МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ЖМУРКО Іван Іванович

Криптографічні алгоритми захисту Інтернет-речей / Cryptographic Algorithms for Internet of Things Security

спеціальність: 125 – Кібербезпека

освітньо-професійна програма –Кібербезпека

Кваліфікаційна робота

Виконав студент групи КБм - 21
І.І. Жмурко

Науковий керівник
к.т.н., доцент Н.Г.Яцків

Кваліфікаційну роботу допущено
до захисту:

«____» _____ ___2023 р.

Завідувач кафедри
_____ В.В.Яцків

ТЕРНОПІЛЬ – 2023

Факультет комп'ютерних інформаційних технологій

Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
____" _____ 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ЖМУРКУ Івану Івановичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Криптографічні алгоритми захисту Інтернет-речей / Cryptographic Algorithms for Internet of Things Security
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 01 грудня 2022 року № 491

2. Строк подання студентом закінченої випускної кваліфікаційної роботи
01 грудня 2023 року.

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

4. Основні питання, які потрібно розробити:
– проаналізувати принципи функціонування та архітектуру Інтернет-речей;
– проаналізувати існуючі загрози безпеці Інтернет-речей;
– дослідити основні аспекти захисту Інтернет-речей;
– дослідити можливості застосування криптографічних методів захисту в умовах обмеженості ресурсів;
– розробити оптимізоване рішення захисту на основі криптографічних алгоритмів для забезпечення цілісності, конфіденційності та автентифікації в контексті Інтернет-речей.

5. Перелік графічного матеріалу у роботі.
– алгоритм вибору оптимального методу шифрування;
– схема алгоритму шифрування PRESENT;
– структура S-блоку.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Дослідження вразливостей та технології Інтернет-речей	12.2022 р. – 03.2023 р.	
2	Аналіз криптографічних методів захисту Інтернет-речей	03.2023 р. – 05.2023 р.	
3	Розробка алгоритму захисту Інтернет-речей на основі блокового шифрування	05.2023 р. – 11.2023 р.	

Студент _____ І.І. Жмурко
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Криптографічні алгоритми захисту Інтернет-речей» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 64 сторінках і містить 26 ілюстрації, 8 таблиць, 2 додатки та 33 джерела за переліком посилань.

Метою роботи є дослідження криптографічних алгоритмів з точки зору їх застосування у захисті Інтернет-речей та розробка адаптивного рішення, яке дозволяє враховувати обмежені можливості пристроїв IoT, забезпечуючи при цьому високий рівень захисту.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи аналізу криптографічних алгоритмів з точки зору їхньої ефективності та оптимізації для застосування в обмежених ресурсах пристроїв IoT, а також експериментальне дослідження для валідації їхньої продуктивності та безпеки в реальних умовах використання.

Результати дослідження: включають розробку оптимальних криптографічних рішень, які враховують специфіку Інтернет-речей, забезпечуючи високий рівень захисту даних та пристроїв при обмежених обчислювальних ресурсах.

Запропоновані рішення можуть бути безпосередньо використані в системах IoT для захисту даних та пристроїв, а також в інших сферах, де є потреба у високому рівні захисту при обмежених ресурсах, таких як медичні пристрої, системи зв'язку або автономні транспортні засоби.

Ключові слова: ШИФРУВАННЯ, АЛГОРИТМИ ЗАХИСТУ, ІНТЕРНЕТ РЕЧЕЙ, МАЛОРЕСУРСНА КРИПТОГРАФІЯ

ABSTRACT

Qualification work on "Cryptographic Algorithms for Internet of Things (IoT) Security" for the degree of "Master" in the specialty 125 "Cybersecurity" educational and professional program "Cybersecurity" is written in 64 pages and contains 26 illustrations, 8 tables, 2 appendices and 33 source according to the list of links.

The purpose of work is to investigate cryptographic algorithms in terms of their application in securing the Internet of Things and to develop an adaptive solution capable of considering the limited capabilities of IoT devices while ensuring a high level of protection.

Research methods. The methods involve the analysis of cryptographic algorithms concerning their efficiency and optimization for application within the limited resources of IoT devices. Additionally, experimental research is conducted to validate their productivity and security in real-world usage scenarios.

Research results: they encompass the development of optimal cryptographic solutions that consider the specifics of the Internet of Things, ensuring a high level of data and device protection within limited computational resources.

The proposed solutions can be directly applied in IoT systems to protect data and devices, as well as in other domains where there is a need for high-level security with limited resources, such as medical devices, communication systems, or autonomous transportation.

Keywords: ENCRYPTION, SECURITY ALGORITHMS, INTERNET OF THINGS, LOW-RESOURCE CRYPTOGRAPHY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AES - Advanced Encryption Standard;

DDoS - Distributed DoS;

DoS - Denial of Service Attacks;

ECC - Elliptic Curve Cryptography;

ECDSA - Elliptic Curve Digital Signature Algorithm;

HMAC - Keyed-Hash Message Authentication Code;

MITM - Man in the Middle;

PKI - Public Key Infrastructure;

RSA - Rivest-Shamir-Adleman;

SHA - Secure Hash Algorithms;

АМРШ - алгоритми малоресурсного шифрування;

IoT - Інтернет-речей;

КА - криптографічні алгоритми;

МАШ - малоресурсе асиметричне шифрування;

МБШ - малоресурсне блокове шифрування;

МРК - малоресурсна криптографія;

МСШ - малоресурсне симетричне шифрувння;

НД - несанкціонований доступ;

ОР - обчислювальні ресурси;

ПЗ - програмне забезпечення;

ПЛІС - програмовані логічні інтегральні схеми;

СМ - сенсорні мережі.

ВСТУП

Актуальність теми. З кожним днем кількість Інтернет-речей (IoT) збільшується. Це можуть бути камери, розташовані на вулицях міста, різноманітні сенсори та датчики, які використовуються в виробництві, медичні пристрої, що моніторять здоров'я, а також різні побутові предмети, які оточують нас щодня [1-5]. Деякі з пристроїв зберігають дуже важливу та приватну інформацію. Наприклад, система замків на дверях квартири зберігає код блокування. Крім того, у медичних системах використовуються такі пристрої, як ЕКГ, смарт-монітори, комп'ютерна томографія та багато інших, які можуть безпосередньо впливати на важливі сфери життя людей. Звідси виникає явна задача досягнення безпеки та приватності даних, які передаються IoT. Розробка абсолютно безпечної системи в IoT є складним завданням [6-9].

По-перше, через те, що системи IoT дуже різноманітні: складаються з різних пристроїв, що мають різні операційні системи, апаратні засоби та використовують різні протоколи. Різноманітність протоколів та стандартів безпеки може призвести до проблеми несумісності між пристроями усунення даного недоліку забезпечить ровиток загальноприйнятих стандартів та протоколів безпеки для спрощення взаємодії пристроїв IoT. По-друге, системи є вкрай масштабними. Вони можуть існувати як у межах однієї квартири, так і розповсюджуватися на міста або навіть країни. Серед значного числа пристроїв IoT, кожен потребує унікальний ключ або сертифікат для безпеки. Управління та розподіл ключів може бути складним завданням. Для вирішення даної проблеми необхідна реалізація систем управління ключами, автоматизація процесу генерації та розподілу ключів. По-третє, що дуже важливе, багато пристроїв IoT мають обмежені ресурси: об’єми пам'яті, обчислювальну потужність, ємність акумулятора та інші.

Все перелічене зумовлює використання криптографічних алгоритмів (КА), як одного із важливих засобів безпеки для шифрування даних [10-12]. Через обмежені обчислювальні можливості та пам'ять багатьох

пристроїв IoT, використання складних КА може бути важким через обмежену потужність обробки даних [13-15]. Проте розробка ефективних алгоритмів та методів криптографії, які використовують мінімальні ресурси пристроїв дозволяє забезпечити надійний рівень захисту. Безпека IoT пристроїв потребує постійних оновлень та вдосконалень, але не всі пристрої можуть бути легко оновлені через їхню обмежену функціональність. Завдяки розробці механізмів для безпечного та ефективного оновлення програмного забезпечення (ПЗ) на пристроях IoT даний недолік може бути усунутий.

Забезпечення безпеки у застосуванні криптографії для IoT вимагає комплексного підходу, який враховує обмеження пристроїв, а також управління ключами, фізичну безпеку та стандартизацію.

Мета і завдання дослідження. Метою кваліфікаційної роботи є дослідження ефективності криптографічних методів захисту IoT та реалізація алгоримту адаптованого для забезпечення безпеки пристроїхв в умовах обмежених ресурсів.

Відповідно до мети кваліфікаційної, передбачено вирішити завдання:

– проаналізувати принципи функціонування та архітектуру Інтернет-речей;

– проаналізувати існуючі загрози безпеці Інтернет-речей;

– дослідити основні аспекти захисту Інтернет-речей;

– дослідити можливості застосування криптографічних методів захисту в умовах обмеженості ресурсів;

– розробити оптимізоване рішення захисту на основі криптографічних алгоритмів для забезпечення цілісності, конфіденційності та аутентифікації в контексті IoT.

Об’єкт дослідження - криптографічні методи захисту IoT.

Предмет дослідження - шляхи підвищення безпеки IoT з використанням криптоалгоритмів.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи аналізу криптографічних алгоритмів з точки зору їхньої ефективності та оптимізації для застосування

в обмежених ресурсах пристроїв IoT, а також експериментальне дослідження для валідації їхньої продуктивності та безпеки в реальних умовах використання.

Наукова новизна отриманих результатів. Розроблено алгоритм адаптивного вибору методу шифрування, що враховує специфіку кожного пристрою IoT та дозволяє забезпечити баланс між рівнем безпеки та обмеженнями ресурсів. Реалізовано алгоритм захисту, що ґрунтується на блоковому шифруванні, який дозволяє забезпечити високий рівень безпеки при мінімальному використанні ресурсів. Синтезовано структуру S-блоку шифрування з оптимізованими характеристиками апаратної складності, що відповідає вимогам малоресурсної криптографії та забезпечує його застосування в пристроях або системах із обмеженими характеристиками.

Практичне значення отриманих результатів. Запропоноване рішення дозволяє налаштовувати методи шифрування для пристроїв IoT, що сприяє високому рівню захисту даних, які передаються. Синтезоване рішення дозволяє досягти високого рівня безпеки при мінімальному споживанні ресурсів, а також може сприяти збільшенню енергоефективності пристроїв IoT. Запропоновані рішення можуть бути використані у різних галузях, де запобігання кібератакам та збереження конфіденційності даних є вирішальними факторами, від систем IoT до фінансових технологій та інших автономних систем.

Публікації та апробація кваліфікаційної роботи.

1. Жмурко І.І. Аналіз криптографічних алгоритмів захисту Інтернет-речей / Збірник матеріалів проблемно-наукової міжгалузевої конференції «Автоматизація та комп’ютерно-інтегровані технології» (АКІТ - 2023), Тернопіль, 2023. - с.112-114.

2. Давлетова А.Я., Жмурко І.І.М Виявлення загроз та захист Інтернет-речей / Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. - с. 39-44.

1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ ІНТЕРНЕТ-РЕЧЕЙ ТА ЇЇ ВРАЗЛИВОСТЕЙ

1.1 Принципи функціонування Інтернет-речей

Технологія IoT - це концепція, що передбачає підключення до мережі інтернет різноманітних фізичних об'єктів, які можуть взаємодіяти між собою та з оточуючим середовищем [1]. Серед компонентів IoT виділяють [2-5]:

- сенсори та датчики - пристрої, які збирають дані від навколишнього середовища чи самого пристрою, наприклад термометри, мікрофони, камери та інші пристрої, які вимірюють різні параметри
- підключення/мережа - забезпечує комунікацію між пристроями чи з Інтернетом через різні технології (Wi-Fi, Bluetooth, LoRa, NB-IoT);
- централізоване сховище, куди передаються та обробляються дані від пристроїв IoT;
- фреймворк прийняття рішень - набір програм, які реагують на зміни в системі і визначають оптимальні сценарії для вирішення проблем або досягнення поставлених цілей;
- інтерфейс користувача - простий та зрозумілий, що дозволяє взаємодіяти з системою IoT.

IoT починається з пристроїв, обладнаних датчиками (рисунок 1.1), які відстежують стан об'єктів та можуть надсилати повідомлення до системи прийняття рішень, видаючи команди на виконання. Потім за допомогою безпровідних з'єднань дані відправляються у хмару. Отримана інформація може аналізуватися на місці, але частіше потрібне хмарне середовище, оскільки датчики мають обмежені ресурси і потребують пункту, куди вони можуть надсилати дані. Після аналізу відбувається відправка вихідних даних. І останній елемент IoT це користувацький інтерфейс, зазвичай мобільний чи веб-додаток, який допомагає взаємодіяти з системою.

Також у процесі може використовуватися штучний інтелект або машинне навчання, яке спрощує та робить процеси більш динамічними. Людина майже не бере участь у процесі обміну даними, а може лише

налаштовувати, надавати інструкції чи мати доступ до даних.

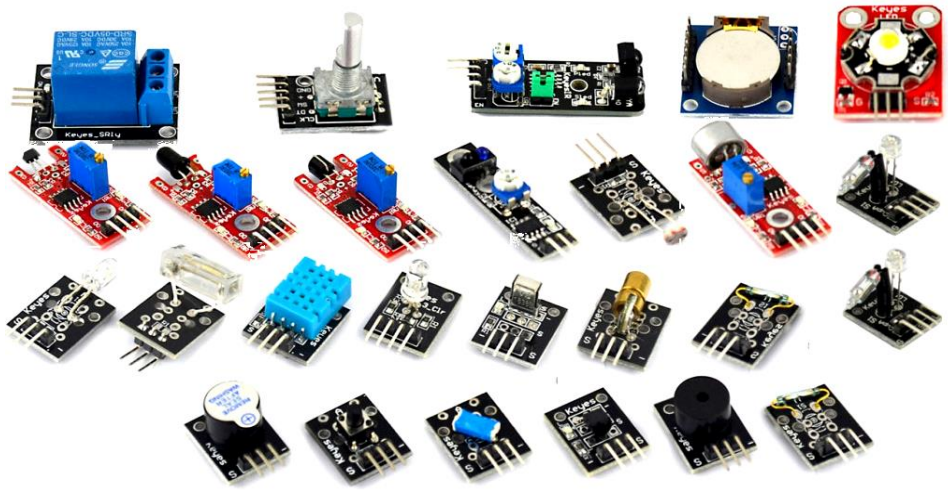

Рисунок 1.1 – Приклад датчиків IoT

Базові елементи, в свою чергу, поділяються на декілька типів елементів: сенсори (рисунок 1.1), актуатори і гейти [2]. Актуатори призначені для взаємодії з оточенням або конкретними об'єктами у ньому, наприклад сервоприводи, динаміки, електронні замки, освітлювальні пристрої та ін. Гейти зазвичай відповідають за логіку обробки інформації від підключених сенсорів. У випадках, коли обробка даних не потребує значного числа обчислювальних ресурсів (ОР), вони можуть приймати рішення самостійно та відправляти команди на актуатори для виконання відповідних функцій (рисунок 1.2).

а) б)

Рисунок 1.2 – Приклад гейтів мікрокомп'ютер (а) або мікропроцесор (б)

Якщо обробка інформації потребує більше ОР, або ці дані потрібно збирати, гейти відправляють їх на сервери для подальшої роботи. Гейти можуть бути мікрокомп'ютерами або мікропроцесорами, які

використовуються в системі. Наприклад, для створення моніторингової системи достатньо використовувати лише сенсори та певний сервер, який буде виступати як гейт. За допомогою датчика руху можна організувати облік кількості людей, які проходять через певний прохід. Додавши актуатор у вигляді динаміка, можна реалізувати звуковий підрахунок перехожих. Ускладнювати конструкцію такої системи можна безкінечно. Але в певний момент з'явиться потреба у тривалому зберіганні зібраної інформації, її аналізі, візуалізації та інших операціях. Для цього необхідні повноцінні сервери, які можуть реалізувати ці операції. Такі сервери разом утворюють хмари, до яких підключаються гейти.

Комунікація сенсорів між собою та гейтами відбувається за допомогою кабелів, або безпровідними каналами (рисунку 1.3) [4].

