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

ВОЛОС Ігор Петрович

Алгоритми захисту даних в мережах Інтернет – речей / Data Security Algorithms in Internet of Things Networks

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
І.П. Волос

Науковий керівник
д.т.н., професор В.В.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

ВОЛОС Ігор Петрович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Алгоритми захисту даних в мережах Інтернет-речей / Data Security Algorithms in Internet of Things Networks
керівник роботи д.т.н., професор В.В. Яцків

затверджені наказом по університету від 1 грудня 2022 року №

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

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

4. Основні питання, які потрібно розробити:
– провести аналіз стану конфіденційності та безпеки даних Інтернету речей;
– розробити класифікацію атак на різних рівнях мереж Інтернет речей;
– дослідити можливість використання фізичної неклонованої функції при захисті IoT;
– розробити процедуру реєстрації пристроїв на основі фізичної неклонованої функції;
– розробити схема автентифікації пристрою Інтернет речей на основі фізичної неклонованої функції.

5. Перелік графічного матеріалу у роботі:
– загрози безпеці Інтернету речей;
– апаратні механізми безпеки;

– використання фізично неклонованої функції в інфраструктурі відкритих ключів;

– модель загрози та рішення для пристроїв IoT;

– етап реєстрації з використанням фізично неклонованої функції;

– процедура автентифікації реєстрації з використанням фізично неклонованої функції.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз архітектури та безпеки Інтернет - речей	12.2022 р. – 03.2023 р.	
2	Атаки на Інтернет- речей та механізми захисту	03.2023 р. – 05.2023 р.	
3	Реалізація та дослідження алгоритмів автентифікації пристроїв Інтернет речей	05.2023 р. – 11.2023 р.	

Студент _____ Волос І.П.
(підпис)

Керівник роботи _____ д.т.н., професор В.В. Яцків
(підпис)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Алгоритми захисту даних в мережах Інтернет-речей» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 82 сторінки і містить 20 ілюстрації, 3 таблиця, 1 додаток та 32 джерел за переліком посилань.

Метою кваліфікаційної роботи є підвищення ефективності алгоритмів захисту даних в мережах Інтернет-речей.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи автентифікації, фізично неклоновані функції, методи проектування.

Результати дослідження. Удосконалено алгоритми автентифікації пристроїв Інтернет-речей на основі фізичної неклонованої функції.

Розроблено модуль для безпечної автентифікації пристроїв на основі фізично неклонованої функції.

Результати роботи можуть бути застосовані при розгортанні мережі Інтернет-речей із захистом від атак на алгоритми автентифікації.

Ключові слова: ІНТЕРНЕТ-РЕЧЕЙ, АВТЕНТИФІКАЦІЇ, ФІЗИЧНА НЕКЛОНОВАНА ФУНКЦІЯ, КОНФІДЕНЦІЙНІСТЬ, АТАКА.

ABSTRACT

Qualification work on "Data Security Algorithms in Internet of Things Networks" for the degree of "Master" in the specialty 125 "Cybersecurity" educational and professional program "Cybersecurity" is written in 82 pages and contains 20 illustrations, 3 tables, 2 appendices and 32 source according to the list of links.

The purpose of the qualification work is to improve the efficiency of data protection algorithms in Internet of Things networks.

Research methods. To solve the tasks in this qualification work, the following methods were used: authentication methods, physically non-cloned functions, design methods.

Research results. Improved authentication algorithms for Internet of Things devices based on physical non-clone function.

A module has been developed for secure authentication of devices based on a physically non-cloned function.

The results of the work can be applied in the deployment of the Internet of Things network with protection against attacks on authentication algorithms.

Keywords: INTERNET OF THINGS, AUTHENTICATIONS, PHYSICAL UNCLONED FUNCTIONS, CONFIDENTIALITY, ATTACK.

ЗМІСТ

ВСТУП

Актуальність роботи. У цифровому світі, який розвивається з великою швидкістю, Інтернет-речей відіграє важливу роль у нашому повсякденному житті, розширюючи можливості інтеграції фізичного та віртуального простору. У наш час Інтернет-речей (Internet of Things, IoT) втілює динамічну глобальну мережеву інфраструктуру, яка пропонує високий рівень доступності, цілісності та взаємодії між різними інтелектуальними пристроями [1].

Хоча прогрес в IoT дуже динамічний, вони також становлять серйозну проблему для безпеки та конфіденційності, враховуючи, що кількість «розумних» пристроїв і комунікацій між ними зростає. Підключення пристроїв до Інтернету створює численні ризики для безпеки, такі як прослуховування бездротового каналу зв'язку, несанкціонований доступ до пристроїв або втручання в роботу пристроїв [2].

Зокрема, кіберзагрози продовжують розвиватися та націлені на пристрої та комунікації Інтернету речей, які були активовані через слабку безпеку мережі та застарілі пристрої. Відповідно, щоб забезпечити надійні послуги, мережа не повинна допускати несанкціонований доступ шляхом перевірки надійних методів зв'язку для надсилання та отримання автентичної інформації та безпечного виконання операцій, передачі та обробки даних в режимі реального часу. Оскільки обсяг даних і їх складність зростають задача захисту IoT залишається надзвичайно актуальною.

Мета і завдання дослідження. Метою роботи є підвищення ефективності алгоритмів захисту даних в мережах Інтернет-речей.

Досягнення визначеної мети передбачає вирішення таких завдань:

– провести аналіз стану конфіденційності та безпеки даних IoT;

– розробити класифікацію атак на різних рівнях мереж Інтернет речей

– дослідити можливість використання фізичної неклонованої функції при захисті IoT;

– розробити процедуру реєстрації пристроїв на основі фізично неклонованої функції;

– розробити схема автентифікації пристрою Інтернет речей на основі фізичної неклонованої функції.

Об’єкт дослідження – процеси захисту даних в мережах Інтернет-речей.

Предмет дослідження – алгоритми та процедури реєстрації та безпечної автентифікації пристроїв Інтернет- речей.

Методи досліджень. Для розв’язання поставлених задач у даній кваліфікаційній роботі використано: методи автентифікації, фізичні неклоновані функції, методи проектування.

Наукова новизна одержаних результатів. Удосконалено алгоритми автентифікації пристроїв Інтернет-речей на основі фізичної неклонованої функції.

Практичне значення отриманих результатів. Розроблено вбудований модуль для безпечної автентифікації пристроїв.

Публікації та апробація КР.

1. Голод Ю.В., Гарматюк В.Р., Волос І.П. Мережеві атаки на інтернет-речей. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 39-41.

2. Дзівак О.А., Мачуляк М.В., Волос І.П. Фізичні атаки на мережі інтернет-речей. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С. 100-102.

1 АНАЛІЗ АРХІТЕКТУРИ ТА БЕЗПЕКИ ІНТЕРНЕТ - РЕЧЕЙ

1.1 Інтернет речей та Інтернет всього

Пристрої Інтернет речей (Internet of Things, IoT), завдяки технологічній революції, бездротовим пристроям та системам зв'язку, мають значну присутність у різних сферах діяльності людини. Інтернет речей став важливою частиною Індустрії 4.0. Здатність перенести фізичні речі в цифровий світ стає більш імовірною завдяки технологіям [1]. Мережі IoT впливають на різноманітні сфери, включаючи домашнє спостереження, повсякденне спостереження здоров'ям, моніторинг промислових об'єктів тощо. IoT поєднує в собі переваги обробки даних, аналітики та використовує можливості Інтернету для прийняття рішень щодо об'єктів реального світу. Це система, в якій інтелектуальні об'єкти пов'язані між собою та мають доступ до Інтернету, як основи взаємозв'язку для збору і обміну інформацією за допомогою «речей». Інтернет речей став одним із важливих напрямків досліджень у всьому світі.

