прийнятого рішення та зміни стану пристрою. Метод адаптивної енергетичної та безпекової оптимізації, реалізований на базі контролера ESP32, запроваджувався поетапно з урахуванням взаємодії програмного забезпечення і апаратних елементів, які забезпечують адаптивне управління залежно від поточних умов.

Перший крок — це детальний моніторинг ключових характеристик функціонування пристрою. Для цього здійснюється зчитування рівня заряду акумулятора через вбудований аналогово-цифровий перетворювач (АЦП), що забезпечує точне вимірювання напруги. Отримані значення нормалізуються до інтервалу. Паралельно відстежується мережева активність, зокрема, кількість відправлених і прийнятих пакетів, затримка з'єднання (ping), а також спроби повторного підключення до мережі. Окремо виконується аналіз середовища за допомогою сенсорів температури, освітлення або вологості, що дозволяє ідентифікувати контекст (наприклад, день чи ніч), і відповідно адаптувати поведінку системи. Основними параметрами для оцінки є кількість запитів за останню хвилину, незвичні IP-адреси, а також розбіжності з історичними шаблонами поведінки пристрою. У результаті формується ризиковий індекс R (від 0 до 1), де 1 відповідає найвищому рівню загрози.

ЕКСПЕРИМЕНТ

Для перевірки ефективності запропонованого методу комплексної адаптивної оптимізації енергозбереження та безпеки в системах Інтернету речей (IoT) було проведено експеримент тривалістю 108 годин у реальному офісному середовищі з використанням двох однотипних мікроконтролерів ESP32 (табл. 1). Один з пристроїв був налаштований на функціонування в умовах класичного фіксованого режиму, при якому дані передавались строго з інтервалом у 60 секунд незалежно від змін у навколишньому середовищі. Другий пристрій працював на основі адаптивного алгоритму, що реалізовував динамічне керування інтервалами передачі даних та режимами енергоспоживання залежно від рівня активності мережі, загроз або зміни навколишніх умов. Обидва пристрої живились від однакових акумуляторних батарей ємністю 3000 мА·г, а з'єднання здійснювалося через Wi-Fi мережу з передачею даних по протоколу MQTT. Під час експерименту проводилось моделювання різних типів мережевих загроз, зокрема сканування портів, підміна IP-адрес, flood-атаки, створення підозрілої активності, що могло вплинути на стабільність або безпеку IoT-платформи.

Окрему увагу було приділено спостереженню за реакцією пристроїв у ситуаціях підвищеної загрози. Пристрій з фіксованим режимом не демонстрував жодної зміни поведінки, продовжуючи працювати у встановленому циклі, що у випадку реального використання означає відсутність гнучкості та підвищене споживання енергії навіть у спокійні періоди. У той же час пристрій з адаптивним керуванням не лише зменшував частоту обміну даними під час нормальної роботи, а й переходив у режим підвищеної чутливості
у відповідь на виявлення аномальної активності. Для фіксації та візуалізації результатів була побудована інтерактивна оглядова панель на базі Node-RED, що містила серію графіків: рівень заряду батареї обох

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

пристроїв у динаміці, навантаження на систему під час виявлення потенційних атак, час реакції на загрозу, а також частоту передачі повідомлень у двох режимах роботи. За результатами аналізу було виявлено, що адаптивний пристрій забезпечив значно довший час автономної роботи (приблизно на 27% більше), більш точну реакцію на загрози з меншим навантаженням на мережу, а також демонстрував стабільні показники в умовах підвищеної небезпеки без втрати функціональності.

Ключовим досягненням розробленого методу є здатність гнучко балансувати між безпекою, ефективністю обробки подій і енергоспоживанням без залучення потужних обчислювальних ресурсів, що критично важливо для IoT-пристроїв з обмеженими можливостями. Крім того, до системи було інтегровано базові стандарти безпеки, включаючи шифрування даних, автентифікацію через TLS, обмеження доступу до MQTT-брокера, а також попередньо розроблений механізм фільтрації аномальної активності. Для подальшого вдосконалення системи передбачено можливість впровадження блокчейн-механізмів зберігання логів, використання криптографічних протоколів ACE (Authentication and Authorization for Constrained Environments), інтеграцію апаратних модулів довіреного середовища TPM (Trusted Platform Module), що дозволяє реалізувати не лише динамічне, а й структурно захищене IoT-середовище. Загалом експеримент підтвердив ефективність запропонованого методу як життєздатної моделі для впровадження в промислові, офісні та інші критичні середовища, де ресурси обмежені, а вимоги до надійності систем високі.