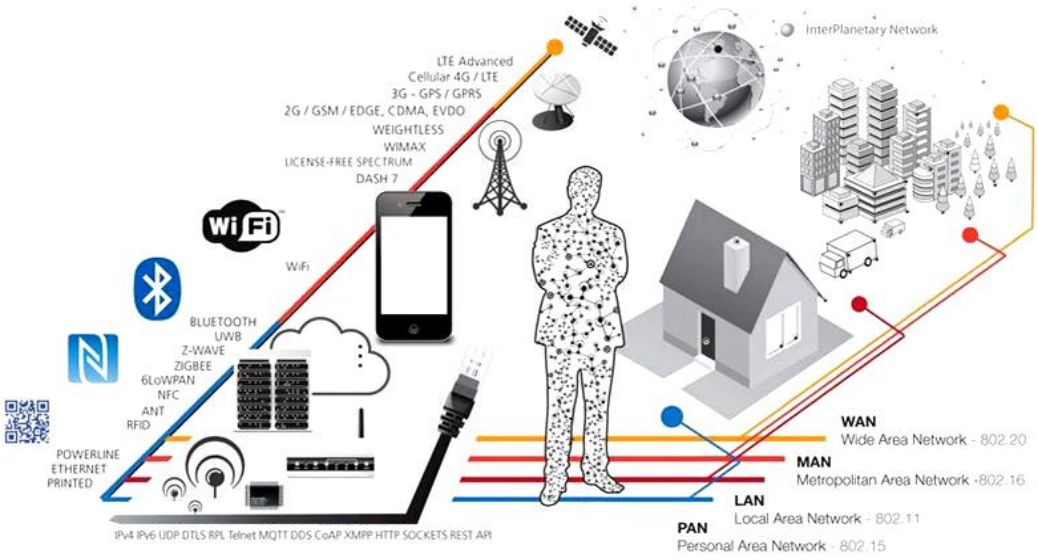

Рисунок 1.3 – Варіанти підключення IoT до існуючих мереж

Безпровідний зв’язок реалізований, як правило, з використанням протоколів Bluetooth LE, LoRa, ZigBee, SigFox та NB-IoT. Найпопулярнішим протоколом є Bluetooth, оскільки ця технологія практично є в будь-якому мобільному пристрої і споживає мінімум енергії. Для передачі даних використовуються також технології GPRS, Wi-Fi або LTE.Зв'язок в IoT може здійснюватися за допомогою різних технологій, таких як 2G/3G/4G, 5G, супутникові зв'язок (VSAT), а також мережі LPWAN, такі як LORA, LTE-M, NB IOT, NB FI і інші [3-6].

IoT складається зі мереж і систем, що пов'язані між собою, кожна з яких розгорнута для вирішення конкретних завдань. Наприклад, в сучасних автомобілях працює декілька мереж: одна керує роботою двигуна, інша - системами безпеки, третя підтримує комунікаціюі т.д. В офісних та житлових будівлях також встановлюється багато мереж для управління опаленням, вентиляцією, кондиціонуванням, телефонним зв'язком, безпекою, освітленням. З розвитком IoT ці мережі з'єднуються для отримання більшого кола можливостей у сфері безпеки, аналітики та управління.

Архітектура IoT включає в основному три рівні (рисунок 1.4).

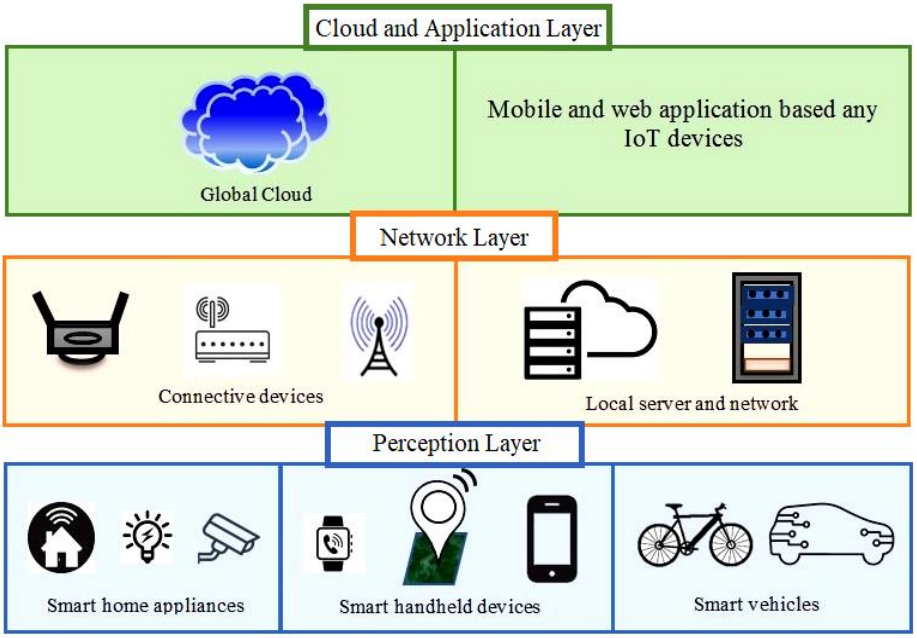

Рисунок 1.4 - Архітектура рівнів IoT

Ключовою ознакою того, що пристрій відноситься до системи IoT, є його здатність самостійно з'єднуватися з Інтернетом або іншими пристроями для передачі інформації за допомогою Wi-Fi, Bluetooth або інших доступних методів. Об'єкти IoT можуть обмінюватися даними в межах однієї кімнати, міста, країни чи навіть у всьому світі. Системи IoT працюють в реальному часі.

Сфери застосування IoTдоволі широкі і різноманітні (рисунок 1.5). За допомогою IoT можна: керувати побутовою технікою вдома, відстежувати рух транспорту, реалізувати практично автономне будівництво та впроваджувати безліч інших малих та глобальних проектів [3-5].

Рисунок 1.5 – Сфери застосування IoT

Основні принципи функціонування IoT включають:

– об'єкти повинні бути здатні підключатися до мережі, надаючи змогу обмінюватися даними та отримувати команди через Інтернет.

– використання різних типів сенсорів для збору різноманітних даних з оточуючого середовища або об'єктів.

– використання різноманітних протоколів зв'язку (Wi-Fi, Bluetooth, Zigbee) для передавання даних між системами чи пристроями.

– аналіз та оброблення отриманої інформації, що дозволяє реагувати на події.

– використання зібраних даних для прийняття рішень та автоматизації різних процесів.

– захист зібраних даних та пристроїв від НД або атак.

Перелічені принципи дозволяють IoT створювати екосистему, де різні об'єкти можуть здійснювати збір, обмін та аналіз даних для покращення різних аспектів нашого життя та робочих процесів.

Очевидна перевага технології IoT полягає у зручності та економії часу. Людей приваблює можливість відстежувати дії всіх пристроїв дистанційно. Проте іноді існує несумісність ПЗ від різних виробників і не завжди вдається об'єднати пристрої в єдину групу через відмінності внутрішніх налаштувань.

Значним недоліком є низький рівень захищеності даних. Кіберзлочинці часто атакують компанії, які використовують IoT-технології, та намагаються зламати системи керування пристроями і бази даних. Тому дослідження у сфері захисту IoT є досить актуальною задачею.

1.2 Загрози безпеці Інтернет-речей та їх реалізації

1.2.1 Типи атак

Система IoT стикається з різними атаками, класифікація яких, за різними критеріями, зображена на рисунку 1.6 [6-9].

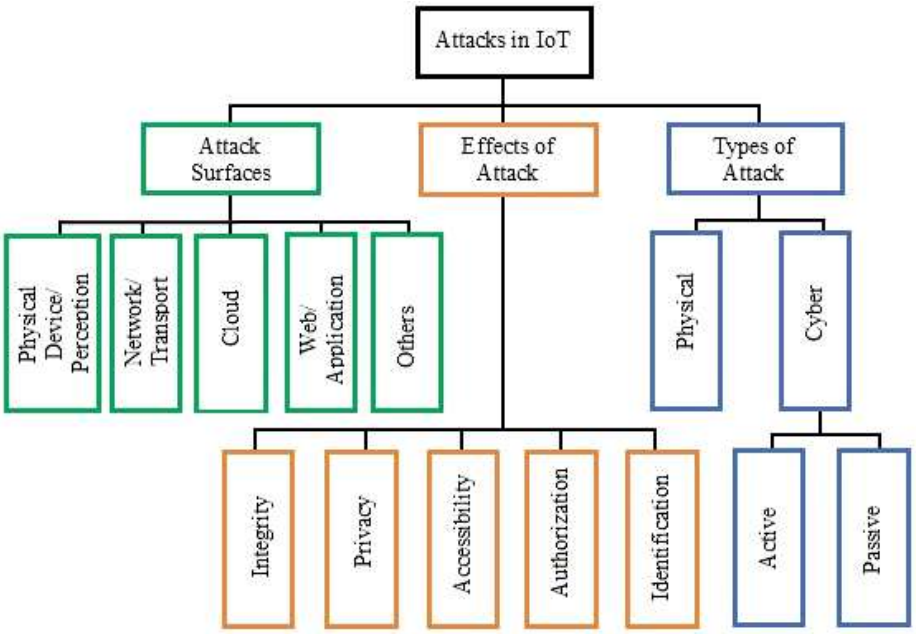

Рисунок 1.6 – Атаки на IoT

Типи атак можна класифікувати в основному як:

- кібератаки стосуються загрози, спрямованої на різні пристрої IoT у безпровідній мережі шляхом злому системи з метою маніпулювання (тобто викрадення, видалення, зміни, знищення) інформації користувача;
- фізичні атакистосуються атак, які фізично пошкоджують IoT пристроїв. Тут зловмисникам не потрібна мережа для атаки на систему. Таким чином, цей тип атак піддається фізичним IoT-пристроям, наприклад, мобільному телефону, камері, датчикам, маршрутизаторам тощо, за допомогою яких зловмисники переривають службу.

Серед кібератак виділяють два типи (таблиця 1.1) активні та пасивні.

Таблиця 1.1 – Кібератаки IoT

Тип	Назва	Вплив
Активна	атаки типу «Hole» та «Sybil», перешкоди (глушіння), спуфінг, DoS, MITM, вибіркоае пересилання, підробка даних, зловмисне ввдення	ідентифікація, авторизація, доступність, конфіденційність, цілісність
Пасивна	перехоплення, аналіз трафіку	конфіденційність

Активні атака відбуваються, коли зловмисник отримує НД до мережі та її відповідної інформації, щоб маніпулювати конфігурацією системи та переривати певні служби. Існують різні способи атакувати безпеку IoT-пристроїв, включаючи порушення, втручання та модифікації під час активних атак. Активні атаки, такі як DoS, Man in the Middle (MITM), атака на sybil, спуфінг, глушіння, вибіркове пересилання, підробка даних та ін. Впливу атаки піддається ідентифікація, авторизація, доступність, цілісність та конфіденційність даних.

Пасивні атаки намагаються зібрати інформацію користувача без його згоди та використати цю інформацію, щоб розшифрувати його приватні захищені дані. Прослуховування та аналіз трафіку є двома основними способами здійснення пасивної атаки через мережу IoT. Прослуховування в основному розгортає IoT-пристрій користувача як датчик для збору та неправильного використання його конфіденційної інформації та місцезнаходження.

Наступні типи атак в основному відносяться до типу кібератак відповідно до їх серйозності на пристроях IoT [5-9].

Атаки відмова у обслуговувані (Denial of Service Attacks) (DoS) головним чином призводять до порушення роботи системних служб шляхом створення кількох надлишкових запитів (рисунок 1.7). Таким чином, користувачу не вдається отримати доступ і спілкуватися з пристроєм IoT, що

ускладнює прийняття правильного рішення. Крім того, DoS-атаки тримають пристрої IoT завжди ввімкненими, що в результаті може вплинути на час роботи батареї. Спеціальний тип атаки під назвою Distributed DoS (DDoS) виникає, коли кілька атак здійснюються з використанням різних IP-адрес для створення численних запитів і підтримки зайнятості сервера. Це ускладнює диференціацію між звичайним трафіком і атакуючим трафіком. Останніми роками унікальний вірус ботнету Інтернет-речей під назвою Mirai знову відповідав за впровадження деструктивних атак DDoS, які пошкодили тисячі пристроїв IoT через втручання.

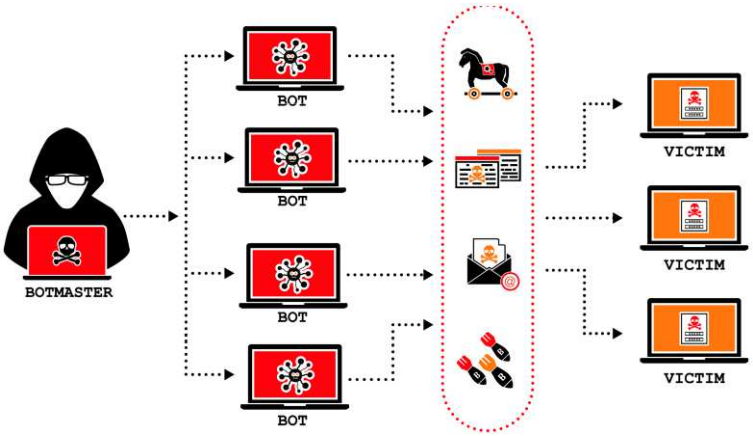

Рисунок 1.7 – DDoS атака

Спуфінг (Spoofing) і Sybil атаки and (Sybil Attacks) головним чином спрямовані на ідентифікацію (RFID та MAC-адреса) користувачів для незаконного доступу до системи в системі IoT (рисунок 1.8). Пакет TCP/IP не має надійного протоколу безпеки, що робить пристрої IoT більш вразливими, особливо для атак підробки. Більше того, ці дві атаки ініціюють подальші серйозні атаки, включно з DoS і атаками «man in the middle».

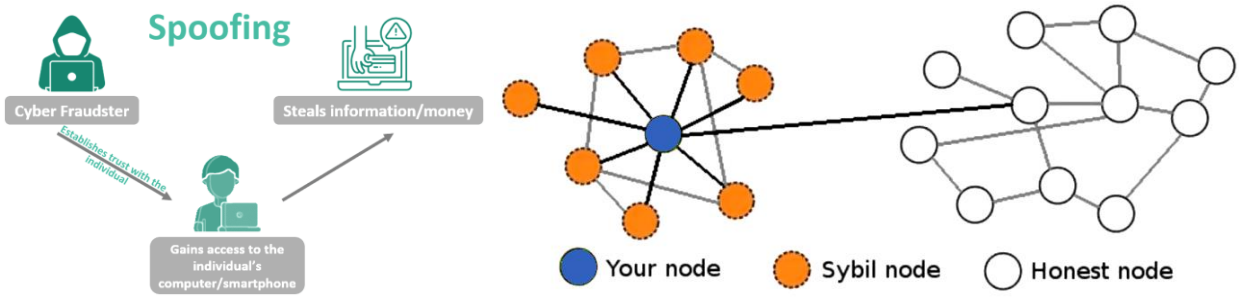

Рисунок 1.8 - Атаки спуфінг і Sybil

Атаки зі створення перешкод (Jamming Attacks) або порушення постійного зв’язоку у безпровідній мережі, надсилаючи небажані сигнали на пристрої IoT, створюють проблеми для користувачів, підтримуючи мережу постійно зайнятою (рисунок 1.9). Крім того, ця атака знижує продуктивність пристроїв IoT, споживаючи більше енергії, знижуючи пропускну здатність, пам’ять тощо. Будучи частиною комунікаційних систем, зловмисники безпосередньо підключені до іншого пристрою користувача, може легко перервати зв’язок шляхом введення підроблених даних, що вводять в оману, щоб маніпулювати вихідною інформацією.

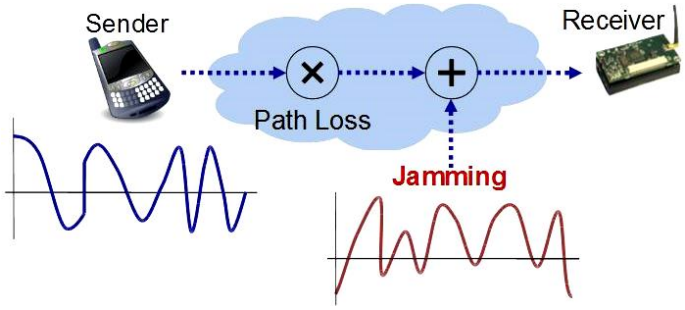

Рисунок 1.9 – Атака із створенням перешкод

Атаки вибіркового пересилання (Selective Forwarding Attacks) з блокуванням вузлів чи пристроїв у мережі або перешкоджанню передачі певних пакетів даних, які спрямовані на спотворення передачі інформації шляхом впливу на пропускну здатність мережі або перешкоджання передачі конкретних повідомлень (рисунок 1.10). Даний вид атаки важко ідентифікувати та уникнути. Вона може спричинити втрату даних, перешкоди в роботі системи або спотворення отриманих результатів.

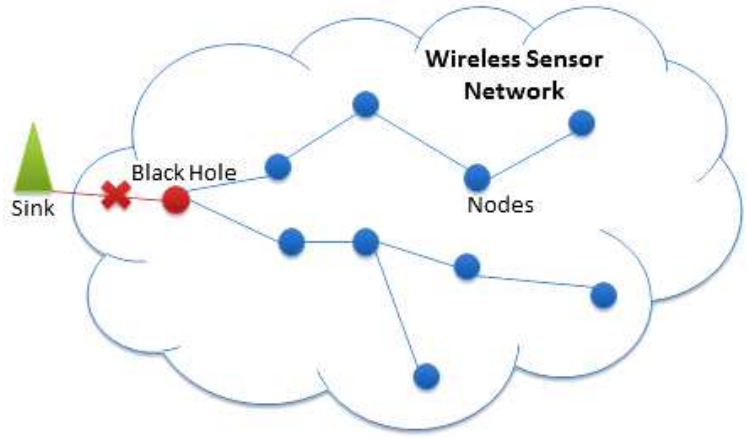

Рисунок 1.10 - Атака з вибірковим пересиланням

Атаки зловмисного введення (Malicious Input Attacks) включають введеня даних чи команд в ПЗ для створення вразливості, наприклад трояни, руткіти, хробаки, рекламне ПЗ та віруси, які спричиняють пошкодження пристроїв IoT, наприклад, фінансові втрати, розсіювання потужності, погіршення продуктивності бездротової мережі (рисунок 1.11).