IoT прагне зв'язати обладнання з Інтернетом, щоб зробити його доступним у будь-який час, будь-де і будь-кому. За допомогою безперебійного підключення та інтелектуальних об'єктів, таких як пральні машини, мікрохвильові печі, лічильники, транспортні засоби, мобільні телефони, холодильники, медичні пристрої тощо, IoT створює чудові програми, такі як інтелектуальні транспортні системи, розумна охорона здоров'я, розумні будинки, розумні міста і т.д. Компанія Ericsson прогнозує, що до 2024 року буде приблизно 30 мільярдів підключених пристроїв, серед яких 20 мільярдів будуть пристрої IoT. Враховуючи широке використання пристроїв IoT, їх застосування і вимоги до безпеки зросли. Крім того, багато пристроїв будуть розміщені в місцевості, які знаходяться без постійного нагляду. Зловмисник може скомпрометувати пристрої та знайти точку входу, щоб скомпрометувати мережу. Дослідницький інтерес до областей IoT значно зріс, і це стало одним із першочергових пріоритетів серед

промисловості та академічних кіл, про що велика кількість наукових публікацій.

Термін «Інтернет речей» також називають «Інтернетом об’єктів». Пристрої IoT мають різні розміри та можливості електронних пристроїв, здатних підключатися до Інтернету. Пристрої IoT можна використовувати в різноманітних сферах, зокрема: виробництво, навколишнє середовище, охорону здоров’я, проживання, електроенергію та зв’язок (рисунку 1.1).

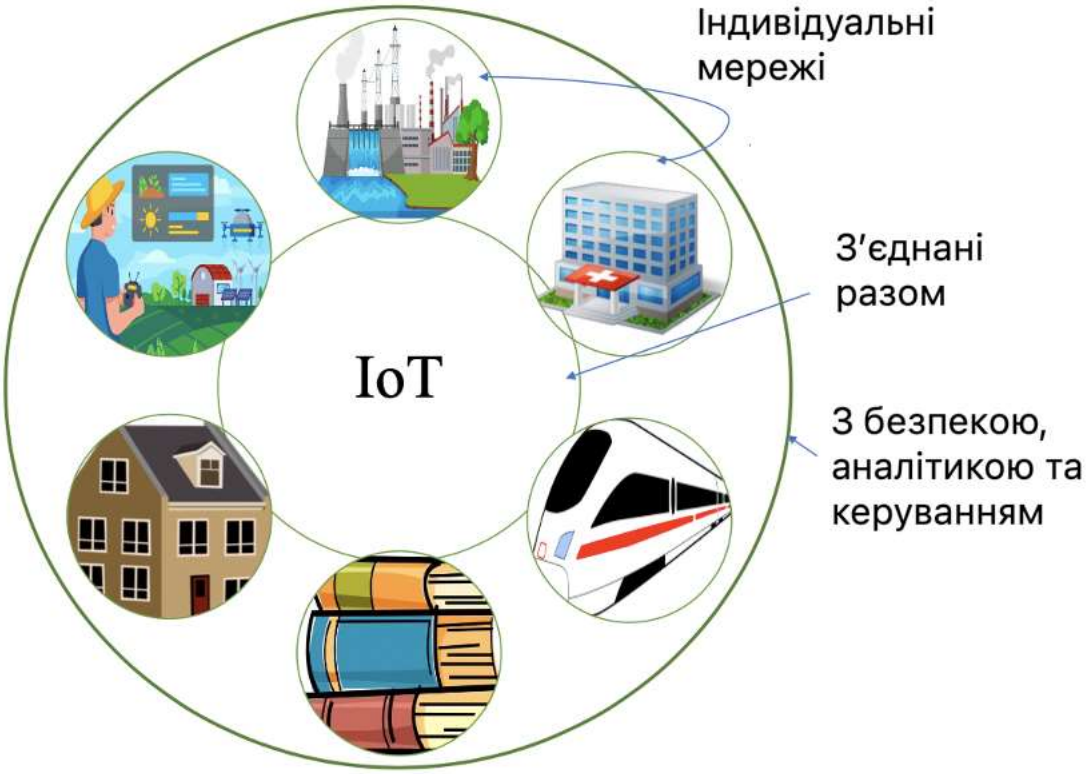


Рисунок 1.1 – Застосування Інтернету речей

На рисунку 1.2 показано елементи та парадигму IoT [1, 2]. Метою Інтернету речей стати розумним завдяки поєднанню як віртуальних, так і фізичних об’єктів. Ці пристрої бачать, чують, думають, обмінюються інформацією та виконують завдання. Інтернет-орієнтоване бачення зосереджується на розвитку мереж на основі протоколу IP, щоб об’єкти могли з’єднуватися та спілкуватися один з одним. У системах IoT попит на потоки великих обсягів даних від датчиків або розумних об’єктів і до них породжує семантико-орієнтоване бачення. У системі IoT, що базується на

сервіс-орієнтованому баченні, зосереджені інтелектуальні послуги та додатки IoT на основі трьох згаданих вище напрямків [3].

Для забезпечення функціональності IoT необхідні шість основних елементів, як показано на рисунку 1.2. Серед шести елементів ідентифікація має важливе значення для визначення назви та відповідності послуг відповідно до їх попиту. Пристрої Інтернету речей збирають дані шляхом вимірювання та надсилають дані в хмару/базу даних для аналізу. Елемент зв’язку використовується для одночасного зв’язування різнорідних об’єктів для обслуговування певних цифрових послуг. Прикладами протоколів зв'язку для IoT є: WiFi, Bluetooth, Zigbee, MQTT, IEEE 802.15.4, OPC-UA, NFC, Z-wave, LoRaWAN та LTE-Advanced. Для обробки використовуються такі апаратні елементи, як мікроконтролери, мікропроцесори, система на кристалі (SoC) і програмована логічні інтегральні (FPGA). Поєднання апаратних і програмних елементів є основою IoT. Метою IoT є надання послуг в будь-який час, будь-де та будь-кому.