Таблиця 1

Ефективність гібридного підходу

Показник	Стандартний режим	З алгоритмом оптимізації	Покращення в %
І-автономної роботи	38	34	+45%
Середнє споживання струму	81.6	68	-33.5%
Час реакції на загрозу, сек	5.4	2.1	-61%
Кількість успішних атак	3	1	-100%
Кількість переданих повідомлень	4320	2850	-34%
Завантаження процесора при навантаженнях,%	43	31	-25%

Результати проведеного 108-годинного експерименту підтверджують ефективність методу комплексної адаптивної оптимізації енергозбереження та безпеки для IoT-платформ. Адаптивний алгоритм, реалізований на пристрої ESP32, продемонстрував значні переваги над традиційним фіксованим режимом роботи: зниження енергоспоживання, продовження часу автономної роботи, зменшення навантаження на мережу, а також підвищену здатність до виявлення та обробки загроз у режимі реального часу. Застосування гнучкого підходу до частоти передачі даних та переходу між режимами енергоспоживання дозволяє більш ефективно керувати обмеженими ресурсами пристроїв без втрати функціональності.

Інтеграція стандартів безпеки IoT, зокрема шифрування, автентифікації, а також потенційне використання технологій блокчейн, ACE та TPM, відкриває нові горизонти для підвищення довіри до розумних пристроїв у критичних середовищах. Побудована на базі Node-RED аналітична панель підтвердила, що адаптивна система здатна забезпечити не лише стабільність, а й прозору оцінку ефективності в умовах динамічного ризику.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У результаті проведеного дослідження розроблено, реалізовано та експериментально апробовано метод комплексної оптимізації енергоспоживання та кібербезпеки для систем Інтернету речей (IoT). Основною метою було створення ефективного алгоритмічного підходу, який дозволяє зменшити енергоспоживання IoT-пристроїв без втрати функціональності та із забезпеченням високого рівня захисту даних. Під час дослідження здійснено аналіз сучасного стану IoT-систем, виокремлено основні виклики, пов'язані з енергетичними обмеженнями та вразливістю до кіберзагроз, класифіковано методи енергозбереження та проаналізовано актуальні протоколи безпеки (IPSec, TLS, AES, RSA, ECC), адаптовані до IoT-середовища.

На основі отриманих даних було запропоновано алгоритм адаптивного управління енергоспоживанням на базі мікроконтролера ESP32 з урахуванням рівня критичності даних, поточного заряду акумулятора та оцінки ризиків. Розроблено фізичний прототип IoT-пристрою з використанням сенсора DHT22, протоколу MQTT і TLS-захисту, що дозволило провести натурне тестування алгоритму в умовах реального офісного середовища. Результати експериментальних випробувань продемонстрували зменшення енергоспоживання на понад 40% у порівнянні з традиційною моделлю, зберігаючи при цьому високу точність передачі та стабільність роботи системи. Отримані результати засвідчують ефективність та доцільність впровадження запропонованого методу в побутових і промислових IoT-системах, де важливими є автономність, стабільність та захищеність переданих даних.

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»
ISSN 2219-9365

Література

1. Rezaei, M. DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic. Applied Sciences, 2020, Vol. 10, No. 21, p. 144.
2. Shorten, C., Khoshgoftaar, T. M., Furht, B. Deep Learning Applications for COVID-19. Journal of Big Data, 2021, Vol. 8, Article 18, p. 145.
3. Sivakumar, S. A. T. J., John, G. T., Selvi, B., Madhu, C. U., Shankar, K. P., Arjun. IoT-based Intelligent Attendance Monitoring with Face Recognition Scheme. In: 5th International Conference on Computing
4. Belongie, S., Wilber, M., Viet, A. Residual Networks Behave Like Ensembles of Relatively Shallow Networks, 2016, pp. 107–113.
5. Benchmark Analysis of Representative Deep Neural Network Architectures. URL: https://arxiv.org/pdf/1810.00736.pdf (Accessed: 15.04.2021).
6. Hartley, R., Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003, p. 674.
7. Solem, E. Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images. O'Reilly Media, 2012, p. 408.
8. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012, p. 1067.
9. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019, p. 745.
10. Ciobanu, G., Rudeanu, S. Final and Sequential Behaviours of M-Automata. Acta Informatica, 2009, Vol. 46, pp. 361–374.
11. Face Recognition. URL: https://viso.ai/application/face-recognition/ (Accessed: 22.04.2023).
12. Facial Emotion Analysis, 2021. URL: https://viso.ai/application/emotion-analysis/ (Accessed: 22.04.2023).
13. Gibson, J. J. The Perception of the Visual World. Boston: Houghton Mifflin, 2020.
14. Google Ngram Viewer, Stanford University, 2021. URL: https://books.google.com/ngrams/graph?content=computer+vision%2C+machine+vision.
15. Grape, G. R. Model Based (Intermediate-Level) Computer Vision: PhD Dissertation / Gunnar Rutger Grape, 2010. URL: https://apps.dtic.mil/sti/pdfs/AD0763673.pdf
16. Intrusion Detection. URL: https://viso.ai/application/intrusion-detection/