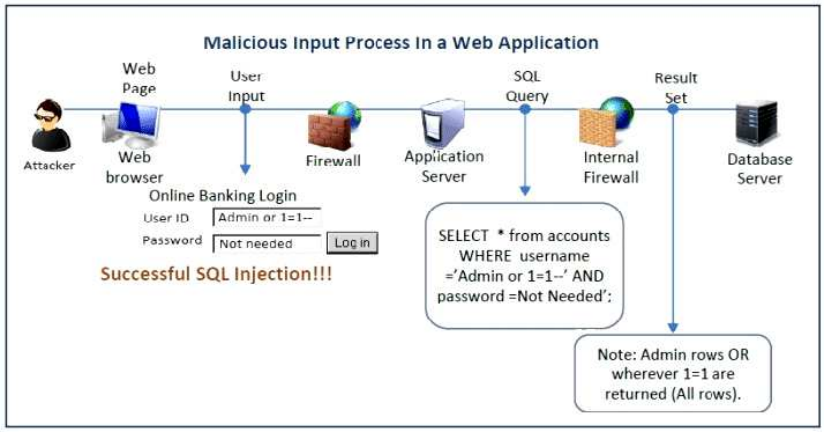

Рисунок 1.11 - Атака зловмисного введення

В атаках типу людина посередині MITM зловмисники Attacks прикидаються частиною комунікаційних систем, де вони безпосередньо підключені до іншого пристрою користувача (рисунок 1.12). Таким чином, вони можуть легко перервати комунікацію, вводячи підроблені та оманливі дані, щоб маніпулювати оригінальною інформацією.

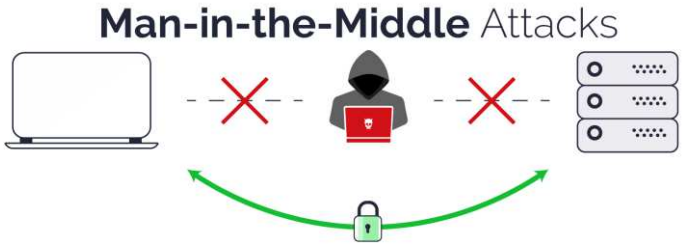

Рисунок 1.12 - Атака Man in the middle

Втручання в дані (Data Tampering) це процес навмисної зміни, модифікації або спотворення даних без належних дозволів чи авторизації. Це може статися як у статичних даних (які зберігаються у системі), так і у даних, що пересилаються або обробляються в реальному часі в мережі. Пристрої IoT, які передають важливу інформацію користувача, наприклад місце знаходження, стан здоров'я, платіжна інформація, знаходяться у

великій небезпеці при зіткненні з цими атаками (рисунок 1.13).

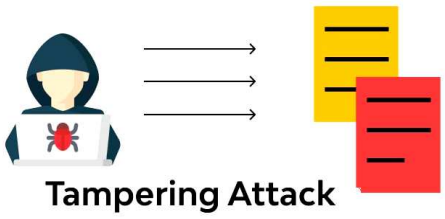

Рисунок 1.13 – Атака втручання в дані

Основні кіберзагрози для IoT включають:

– DDoS-атаки з перенавантаженням серверів, що призводять до відмови в обслуговуванні через надмірний обсяг запитів;

– витік конфіденційних даних - НД до особистих даних, який може призвести до їх використання в шахрайських цілях;

– недостатня кібербезпека - використання слабких паролів, застарілих програмних захистів, недостатнє оновлення ПЗ створюює уразливості для вторгнень;

– злам пристроїв - незаконний доступ до підключених пристроїв, що дозволяє злочинцям керувати ними.

– фізичні загрози - якщо пристрої залишаються фізично доступними, вони можуть стати об'єктом крадіжок або зламу систем безпеки;

– маніпулювання даними - несанкціоновані зміни інформації, що використовується для прийняття рішень, які можуть вплинути на їхню достовірність.

– спам та фішинг - використання IoT-пристроїв для розсилки спаму або проведення фішинг атак для викрадення особистої інформації.

1.2.2 Рівні атак

Архітектура IoT включає в основному три рівні, які були наведені на рисунку 1.4. Однак для більш точного опису можливих атак виділяють чотири потенційні рівні атак на IoT (рисунок 1.14), де рівень опрацювання і зберігання даних розглядаються окремо. В таблиці 1.2 наведено перелік можливих типів атак на кожному з рівнів [6-9].

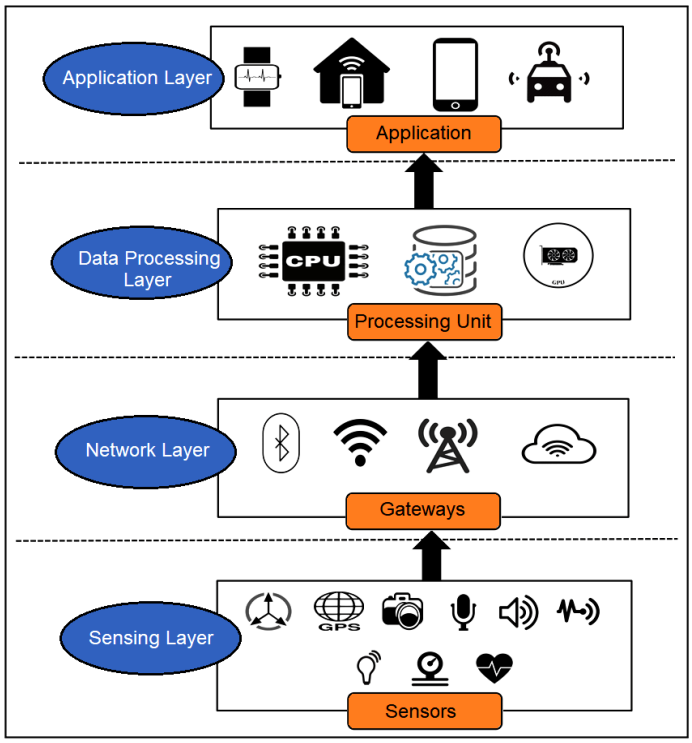

Рисунок 1.14 – Рівні атак

Таблиця 1.2 – Можливі атаки на різних рівнях IoT

Рівень атаки	Назва атаки
Рівень додатків	DDoS, відмова підтвердження (repudiation), шкідливий вузол, пошкодження даних, перехоплення, bluesnarfing, bluejacking
Рівень зберігання та обробки даних	DDoS, викрадення сесії, Атака на виснаження ресурсів (Exhaustion attack), Атака на переповнення (Flooding Attack), зловмисні введення, інсайдерська атака
Рівень мережі	DDoS, атаки типу «hole» та « sybil», вибіркове пересилання, перехоплення, спуфінг, аналіз трафіку, створення першкод, MITM, атаки на маршрутизацію
Рівень фізичних пристроїв	DDos, перехоплення, підробка (counterfeiting), радіоперешкоди, фізична атака, атака захоплення вузла, відстеження користувачів

Атаки на рівень фізичних пристроїв відомі як прямі атаки системи IoT, оскільки вони несуть конфіденційну та важливу інформацію користувачів.

Крім того, зловмисники можуть легко отримати доступ до фізичного рівня пристроїв IoT. RFID-мітки, датчики, приводи, мікроконтролери, зчитувачі RFID є деякими одиницями певних фізичних пристроїв, що використовуються для ідентифікації, зв'язку, збору та обміну інформацією [13]. Ці частини вразливі до DoS, прослуховування, радіоперешкод та найбільш небезпечними є фізичні атаки.

Атаки на рівень мережі або транспортний рівень реалізуються через фізичні пристрої, що підключаються через мережеві служби, включаючи провідні та безпровідні та бездротові мережі в системах IoT. Сенсорні мережі (СМ) відіграють важливу роль у розробці мережі IoT. Таким чином, СМ повинна бути інтегрована для створення широкомасштабної поверхні IoT, є потенційною мішенню для різних типів атак, оскільки інформація користувача передається відкрито через СМ без використання надійних протоколів безпеки. Щоб здійснити атаку на рівні мережевої служби, зловмисники завжди намагатимуться знайти відкриті порти або слабкий протокол маршрутизації для доступу до мережі користувача, використовуючи свою IP-адресу, шлюз і MAC-адресу для маніпулювання конфіденційною інформацією. Мережеві атаки схильні до DoS, глушіння, MITM, підробки, Sybil-атак, вибіркової переадресації, аналізу трафіку, інтернет-атак, атак маршрутизації тощо.

Наступним видом є атаки на хмарний рівень. Окрім власних сховищ для зберігання, пристрої IoT використовують систему could, яка з'єднує більшість інтелектуальних пристроїв і має необмежений об'єм пам'яті. Ця технологія хмарних обчислень дозволяє віддалено ділитися своїми збереженими ресурсами для інших користувачів. Таким чином, хмарні обчислення стали базовою платформою для пристроїв IoT для транспортування та збереження даинх користувача. Крім того, ця служба може зробити системи IoT динамічними та оновлювати їх у реальному часі. Тому, користувачів, які використовують подібні хмари, можуть зламати, викрасти та маніпулювати даними за допомогою атак даного рівня. Крім того, на хмарних рівнях можуть виявлятися DoS, інсайдерські атаки та

зловмисні атаки.

Атаки на рівень Web та додатків дають змогу віддалено отримувати доступ до різних пристроїв IoT і керувати ними. Їх зростання пояснюється тим, зо за останні десятиліття інтелектуальна технологія розвивається дуже швидко, що призводить до збільшення попиту на пристрої IoT для віддаленого доступу та керування розумними пристроями, такими як розумні автомобілі, побутові пристрої, освітлення і фітнес-пристрої. Web- та мобільні додатки. Пристрої IoT підключаються до мережі через сервери та хмари за допомогою ПЗ на основі web-мобільного ПЗ. Крім того, технологія реального часу робить пристрої IoT більш живими за допомогою розумних технологій. Розумні пристрої, такі як гаджети на базі операційної системи Android, привернули увагу ринку завдяки своїй відносно простій і відкритій архітектурі та інтерфейсу програмування додатків. Таким чином, треті сторони можуть легко завантажувати свої програми в хмару, що створює спосіб для розробників зловмисного ПЗ запускати різні шкідливі атаки для доступу до пристроїв IoT з/без дозволу користувача. Тому, розумні пристрої, які використовують Web- та мобільні додатки, вразливі до DoS, пошкодження даних, крадіжки, блуджекінгу, блюснарфінгу тощо.

Інші атаки ініціюються системами IoT через інтелектуальну технологію, на яку атакують взаємозалежні, взаємопов’язані та соціальні системи IoT. Атаки, спричинені взаємозалежними системами IoT, стосуються випадків, коли зловмиснику не потрібно ідентифікувати пристрій користувача для атаки. Наприклад, розумний будинок має різні види датчиків, які контролюють температури, кондиціонування, систему освітлення. Ці датчики також залежать від інших датчиків, які підключені до хмар для оновлення та роботи в ойлайн режимі. Більшість пристроїв IoT пов’язані через глобальну мережу, яка створює широкий спектр атак різних рівнів для пристроїв IoT, це збільшує потенціал різних типів атак

Будь-які вразливості можуть легко поширитися на інші пристрої IoT завдяки взаємопов’язаним системам. Атаки на соціальному рівні є новою системою IoT через збільшення кількості соціальних сайтів, які залучають

користувача до обміну своєю особистою інформацією з іншим користувачем. Таким чином, ці соціальні сайти можуть використовувати інформацію користувача для будь-яких незаконних дій.

1.2.3 Наслідки атак

Атаки на IoT можуть призвести до серйозних наслідків, оскільки пристрої зазвичай зв'язані з інфраструктурою, яка контролює різні аспекти життя. Основними проблемами цього аспекту є цілісність, конфіденційність, інтеграція обмінюваної інформації, конфіденційність користувача та автентичність. При розробці будь-якого протоколу безпеки, щоб протистояти атакам для системи IoT, слід враховувати їх особливості [16-19]. Наслідки IoT атак щодо різних аспектів наведено на рисунку 1.15.

Ідентифікація відбувається перед авторизацією користувача в мережі IoT. Для роботи з хмарним сервером клієнтам необхідно зареєструватися. Однак компроміси та надійність систем IoT створюють проблеми для ідентифікації. Атаки Sybil і спуфінг порушують безпеку мережі, і зловмисники можуть легко отримати доступ до сервера без належної ідентифікації. Таким чином, необхідна ефективна схема ідентифікації для системи IoT, яка може забезпечити захист, маючи системні обмеження.

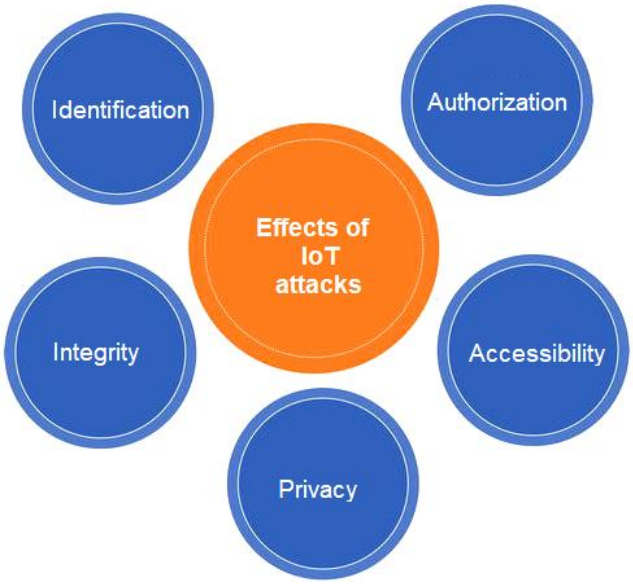

Рисунок 1.15 – Наслідки атак IoT

Авторизація стосується доступу користувача до системи IoT. Дозвіл дається лише авторизованим клієнтам вводити, контролювати та використовувати інформаційні дані мережі IoT, а також виконувати лише їх команди. Справді складно підтримувати всі журнали користувачів і надавати доступ на основі інформації, оскільки користувачі обмежуються не лише людьми, але й датчиками, машинами та службами. Крім того, формування сильного захисного середовища є складним завданням при обробці великих наборів даних клієнта

Доступність гарантує, що послуги системи IoT завжди надаються авторизованим користувачам. Це одна з важливих вимог для створення ефективної мережі IoT, тоді як атаки DoS і створення перешкод порушують цю службу, створюючи непотрібні запити та підтримуючи мережу зайнятою. Отже, потрібен надійний протокол безпеки, щоб служби пристроїв IoT були доступними для їхніх клієнтів без будь-яких перерв.

Конфіденційність це єдиний фактор, з яким пов’язані як активні, так і пасивні атаки в системі IoT. Оскільки все, включаючи конфіденційну та особисту інформацію, медичні звіти, фінансові дані тощо, зберігається та безпечно передається через Інтернет за допомогою різних пристроїв IoT, які не повинні бути розголошені будь-якими неавторизованими користувачами. Однак важко зберегти конфіденційність більшості даних від третіх сторін, оскільки зловмисники можуть ідентифікувати фізичне місцезнаходження, відстежуючи пристрій IoT і розшифровувати інформацію.

Цілісність гарантує, що змінювати інформацію пристроїв IoT під час використання безпровідної мережі для зв’язку можуть тітьки користувачі, які авторизувалися. Ця вимога є фундаментальною для безпеки системи IoT, щоб захистити її від різноманітних атак зловмисного введення, таких як атаки SQL-ін’єкцій. Якщо ця функція якимось чином порушена нерегулярною перевіркою під час зберігання даних у пристроях IoT, це вплине на їх функціональність у довгостроковій перспективі. Бувають випадки, коли це може не тільки розкрити конфіденційну інформацію, але й вплинути на життя людини.

1.3 Основні аспекти захисту Інтернет-речей

IoT генерує великий потік інформації, який потрібно шифрувати на кінцевих пристроях, а вони досить часто характеризуються обмеженнями у споживанні енергії та вартості, але при цьому мають забезпечувати належний рівень безпеки.

Загалом можна виділити 4 рівні захисту, кожен з них відіграє ключову роль у системі IoT (рисунок 1.16) [6-9, 17-19].