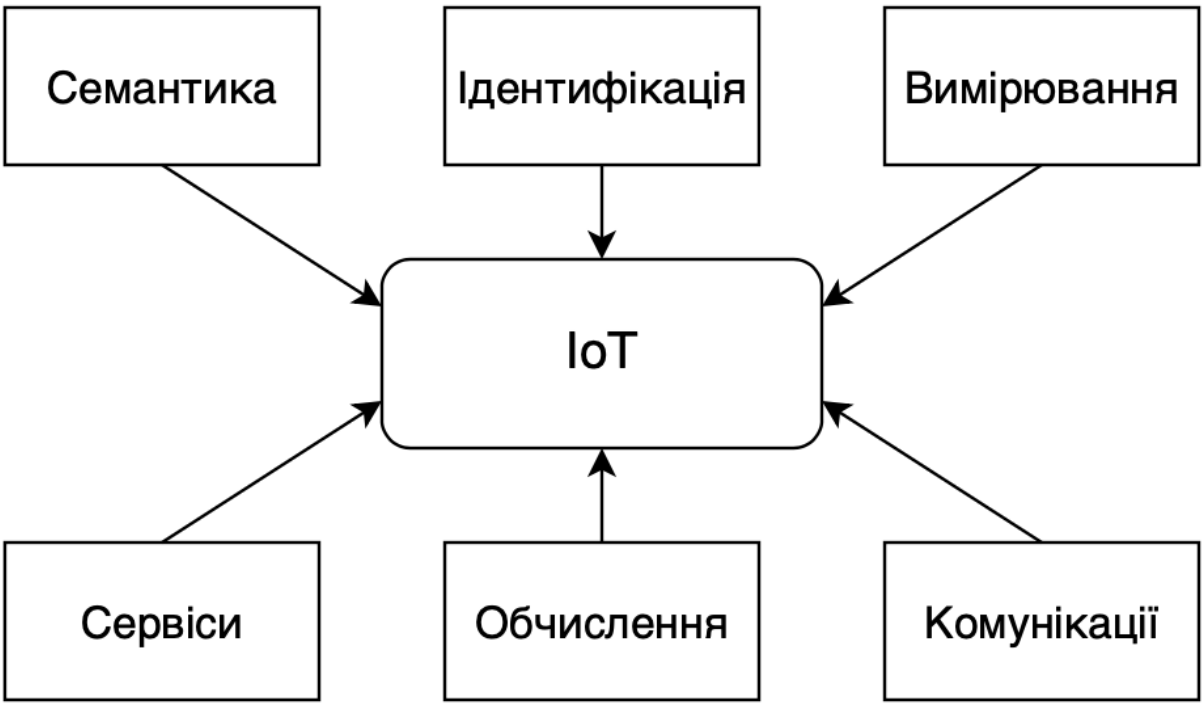


Рисунок 1.2 – Елементи та парадигма Інтернету речей

Загальна архітектура системи IoT має чотири рівні: сприйняття, мережа, обробка та прикладний рівень, які показані на рисунку 1.3. Пристрої на рівні сприйняття, такі як датчики різних типів, сканери радіочастотної ідентифікації (RFID), камери спостереження, модулі глобальної системи позиціонування (GPS), конвеєрні системи, промислові роботи, відповідають за умови моніторингу, збір даних тощо. Мережевий рівень відповідає за передачу даних до системи обробки наступного рівня та складаються з різних систем зв'язку, таких як WiFi, Bluetooth, Zigbee, LTE та протоколів IPv4 та IPv6. Хмарні сервери та бази даних відповідають за аналіз даних, обчислення, прийняття рішень і зберігання великої кількості даних. Прикладний рівень забезпечує потреби кінцевих користувачів.

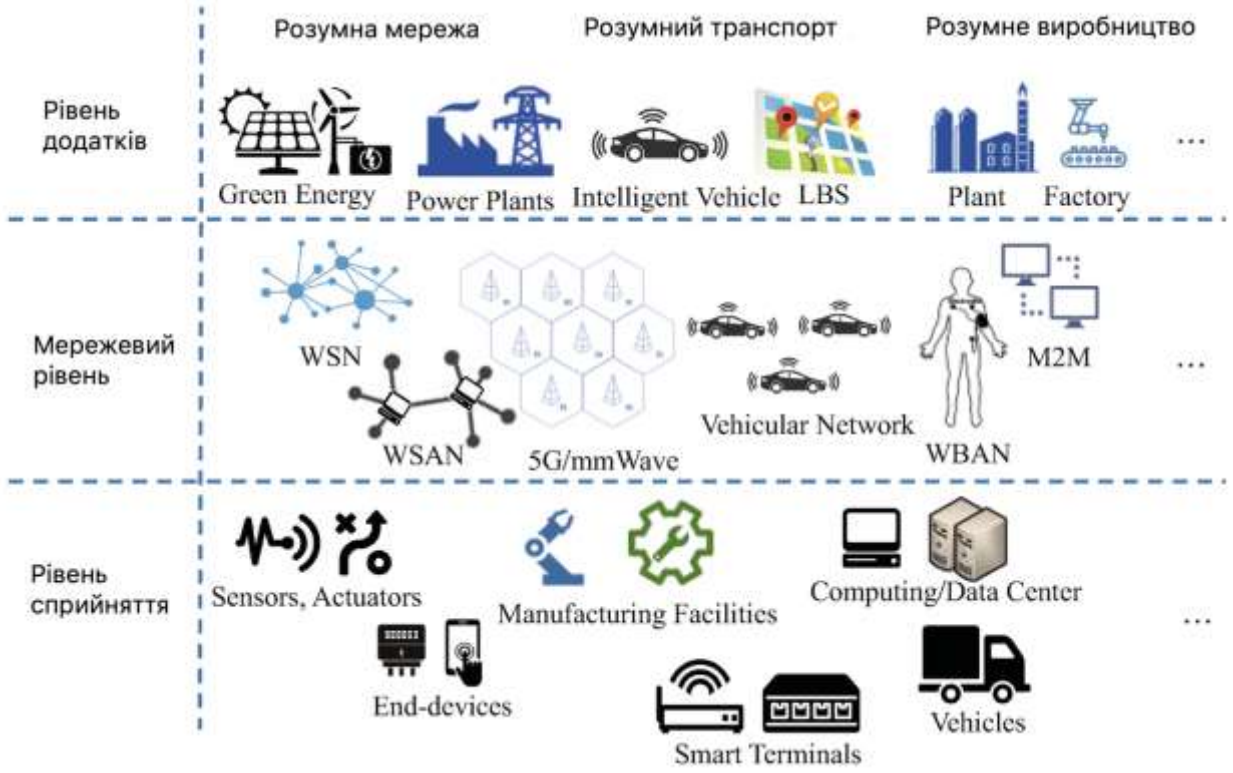


Рисунок 1.3 – Загальна архітектура IoT

IoT є основою різноманітних програм, таких як обробна промисловість, інтелектуальна охорона здоров'я, інтелектуальний транспорт, розумна мережа, розумне місто тощо. «Інструменти», «Взаємозв'язки» та «Інтелект» є важливими елементами розумних програм для IoT. З іншого боку, IoT діє як

інтегрований компонент Інтернету всього (IoE). Концепцію IoE спочатку запропонувала Cisco у 2013 році. Основною метою технології IoE є перетворення зібраних даних в інформацію або прийняття відповідних рішень на основі даних. Метою IoE також є створенню нових можливостей, навичок і досвіду, щоб стати автономною системою. На рисунку 1.4 представлені основні «чотири стовпи» IoE: люди, дані, процеси та речі [2].

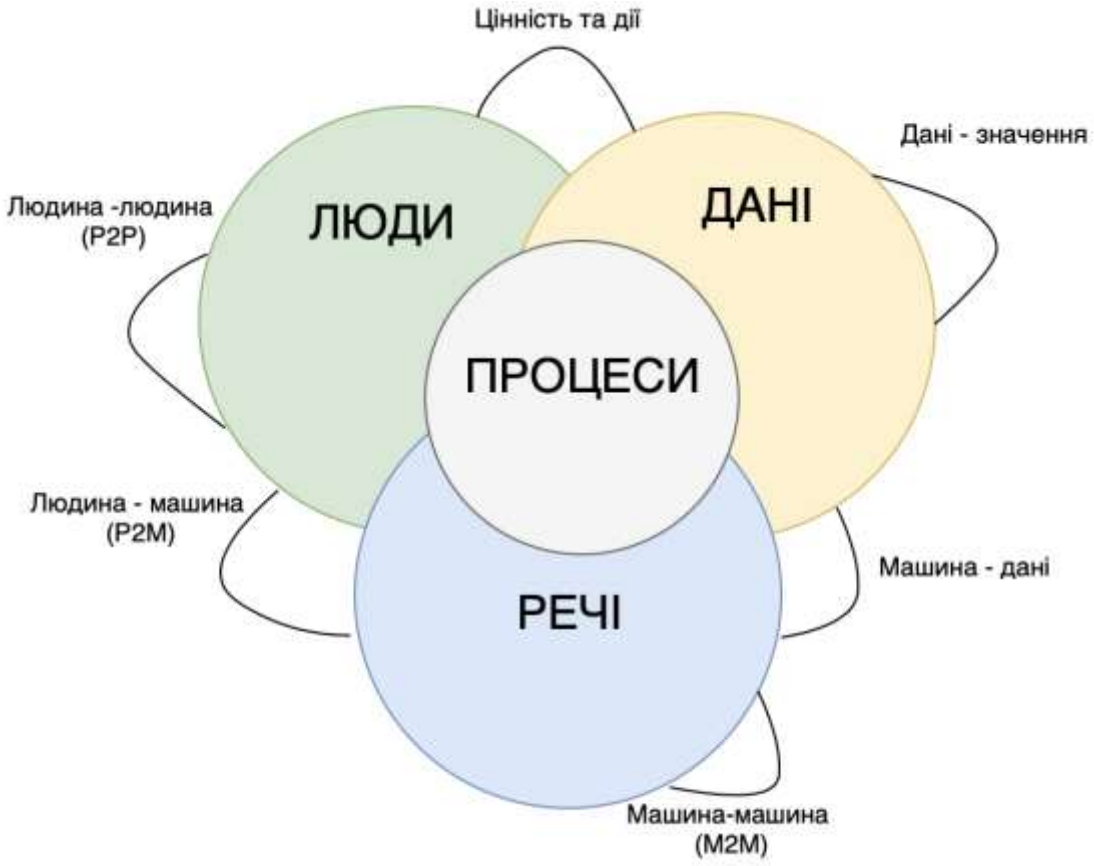


Рисунок 1.4 – Компоненти Інтернету всього

1. Люди в системі є критично важливим елементом середовища IoE. З появою IoE люди діляться своїми особистими думками за допомогою незліченних нових способів спілкування, таких як соціальні мережі, розумні датчики, які збирають дані, розумні годинники тощо. Ці дані передаються на сервери для аналізу та надання відповідної інформації відповідно до їхніх особистих, системних чи галузевих бізнес-вимог. Інформація допомагає

людям або системі швидко вирішувати відкриті питання або приймати рішення.