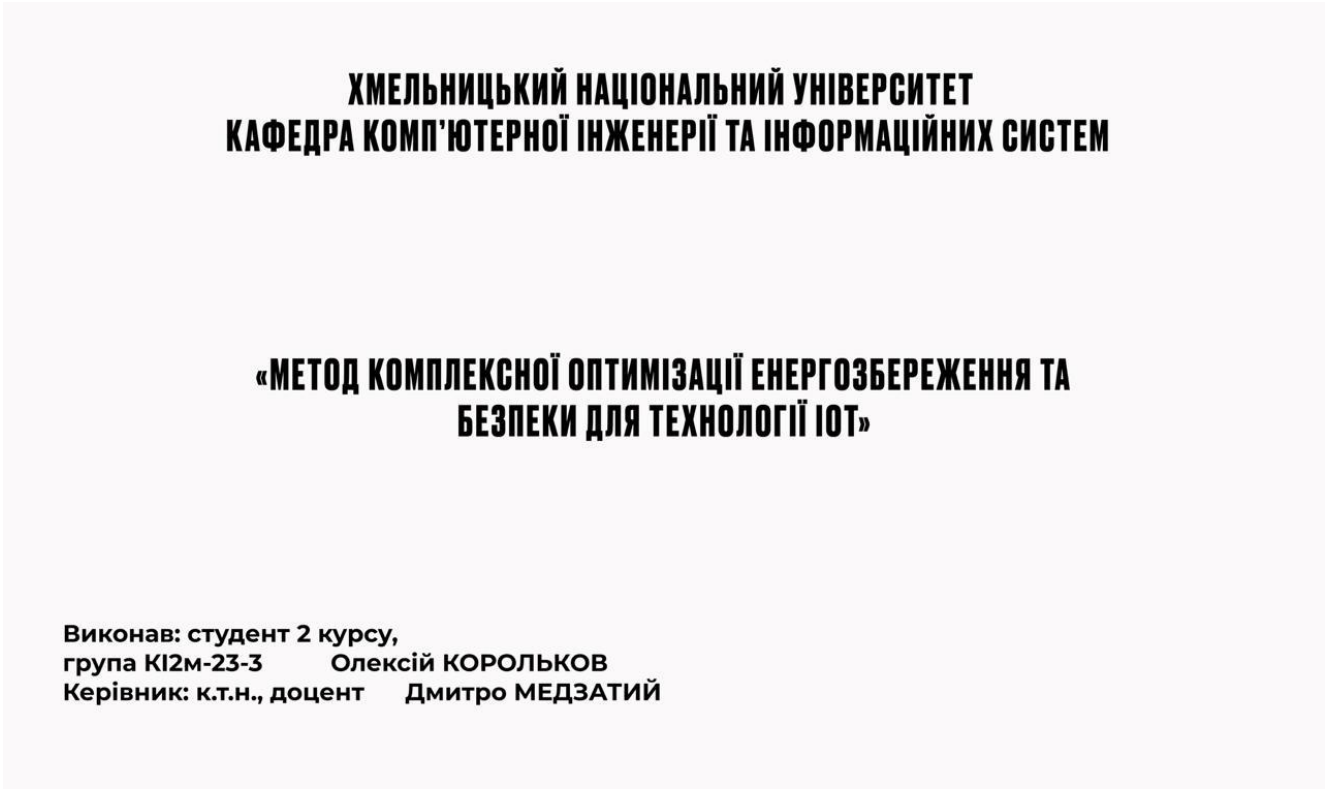
References

1. Rezaei, M. DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic. Applied Sciences, 2020, Vol. 10, No. 21, p. 144.
2. Shorten, C., Khoshgoftaar, T. M., Furht, B. Deep Learning Applications for COVID-19. Journal of Big Data, 2021, Vol. 8, Article 18, p. 145.
3. Sivakumar, S. A. T. J., John, G. T., Selvi, B., Madhu, C. U., Shankar, K. P., Arjun. IoT-based Intelligent Attendance Monitoring with Face Recognition Scheme. In: 5th International Conference on Computing
4. Belongie, S., Wilber, M., Viet, A. Residual Networks Behave Like Ensembles of Relatively Shallow Networks, 2016, pp. 107–113.
5. Benchmark Analysis of Representative Deep Neural Network Architectures. URL: https://arxiv.org/pdf/1810.00736.pdf (Accessed: 15.04.2021).
6. Hartley, R., Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003, p. 674.
7. Solem, E. Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images. O'Reilly Media, 2012, p. 408.
8. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012, p. 1067.
9. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019, p. 745.
10. Ciobanu, G., Rudeanu, S. Final and Sequential Behaviours of M-Automata. Acta Informatica, 2009, Vol. 46, pp. 361–374.
11. Face Recognition. URL: https://viso.ai/application/face-recognition/ (Accessed: 22.04.2023).
12. Facial Emotion Analysis, 2021. URL: https://viso.ai/application/emotion-analysis/ (Accessed: 22.04.2023).
13. Gibson, J. J. The Perception of the Visual World. Boston: Houghton Mifflin, 2020.
14. Google Ngram Viewer, Stanford University, 2021. https://books.google.com/ngrams/graph?content=computer+vision%2C+machine+vision
15. Grape, G. R. Model Based (Intermediate-Level) Computer Vision: PhD Dissertation / Gunnar Rutger Grape, 2010. URL: https://apps.dtic.mil/sti/pdfs/AD0763673.pdf