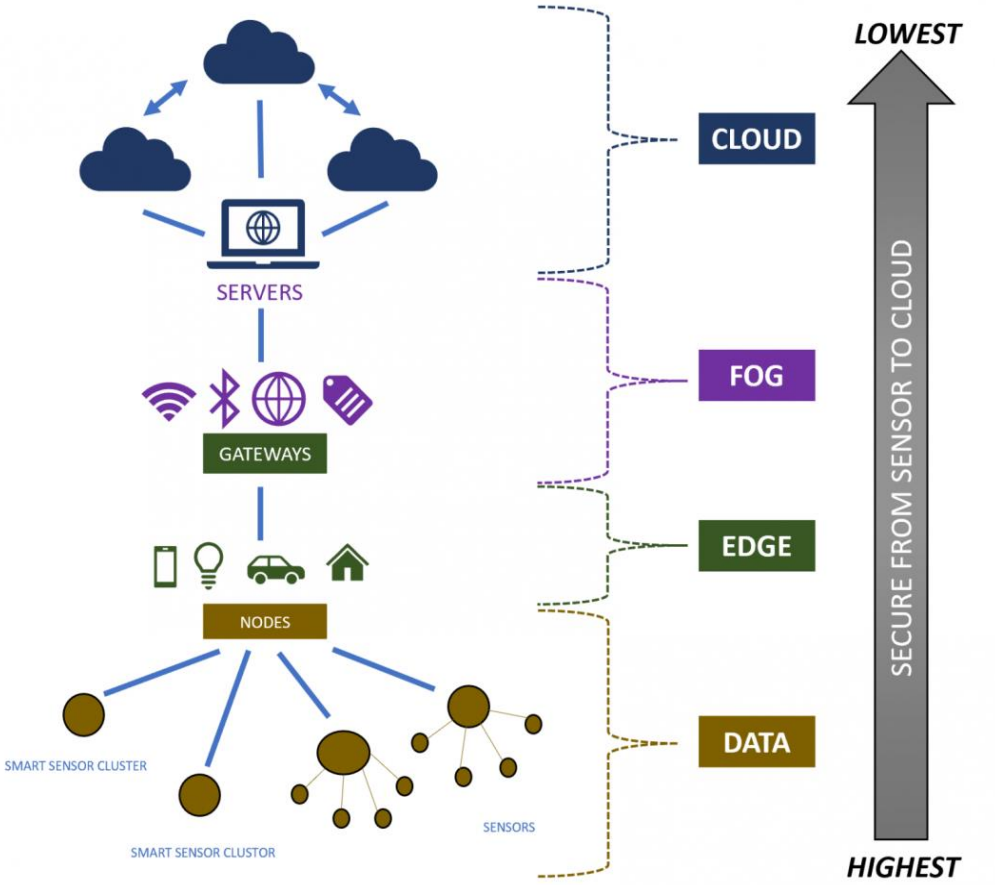

Рисунок 1.16 - Рівень захисту архітектури IoT

Як видно з рисунку 1.16 найнижчий рівень захисту спостерігається у хмарних сховищах, що призводить до найменшого рівня захисту даних, що передаються системою.

На рівні даних (DATA) важливо захистити дані, які отримуються від сенсорів шляхом шифрування, захисту від НД та змін, а також забезпечення цілісності даних. Рівень межі (EDGE) - місце, де дані переходять з пристроїв

IoT до основної мережі. Безпека на даному рівні забезпечується на рівні мережі, аутентифікації пристроїв, захистом від атак, що спрямовані на мережеві протоколи. Рівень передачі даних (FOG) відповідає за передавання та опрацювання даних. Тут важливо використовувати криптографічні методи для захисту при передаванні даних між різними вузлами та забезпечити їх цілісність під час опрацювання. Рівень хмари (CLOUD) - місце де дані великих обсягів, які поступають з пристроїв,зберігаються, обробляються та аналізуються, тому необхідний їх захист від НД, шифрування, резервне копіювання, контроль доступу та моніторинг вразливостей.

Кожен з рівнів потребує вдосконалення та захисту з врахуванням його функцій та ролі в загальній системі IoT. Важливо забезпечити комплексний захист на кожному рівні, щоб уникнути слабких точок у системі і реалізувати безпеку всієї інфраструктури на високому рівні. Охоплення всіх аспектів захисту IoT є складним та широким завданням, проте можна виділити основні напрямки:

1. Шифрування для захисту даних, які передаються пристроями та їх безпечного зберігання, обміну й управління ключами шифрування.

2. Використання методів ідентифікації, автентифікація та авторизації, засобів перевірки ідентичності пристроїв та користувачів, які взаємодіють у мережі IoT.

3. Використання механізмів контролю доступу, які визначають, хто має доступ до певних ресурсів і функцій пристроїв.

4. Використання методів виявлення потенційних загроз та вразливостей в системі IoT.

5. Реалізація фізичного захисту та безпеки мережі, зокрема використання засобів для того, щоб забезпечити захист пристроїв щодо фізичних втручань та доступу до них, а також заходів для забезпечення безпеки мережі, в т.ч. захист від перехоплення даних.

6. Регулярні оновлення ПЗ для усунення вразливостей та удосконалення безпеки.

7. Моніторинг та реагування на вразливості, зокрема впровадження

заходів для запобігання та протидії атакам, в т.ч. захист від DDoS, зламів, витоків інформації та інших загроз.

Застосування методів і засобів захисту у IoT стикається з кількома проблемами безпеки через специфіку цієї області, зокрема:

- велика кількість інтернет-підключень, оскільки пристрої приєднуються та від'єднуються одночасно, тому потрібно переконатися, що інформація не витікає з жодного із з'єднань при передачі.
- проблема заряду у IoT пристроях, тому, що вони працюють на низькому рівні заряду, однак повинні передавати дані з мінімальними затримками;
- вразливості пристроїв мають пряму залежність рівню їх інтелектуальності, тобто чим розумнішим і складнішим є пристрій, тим більше потенційних вразливостей в ньому, які іноді важко виявити дослідникам та тестувальникам.

Зазвичай атаки на пристрої IoT можуть бути різноманітні, тому для їх захисту необхідно використовувати комплексний підхід, включаючи як цифрові так і фізичні заходи захисту, що разом допомагають убезпечити пристрої та мережу IoT від можливих загроз.

Поширеними засобами для захисту при DDoS-атаках є використання систем фільтрації трафіку, розпізнавання та блокування недійсних запитів, мережеві файерволи та системи виявлення вторгнень (IDS/IPS). Експлойт ПЗ – передбачає використання відомих вразливостей у ПЗ для отримання доступу до пристрою IoT. Проте регулярні оновлення ПЗ, патчі для виправлення вразливостей, використання захисту від вірусів та шкідливих програм дозволяють виправляти вразливості та усувати відомі слабкі місця. Для захисту використовуються ПЗ для виявлення й запобігання атак, наприклад фішингу, вірусів чи розподіленій відмові в обслуговуванні. Ефектимним засобами захисту від MITM атак є аутентифікація та авторизація користувачів або пристроїв, а також належний контроль прав доступу, використання шифрування трафіку, захищених каналів зв'язку.

Для попередження фізичного доступу до пристрою IoT для отримання

несанкціонованого контролю над ним використовуються з захищені корпуси, обмеження фізичного доступу до пристроїв. Важливим є також захист з використанням біометричних методів, двофакторної перевірки або фізичних бар'єрів та обмежень доступу, блокування портів для підключення зовнішніх пристроїв тощо. Для захисту від брутфорс атак, спроб зламати паролі або ключі шляхом послідовного випробування різних комбінацій, необхідно щоб пароль був складним та унікальним, використовувати обмежену кількість спроб його вводу, двофакторна аутентифікація.

Постійний аналіз мережі допомагає виявляти підозрілі або несподівані активності, що свідчать про можливу атаку на систему IoT. Фільтри пакетів та права доступу використовуються для блокування небажаного мережевого трафіку, що можуть бути ознаками аномальної активності або потенційної загрози безпеці. Це дозволяє системі виявляти незвичайні паттерни чи атаки та реагувати на них. Також, шифрування трафіку між пристроями IoT та центральними системами через використання віртуальної приватної мережі (VPN) є ефективним способом, щоб забезпечити конфіденційність та попередити перехоплення чи зміну інформації при передаванні її між пристроями. Дотримання встановлених стандартів безпеки, застосування політик, які забезпечують захист особистої інформації користувачів та обмеження доступу до неї дозволяють забезпечити надійне та безпечне середовище для пристроїв IoT та їх користувачів.

Дані методи дозволяють досягнути досить високого рівня безпеки для систем IoT, допомагать вчасно реагувати на потенційні загрози та підозрілу активність.

2 АНАЛІЗ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ІНТЕРНЕТ-РЕЧЕЙ

2.1 Криптографічні алгоритми захисту даних

«Криптографія - є наукою про забезпечення конфіденційності, цілісності та автентичності інформації шляхом застосування різних методів та технік» [11]. КА представляють різні підходи до захисту даних та комунікаційних процесів в IoT, кожен з яких має свої характеристики та відповідність для різних сценаріїв застосування (рисунок 2.1) [10-13].

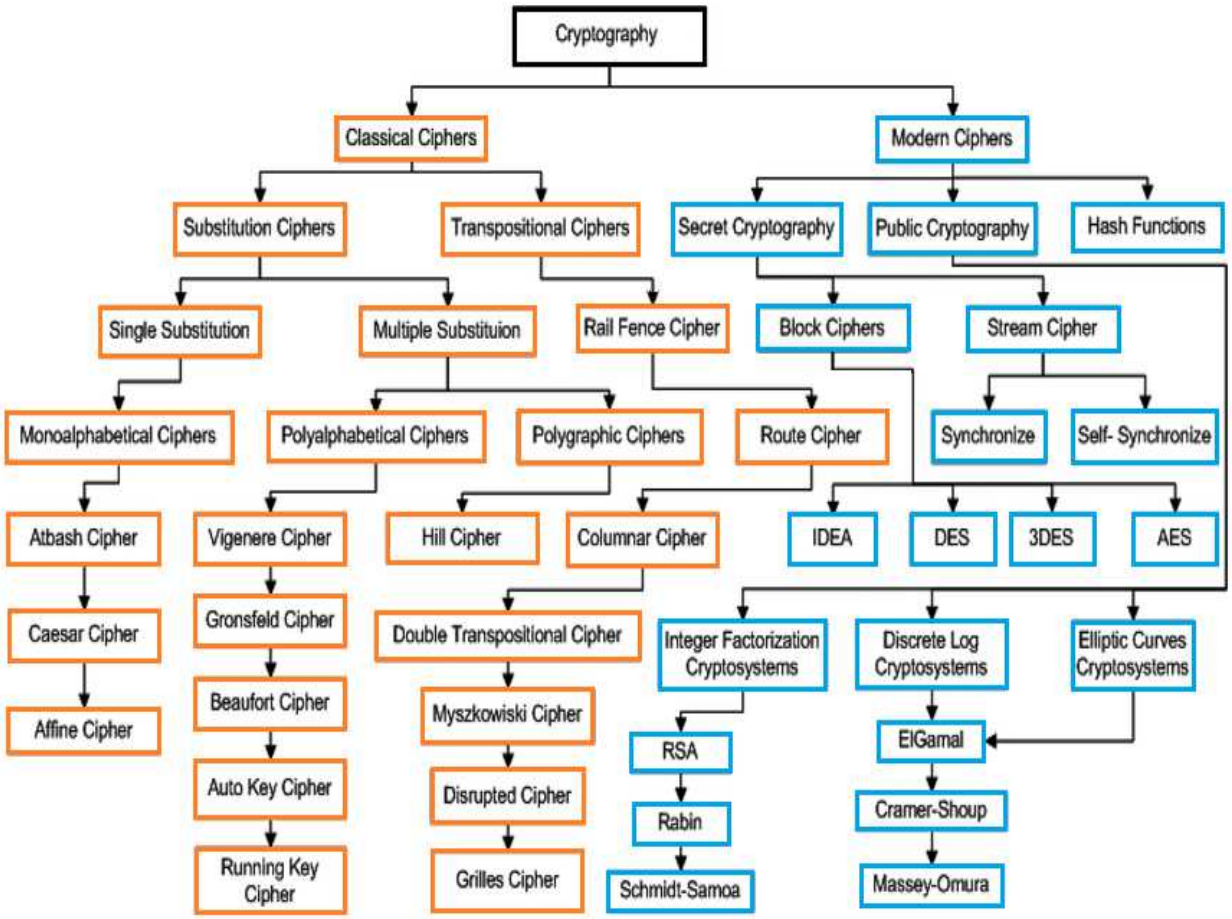

Рисунок 2.1 – Криптографічні алгоритми

Основні КА можна умовно поділити на класичні (традиційні) та сучасні [19]. До першої групи відносяться підстановка, заміна символів або блоків символів на інші відповідно до певного ключа (шифрувальної таблиці) та транспозиція, перестановка символів або блоків символів за певним правилом, не змінюючи самі символи. До другої – шифрування, хешування, цифрові підписи та ін. Сучасні методи криптографії, особливо асиметричне

шифрування, забезпечують вищий рівень безпеки порівняно з класичними методами. Класифікація основних алгоритмів шифрування, що застосовуються в контексті захисту IoT зображена на рисунку 2.2 [6-9].

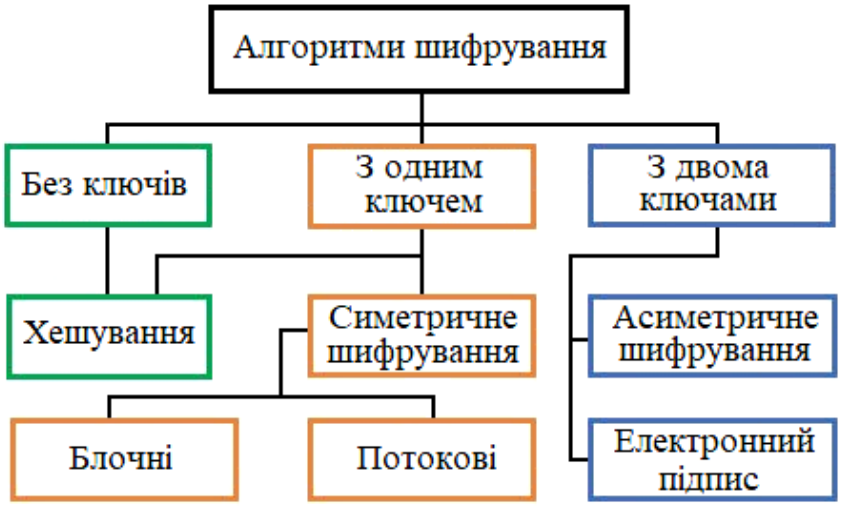

Рисунок 2.2 – Класифікація алгоритмів шифрування

«AES (Advanced Encryption Standard) - симетричний шифр, який використовується для шифрування та розшифрування даних, що є одним з найбільш поширених алгоритмів шифрування» [19]. Застосовується для захисту конфіденційності даних, які передаються між пристроями IoT.

Перевагою є висока швидкість та ефективність, він важається надійним для багатьох застосувань. Недоліками AES є те, що для роботи з довшими ключами безпеки необхідно більше ресурсів. Також він може бути вразливим до деяких атак, якщо не використовується належним чином.

«RSA (Rivest-Shamir-Adleman) - асиметричний шифр, який використовує пару ключів - приватний та публічний - для шифрування та розшифрування даних»[10]. Зазвичай використовується для підпису даних та обміну ключами для подальшого симетричного шифрування.

Перевагою є цього ефективність для підписування та обміну ключами, оскільки він заснований на складності факторизації великих простих чисел. Недоліком є те, що він повільний у порівнянні з іншими симетричними алгоритмами, особливо для операцій шифрації / дешифрації та ресурсозатратний.

«ECC (Elliptic Curve Cryptography) - використовує математичні

властивості еліптичних кривих для створення ключів та шифрування даних» [20]. Часто використовується через обмежені ресурси пристроїв, оскільки необхідна менша кількість ОР, але при цьому дозволяє забезпечити безпеку на високому рівні. Використовує менше ОР порівняно з RSA.

Перевагою є можливість висока ефективність алгоритму для пристроїв з обмеженістю ресурсів. Проте він залежить від безпеки еліптичних кривих, що можуть бути під загрозою в разі невдалого вибору кривих або параметрів.

«ECDSA (Elliptic Curve Digital Signature Algorithm) - КА для створення цифрових підписів на основі еліптичних кривих» [19].

SHA (Secure Hash Algorithms) - хеш-функції, які перетворюють вихідні дані в унікальний, фіксованої довжини хеш-код, який важко змінити назад у вихідні дані. Наприклад, функції MD5, SHA-256. Алгоритм використовується для перевірки цілісності даних та підтвердження їх автентичності [21].

Переваги полягають у швидкості та ефективності перевірки, проте він вразливий до колізій, ситуацій, коли два різних вхідних повідомлення дають одинаковий хеш для деяких версій алгоритмів.

Використання хеш-функцій або цифрових підписів для перевірки цілісності даних під час їх передачі в мережі IoT. Це дозволяє виявити будь-які зміни в інформації під час передачі.

HMAC (Keyed-Hash Message Authentication Code) - конструкція аутентифікації повідомлення, яка використовує хеш-функцію у поєднанні з секретним ключем [22]. Використовується для перевірки цілісності та достовірності повідомлень в мережі IoT. Забезпечує аутентифікацію повідомлень, відомий своєю безпечністю та використанням хеш-функцій але вимагає безпечного обміну ключами для ефективної роботи.

Використання електронних цифрових підписів для підтвердження та авторизації даних, переданих між пристроями в мережі IoT може забезпечити контроль доступу та ідентифікацію користувачів.

PKI (Public Key Infrastructure) використовується для забезпечення безпеки мережі та підтвердження ідентичності за допомогою цифрових сертифікатів, які містять публічні ключі [19]. Цифрові сертифікати

використовуються для підтвердження ідентичності пристроїв перед взаємодією в мережі IoT. Наприклад, підключення нового пристрою до мережі шляхом обміну сертифікатами для перевірки автентичності.