2. Дані передаються як в традиційній мережі IoT. Дані зібрані пристроями, можуть надсилатися безпосередньо або після початкового перетворення на крайовому рівні. Необроблені дані, отримані або згенеровані пристроєм, не мають значення. Проте, коли ці фрагменти даних перетворюються, підсумовуються, класифікуються та аналізуються самим пристроєм або хмарним сервером на периферійному рівні, вони стають корисним вмістом, який може відстежувати та контролювати численні системи, приймати точні та швидші рішення та надавати корисні рішення.

3. Процес, заснований на різних системах, таких як штучний інтелект, глибоке навчання, соціальні мережі, комп'ютерний зір або інші технології, допомагає доставити належну інформацію призначеним людям та/або пристроям в певний час. За допомогою цього процесу інформація буде отримана з даних, а передача даних буде контролюватися через мережу. Метою процесів є отримання оптимального результату для подальшої обробки та прийняття рішень.

4. Речі стикаються з визначенням IoT. Різні типи чутливих елементів вбудовані в фізичні елементи, які служать для збору даних. Різні пристрої повинні мати можливості зв'язку, бездротові чи дротові, для передачі згенерованих і оброблених даних до потрібного пункту призначення через систему.

Система IoE – це поєднання блоків центру обробки даних, інтелектуальної мережі та підключених пристроїв.

Віртуальний центр обробки даних складається з операційної системи, програмного забезпечення віртуалізації робочого столу тощо. Він зв'язується з інтелектуальною мережею для надання послуг підключеним пристроям (розумним датчикам, пристроям, приводам, мобільним терміналам, носимим пристроям тощо) і, зрештою, людям. Підключені пристрої поділяються на три категорії: людина-людина, машина-машина та машина-людина. Основою

є волоконно-оптична або бездротова мережа, яка гарантує високу швидкість мереж, щоб забезпечити високу присутність, низьку затримку та відмінну якість послуг IoE.

1.2 Область застосування Інтернет-речей

Програми IoT можна використовувати різними способами, щоб допомогти системам і підприємствам спростити, покращити, автоматизувати та контролювати процеси. IoT також можна використовувати для доставки важливих даних, ефективності діяльності або факторів навколишнього середовища, які потрібно постійно та дистанційно контролювати. Тому додатки IoT можуть допомогти у створенні нових систем і бізнес-стратегій, а також надати підприємствам миттєві дані, необхідні для створення продуктів і послуг.

1.2.1 Розумне місто

Розумне місто – це технологічно розвинений регіон, який збирає інформацію за допомогою різних електронних методів, технологій розпізнавання голосу та датчиків. Дані використовуються для успішної обробки активів, послуг і програм; у свою чергу, інформація використовується для безперебійної роботи в усьому місті. Дані, отримані від членів громади, обладнання, споруди та активи, обробляються та аналізуються для відстеження та підтримки транспортної інфраструктури, енергетичних установок, комунальних послуг, підключення до системи водопостачання, управління відходами, запобігання злочинності, управління даними, освітні установи, бібліотеки, заклади охорони здоров'я та інші громадські програми. Розумне місто – це набір різних датчиків і обладнання для моніторингу, звітування та обробки для ефективного управління ресурсами інфраструктури. Використовуючи інформацію, зібрану від

бездротових датчиків, система навчатиметься та прийматиме рішення, щоб забезпечити людям корисні результати. Порівняно з охороною здоров'я, водопостачанням і наглядом за навколишнім середовищем у нинішніх міських районах, розумне місто зможе краще зв'язати громадян і необхідні послуги. Необхідно захистити як конфіденційність мешканців, так і цілісність інформаційної системи. Конфіденційна інформація збирається датчиками, що робить її вразливою для кібератак.

1.2.2 Інтернет медичних речей (IoMT)

Інтеграція функцій здоров'я в пристрої IoT створило середовище IoMT. З розвитком технологій збільшується використання пристроїв IoMT. Крім того, ситуація з COVID-19 обмежує особисті зустрічі пацієнтів і лікарів. Пандемія створила нову еру IoMT для надання лікування пацієнтам. IoMT створює мережу людей і медичних пристроїв (бездротових медичних пристроїв та імплантованих медичних пристроїв). Він використовує бездротовий зв'язок для обміну даними про здоров'я з медичними установами, такими як лікарі, лікарні, медичні експерти тощо. З розвитком мікроелектроніки медичні пристрої стали інтелектуальними та можуть контролювати та повідомляти про фізичні стани, такі як артеріальний тиск, серцебиття, рівень кисню тощо. Пристрої можна розмістити в тілі у вигляді годинників, ременів, взуття, одягу, намиста тощо. Крім того, IoMT став найбільш значущою розробкою у медичному секторі, оскільки він залучає не лише людей похилого віку, але й усіх хворих літніх людей до постійного моніторингу та лікування. Багато систем охорони здоров'я в усьому світі застосовують систему IoMT для надання лікування. Проте, згідно з дослідженням CyberMDX 2020 року, майже 50% обладнання IoMT піддається атакам. Мережа IoMT відрізняється від інших систем тим, що вони потенційно можуть вплинути на життя пацієнтів і створити проблеми з конфіденційністю, якщо їхні особи будуть розголошені. Підтримка безпеки та конфіденційності є головною задачею системи IoMT. Згідно з

дослідженням компанії Critical Insights, що займається кібербезпекою, у 2021 році кількість інцидентів у сфері кібербезпеки досягла найвищого рівня за весь час, ставши під загрозу рекордну кількість особистих даних про здоров'я пацієнтів. У 2021 році атаки на систему охорони здоров'я завдали шкоди 45 мільйонам людей, порівняно з 34 мільйонами у 2020 році. Згідно з дослідженням, лише за три роки кількість зламаних даних зросла втричі з 14 мільйонів у 2018 році.

1.2.3 Smart Grid

Розумна мережа – це електрична система, яка містить кілька ефективних та енергоефективних функцій, таких як інфраструктура для інтелектуального вимірювання, інтелектуальні панелі живлення, розумне обладнання, система керування, альтернативна/відновлювана енергія тощо. Термін «розумна мережа» стосується концепції, яка об'єднує всю систему виробництва та розподілу електроенергії в одному кадрі. Це система електроенергії, побудована на основі цифрових технологій, яка використовує двосторонній цифровий зв'язок для постачання електроенергії споживачам. Іншими словами, Smart Grid – це мережа, яка робить всю систему розумнішою або чистішою. Чиста енергія зараз користується великим попитом у всьому світі. У 2003 році вперше термін «Smart Grid» був викладений Майклом Т. Берром. Технологія Smart Grid дозволяє здійснювати моніторинг, координацію та контроль електричної мережі в реальному часі через комунікаційні мережі між фізичними компонентами, що забезпечує більш ефективне та економічне управління мережею. Широка доступність підключення до Інтернету в більшості будинків зробила інтелектуальну мережу більш життєздатною для впровадження. Інтелектуальна мережа складається з диспетчерського контролю та збору даних (SCADA), системи енергоменеджменту, мережевих систем зв'язку та розподілених енергетичних ресурсів (DER). У системі розумної електромережі конфіденційність і безпека даних користувачів є ключовими та складними

проблемами. Кіберфізичний експлойт – це збій безпеки в кіберпросторі, який негативно впливає на фізичне середовище CPS. За останні роки в цьому секторі було зареєстровано низку серйозних кіберфізичних інцидентів. Комп'ютерний черв'як під назвою «Stuxnet» використовував чотири недоліки нульового дня та криптографічно підписані сертифікати, щоб уникнути виявлення вторгнення. Він влучив в іранський комплекс зі збагачення ядерного палива в червні 2010 року, де цілями були програмовані логічні контролери (PLC) системи SCADA [4]. У грудні 2015 року три українські електророзподільні компанії були зламані в ході злагодженої операції. Тридцять підстанцій були знеструмлені майже на три години, в результаті чого 225 000 споживачів зазнали глобальних знеструмлень. Щоб зірвати заяви про збої, була здійснена телефонна спроба DoS, у той час як облікові записи авторизованих учасників віртуальної приватної мережі були викрадені за допомогою вірусу Black-Energy3. В даний час використовуються різні методи на основі штучного інтелекту для виявлення та захисту механізму системи безпеки інтелектуальної мережі.