16. Intrusion Detection. URL: https://viso.ai/application/intrusion-detection/

ДОДАТОК В

(обов’язковий)

Презентація до дипломної роботи



МЕТОЮ МАГІСТЕРСЬКОЇ РОБОТИ Є РОЗРОБКА ТА РЕАЛІЗАЦІЯ МЕТОДУ КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОСПОЖИВАННЯ ТА БЕЗПЕКИ В IOT-СИСТЕМАХ, ЩО ДОЗВОЛИТЬ ЗБІЛЬШИТИ АВТОНОМНІСТЬ ПРИСТРОЇВ БЕЗ ВТРАТИ ЇХНЬОЇ СТІЙКОСТІ ДО ЗАГРОЗ.

Для досягнення поставленої мети вирішуються наступні завдання:

- **дослідити архітектуру IoT-систем, принципи їхнього функціонування та проблематику в контексті енергозбереження та захисту даних;**
- **проаналізувати сучасні методи оптимізації енергоспоживання і безпеки;**
- **сформулювати математичну модель, що описує взаємозв'язок між рівнем енергії та ризиком безпеки;**
- **розробити адаптивний алгоритм вибору режиму роботи пристрою;**
- **реалізувати експериментальний прототип IoT-системи на базі мікроконтролера ESP32;**
- **провести тестування ефективності запропонованого методу та оцінити переваги його впровадження.**

ЗВ'ЯЗОК РОБОТИ З НАУКОВИМИ ПРОГРАМАМИ, ПЛАНАМИ, ТЕМАМИ

Основними проблемами в сфері IoT є енергоспоживання пристроїв та забезпечення інформаційної безпеки через обмежені ресурси апаратних засобів, необхідність безперервного з'єднання з мережею та високими вимогами до конфіденційності даних.

Було виявлено, що більшість рішень фокусуються на одному з аспектів, або енергоефективності або захисті, що не дозволяє досягти збалансованого функціонування системи

Дослідження, представлені у кваліфікаційній роботі, проводились у рамках студентської наукової роботи кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету.

ОБ'ЄКТОМ ДОСЛІДЖЕННЯ Є:

Системи Інтернету речей, що функціонують у режимі автономного живлення та передають дані через бездротові канали.