Протокол для обміну ключами Diffie-Hellman дозволяє двом сторонам безпечно обмінюватися секретними ключами через незахищенну мережу. Зашифровані з'єднання між девайсами, наприклад, застосування протоколів шифрування, таких як TLS або SSL, використовуються для захисту передавання інформації між різними пристроями в мережі IoT.

Ці КА є основою для забезпечення безпеки і захисту даних та застосовуються у різних областях, включаючи захист IoT, електронну комерцію, забезпечення безпеки мережі та багато інших.

2.2 Оцінка ефективності застосування криптографічних методів захисту в умовах обмеженості ресурсів

Оцінка ефективності КА включає різні аспекти, такі як швидкість обчислення, витрати ресурсів пристроїв, безпека та виконання конкретних функцій.

Порівняння різних КА, з врахуванням розміру ключа, стійкості, продуктивності та вимог до ресурсів, наведена в таблиці 2.2. Розмір ключа вказаний в бітах. Стійкість оцінюється як висока, середня або вразлива залежно від здатності алгоритму витримувати різні види атак. Продуктивність оцінюється як висока, середня або низька в залежності від швидкодії алгоритму та вимог до ОР. Вимоги до ресурсів оцінюються як високі, середні або низькі в залежності від ресурсів, необхідних для ефективного виконання алгоритму.

З точки зору швидкодії максимально ефективними є AES та ECC, проте ECC вимагає менше ОР. RSA відносно повільний у порівнянні з деякими іншими алгоритмами, особливо для операцій зашифровування та розшифровування. З огляду на використання ОР при ECC, зазвичай, використовується менше ОР та обсягу пам'яті для роботи. RSA, в свою чергу,

може вимагати більше ресурсів, особливо для довгих ключів.

Таблиця 2.1 – Порівняння криптографічних алгоритмів захисту

Алгоритм	Тип алгоритму	Розмір ключа	Стійкість	Продуктивність	Вимоги до ресурсів
DES	Симетричний	56 біт	Вразливий	Висока	Низькі
AES	Симетричний	128, 192, 256 біт	Висока	Висока	Середні
RSA	Асиметричний	1024+ біт	Висока	Низька	Високі
ECC	Асиметричний	160-521 біт	Висока	Висока	Низькі
SHA-256	Хеш-функція	-	Висока	Висока	Низькі

Таблиця 2.1 демонструє загальну оцінку алгоритмів на підставі визначених критеріїв, проте при виборі конкретного КА необхідно враховувати сферу їх застосування, вимоги щодо безпеки та характеристики пристроїв IoT.

AES, RSA та ECC мають широку підтримку та реалізацію в різних системах, тому вони використовуються в багатьох сучасних застосунках. Інші алгоритми можуть бути складними для реалізації на пристроях у яких обмежені ресурси через їхню складність чи обсяг.

RSA та ECC вважаються алгоритмами, які важко піддавати криптоаналізу, AES має безпеку достатньо високого рівня. На відміну від зазначених, SHA може бути вразливим до колізій (ситуацій, коли два різних вхідних повідомлення дають одинаковий хеш) для деяких версій алгоритмів.

Кожен алгоритм характеризується певними перевагами, тому вибір конкретного для застосування в IoT залежить від конкретних вимог до безпеки, ресурсів пристроїв та обсягів інформації, які необхіфдно

зашифрувати та обробити.

Приклад використання КА в IoT наведено на рисунку 2.3 [23].

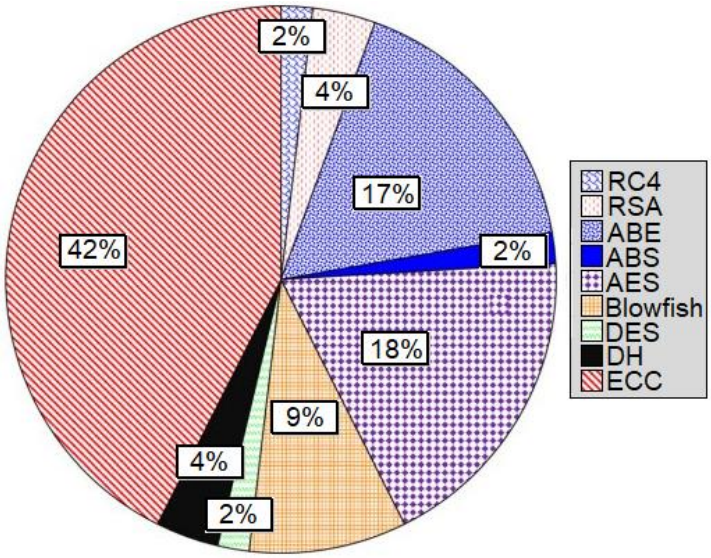

Рисунок 2.3 – Використання криптографічних алгоритму в IoT

Для кращого розуміння проблеми застосування та аналізу різних криптографічних методів в IoT стосовно врахування обмеженості ресурсів, необхідно розглянути архітектуру IoT, розподілену на чотири логічні рівні, які можуть взаємодіяти як вертикально, так і горизонтально (рисунок 2.4).

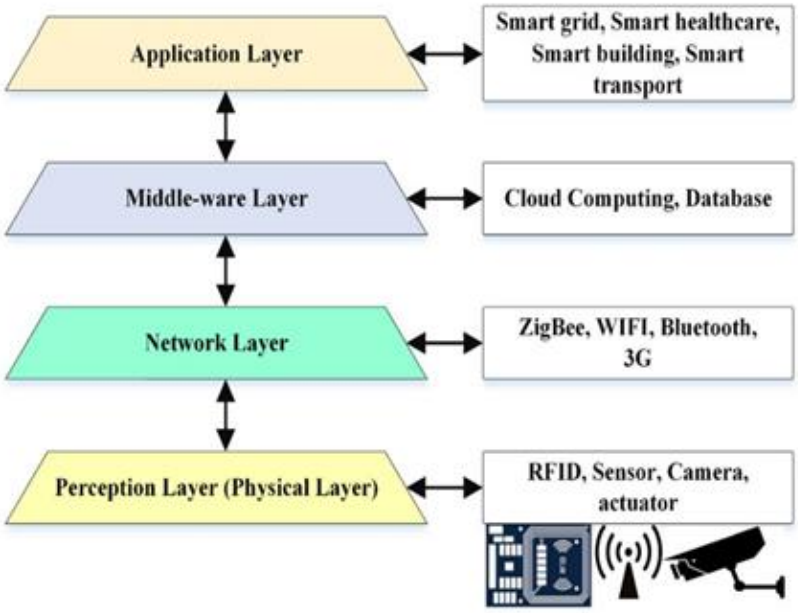

Рисунок 2.4 - Чотирирівнева структура IoT

Особливими характеристиками кожного з рівнів в IoT є:

– рівень додатків - де IoT зустрічається з користувачем.

− рівень обчислювального ядра реалізує обробку даних (великих обсягів), використання алгоритмів аналітики, зберігання інформації, симуляцію IoT в реальному часі та дистанційне управління.

− рівень комунікацій на якому, за допомогою провідної та безпровідної передачі даних, різні комунікаційні протоколи, мережі та їх архітектури, пристрої взаємодіють як між собою, так і з хмарним сховищем.

− рівень фізичних пристроїв - збір даних датчиками (невеликі обсяги), можливість дистанційного управління, низькопотужні та обмежені ресурсами пристрої, енергоефективність.

Кожен із вказаних рівнів архітектури IoT неодмінно пов'язані із безпекою (таблиця 2.2).

Таблиця 2.2 – Рівні структури IoT

Рівень додатків	персоналізовані додатки; сфери застосування.	Безпека IoT
Рівень IoT хмари та обчислювального ядра	обробка та зберігання Big Data; управління пристроями; застосування алгоритмів машинного навчання.	
Рівень мережі	передача даних; комунікаційні протоколи; мережеві протоколи та з’єднання.	
Рівень фізичних пристроїв	сенсори; датчики; RFID комунікації; дані.	

Використання КА в першу чергу стосується рівня фізичних пристроїв і взаємодії з ним. Датчики отримують дані, які потрібно передавати. Без шифрування дані можна легко перехопити, просто прослуховуючи канал передачі. На даному рівні КА вирішують наступні задачі:

- конфіденційність - захист від НД до інформації. Дані, зашифровані алгоритмами, залишаються конфіденційними, доступними лише авторизованим користувачам.
- цілісність - гарантує, що дані залишилися незмінними під час передавання, тобто ніхто не може втрутитися та змінити їх у процесі передачі.
- автентичність повідомлення - дозволяє перевірити, що отримане повідомлення є дійсним і було створено певним автором.

Незважаючи на існуючі надійні методи шифрування, існують певні обмеження, які можуть призвести до пошуку нових рішень, зокрема можливості щодо обробки складних алгоритмів шифрування через обмежену пам'ять та обчислювальну потужність пристроїв IoT. Деякі алгоритми шифрування можуть мати високі вимоги до ОР і це може бути недоцільним для пристроїв із обмеженими можливостями.

У контексті IoT методи криптографії повинні бути легкими для використання на пристроях у яких обмежені ресурси, ефективними у аспекті опрацювання даних та достатньо безпечними для захисту комунікації та даних. Тому, для IoT особливо важливі такі криптографічні методи:

- шифрування, наприклад, асиметричне оскільки його використання вимагає менше ОР, але забезпечує безпеку обміну ключами та зашифрованих повідомлень, зокрема вибір алгоритму ECDSA може бути ефективним для пристроїв IoT;
- хеш-функцій, які використовуються для перевірки цілісності даних, але не вимагають значних ОР, наприклад для пристроїв IoT може застосовуватися SHA-256;
- цифрові сертифікати для автентифікації та забезпечення безпеки обміну даними, що відповідає ресурсам пристроїв IoT;
- протоколи обміну ключами із оптимізацією ОР, які є безпечними, але не вимагають великих обчислювальних витрат, наприклад, протоколи типу DTLS або Lightweight MQTT Security (LwM2M).

Загалом, оптимальний вибір криптографічного методу для IoT

залежить від конкретного використання, обмежень пристроїв та ресурсів, а також потреб у безпеці та ефективності взаємодії між пристроями. Тому, алгоритми шифрування для IoT мають бути ефективними, маючи низький рівень обчислювального навантаження, а також забезпечувати необхідний рівень безпеки для захисту передаваних даних.

2.3 Механізми малоресурсної криптографії

Класична криптографія стає практично непридатною для використання, наприклад, RSA потребує великого розміру ключа та значних обчислювальних потужностей. Тому використовують так звану «малоресурсну криптографію» (Lightweight Cryptography) (МРК), яка більш сумісна з оточенням IoT [14, 15].

МРК - галузь криптографії, яка спеціалізується на розробці КА і протоколів, спроектованих для пристроїв в яких обмежені ресурси, таких як мікроконтролери, мікропроцесори, програмовані логічні інтегральні схеми (ПЛІС), IoT-пристрої, сенсори тощо. Основною мето є забезпечення ефективного рівня безпеки за обмежених умов обчислювальних, енергетичних та обсягових обмежень.

Залежно від призначення, виділяють такі типи МРК: для ПЗ та апаратного забезпечення, зокрема [13-15, 19]:

- блокові шифри, що включають алгоритми шифрування, які працюють з фіксованими блоками даних та ключами, наприклад, AES, які можуть бути оптимізовані для легкості обчислення та використовуватися в малих вбудованих системах;
- малоресурсні хеш-функції - призначені для створення фіксованих рядків символів (хешів) з вхідних даних та можуть бути оптимізовані для використання на пристроях низького використання ресурсів;
- малоресурсні потокові шифри - відрізняються від блокових шифрів тим, що вони шифрують великі потоки даних без поділу на блоки. Їх особливість - працездатність з великими обсягами інформації та ефективність

в умовах обмежених ресурсів.

Класифікація МРК показана на рисунку 2.5.

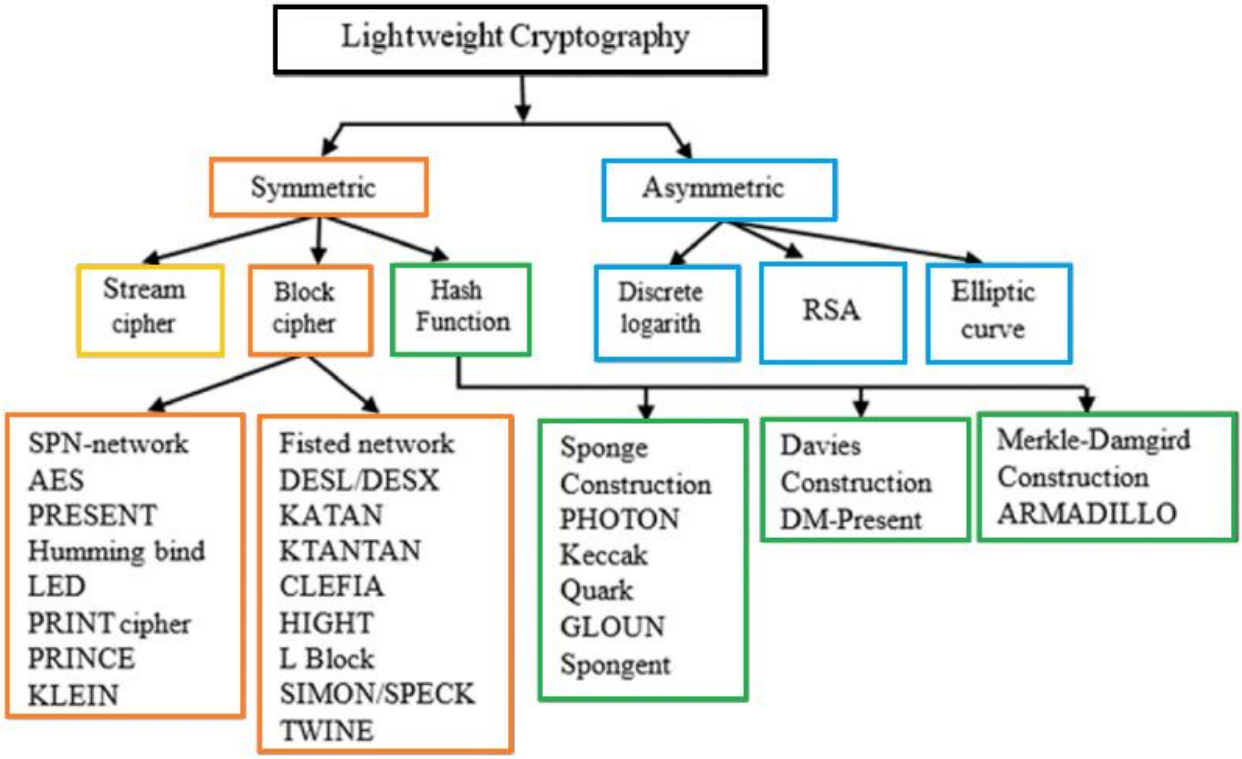

Рисунок 2.5 - Класифікація МРК

При використанні перелічених способів необхідний час для ініціалізації, під час якої шифрування не відбувається.

Основними проблемами в аспекті захисту IoT є стійкість та конфіденційність даних, інтеграція обмінюваної інформації, конфіденційність та автентичність користувача. Жорсткі обмеження на внутрішні ОР пристроїв IoT утруднюють або роблять неможливим використання класичних КА, що зумовлює використання МРК, яка поєднує надійні та ефективні алгоритми з мінімальними витратами на обладнання.

Актуальним завданням є оптимізація параметрів малоресурсного блокового шифрування (МБШ) [15-17, 24-30]. Для систематизації основних підходів до побудови існуючих МБШ необхідно оцінити наступні їх характеристики:

- нелінійність всіх координатних функцій циклового перетворення;
- властивості перемішування циклового перетворення;
- стійкість ознаки нелінійності перетворення.

Для оцінки нелінійності блокового шифрування використовується поняття показника сильної нелінійності (exponent of strong non-linearity), що є найменшою кількість циклів, при якому всі координатні булеві функції циклового перетворення є нелінійними.

При побудові МБШ застосовуються такі архітектурні рішення, що відрізняють їх від класичних блокових шифрів:

- зменшення розмірів основних параметрів алгоритму, наприклад розмір блоку з 128 до 64 біт, використання ключів довжиною 64, 80 і 128 біт;
- використання спрощеного ключового розкладу;
- проектування алгоритмів на основі широко застосовуваних операцій, які здійснюють елементарні лінійні/нелінійні перетворення;
- зменшення розмірів даних, що використовуються в конкретних операціях, наприклад відмова від 8-бітових s-боксів на користь 4-бітових;
- використання необтяжливих щодо ресурсомісткості, але ефективних перетворень (бітові перестановки, регістри зсуву тощо).

На сьогоднішній день відомо досить багато МБШ і на основі SP-мереж (Substitution–Permutation Network), і мереж Фейстеля (Feistel Network) [28]. Обидва підходи мають свої позитивні характеристики та проблеми у контексті побудови алгоритмів в умовах обмежених ресурсів.

З метою оцінки можливостей оптимізації параметрів та пошуку нових рішень проведено дослідження алгоритмів PRESENT [26] та CLEFIA [27], що включені у міжнародний стандарт ISO/IEC 291922:2012 [31], а також нових алгоритмів, LILLIPUT [28], MIDORI [29] та SKINNY [30].