1.2.4 Інтернет транспортних засобів (IoV)

З розвитком промисловості кількість транспортних засобів стрімко збільшується. Збільшення транспортних засобів викликає занепокоєння щодо безпеки, що запускає безпечний зв'язок. IoV очолить індустрію 4.0. Безперечно, IoV буде яскравим і прибутковим у майбутньому, пропонуючи покращену безпеку на дорозі, зменшений вплив на навколишнє середовище, краще використання простору та контроль витрат. IoV, який часто називають «розумним транспортом» або «підключеними автомобілями», являє собою структуру, що складається з транспортних засобів, смартфонів і носіїв, придорожнього обладнання та мережі. Люди, автомобілі та численні пристрої IoT, які є частиною транспортної системи, спілкуються через IoV. Постраждали транспорт, виробництво, енергетика, програмне забезпечення та інші галузі. Екосистема IoV включає апаратне забезпечення, програмне

забезпечення, послуги та численні мережеві технології, починаючи від Bluetooth і стільникового зв'язку до Wi-Fi і 5G, а також кілька типів зв'язку (V2V, V2X і так далі). Системи зв'язку «транспортний засіб – транспортний засіб» і «транспортний засіб – інфраструктура» об'єднуються для створення автомобільної спеціальної мережі або VANET, а термін IoV розвинувся з позначення VANET. Комбінація функціональних можливостей, таких як датчики, платформи керування та різні комп'ютерні ресурси, робить кожен транспортний засіб в IoV інтелектуальним об'єктом. Кожен транспортний засіб з'єднується з будь-яким об'єктом через комунікаційну архітектуру V2X. Метою IoV, також відомого як V2X, є безпечне водіння шляхом зменшення кількості аварій, зменшення заторів, надання інформації про маршрути з низьким трафіком та надання інших інформаційних послуг. Кожен транспортний засіб у мережі IoV взаємодіє з усіма іншими речами, які можуть на нього впливати. V2X в основному включає зв'язок «транспортний засіб – транспортний засіб» (V2V), «транспортний засіб – датчики» (V2S), «транспортний засіб – інфраструктура» (V2I), «транспортний засіб – мережа» (V2N) і «транспортний засіб – пішохід» (V2P). Тим не менш, дороги можуть бути захоплені шляхом модифікації або зміни даних або прийняття неправильних рішень через отримання жартівливих даних. Щоб уникнути подібних ситуацій, необхідно розробити надійну структуру автентифікації, яка може протистояти вразливостям системи безпеки та проводити перевірку за мілісекунди.

1.3 Конфіденційність і безпека даних Інтернету речей

Захищений зв'язок є головним фактором збереження конфіденційності даних у різних типах архітектури IoT та IoE. Положення, пов'язані зі збором даних, зберіганням у пам'яті та обміном, мають дотримуватися таким чином, щоб забезпечити конфіденційність особистих даних користувача.

Безпеку можна покращити за допомогою безпечного керування ключами і функції фізичного неклонування. В таблиці 1.1 показано чотири домени, які визначають різні концепції безпеки мережі на основі IoT [5].

Таблиця 1.1 – Класифікація безпеки IoT

Таксономія безпеки в Інтернеті речей			
Дані	Зв’язок	Архітектура	Застосування
Конфіденційність Довіра	Обмін інформацією Спільне використання інформації	Аутентифікація Авторизація	Аутентифікація Авторизація Обмін ресурсами Створення довіри

Оскільки Інтернет речей стає невід’ємною частиною нашого повсякденного життя, використання пристроїв на основі Інтернету речей стрімко зростає. Прогнозується, що через постійний розвиток урбанізації 70% пристроїв будуть пристроями на основі Інтернету речей. CISCO прогнозує, що до 2025 року буде використано пристроїв на 14,4 трильйона доларів. Трафік M2M зростає, і очікується, що до 2022 року він становитиме до 45% від усього Інтернет-трафіку. Інше дослідження показує, що до 2025 року світова економіка додатків для охорони здоров’я на основі Інтернету речей сприятиме зростанню приблизно на 1,1–2,5 трильйона доларів США на рік. Це змінить світову економіку, і оцінюваний вплив до 2025 року становитиме від 2,7 до 6,2 трильйонів доларів [5]. Зростання кількості пристроїв IoT приваблює зловмисників, щоб отримати доступ для досягнення своїх цілей. За даними Symantec, у 2022 році кількість кібератак зросла на 200% порівняно з 2021 роком і склала приблизно 3 мільярди атак [6].

1.3.1 Обмеження та вразливості пристроїв IoT

Підключені пристрої піддаються різним видам загроз, і їх кількість зростає з кожним днем. До пристроїв малої потужності не можна застосувати

традиційні методи забезпечення безпеки. Щоб зберегти безпеку пристроїв і конфіденційність споживачів, важливо заблокувати доступ зловмисників до пристроїв або мережі.

Застосувати звичайні підходи до безпеки в мережах або пристроях IoT складно, оскільки вказані пристрої зазвичай мають обмежені ресурси. Основні обмеження безпеки пристроїв IoT наведені в таблиці 1.2 [7].

Таблиця 1.2 – Обмеження безпеки пристроїв IoT

Апаратні обмеження	Обмеження програмного забезпечення	Обмеження мережі
Обчислювальні та енергетичні обмеження Обмеження пам'яті Упаковка, захищена від втручання	Вбудоване програмне обмеження Динамічний патч безпеки	Мобільність Масштабованість Множинність засобів зв'язку Багатопротокольна мережа Динамічна топологія мережі

1.3.2 Поверхні атак IoT

Із зростанням кількості та різноманітності пристроїв IoT поверхня атаки значно збільшилася. Інструменти (наприклад, мережі та протоколи) та сутності складають поверхню атаки (тобто пристрої, методи та інформація). Поверхня атаки визначається підключенням компонентів системи, а також політиками, які керують дозволом пристрою для доступу до системи.. Можливими поверхнями для атаки можуть бути адміністративний інтерфейс, веб-інтерфейс пристрою/хмари, механізми оновлення, мобільні програми, фізичні інтерфейси, мікропрограмне забезпечення пристрою, пам’ять пристрою тощо. Поверхня атаки групує численні місця, якими зловмисник може скористатися для отримання доступу до системи та викрадення/витоку/зміни інформації. За кожною поверхнею атаки стоять

певні елементи та функції пристроїв мережі IoT, у яких є набір недоліків безпеки. Визначивши поверхню атаки, можна визначити ризики безпеки та потенційно вразливі області, де потрібен захист глибокого рівня. Велика кількість поверхонь атак, які зловмисник може використовувати для виконання своїх шкідливих операцій, безсумнівно, є стимулом для розробки ефективних рішень безпеки. Крім того, через обмеженість ресурсів вузлів IoT звичайні заходи безпеки неможливо реалізувати, що ставить під загрозу всю мережу. Ботнет Mirai та його похідні, які можуть взяти під контроль пристрої IoT і запустити нищівну DDoS-атаку, є чудовими прикладами таких ризиків.