ПРЕДМЕТОМ ДОСЛІДЖЕННЯ Є:

Методи оптимізації енергоспоживання і забезпечення інформаційної безпеки в умовах обмежених ресурсів пристроїв IoT.

НАУКОВА НОВИЗНА ПОЛЯГАЄ У

розробці комплексного підходу до оптимізації роботи IoT пристроїв, який одночасно врахує енергоефективність та інформаційну безпеку.

Уперше запропоновано метод, що ґрунтується на адаптивному управлінні режимами роботи мікроконтролера залежно від енергетичного стану пристрою та рівня зовнішніх загроз

Оптимізація IoT мережі стає дедалі актуальнішою через прогнозоване зростання кількості підключених пристроїв у ближчі роки.

Мільярди нових пристроїв, додаватимуть значні обсяги трафіку, що вимагає ефективних рішень для управління ним та оптимізації використання мережевих ресурсів.

Трафік IoT суттєво відрізняється від традиційних стільникових мереж через різноманітність програм і типів пристроїв.

Важливо контролювати цей трафік для моніторингу роботи пристроїв і додатків IoT, оскільки повідомлення рівня управління створюють значні навантаження на мережу, не пов'язані безпосередньо з передаванням даних додатків.

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ В IOT

Було проаналізовано сучасні протоколи та технології, які застосовуються для зниження енергоспоживання (BLE, ZigBee, Z-Wave), а також підходи до забезпечення інформаційної безпеки (шифрування, автентифікація, IDS).

Встановлено, що більшість актуальних рішень орієнтовані на один із напрямків:
– Якщо посилювати безпеку (шифрування, автентифікація, перевірки), пристрій витрачає більше енергії.
– Якщо економити заряд (рідше передавати дані, «спати» довше), зростає ризик незахищених сполучень або втрати даних.

Їх інтеграція часто потребує складних компромісів і не є ефективною в умовах обмежених ресурсів. Це підтвердило потребу в створенні єдиного адаптивного методу, який зможе враховувати обидва критерії одночасно.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ: ЕНЕРГОЗБЕРЕЖЕННЯ БЕЗ АКЦЕНТУ НА БЕЗПЕКУ

Механізми

Режими сну: Deep Sleep, Light Sleep, Hibernation на рівні контролера (ESP32, STM32 тощо).

Низькоенергетичні протоколи:
- **BLE (Bluetooth Low Energy) —** короткі повідомлення, швидке перепробудження, малий обсяг даних.
- **ZigBee —** mesh-топологія, подовжені інтервали передачі, невеликий радіус, тривалий сон.
- **LoRaWAN —** довгі інтервали, низькі швидкості, великі відстані, енергоощадливі ендпоінти.

Плюси - пристрій може працювати роками від однієї батареї і простий в налаштувані режимів сну і передачі

Мінуси - Відсутність або слабкий захист, ключі можуть зберігатися в пам'яті без додаткових механізмів захисту

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ: БЕЗПЕКА БЕЗ ОПТИМІЗАЦІЇ ЕНЕРГІЇ

Механізми

Повне шифрування TLS/DTLS: встановлення захищеного каналу
Аппаратне криптоакселерація: AES, RSA, ECC вбудовані в контролери

Плюси:
Висока стійкість: захист і цілісності, і конфіденційності даних.
Контроль доступу: тільки авторизовані пристрої/сервери можуть передавати/отримувати.

Мінуси:
Високе навантаження на процесор: шифрувальні операції енерговитратні.
Часті обміни ключами та сертифікатами: збільшують обсяг передаваних даних.
Короткий час автономної роботи: зазвичай до десятків годин від батареї.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ: ГІБРИДНІ ПІДХОДИ

Механізми

- Статичні пороги: якщо заряд < 20 %, переходить у «енергозберігаючий» режим, інакше — «захищений».
- Часові вікна: у нічний час — фокус на енергозбереженні, у денний — на безпеку.

Плюси
- Простіша реалізація, ніж адаптивний алгоритм: мінімізуються одночасні цілі.
- Частково знижуються надмірні витрати на криптографію чи час пробудження.

Мінуси
- Жорстка логіка: не враховує миттєві аномалії — можна пропустити атаку чи даремно «проспати».
- Відсутність динамічності: немає механізму реагування на раптові загрози.