Показник сильної нелінійності ітеративного блокового алгоритму шифрування визначчатиметься через мінімальну кількість циклів, необхідну для того, щоб кожна координатна функція вихідного блоку була нелінійною. Проведено експериментальну оцінку показника сильної нелінійності.

Якщо перетворення є нелінійним, то існують такі значення $x, x', a \in V_n$, $a \neq 0$, що задовільняють умову

$$f(x) + f(x+a) \neq f(x') + f(x'+a) \quad (1)$$

Під час експерименту для випадково обраної значень x,x′,a

шифруються вектори $x, x', x+a, x'+a$ при певному значенні числа циклів шифрування. Якщо знайдеться значення, коли кожна координатна функція задовільняє умову (1), то показник сильної нелінійності не перебільшує число циклів шифрування.

Варто зазначити, що суперпозиція нелінійних функцій може дати систему, що містить лінійні рівняння. Наприклад, розглянемо таку систему:

$$\begin{cases} \mathrm{f}_1(x, y) = x \oplus y \oplus xy, \\ f_2(x, y) = x \oplus xy. \end{cases} \quad (2)$$

Виконавши підстановку функцій f_1, f_2 замість аргументів x, y отримаємо

$$\begin{cases} f_1(f_1(x, y), f_2(x, y)) = x \oplus y \oplus xy, \\ f_2(f_1(x, y), f_2(x, y)) = y. \end{cases} \quad (3)$$

Експериментально перевірено, що для всіх розглянутих алгоритмів нелінійність координатних функцій вихідного блоку зберігається протягом 500 циклів.

Ця властивість є важливою, оскільки МБШ використовуються при побудові ключового розкладу або хеш-функцій, наприклад, PRESENT.

Для оцінки властивостей перемішування, застосовується матрично-графовий підхід [32], при якому суттєва залежність координат вихідних векторів від координат вхідних векторів кодується матрицею перемішування порядку n: елемент матрици $\mathrm{m_{ij}}$ дорівнює 1 де $i, j \in \{1, ..., n\}$, якщо є істотна залежність *j*-ї координатної функції виходу від *i*-ї координати входу, і 0-у іншому випадку.

Матриця називається матрицею перемішування або матрицею істотної залежності. Для ітеративних перетворень оцінка властивостей перемішування полягає у вивченні примітивності матриць та визначенні їх експонентів. Матриця примітивна, якщо певний її степінь не містить нульових елементів. Найменший з таких степінь називається експонентом матриці.

Для дослідження властивостей перемішування, МБШ розглядалися

шифри, для яких отримані значення експонентів матриць перемішування, побудованих для циклових функцій. За допомогою ПЗ здійснено послідовне зведення матриць до ступеня та отримано значення їх експонентів.

Перемішуючі властивості оцінені експериментальним способом, тобто отримана оцінка понад показник досконалості, який дорівнює найменшому числу циклів, при якому кожна координатна функція істотно залежить від кожної вхідної координати. В ході експерименту для кожної вхідної координати було обрано 20 пар сусідніх до неї випадкових векторів $(x_1, x_1'), \ldots, (x_{20}, x_{20}')$, які шифруються при певному значенні числа циклів шифрування. При виконанні умови

$$(f(x_1) \oplus f(x_1')) \vee \ldots \vee (f(x_{20}) \oplus f(x_{20}')) = e, \quad (4)$$

де *e*-вектор з одиниць відповідної довжини, то показник досконалості не перевищує поточного числа циклів.

Результати дослідження наведено у таблиці 2.3.

Таблиця 2.3 – Результати дослідження МБШ

Шифр	Кількість циклів	Розмір матриці	Експонент матриці	Показник досконалості	Показник сильної нелінійності
Present	32	64x64	3	4	1
Midori	16, 20	64x64	3	3	1
Skinny	32, 36, 40, 48, 56	64x64	6	6	1
Clefia	36, 44, 52	128x128	5	5	2
Liliput	30	64x64	5	5	2

Визначені показники сильної нелінійності циклових перетворень МБШ, показники досконалості та експоненти матриць суттєвої залежності, значення яких суттєво менші від числа циклів шифрування, що вказує на потенціал для можливої оптимізації алгоритмів шляхом скорочення числа циклів шифрування. Експериментально перевірено, що у всіх

розглянутих шифрах нелінійність кожної координатної функції вихідного блоку зберігається протягом великої кількості циклів шифрування, що вказує на можливість їх використання при побудові ключового розкладу та хеш-функцій.

2.4 Вимоги до криптографічного захисту Інтернет-речей

Проведені дослідження дозволяють сформулювати основні вимоги стосовно криптографічного захисту IoT. Вони, зокрема, включають:

- стійкість до кіберзагроз - криптографічний алгоритм повинен бути стійким до різних видів кібератак, таких як перехоплення, підробка та атаки зміни даних;
- ефективність при умові обмежених ресурсів - алгоритм має бути ефективним у використанні ресурсів обчислювальної потужності, використання пам'яті та споживання енергії для задоволення потреб пристроїв IoT;
- відповідність стандартам безпеки - вимагається відповідність до визнаних стандартів безпеки, що дозволяє співпрацювати та інтегруватися з іншими системами безпеки
- масштабованість - алгоритм повинен бути масштабованим для різних варіантів застосування в IoT, від пристроїв у яких існують обмеження щодо характеристик та можливостей до більш потужних систем;
- управління ключами та сертифікатами – захист повинен передбачати розробку механізмів управління ключами та сертифікатами, що повинні бути ефективними та забезпечувати безпеку обміну ключами та автентифікації;
- стандартизація та сумісність - важливо, щоб запропонований алгоритм враховував різнорідність пристроїв та мереж, був сумісним і стандартизованим для забезпечення взаємодії з іншими засобами й системами безпеки;
- підтримка оновлень та адаптабельність – повинна передбачатися

можливість оновлення та адаптації до нових вимог безпеки, зокрема до зміни обставин та з'явлення нових загроз.

Ці вимоги враховують потреби безпеки у мережі IoT, відповідають обмеженим ресурсам пристроїв та забезпечують стійкість до різних видів загроз безпеці.

2.5 Алгоритм вибору оптимального способу шифрування

Одниим з найважливіших засобів для реалізації безпеки є спосіб шифрування даних. Проте їх впровадження у підключені пристрої має як переваги, так і труднощі, які варто врахувати. Серед недоліків основними є обмеженість обчислювальних можливостей та пам'ять і використання складних КА може бути не можливим через низьку потужність обробки даних.

Деякі алгоритми можуть вимагати багато енергії для шифрування або розшифрування даних, який не може бути прийнятним для пристроїв з невеликими джерелами енергії. Важливим є враховування сумісность різних пристроїв та стандартизацію криптографічних протоколів для забезпечення правильної взаємодії та безпеки в мережі IoT.

Переваги, які досягаються за рахунок застосування шифрування даних включають захист інформації, якою обмінюються пристрої, від перехоплення, підтвердження ідентичність пристроїв та користувачів, що зменшує ризик НД. Також забезакпення цілісності даних та виявлення будь-яктх змін при передачі завдяки використанню хеш-функцій, а відповідно обрані та правильно налаштовані КА можуть забезпечити стійкість до різних видів кібератак.

Для оцінки ефективності алгоритмів малоресурсного шифрування (АМРШ) в IoT проаналізуємо, як вони враховують обмежені ресурси підключених пристроїв. АМРШ, як правило, буває двох типів.

Симетричні - використовують один ключ як для шифрації, так і для дешифрування повідомлень. Деякі приклади малоресурсного симетричного

шифрувння (МСШ) включають Simon, Speck, KLEIN, й інші [24-32].

Асиметричні використовують два ключі - публічний та приватний, наприклад, такі як ECDSA, також можуть адаптовуватися для малоресурсних умов. АМРШ проектуються так, щоб забезпечувалася висока стійкість і ефективність, враховуючи обмежені ресурси, які характерні для пристроїв IoT.

Підбираючи параметри, такі як вибір блокового або потокового шифру, щодо розмірів ключа та блоку, структури блокового шифру та кількості циклів, можна реалізувати малоресурсні симетричні алгоритми. Важливо враховувати обмеженість ресурсів компонентів IoT під час адаптації класичних алгоритмів, наприклад, AES, до вимог і можливостей IoT пристроїв. Характеристики МСШ наведено в таблиці 2.4.

Таблиця 2.4 – Малоресурсні симетричні алгоритми

Алгоритм	Довжина коду	Структура побудови шифру	Кількість циклів	Розмір ключа	Розмір блоку	Потенційні атаки
AES	2606	SPN	10	128	128	Атака посередника
HEIGHT	5672	GFS	32	128	64	Атака з переповненням
TEA	1140	Feistel	32	128	64	Атака на звязаних ключах
PRESENT	936	SPN	32	80	64	Диференціальна атака
RC5	Не фіксовано	ARX	20	16	32	Диференціальна атака

У контексті IoT МСШ (таблиця 2.3) більш підходять завдяки швидкості операцій, які в основному ґрунтуються на операціях XOR та перестановок.

Вони мають високу швидкість обробки та не вимагають значних ресурсів.Асиметричні шифри менше підходять для АМРШ, але все ж потрібно враховувати їхню роль. КА з асиметричним ключем відомі як КА з відкритим ключем, оскільки в цьому методі потрібна пара відкритих та закритих ключів.

Малоресурсе асиметричне шифрування (МАШ) складне у роботі та не менш ефективне за критерієм часу (таблиця 2.5). Останнім часом акцент щодо МРК зсунувся в бік КА з асиметричним ключем, проте результати ще не такі стабільні та продуктивні, як у криптографії з симетричним ключем.

Таблиця 2.5 –Характеристики МАШ та види потенційних атак

Алгоритм	Розмір ключа	Довжина коду	Атаки
RSA	1024	900	Модульна атака
ECC	160	8838	Атака по часу

В таблиці 2.6 наведено порівняння симетричного та асиметричного способів АМРШ.

Таблиця 2.6– Порівняння малоресурсних алгоритмів шифрування

Характеристики	МСШ	МАШ
1	2	3
Ключ	Один спільний секретний ключ	Пара відкритого для передачі ісекретного ключів
Астекти безпеки	Конфіденційність, автентифікація	Конфіденційність
Швидкість та складність	Швидший за асиметричні	Передача та обчислення двох ключів потребує ресурсів та часу

Продовження таблиці 2.6

1	2	3
Апаратна реалізація	Менш складні, потребують менше ресурсів, оскільки операція XOR є достатньо простою	Досить складні тому потребубть більше ресурсів на апаратному рівні
Недоліки	Вразливі до великої кількості атак, ключ набато лакше перехопити	Швидкість та складність
Приклади	Блокові шифри AES, DES, Blowfish, TEA Потокові Trivium, Chacha	RSA, DSA, ECC

Для визначення який спосіб шифрування, у якому випадку застосовувати краще, запропоновано алгоритм, що дозволяє врахувати різноманітні ресурси пристроїв IoT. Він полягає у комбінуванні різних типів шифрування для створення більш оптимального та пристосованого до умов специфіки IoT. Однією із основних переваг запропонованого алгоритму є гнучкість у виборі оптимального способу шифрування, що відповідає конкретним потребам та обмеженням певного IoT пристрою.

Наприклад, можна використовувати більш витратні методи там, де ресурси дозволяють, та переходити до варіантів з меншими вимогами для пристроїв у яких обмежені ресурси.

Алгоритм дозволяє враховувати індивідуальні потреби кожного IoT пристрою та забезпечує більш гнучкий та оптимальний підбір методів шифрування, адаптованих до конкретних умов його функціонування.

На рисунку 2.4 наведена схема алгоритму вибору оптимального методу шифрування для пристроїв IoT. Входом є параметри пристрою IoT, а виходом - відповідний АМРШ. На початковому етапі вводяться параметри конкретного пристрою, зокрема розмір даних (РД), потужність джерела живлення - батареї (ПБ), об’єм пам'яті (ОП), та обчислювальні ресурси (ОР).

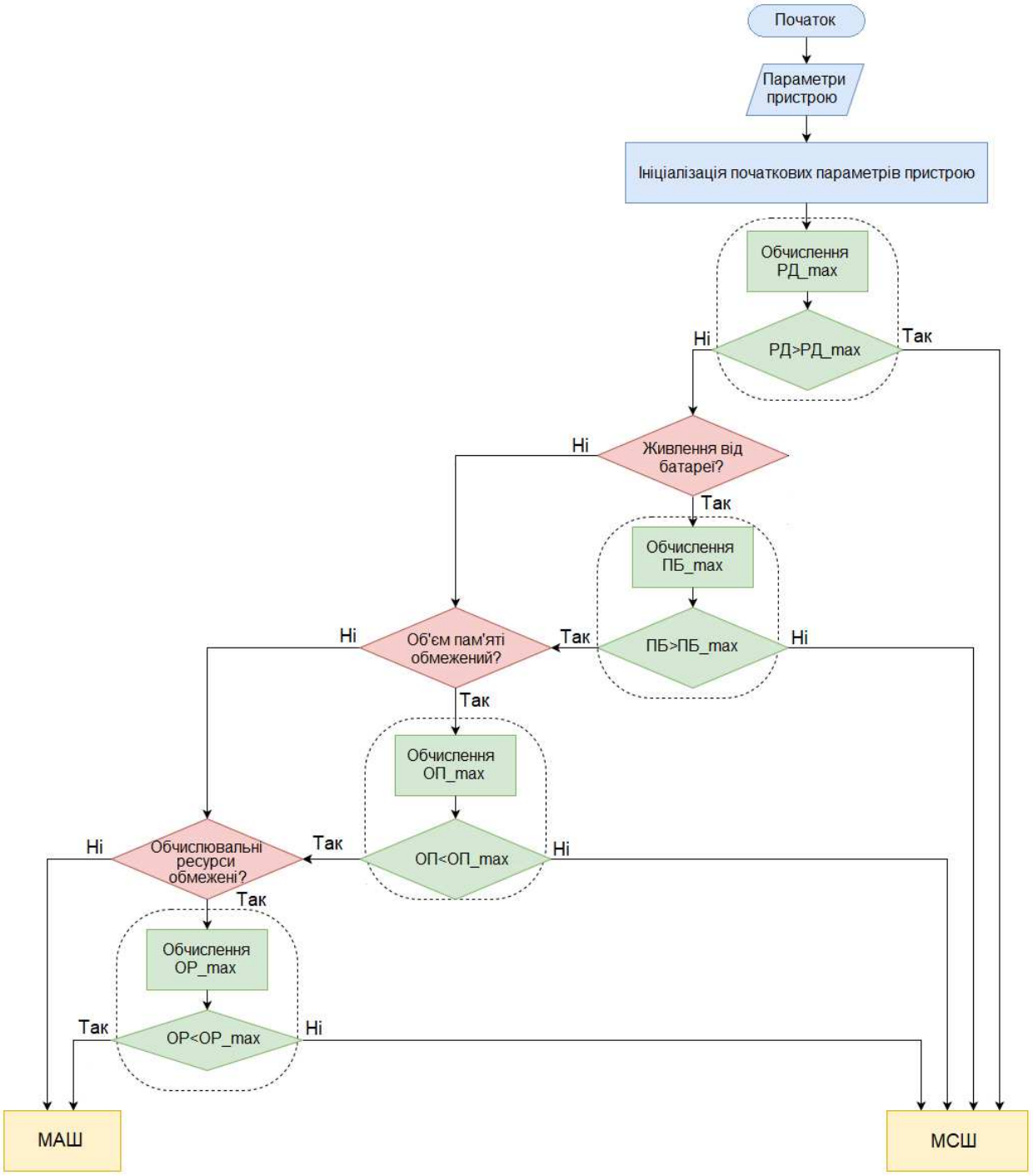

Рисунок 2.6– Алгоритм вибору оптимального методу шифрування

Далі відбувається ініціалізаціця параметрів, тобто присвоєння об’єкту початкових значень. Розрахунок граничних значень кожного з параметрів відбувається за певним алгоритмом або є зазадалегідь відомими характеристиками пристрою.

На першому етапі аналізується РД, що передаються у мережі. Вибір алгоритму шифрування відбувається з врахуванням РД. Якщо РД перевищує максимальне значення РД_max, вони вважаються даними великого обсягу,

для яких рекомендується використання МСШ. Це обумовлене більшою складністю та великим обсягом обробки даних при використанні складних алгоритмів на пристроях Інтернет-речей. У іншому випадку обсяг даних визначається як не великий та відбувається перехід до наступного етапу аналізу.

За аналогією відбувається аналіз інших параметрів конкретного пристрою IoT, такі як обсяги пам'яті, обчислювальна потужність та ємність акумулятора. Їх врахування дозволяє визначити, краще використовувати МСШ чи МАШ. Наприклад, якщо ОР обмежені, важливо використовувати менш витратні способи шифрування.

В результаті, на основі конкретних характеристик пристрою, отримаємо оптимальне рішення щодо вибору способу шифрування, яке враховує специфіку кожного пристрою IoT та дозволяє забезпечити баланс рівня безпеки та обмеженнями ресурсів.