1.3.3 Вразливості Інтернет - речей

Із зростанням кількості пристроїв Інтернету речей загрози безпеці також зростають, оскільки зловмисники отримують шанс маніпулювати величезною кількістю даних. Без належної безпеки пристрої IoT будуть вразливі до витоку конфіденційних даних. Крім того, пристрої IoT уразливі до нападів і ризиків безпеці, оскільки їм бракує необхідної вбудованої безпеки для боротьби із загрозами через їх низьку ціну, мінімальну потужність і низькі обчислювальні можливості, а також через неоднорідність і масштаб мережі. Пристрої IoT вразливі до загроз не лише з технічних аспектів, але й через діяльність користувачів. Ось кілька причин, чому ці розумні пристрої все ще знаходяться в зоні ризику [8].

1. Обмежені апаратні та обчислювальні можливості: ці пристрої розроблені для конкретних програм, які вимагають лише обмежених можливостей обробки, залишаючи мінімальну область для безпеки та захисту даних для інтеграції.

2. Гетерогенна технологія передачі даних: ці пристрої спілкуються з різними типами пристроїв і часто використовують різні технології зв’язку, що ускладнює встановлення єдиних заходів захисту та протоколів.

3. Компоненти пристрою вразливі: мільйони розумних пристроїв можуть бути пошкоджені незахищеними або застарілими елементами.

4. Користувачі недостатньо обізнані про безпеку: через брак знань користувачів про безпеку розумні пристрої можуть бути піддані зонам ризику та ймовірним атакам. Багато пристроїв IoT дозволяють користувачам інтегрувати програми сторонніх розробників, що також може поставити пристрій у небезпечну зону.

5. Слабка фізична безпека: на відміну від центрів обробки даних Інтернет-сервісів, не лише користувачі, а й інші особи з поганими намірами мають фізичний доступ до основної частини компонентів IoT.

Проблеми безпеки поділяються на загрози програмного рівня та загрози апаратного рівня, як показано на рисунку 1.5.

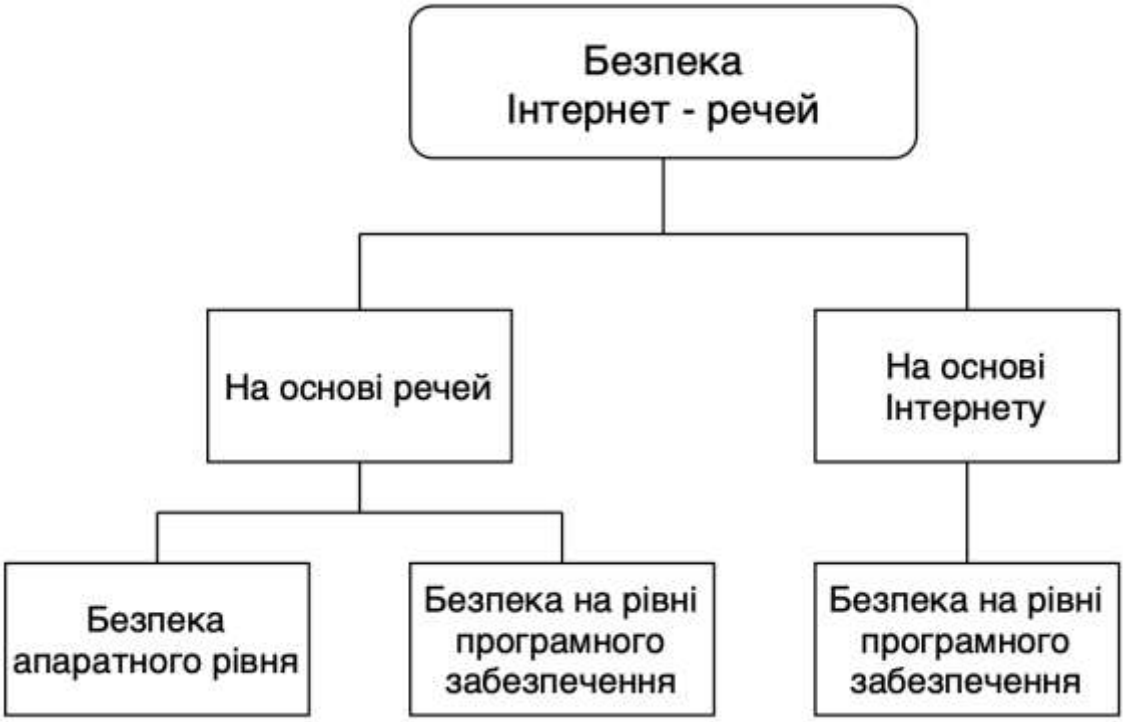


Рисунок 1.5 – Атака на апаратне та програмне забезпечення

Витік інформації, незаконний доступ тощо, визначаються як атаки на рівні програмного забезпечення, щоб змусити систему працювати неправильно та зібрати бажану інформацію, як-от дані кредитної картки, пароль тощо. Брандмауер, оновлена база даних вірусів і використання найновішого програмного забезпечення можуть обмежити атаки на основі

програмного забезпечення. Не тільки атаки на рівні програмного забезпечення, але й атаки на рівні апаратного забезпечення є помітним полем для зловмисників. Щоб побудувати повністю захищене апаратне забезпечення, необхідно розробити захищені інтегральні схеми (IC) або SoC. Введення однієї шкідливої схеми під час виготовлення може призвести до компрометації системи, і це може бути непомітним для розробників.

Верхні елементи піраміди більш вразливі з мінімальним ймовірним впливом, а елементи нижчого рівня мають протилежні характеристики. Можна стверджувати, що кіберзагрози шукають більшого контролю та можливостей на нижчому рівні.

У компонентах системи є дефекти, які роблять її вразливою та розширюють поверхню атаки. Зокрема, зловмисник прагне використовувати апаратне або програмне забезпечення системи IoT, щоб отримати доступ для виконання своїх зловмисних дій. У звіті HP сказано, що 50% комерційно доступного IoT має значний недолік безпеки. Важливо запобігати та реагувати на раніше перераховані вразливості, оскільки вони можуть розкрити конфіденційну інформацію в системах IoT. Оскільки мережа IoT піддається різним видам атак, це складне завдання для аналізу безпеки. Тим не менш, величезний обсяг даних, створених середовищами IoT, сприяє підвищенню рівня безпеки всієї системи.

1.3.4 Вимоги безпеки Інтернет - речей

Щоб убезпечити обладнання, необхідно розуміти цілі безпеки. Конфіденційність, цілісність і доступність відомі як тріада CIA. У таблиці 1.3 перераховані цілі безпеки запропоновані IAS, а також приведено їх визначення [9].

Таблиця 1.3 – Вимоги безпеки IoT

Вимоги безпеки	Визначення
Конфіденційність	Процес, під час якого суворо зберігається таємниця та конфіденційність інформації, що транслюється в ефірі та зберігається, і доступ до неї мають лише дозволені об’єкти або користувачі.
Цілісність	Процес, у якому не відбувається змінення даних І і забезпечується точність.
Невідмовність	Процедура, за допомогою якої система IoT перевіряє легітимність і походження події.
Доступність	Процес забезпечення доступності послуг для тих, хто їх потребує, навіть якщо сталося відключення електроенергії або поломка.
Приватність	Метод, за допомогою якого система IoT має доступ до приватних даних, дотримуючись правил і політик.
Аудит конфіденційності	Процес, за допомогою якого система IoT відстежує свої дії.
Відповідальність	Механізм, за допомогою якого користувачі системи IoT відповідатимуть за свої дії.
Надійність	Метод, за допомогою якого система IoT може підтвердити ідентифікацію особи та встановити довіру до третьої сторони.

Конфіденційність пов'язана з набором правил, які встановлюють критерії для уповноважених осіб, які мають доступ до інформації. Цілісність – ще одна характеристика, яка забезпечує надійні послуги, щоб пристрої IoT отримували лише законні команди та інформацію. Доступність гарантує, що функціональні можливості IoT доступні законним об’єктам і користувачам у будь-який час і в будь-якому місці.