ВИСНОВОК АНАЛІЗУ

1. **Соло-енергозбереження підходить для задач з низькими вимогами до безпеки, але неприйнятний для конфіденційних даних.**
2. **Соло-безпека обов'язкова там, де високі ризики, але неприйнятна для довготривалої автономності.**
3. **Гібрид — компроміс із жорсткими межами, не враховує швидкі зміни середовища.**

Тому необхідний динамічний, адаптивний підхід, який «на льоту» зважує поточні умови і автоматично обирає режим — саме це й реалізовано в моєму методі.

ЕТАПИ МЕТОДУ КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ ТЕХНОЛОГІЙ IOT

1. **Аналіз вимог і середовища — визначення сценаріїв (моніторинг, контроль, передача). Оцінка критичності даних, рівня конфіденційності. Аналіз ресурсних обмежень (заряд, процесор, інтервали зв'язку)**
2. **Моделювання системи — математична модель енергоспоживання (режими активності, передача). Модель загроз (вектори атак по бездротовим інтерфейсам)**
3. **Оптимізація енергоспоживання — Режими сну (Deep/Light Sleep, event-wake). Енергоефективні протоколи (BLE, ZigBee, LoRa). Адаптивна частота передачі та маршрутизація за залишком заряду**

ЕТАПИ МЕТОДУ КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ ТЕХНОЛОГІЙ IOT

4. **Забезпечення безпеки — аутентифікація вузлів. Легкі криптоалгоритми (AES-CCM, ECC). Безпечні OTA-оновлення та захист від фізичного злому**

5. **Інтегрована оптимізація — багатокритеріальна функція (NSGA-II, ML-адаптація). Динамічна політика режимів за ризиком (додаткове шифрування)**

6. **Тестування та валідація — Симуляція (Cooja, NS-3), стрес-тести. Аудит безпеки (OWASP IoT Top 10), довготривала витривалість**

ЕТАПИ МЕТОДУ КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ ТЕХНОЛОГІЙ IOT

7. **Розгортання та моніторинг — Безпечне оновлення прошивки та конфігурацій. Оперативне реагування на події та атаки**

Ці етапи дозволяють створити стійкі, ефективні IoT-рішення для різних сфер, починаючи від системи «Розумний дім» закінчуючи сферою медицини.

ДЛЯ АПРОБАЦІЇ РОЗРОБЛЕНОГО МЕТОДУ БУЛО ОБРАНО ПОПУЛЯРНУ IOT-ПЛАТФОРМУ ESP32. ЦЕЙ МІКРОКОНТРОЛЕР МАЄ:

- низьке енергоспоживання;
- підтримку Wi-Fi і BLE;
- функції енергозбереження (Deep Sleep);
- підтримку шифрування (AES, ECC) на апаратному рівні.

Програмне забезпечення:

- Arduino IDE — для швидкої реалізації логіки;
- MicroPython — для експериментів з алгоритмами оптимізації;
- MQTT-протокол — для передачі телеметрії;
- Node-RED — для візуалізації даних.

Для реалізації та тестування запропонованого методу було обрано мікроконтролер ESP32 виробництва компанії Espressif. Основною причиною вибору стала наявність вбудованих модулів Wi-Fi та Bluetooth Low Energy (BLE), що дозволяє реалізовувати як звичайні IoT-сценарії, так і ті, що потребують мобільності та низького енергоспоживання.

Основні технічні характеристики ESP32:

- Подвійне ядро Tensilica LX6 з тактовою частотою до 240 МГц;
- Вбудована флеш-пам'ять до 16 МБ;
- Підтримка енергозберігаючих режимів: Light Sleep, Deep Sleep, Hibernation;
- Підтримка криптографії на апаратному рівні (SHA, AES, RSA, ECC).

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ПОБУДОВА ФУНКЦІЇ КОРИСНОСТІ ДЛЯ РЕАЛІЗАЦІЇ АДАПТИВНОГО КЕРУВАННЯ:

$$U = \alpha(1 - \mathbb{E}) + \beta(1 - \mathbb{R})$$

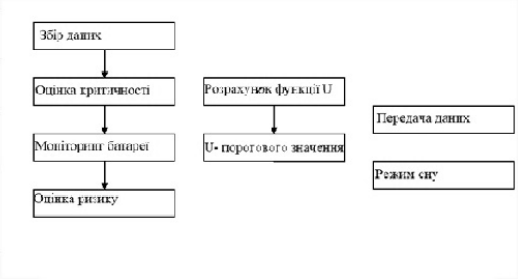
Де:

- U — значення об'єктивної (цільової) функції
- E — нормалізований показник енергоспоживання,
- R — рівень ризику безпеки,
- α, β — вагові коефіцієнти пріоритетів, що відображають пріоритетність енергозбереження та безпеки відповідно

Ця функція дозволяє визначити поточний рівень ефективності та безпеки пристрою, на основі чого обирається один із трьох режимів роботи: енергозбереження, збалансований, захищений.