3. РОЗРОБКА АЛГОРИТМУ ЗАХИСТУ ІНТЕРНЕТ-РЕЧЕЙ НА ОСНОВІ БЛОКОВОГО ШИФРУВАННЯ

3.1 Структура алгоритму блокового алгоритму шифрування

Сучасні алгоритми захисту інформації, зокрема, шифрування, розроблені для використання в комп'ютерах у складі програмних комплексів, не враховуючи оптимізації на рівні апаратного забезпечення. Це ускладнює використання більшості існуючих КА у пристроях в яких обмежені обчислювальні можливості, обмеженим обсягом пам'яті та низьким споживанням енергії. Це можуть бути, системи моніторингу, промислові сенсорні чи вбудовані системи, які, також, називають малоресурсними.

Варто зазначити IoT, який є бездротовою самоконфігуруючою мережею між різними об'єктами різних класів, такими як побутові прилади, транспортні засоби, інтелектуальні датчики та мітки радіочастотної ідентифікації RFID [13]. Системи RFID саме й ставлять найвищі вимоги, оскільки в основному є системами з обмеженою доступною площею та живуть за рахунок електромагнітного поля.

Методами захисту даних у таких системах відповідно є методи МРК або криптографічні техніки та алгоритми, які спеціально розроблені для застосування в пристроях або системах із обмеженими характеристиками, які забезпечують високий рівень безпеки при мінімальному використанні ресурсів, наприклад АМРШ, оптимізовані версії відомих алгоритмів або енергоефективні методи шифрування, що враховують обмежені можливості пристроїв.

Стандартні підходи до розв'язання проблеми створення ефективних методів МРК включають:

- використання класичних КА, якщо це можливо.
- модифікація класичних КА з пристосуванням до апаратних особливостей та обмежень малоресурсних систем;
- розробка нових спеціалізованих рішень на методологічному, алгоритмічному та програмно-апаратному рівнях.

У кожного із цих підходів є свої недоліки. На даний час, як правило, більшість рішень у даній області відносяться до третього підходу та демонструють непогані результати. Проте, що при адаптації КА до особливостей апаратної бази в умовах обмежених ресурсів можуть виникати небажані наслідки, такі як додаткові слабкі місця алгоритму або зниження їх загальної стійкості.

У МРК в основному використовуються як блокові, так і потокові алгоритми [13-15, 25-30]. Класичними алгоритмами, спрямованими на апаратну реалізацію є алгоритм потокового шифрування MICKEY, симетричний алгоритм синхронного потокового шифрування Trivium, алгоритм потокового шифрування GRAIN, симетричні алгоритми блокового шифрування DESL і PRESENT. Також існують нові алгоритми KATAN і KTANTAN, а також MIBS і TWIS HIGHT або mCrypton, які поки що не досить добре досліджені. Тому, через індивідуальні особливості, вони не мають широкого застосування.

Кожен із зазаначених алгоритмів має свої обмеження. Наприклад, алгоритм Trivium потребує для своєї реалізації на кристалі площу, яка перевищує допустимі обмеження в 1,5 рази. Алгоритм GRAIN у малоресурсній версії допускає атаки на пов'язані ключі. MICKEY не є повністю стійким до всіх видів атак, також існують питання стосовно його надійності.

Серед блокових алгоритмів, DESL, розроблений на основі всесвітньовідомого алгоритму DES, є вдалим рішенням у МРК завдяки тому, що останній вже розроблявся з урахуванням апаратної реалізації. Авторами DESL доведено, що зміни, внесені в алгоритм під час його адаптації, не впливають на його стійкість до атак у межах диференційного та лінійного криптоаналізу. Єдиним серйозним недоліком є довжина ключа, яка становить 56 бітів та відкриває можливість для його розшифрування за допомогою потужної багатопроцесорної системи шляхом повного перебору протягом декількох днів.

Альтернативою DESL є шифр PRESENT, що відрізняється

компактністю та ефективністю. Розроблений групою науковців з Німеччини, Данії та Франції PRESENT був представлений на конференції CHES-2007 та включений до стандарту ISO/IEC 29192-2:2012 [31].

Автори алгоритму акцентували увагу на його спеціалізованому застосуванні в областях, де не підходять загальновживані алгоритми, наприклад AES. Апаратна реалізація PRESENT вважається одною з найбільш компактних серед існуючих [33]. Крім того, модифікації цього алгоритму знайшли своє використання в інших ресурсозалежних пристроях. Наприклад, H-PRESENT-128 є найбільш компактною з відомих хеш-функцій, а інші його модифікації застосовуються як генератори псевдовипадкових чисел для схеми crypto-GPS.

PRESENT є блоковим шифром і базується на SP-мережі з блоками інформації розміром 64 біти, ключами завдовжки 80 або 128 біт та складається з 31+1 циклу шифрування. На рисунку 3.1 наведена узагальнена схема алгоритму, описана псевдокодом.

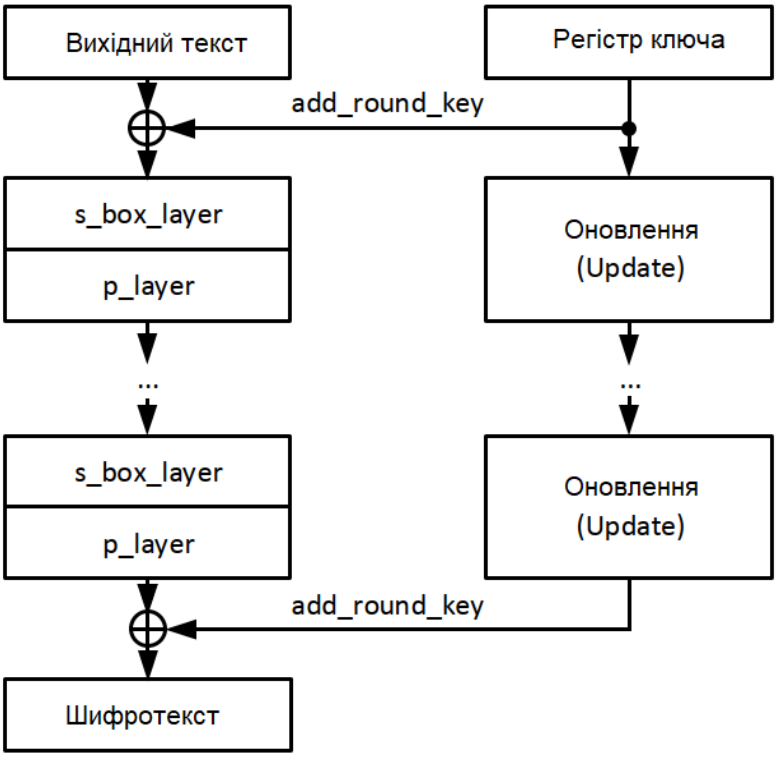

Рисунок 3.1 - Узагальнена схема PRESENT

Кожний цикл включає операцію XOR з цикловим ключем K_i. Ключ має довжину 64 біти і визначається функцією оновлення ключа (Update). Далі

відбувається розсіюючe перетворення, що полягає у проходженні блоку через 16 однакових *S*-блоків, розмір кожного 4 біти, складених так, щоб максимально підвищити стійкість алгоритму до лінійного і диференційного криптоаналізу. Після цього біти в блоці переставляються (міксуються).

Алгоритм не є великим за обчислювальною складністю. Для функції add_round_key() потрібно виконати операцію $b_j \to b_f \oplus k_{if}$ де j змінюється від 0 до 63, а i від 1 до 32. Блок s_box_layer виконуєрозсіювання вхідних даних шифрування по всьому обсязі алгоритму, що дозволяє забезпечувати додатковий рівень шифрування. Блок p_layer здійснює переміщення біта i на позицію $P_{(i)}$. Ключ задається користувачем у спеціальному регістрі ключа K. $K_i = K_{63}, K_{62}, K_{61}, \ldots, K_2, K_1, K_0$, отже, містить 64 біти лівої частини значущих бітів K. Таким чином, на i-му циклі, $K_i = K_{63}, K_{62}, K_{61}, \ldots, K_2, K_1, K_0 = K_{79}, K_{78}, K_{77}, \ldots, K_{18}, K_{17}, K_{16}$. Регістр ключа оновлюється в ході алгоритму наступним чином:

$$K_{79}, K_{78}, K_{77}, \ldots, K_2, K_7, K_0 = K_{18}, K_{17}, \ldots, K_{18}, K_{19}, \ldots, K_{20}, K_{19};$$

$$K_{79}, K_{78}, K_{77}, K_{76} = S[K_{79}, K_{78}, K_{77}, K_{76}];$$

$$K_{19}, K_{18}, K_{17}, K_{16}, K_{15} = K_{19}, K_{18}, K_{17}, K_{16}, K_{15} \oplus round\_counter.$$

Ключ зміщується на 61 позицію вліво, 4 біти лівої значущої частини проходять через *S*-блок, а над молодшими правими значеннями циклічного лічильника i та бітами $K_{19}, K_{18}, K_{17}, K_{16}, K_{15}$ регістра ключа K виконується операція додаваня по модулю 2 або виключаюче або..

Розглянемо приклад оновлення ключа в алгоритмі PRESENT:

- початковий ключ K - довжиною 80 біт ($K_0 - K_{79}$).
- ключ зсувається вліво на 61 позицію, тобто, $K_0 \to K_{61}$, $K_1 \to K_{62}$, і так далі;
- 4 біти лівої частини ($K_{79} - K_{76}$) проходять через *S*-блок (це нелинійне перетворення), отримуючи нові значення, замінюючи частину ключа $K_{79}, K_{78}, K_{77}, K_{76}$;
- оновлення лічильника циклів (round_counter): відбувається

додавання по модулю 2 певної кількості бітів із $K_{19}, K_{18}, K_{17}, K_{16}, K_{15}$ до round_counter.

Припустимо, що ми маємо ключ початково у вигляді:

```
K: K0 K1 K2 ... K75 K76 K77 K78 K79
```

Після кожного циклу оновлення ключа він може виглядати наступним чином:

```
Shift left 61 bits: K61 K62 ... K16 K17 K18 K19
S-box on K79-K76
XOR with round_counter on K19-K15
```

Ці операції виконуються для оновлення ключа на кожному циклі алгоритму шифрування PRESENT.

Фрагмент коду програмної реалізації алгоритму PRESENT, де генеруються ключі циклів та виконуються операції шифрування для кожного циклу, матиме вигляд:

```
generate_round_keys()

for i in range(1, 32):
    add_round_key(state, round_keys[i])
    s_box_layer(state)
    p_layer(state)

add_round_key(state, K32)
```

В цьому фрагменті спочатку генеруються ключі циклів за допомогою функції generate_round_keys(). Далі виконується цикл, де функції застосовуються до state для кожного з 31 циклу: add_round_key - додає ключ і до поточного стану шифрування; s_box_layer - застосовує заміну S-блоків; p_layer - виконує перемішування (перестановку) бітів.

На завершення застосовується останній ключ циклу K_{32} за допомогою функції add_round_key. Цей код представляє один цикл алгоритму шифрування PRESENT, який включає в себе основні етапи шифрування, такі як додавання ключа, заміну S-блоків та перемішування бітів у стані шифрування.

Загальна структура програмної реалізації, на мові програмування Python, яка може служити основою для реалізації алгоритму PRESENT. Наведена в додатку А. Деталі процесу генерації раундових ключів можуть залежати від конкретної реалізації алгоритму, включаючи розмір ключа, кількість циклів і внутрішні операції перетворення ключа. Тому наведений лише загальний опис кроків, оскільки саме визначення та реалізація цих кроків на практиці відрізняються алгоритм від алгоритму.

Даний алгоритм досить захищений від атак на основі пов'язаних ключів, слайд-атак та інших типових методів атак на криптосистеми. Рішення може бути відображене у вигляді схеми (рисунок 3.2), позначено: БПП та БГП – блоки перетворення та генерації повідомлення відповідно, БЗС – блок зберігання 64-бітного слова, БРК та БЗК – відповідно блоки розділення та зберігання ключа.

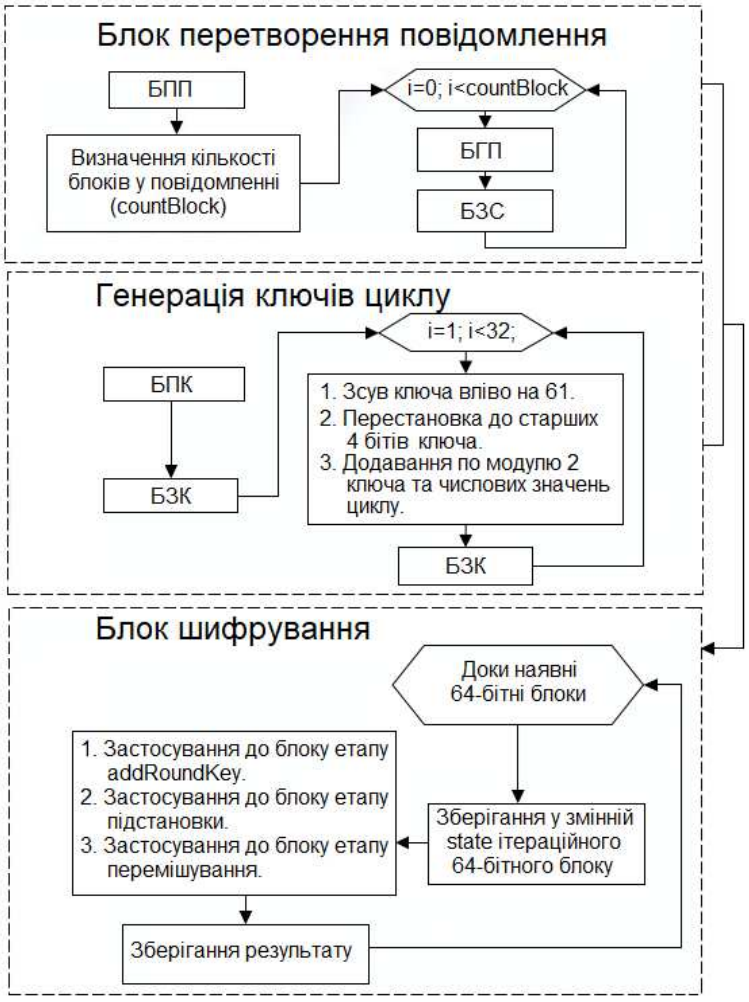

Рисунок 3.2 - Граф-схема алгоритму PRESENT для програмного рішення

3.2 Апаратна реалізація блокового алгоритму шифрування

Для оцінки ефективності алгоритму було створено ряд практичних рішень, що включають програмні коди та апаратну реалізацію на основі ПЛІС. Програмне рішення адаптоване для технології .NET Micro Framework та може застосовуватися на 32- та 64-розрядних мікроконтролерах з архітектурою ARM7, ARM9 та Blackfin.

Апаратне рішення алгоритму реалізоване на ПЛІС від компанії ALTERA Cyclone II EP2C20F484C7 з робочими частотами 27, 50 та 100 МГц у вигляді окремого обчислювального ядра. По суті, це є система на кристалі, яка відповідає класичним вимогам малоресурсної криптографії. Структура конфігураційного проекту для апаратної реалізації алгоритму PRESENT показана на рисунку 3.3.

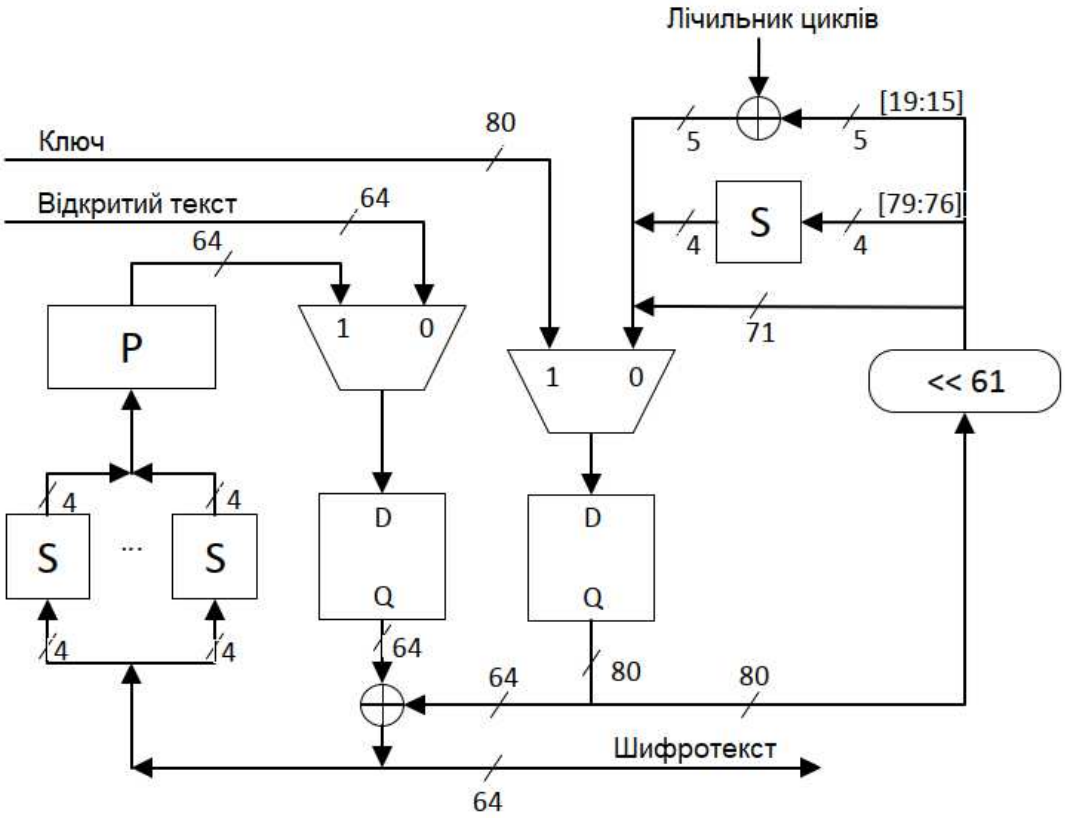

Рисунок 3.3 - Структура конфігураційного проекту системи на кристалі

Під час апаратної реалізації PRESENT, специфічні вимоги пред'являються до його ресурсозберігаючого шифрування: площа апаратного блоку на кристалілі максимум 3000 GE, де GE (gate equivalent) - це площа,

зайнята на кристалі одним логічним елементом типу 2I-HE.