2 АТАКИ НА ІНТЕРНЕТ- РЕЧЕЙ ТА МЕХАНІЗМИ ЗАХИСТУ

2.1 Класифікація атак безпеки на різних рівнях мереж Інтернет-речей

Кіберзлочинці шукають вразливі місця пристроїв і використовують їх, щоб отримати перевагу в проведенні атак. Атаки на IoT поділяються на чотири категорії (рисунок 2.1) [10, 11]:

1) фізичні атаки або атаки сприйняття;

2) мережеві атаки,

3) атаки на програмне забезпечення або програми;

4) атаки на шифрування.

Рисунок 2.1 –Загрози безпеці Інтернету речей

2.1.1 Фізичні атаки

Ці атаки є результатом порушень обладнання системи IoT, наприклад датчиків пристроїв IoT, і зловмисники отримують доступ через близькість, наприклад, вставляючи USB-накопичувач. За оцінками, 70% усіх кібератак починаються зсередини, навмисно чи випадково з боку людей. Ці атаки можуть обмежити термін служби або функціональність апаратного забезпечення. До фізичних відносяться наступні атаки.

1. Перешкоди у вузлах БСМ: атака на перешкоди на вузлах у БСМ здійснюється зловмисними вузлами в мережі, які заважають, порушують або глушать радіосигнали, що використовуються, надсилаючи непотрібну інформацію у використовуваному діапазоні частот. Залежно від потужності

джерело перешкод може вивести з ладу всю систему або лише її невелику частину. Це блокування може бути тимчасовим, періодичним або постійним.

2. Фізична шкода: атака фізичної шкоди – це атака, яка призводить до фізичного пошкодження цільової інфраструктури. Зловмисник може отримати доступ до пристроїв, фізично пошкодивши об'єкти мережі IoT. Цього типу атаки можна уникнути, якщо захистити зону, де розташовані пристрої IoT.

3. Втручання в вузол: атака втручання в вузол – це атака, під час якої зловмисник пошкоджує вузол IoT. Вузол може бути замінений або частина апаратного забезпечення може бути скомпрометована. Зламаний вузол може бути створений шляхом заміни вузла, і зловмисник може контролювати цей вузол. Отримавши доступ, зловмисник може змінити конфіденційну інформацію, наприклад, спільні криптографічні ключі/облікові дані (якщо такі є) або таблиці маршрутизації, а також порушити функціональність вищих рівнів зв'язку.

4. Соціальна інженерія: під час атак соціальної інженерії зловмисна діяльність здійснюється через взаємодію з людиною. Соціальна інженерія – це психологічна маніпуляція з метою змусити користувачів системи Інтернету речей виконати певні дії або надати конфіденційну інформацію, яка служитиме їхнім цілям. Зловмисник починає зі збору базових даних, необхідних для здійснення атаки. Потім зловмисник завойовує довіру жертви і отримує необхідну інформацію.

5. Ін'єкція зловмисного вузла: у цій атаці зловмисник фізично розгортає зловмисний вузол у мережі IoT, який збирає інформацію для зловмисника або змінює дані зв'язку, щоб передавати неправильну інформацію іншим вузлам. Таким чином, зловмисник контролює передачу та отримання потоку даних і, нарешті, роботу вузлів.

6. Атака із затримкою сну: пристрої IoT запрограмовані на виконання режиму сну, щоб залишатися в режимі низького енергоспоживання настільки довго, наскільки це можливо, без негативного впливу на програми вузла;

отже, це подовжує термін служби акумулятора. Обчислювальні пристрої, такі як сенсорний вузол, які живляться від батареї, сприйнятливі до атак скорочення часу сну. Зловмисник взаємодіє з вузлом таким чином, щоб вузол-жертва залишався активним, що призводить до більшого споживання електроенергії та виходу з ладу.

7. Впровадження шкідливого коду: зловмисник атакує пристрій фізично та впроваджує шкідливий код, щоб скомпрометувати систему та отримати доступ. Це можна зробити, вставивши USB зі зловмисною програмою у вузол або вставивши канали зв'язку за допомогою таких методів, як: 1) вставлення зловмисних кодів, які здаються законними; 2) модифікація кодів/пакетів даних після захоплення; 3) заміна раніше змінених повідомлень між вузлами. Мета атаки з ін'єкцією зловмисного коду може бути різною, наприклад, викрадення даних, отримання контролю над усією чи частковою системою та розповсюдження хробаків.

8. Радіочастотні перешкоди: RFID – це технологія автоматичної ідентифікації, у якій зв'язок здійснюється за допомогою радіочастот (RF) з ідентифікаційним кодом (ID). Під час цієї атаки сигнал скомпрометований шляхом створення та надсилання шумових сигналів через радіочастотний сигнал, який використовується для зв'язку RFID. Шум спричинить перешкоди для сигналів RFID.

9. Клонування тегів: під час атаки клонування зловмисник хоче мати тег, який матиме ті самі характеристики, що й оригінальний тег, і зрештою зможе замінити його. Під час цієї атаки зловмисник зможе скопіювати інформацію електронної мітки RFID або смарт-карти на клоновану мітку за допомогою зворотного проектування або безпосередньо з середовища розгортання. Перехоплення, підслуховування та інші технології використовуються в атаках клонів, щоб отримати дані з оригінального тегу, включаючи кодування та інформацію про клієнта.

10. Підслуховування: атака підслуховування, також відома як перехоплення або стеження, є крадіжкою інформації, оскільки вона

передається за допомогою бездротового зв’язку. Для підслуховування зловмисник може використовувати антену для отримання переданих даних у системі RFID. Щоб досягти успіху, зловмисник знаходить слабке з’єднання, щоб використовувати його для перенаправлення мережевого трафіку.

11. Підробка тегів: під час атаки підробки тегів метою зловмисників є змінити ідентичність тегу. Зловмисники можуть підробити мітку в системі RFID за допомогою методу маніпуляції з міткою. Зловмисники отримають доступ до каналу зв’язку, змінивши тег пристрою IoT.

12. Атака через збій: зловмисник може використовувати більше потужності, ніж дозволений діапазон. Ця операція може вивести пристрої з ладу.

13. Реплікація об’єктів: так як пристрої IoT не контролюються фізично у віддалених місцях, зловмисник може фізично вставити новий пристрій у мережу. Наприклад, зловмисний пристрій може бути доданий шляхом копіювання ідентифікаційних даних законного пристрою. У результаті така атака може призвести до значного зниження продуктивності мережі. Окрім зниження продуктивності, шкідливий об’єкт може просто пошкодити або неправильно спрямувати отримані пакети, надаючи зловмиснику доступ до конфіденційної інформації та витягуючи секретні ключі.

14. Апаратний троян: під час цієї атаки зловмисник спеціально модифікує інтегральну схему. Атаки апаратних троянів плануються на етапі проектування та залишаються неактивними, доки розробник не активує подію.

2.1.2 Атака на шифрування

Атака шифрування полягає в тому, щоб зробити вразливою алгоритм шифрування, який використовується в системі IoT [12].

1. Атака «людина посередині»: атака «людина посередині» – це кібератака, коли зловмисник займає позицію між двома користувачами на лінії зв’язку та ділиться ключами з обома користувачами. Зловмисник може

перехопити сигнал, який обидва користувачі надсилають один одному, і може зашифрувати або розшифрувати дані за допомогою ключів, якими він ділиться з ними обома. Зловмисник також може змінити зв'язок між двома сторонами без їх відома, які думають, що вони надсилають дані одна одній.

2. Атаки по бічному каналу: фізичні характеристики пристроїв IoT (наприклад, енергоспоживання, час виконання, електромагнітні витоки, системні збої тощо) можуть виявити конфіденційну інформацію. Під час роботи пристроїв IoT зловмисник виконує різні тести для отримання конфіденційної інформації. Іноді потрібно мати технічні знання про внутрішній принцип роботи системи, яка буде використовуватися. У загальнодоступних криптографічних системах, таких як RSA, вилучення інформації з поведінки пристрою є звичайним. RSA виконує шифрування та дешифрування повідомлень за допомогою закритих і відкритих ключів на основі модульних операцій і великих експоненційних значень. Зловмисник виконує аналіз затримки, щоб дізнатися, скільки часу потрібно для обчислення експоненціальних результатів.