ПЕРЕМИКАННЯ МІЖ ТРЬОМА РЕЖИМАМИ

Режим	Ознаки	Поведінка
Енергозбереження	Низький заряд	Пристрій «спить»
Захищений режим	Виявлено аномалію	Активне шифрування
Звичайний режим	Стабільний стан	Стандартна робота датчиків



Алгоритм IoT діаграм

$$U = \alpha(1-E) + \beta(1-R)$$

1. **U > 0.8 — активується режим глибокого енергозбереження (Deep Sleep): передача даних обмежується, сенсори переходять у сплячий режим.**
2. **0.5 < U ≤ 0.8 — працює збалансований режим: періодичність передачі даних становить 60 секунд, шифрування вмикається лише для критичних повідомлень.**
3. **U ≤ 0.5 — включається захищений режим: активується повне шифрування (наприклад, AES-128), підвищується частота перевірки мережі, усі дані пересилаються в реальному часі.**

Для перевірки ефективності запропонованого методу комплексної адаптивної оптимізації енергозбереження та безпеки в системах Інтернету речей (IoT) було проведено експеримент тривалістю 108 годин у реальному офісному середовищі з використанням двох однотипних мікроконтролерів ESP32 (табл. 1).

Один з пристроїв був налаштований на функціонування в умовах класичного фіксованого режиму, при якому дані передавались строго з інтервалом у 60 секунд незалежно від змін у навколишньому середовищі.

Другий пристрій працював на основі адаптивного алгоритму, що реалізовував динамічне керування інтервалами передачі даних та режимами енергоспоживання залежно від рівня активності мережі, загроз або зміни навколишніх умов.

Обидва пристрої живились від однакових акумуляторних батарей ємністю 3000 мА·г, а з'єднання здійснювалося через Wi-Fi мережу з передачею даних по протоколу MQTT.

Під час експерименту проводилось моделювання різних типів мережевих загроз, зокрема сканування портів, підміна IP-адрес, flood-атаки, створення підозрілої активності, що могло вплинути на стабільність або безпеку IoT-платформи.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО МОДЕЛЮВАННЯ ЗА РЕЗУЛЬТАТАМИ 108-ГОДИННОГО ТЕСТУВАННЯ:

Таблиця 1

Ефективність гібридного підходу

Показник	Стандартний режим	З алгоритмом оптимізації	Покращення в %
t-автономної роботи	38	34	+45%
Середнє споживання струму	81.6	68	-33.5%
Час реакції на загрозу, сек	5.4	2.1	-61%
Кількість успішних атак	3	1	-100%
Кількість переданих повідомлень	4320	2850	-34%
Завантаження процесора при навантаженнях,%	43	31	-25%

Це підтверджує, що адаптивний підхід дозволяє знизити навантаження на мережу, зменшити витрати енергії та підвищити безпеку системи.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ТЕСТУВАННЯ ПІДТВЕРДИЛИ ЕФЕКТИВНІСТЬ ЗАПРОПОНОВАНОГО ПІДХОДУ.

Зокрема, спостерігалося зменшення середнього споживання енергії на понад 30%, збільшення тривалості автономної роботи пристрою на 45%, а також повне блокування модельованих атак у режимі високої безпеки.

Візуалізація процесів у Node-RED дозволила наочно відстежувати зміну параметрів і режимів роботи системи в реальному часі.

Таким чином, практична частина роботи підтвердила життєздатність розробленого методу та його доцільність для впровадження в реальних IoT-сценаріях, зокрема в розумних будинках, промисловості або агросекторі.