Виходячи з цієї вимоги, важливо відзначити, що зберігання таблиць заміни для S-блоків у окремих регістрах не є оптимальним рішенням щодо апаратних витрат через те, що регістри будуються на основі D- або T-тригерів, які займають від 5GE до 12GE.

Існує можливість значного зменшення кількості використовуваних логічних елементів за рахунок реалізації блоків заміни як логічних функцій з кількох аргументів. S-блоки шифру PRESENT мають ідентичну структуру, тому системи рівнянь для таких блоків будуть мати однаковий вигляд.

$$S_0(x) = x_3x_2\overline{x_1}x_0 + \overline{x_3}x_2\overline{x_1x_0} + \overline{x_3x_2x_1}x_0 + x_3x_1\overline{x_0} + \\ + \overline{x_3}x_2x_1x_0 + \overline{x_3x_2}x_1x_0 + x_3\overline{x_2x_0};$$

$$S_1(x) = \overline{x_3}x_2x_1\overline{x_0} + x_3x_2x_0 + \overline{x_2}x_1\overline{x_0} + x_3\overline{x_2}x_1x_0 + \overline{x_3x_2}x_1x_0 + x_3\overline{x_2x_0};$$

$$S_2(x) = \overline{x_3x_2x_1x_0} + \overline{x_3x_2}x_1x_0 + x_3x_2\overline{x_1} + \overline{x_3}x_2x_1x_0 + \overline{x_2}x_1\overline{x_0} + x_3\overline{x_2}x_1x_0;$$

$$S_3(x) = \overline{x_3}x_2x_1\overline{x_0} + \overline{x_3x_2x_1x_0} + \overline{x_3}x_2\overline{x_1x_0} + x_3\overline{x_2}x_1 + \overline{x_3}x_2x_1x_0 + \\ + x_3\overline{x_2x_1}x_0 + \overline{x_3x_2}x_1x_0.$$

де x_3, x_2, x_1, x_0 - біти вхідного вектора,

S_3, S_2, S_1, S_0 - біти вихідного вектора.

Процес синтезу таких рівнянь з їх подальшою мінімізацією є досить трудомістким, і складність лише зростає зі збільшенням розрядності S-блоку. Крім того, ручне визначення блоків заміни є нераціональним стосовно швидкості розробки та інтеграції в процес проектування. Тому сучасні системи автоматизованого проектування зазвичай можуть виробляти синтез подібних логічних схем за описом, представленим у мовах опису апаратури, таких як Verilog і VHDL. Зокрема, для мови Verilog можна скористатися директивою function для синтезу комбінаційних схем.

В порівнянні цих двох підходів до синтезу блоків заміни в середовищі Altera Quartus II для ПЛІС Altera Cyclone II не виявлено суттєвих переваг ручної розробки перед повністю автоматичною. У обох випадках S-блоки мають апаратну структуру, яка зображена на рисунку 3.4.

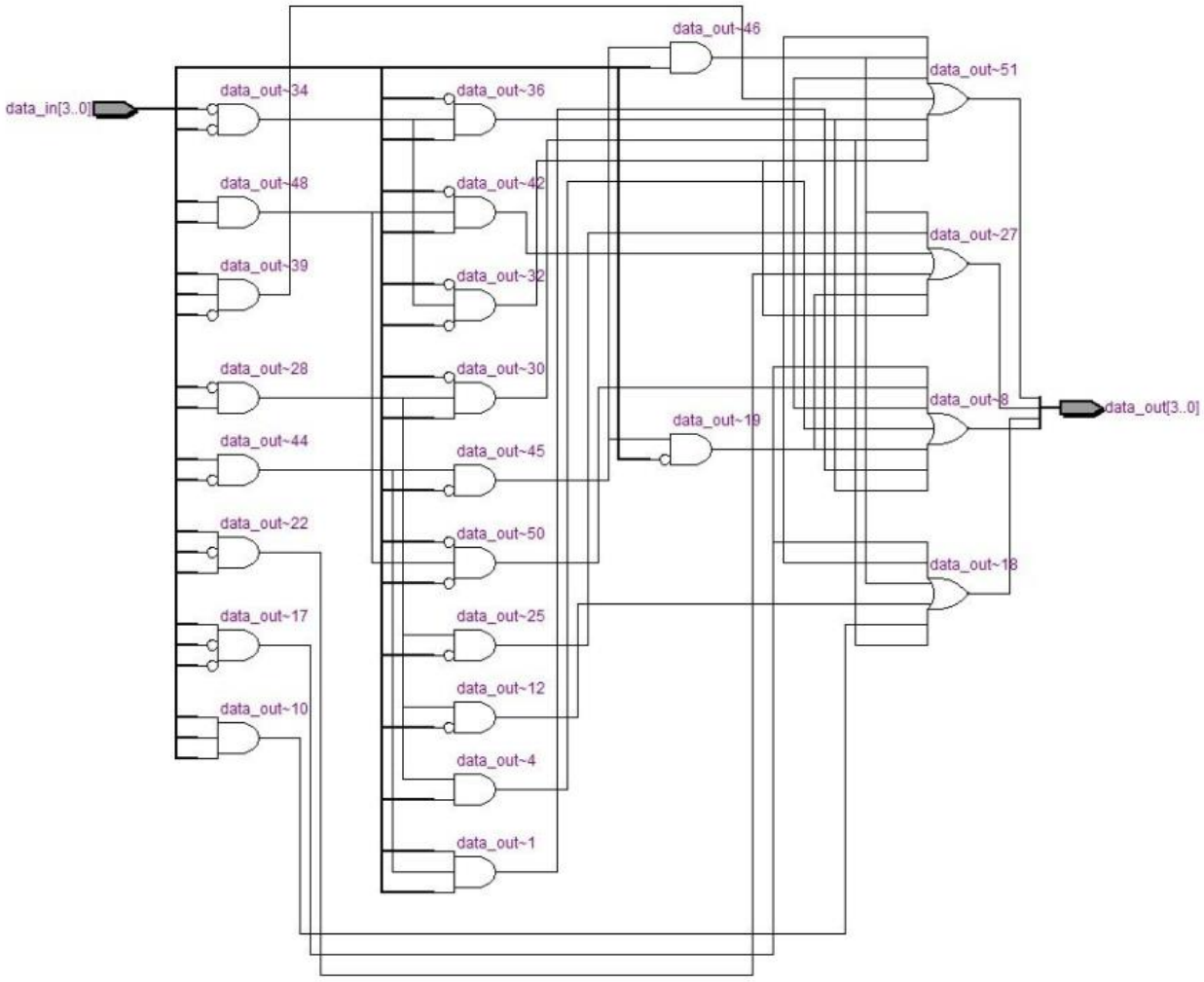

Рисунок 3.4 - Структура S-блоку

Проведений аналіз отриманого рішення показав, що максимальна доступна частота роботи схеми алгоритму для такої системи на кристалі визначається максимально допустимою частотою ПЛІС і становить 160 МГц, кількість задіяних логічних елементів складає 297, кількість задіяних блоків пам'яті - 0.

Ця схема алгоритму має пропускну здатність, яка при тактовій частоті 20 МГц становить 37,6 Мбіт/с, а при тактовій частоті 160 МГц - 301,2 Мбіт/с. Кількість тактів на блок дорівнює 34, а ефективність рішення - 126599 біт/сек на елемент. Щодо споживаної потужності: при 20 МГц вона коливається від 20 до 50 мВт, а при 160 МГц - від 47 до 114,77 мВт.

Отримано важливі результати: досліджено малоресурсний алгоритм PRESENT, розраховано його обчислювальну складність, створено програмне

рішення, достатньо ефективне для використання у вбудованих пристроях. Крім того, синтезовано апаратний блок для системи на кристалі, що задовольняє всі вимоги МРК. Проведено його налагодження, моделювання та необхідні експерименти, що підтвердили його працездатність, ефективність і можливість практичного застосування.

ВИСНОВКИ

В роботі запропоновоно рішення, яке дозволяє враховувати обмежені можливості пристроїв IoT, забезпечуючи при цьому високий рівень захисту. При цьому отримано наступні результати:

1. Проведений аналіз архітектури та принципів функціонування технології IoT, що дозволив визначити ключові елементи та взаємозв'язки в системі IoT.

2. Проаналізовано загрози безпеки IoT та механізмів їх реалізації з метою розуміння сценаріїв потенційних вразливостей для інфраструктури IoT.

3. Аналіз різних типів атак та можливості їх реалізації на різних рівнях IoT надав глибоке розуміння потенційних ризиків та їх можливі наслідки. Визначено, що основними проблемами в аспекті захисту IoT є стійкість та конфіденційність даних, інтеграція обмінюваної інформації, конфіденційність та автентичність користувача.

4. Дослідження основних аспектів захисту технології IoT дозволили запропонувати стратегії та методи захисту на кожному з рівнів з врахуванням його функцій та ролі в загальній системі IoT, що дозволить забезпечити високий рівень безпеки всієї інфраструктури.

5. Проведені дослідження криптографічних алгоритмів захисту даних та комунікаційних процесів в IoT, зокрема сучасних методів криптографії, дозволив визначити їх переваги, недоліки та обмеження щодо їх використання, оскільки жорсткі обмеження на внутрішні ОР пристроїв IoT утруднюють або роблять це неможливим.

6. Проведена оцінка ефективності застосування криптографічних методів захисту в умовах обмеженості ресурсів пристроїв IoT, показала, що вони повинні бути ефективними, маючи низький рівень обчислювального навантаження, а також забезпечувати високий рівень безпеки для захисту даних, що передаються. Встановлено, що оптимальним в середовищі IoT є використання малоресурсної криптографії, оскільки вона адаптується до

обмежень у ресурсах обчислювальної потужності, енергії та обсягу пам'яті.

7. В результаті проведених досліджень запропоновано алгоритм що дозволяє врахувати різноманітні ресурси пристроїв IoT. Він полягає у комбінуванні різних типів шифрування для створення більш оптимального та пристосованого до умов специфіки IoT. Однією із основних переваг запропонованого алгоритму є гнучкість у виборі оптимального способу шифрування, що відповідає конкретним потребам та обмеженням певного IoT пристрою.

8. В роботі запропоновано програмно-апаратну реалізацію на основі ПЛІС малоресурсного блокового алгоритму PRESENT, який є досить стійким та захищеним від атак на основі пов'язаних ключів, слайд-атак та інших типових методів атак на криптосистеми. Розраховано його обчислювальну складність. Синтезовано апаратний блок для системи на кристалі, що задовольняє всі вимоги МРК. Рішення адаптоване для технології .NET Micro Framework та може застосовуватися на 32- та 64-розрядних мікроконтролерах з архітектурою ARM7, ARM9 та Blackfin є достатньо ефективним для використання у вбудованих пристроях.

ПЕРЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жураковський Б.Ю., Зенів І.О. Технології інтернету речей. Навчальний посібник.- КПІ ім. Ігоря Сікорського. –Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

2. What is the internet of things?. [Електронний ресурс] - Режим доступу: https://www.ibm.com/topics/internet-of-things

3. Internet of things (IoT). [Електронний ресурс] - Режим доступу: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

4. Hagar Jon Duncan. IoT System Testing: An IoT Journey from Devices to Analytics and the Edge.- Apress Media LLC, 2022. - 323 p.

5. Інтернет речей. Новомодне захоплення чи технологія, що змінює світ?. [Електронний ресурс] - Режим доступу: https://sites.google.com/view/bezpecnyj-internet/можливості-інтернету/інтернет-речей-та-смарт-технології

6. Cheruvu Sunil, Kumar Anil, Smith Ned, Wheeler David M. Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment.-Springer / Apress, 2020. - 488 p.

7. Shandilya Shishir. Internet of Things Security: Fundamentals, Techniques and Applications.- River Publishers, 2018. - 164 p.

8. Gupta Brij B., Quamara Megha. Internet of Things Security: Principles, Applications, Attacks, and Countermeasures.-CRC Press, 2020. - 115 p.

9. Ziegler Sebastien. Internet of Things Security and Data Protection.- Springer, 2019. - 221 p.

10. Achary R. Cryptography And Networking Security: An Introduction.- Mercury Learning and Information, 2021. - 652 p.

11. Rubin F. Secret Key Cryptography: Ciphers, from Simple to Unbreakable.-Manning Publications, 2022. - 225 p.

12. Gupta B., Mamta. Secure Searchable Encryption and Data Management.- CRC Press, 2021. - 117 p.

13. Cole P.H., Ranasinghe D.C. (eds.) Networked RFID Systems and

Lightweight Cryptography. Raising Barriers to Product Counterfeiting.- Springer, 2008. - 349 p.

14. Bogdanov Andrey. Lightweight Cryptography for Security and Privacy.- Springer, 2017. - 155 p.

15. Ramakrishnan Srinivasan (ed.) Cybersecurity Lightweight Cryptographic Techniques.- ITexLi, 2022. - 129 p.

16. James A., Seth A., Mukhopadhyay S.C. IoT System Design: Project Based Approach.-Springer, 2022. - 291 p.

17. Patel Chintan, Doshi Nishant. Internet of Things Security: Challenges, Advances, and Analytics.- Auerbach Publications, 2018. - 261 p.

18. Безпека та Інтернет речей - пов'язані разом. [Електронний ресурс] - Режим доступу: https://worldvision.com.ua/articles/bezopasnost-i-internet-veshchey-svyazani-vmeste

19. Banoth R., Regar R. Classical and Modern Cryptography for Beginners.- Springer, 2023. - 230 p.

20. Спеціалізовані комп’ютерні технології в інформатиці / під загальною редакцією Я. М. Николайчука. - Тернопіль: ТзОВ «Терно-граф», 2017.- 913 с.

21. Остапов С.Е., Євсеєв С.П., Король О.Г. Кібербезпека: сучасні технології захисту. Навчальний посібник для студентів вищих навчальних закладів. – Львів: «Новий Світ-2000», 2020 . – 678 с.

22. Бобало Ю.Я., Горбатий І.В. Інформаційна безпека.-Навчальний посібник. - Львів : Видавництво Львівської політехніки, 2019. - 580 с.

23. Security of internet of things based on cryptographic algorithms: a survey - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Use-of-cryptography-algorithms-in-IoT_fig2_348537924

24. Easttom Chuck. Computer Security Fundamentals.-4th Edition. - Pearson Education Inc., 2020. - 512 p.

25. Mohammad Shah, I.N.; Ismail, E.S.; Samat, F.; Nek Abd Rahman, N. Modified Generalized Feistel Network Block Cipher for the Internet of Things.

Symmetry 2023, 15, 900. https://doi.org/10.3390/sym15040900

26. Bogdanov A., Knudsen L., Leander G., et al. PRESENT: An ultra-lightweight block cipher //CHES 2007. LNCS. 2007. V.4727. P.450–466.

27. Shirai T., Shibutani T., Akishita K., et al. The 128-bit blockcipher CLEFIA // FSE 2007. LNCS. 2007. V.4593. P.181–195.

28. Thierry P., Julien F., Marine M., and Gaël T. Extended generalized Feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput // IEEE Trans. Computers. 2015. V.65. Iss.7. P.99.

29. Banik S., Bogdanov A., Isobe T., et al. Midori: a block cipher for low energy // ASIACRYPT 2015. LNCS. 2015. V.9453. P.411–436.

30. Beierle C., Jean J., Kolbl S., et al. The SKINNY family of block ciphers and its low-latency variant MANTIS // CRYPTO 2016. LNCS. 2016. V.9815. P.123–153.

31. Міжнародний стандарт ISO/IEC 291922:2012 https://www.iso.org/ru/standard/56552.html

32. I. Yousif, S. Hussein, H. Hoomod, Q. Mohammed New Hybrid Lightweight Data Encryption Algorithm for Operation System Protocol in Internet of Thing Environment / International Journal Of Latest Technology In Engineering & Management (IJLTEM).- Volume 8.- Issue 5, 2023.-P. 13-21.

33. Parthasarathy R., Saravanan P., Efficient Hardware Implementation of PRESENT Lightweight Cipher,"2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 556-561.