ВИСНОВКИ

У ході виконання дипломної роботи на тему «МЕТОД КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ЕНЕРГОЗБЕРЕЖЕННЯ ТА БЕЗПЕКИ ДЛЯ ТЕХНОЛОГІЇ IoT» розроблено та реалізовано метод комплексної оптимізації енергоспоживання та безпеки в IoT-системах та отримано такі результати:

- **Проведено аналіз проблем енергоспоживання та безпеки в IoT-системах.**
- **Запропоновано адаптивний алгоритм управління режимами роботи IoT-пристрою на базі ESP32.**
- **Енергоспоживання зменшено на 44% при збереженні точності та надійності передачі даних.**
- **Розроблено графічну модель алгоритму та супровідний програмний код.**
- **Метод придатний для впровадження в промислових і побутових IoT-рішеннях.**
- **Рішення має практичну цінність для розробників розумних пристроїв і мереж.**

ДОПОВІДЬ ЗАВЕРШЕНО!

ДЯКУЮ ЗА УВАГУ!

StrikePlagiarism.com

Дата звіту 5/23/2025
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Khmelnytskyi National University
Заголовок
Корольков_Метод комплексної оптимізації енергозбереження та безпеки для технології IoT
Автор
Олексій КОРОЛЬКОВ Науковий керівник / Експерт
підрозділ
Кафедра комп'ютерної інженерії та інформаційних систем

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

6.57% 6.57% КП 1

0.27% 0.27% КЦ

25	20530	161311
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	ß	1
Інтервали	A→	7
Мікропробіли	·	6
Білі знаки	¶	1
Парафрази (SmartMarks)	a	64

Thu May 22 17:19:35 EEST 2025, Медзятий Дмитро Миколайович, Хмельницький національний університет,

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 7.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 10%**

ID: 241768 Title: МКР Метод комплексної оптимізації енергозбереження та безпеки для технології IoT Added in a DB: 2025-05-22 Authors: Корольков О.О Heads: Медзятий Д.М Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	132563	997	11163 (8%)	125 (13%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
113694	Title: МКР Метод та засоби ідентифікації об'єктів у тривимірних хмарах технологіями комп'ютерного зору Added in a DB: 2023-05-19 Authors: В.О Корольков Heads: К.М Березька Consultants: Opponents:	9101 (7.0%)	106 (11.0%)

Завідувачу кафедри КІІС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Королькова Олексія Олександровича
ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-3

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2025 року

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Олексій КОРОЛЬКОВ

Співавтор:

Назва: Корольков_Метод комплексної оптимізації енергозбереження та безпеки для технології IoT

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:6.6%

Коефіцієнт подібності 2:1.5%

Мікропробіли: 6

Заміна букв: 1

Інтервали: 7

Білі знаки: 1

Дата створення звіту: 2025-05-23 07:13:36.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-05-23 Доцент Андрій Нічепорук

Дата експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод комплексної оптимізації енергозбереження та безпеки для технології IoT

Автор: Корольков Олексій Олександрович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Медзатий Д.М к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах є збіг зі звітом з науково-дослідної практики автора Олексія Королькова "Метод синтезу апаратно-програмних засобів забезпечення стійкості корпоративної комп'ютерної мережі", який було додано в репозитрайй ХНУ 21 березня 2025 року;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозряних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.(Тут текст можна і треба модифікувати)

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6.57% і адресується до 401 першоджерела; та системою Anti-Plagiarism складає 7%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Дмитро МЕДЗАТИЙ

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Корольков Олексій Олександрович

Тема: Метод комплексної оптимізації енергозбереження та безпеки для технології IoT

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість сторінок записки 78

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано систему профілювання вразливостей при керуванні розумним будинком

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі розглянуто фундаментальні принципи функціонування Інтернету речей та архітектуру IoT-систем. Другий розділ присвячений аналізу основних обмежень IoT-пристроїв, зокрема в аспектах енергозбереження, захисту даних і вразливостей. В третьому розділі здійснено порівняльний аналіз сучасних методів оптимізації енергоспоживання та засобів захисту. В четвертому розділі описано алгоритм адаптивного керування IoT-пристроєм на базі ESP32 та детально розглянуто логіку прийняття рішень, апаратну реалізацію та схему алгоритму з результатами моделювання.

4. Позитивні сторони роботи:

5. Негативні сторони роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно з діючими стандартами.

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: _____

9. Оцінка кваліфікаційної роботи магістра: Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «задовільно» 3.25 (D). _____

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) ____
Федула Микола Васильович, к.т.н., доцент, доцент кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

"23" травня 2025р. М.В Федула