Зловмисники можуть отримати конфіденційну інформацію, таку як секретні ключі, вивчаючи час обчислення та використовуючи знання техніки реалізації. Більшість пристроїв IoT запровадять такі заходи безпеки, як шифрування, щоб захистити конфіденційну інформацію з міркувань безпеки. Однак за допомогою атаки по бічному каналу механізм безпеки може бути зламаний.

3. Атаки криптоаналізу: зловмисник під час атаки криптоаналізу вивчає зашифрований текст, шифри та криптосистеми з метою пошуку ключа шифрування, який використовується шляхом зламу схеми шифрування системи. Зловмисник зламує системи криптографічної безпеки та отримує доступ до зашифрованих повідомлень, навіть не знаючи джерела відкритого тексту, ключа шифрування або алгоритму, який використовується для його шифрування . Захищене хешування, цифрові підписи та інші криптографічні алгоритми також є цілями цієї атаки.

2.1.3 Мережеві атаки

Мережеві атаки зосереджені на мережі систем IoT для віддаленого доступу до великих обсягів даних [12].

1. Атаки аналізу трафіку: на відміну від атак підслуховування, зловмиснику не потрібно скомпрометувати вихідні дані, зловмисник прослуховує мережу, щоб отримати деяку інформацію, використовуючи програми аналізу, такі як сканування портів, аналіз пакетів.

2. Клонування RFID: створення копії RFID користувача без відома є ще одним способом зламу системи доступу RFID. Навіть без фізичного доступу до RFID-картки зловмисник може клонувати RFID-мітку, скопіювавши дані з RFID-мітки жертви на іншу RFID-мітку. Зловмисник може отримати інформацію та записати дані в подібний порожній RFID, використовуючи готові компоненти, стоячи на відстані кількох метрів. Цілісність системи буде порушена, оскільки результатом клонування є обмін ідентичними тегами.

3. Підробка RFID: на відміну від клонування RFID, під час атаки підробки RFID зловмисник фізично не копіює тег RFID. Технічно атаки клонування та спуфінгу здійснюються паралельно. У цьому типі атаки зловмисник імітує дійсну мітку RFID, щоб отримати свої привілеї, зчитує та записує передачу даних із мітки RFID. Зловмисник може отримати повний контроль над системою, видаючи себе за легітимне джерело та надсилаючи власні дані, які включають ідентифікатор автентичного тегу. Атаки підробки відбуваються, коли хакер успішно займає позицію авторизованого користувача в системі.

4. Несанкціонований доступ RFID: у RFID можуть бути доступні різні рівні функцій безпеки. Якщо в системі RFID не використовуються належні механізми автентифікації, зловмисник може отримати доступ до тегів. Зловмисник може просто читати, редагувати або навіть знищувати дані на пристроях RFID для власної вигоди.

5. Людина в центрі атаки: зловмисник розміщується в інтерфейсі між двома датчиками, збираючи приватні дані та порушуючи конфіденційність, підслуховуючи або видаючи одного з клієнтів, щоб виглядало, ніби відбувається звичайний потік інформації. Метою цієї атаки є отримання особистої інформації або несанкціонована зміна пароля. Ця атака покладається виключно на мережеві протоколи зв'язку системи IoT [9].

6. Відмова в обслуговуванні: щоб здійснити успішну атаку на відмову в обслуговуванні, зловмисник переповнює мережу IoT великою кількістю запитів, що призводить до значної кількості трафіку даних; це триває до тих пір, поки ціль не зможе відповісти або просто не вийде з ладу. У цій атаці законні користувачі не можуть використовувати мережеві ресурси для доступу до інформації, оскільки всі доступні ресурси вичерпані, що робить мережеві ресурси недоступними для користувачів. Крім того, незашифровані дані багатьох користувачів також можуть бути розкриті.

7. Атака sinkhole: під час атаки sinkhole зловмисник обманює систему, заманюючи весь потік даних із сусідніх вузлів БСМ у скомпрометований вузол. Зловмисник використовує скомпрометований вузол для залучення мережевого трафіку шляхом передачі шахрайської інформації про маршрутизацію. Метою зловмисника є порушення цілісності системи, а також переривання роботи мережі.

8. Атаки з інформацією про маршрутизацію: під час атаки з інформацією про маршрутизацію зловмисник використовує скомпрометований вузол або групу скомпрометованих вузлів, щоб створити або змінити інформацію про маршрутизацію. Метою атаки є обфускація системи та створення петель маршрутизації, дозвіл або відхилення трафіку, зміна пункту призначення, надання підроблених повідомлень про помилки, скорочення чи розширення вихідних шляхів або навіть розділення мережі.

9. Атака Sybil: атаки Sybil частіше зустрічаються в мережах з великою кількістю клієнтів. Один вузол, який незаконно отримує ідентифікаційні дані багатьох інших вузлів, називається шкідливим вузлом. Зловмисник

використовує ідентифікаційні дані інших вузлів, змушуючи сусідні вузли отримувати фальшиву та невірну інформацію. Зловмисник може взяти участь у розподіленому алгоритмі, такому як вибори, де один вузол sybil ідентифікується декілька разів. Його також можна вибрати як частину шляху маршрутизації, що призводить до більшої відстані маршрутизації.

10. Повторна атака: зловмисники отримують інформацію шляхом підслуховування повідомлень двох сторін, а зловмисний вузол повторно надсилає старі пакети в систему загалом у вигляді трансляції або на певний набір пристроїв. Коли інші вузли отримують ці повідомлення, вони оновлюють свої таблиці маршрутизації відповідно до цієї простроченої інформації та відповідають незалежно від того, передає відправник нові пакети чи ні. Таблиця маршрутизації та топологія мережі також застаріють, і через величезну кількість відтворюваних пакетів споживатиметься як пропускна здатність, так і енергія. Це призведе до того, що діяльність мережі буде припинено раніше, ніж очікувалося.

11. Атака шантажу: під час атаки шантажу скомпрометований вузол усуває законний вузол із мережі, оголошуючи, що законний вузол є шкідливим вузлом. Якщо скомпрометований вузол здатний заблокувати велику кількість вузлів, мережа стане нестабільною.

12. Атака Blackhole: у цій атаці зловмисний вузол замість того, щоб пересилати всі пакети, може скинути їх, і він може скинути весь трафік даних навколо зловмисного вузла. Цей напад також називають «егоїзмом». Його вплив найвищий, якщо шкідливий вузол є приймачем даних.

2.1.4. Атаки на програмне забезпечення або програми

Програмне забезпечення або додаток є четвертою категорією, яка підвищує небезпеку IoT. Зловмисне програмне забезпечення впроваджується в програму мережі, щоб почати цю атаку. Ця шкідлива програма може порушувати та контролювати операції, поширювати віруси, пошкоджувати або викрадати дані тощо [13, 14].

1. Вірус і хробаки: Вірус – це невелике комп’ютерне програмне забезпечення, яке може розмножуватися й заражати інші комп’ютери так само, як і справжній вірус. Комп’ютерний черв’як подібний до вірусу тим, що він поширюється сам по собі, а не маскується всередині іншого програмного забезпечення.

2. Шкідливі сценарії: зазвичай у мережі IoT є підключення до Інтернету. У цій атаці зловмисник вводить в оману користувача, який контролює шлюз, відвідуючи прибуткові рекламні оголошення або веб-сайти, а потім запускаючи виконувані сценарії Active-X зі зловмисними змінами в різних частинах системи, що може призвести до вимкнення системи. або викрадення даних.

3. Шпигунське та рекламне ПЗ: для розповсюдження зловмисного програмного забезпечення та шкідливого коду зловмисники атакують пристрої, такі як Інтернет речей, використовуючи заводські облікові дані користувача за замовчуванням, підбираючи паролі, зламуючи та маніпулюючи слабкими конфігураціями. Як тільки зловмисник отримає повний контроль, він може розпочати DDoS-атаку на ціль. Зловмисник може вставити шкідливе програмне забезпечення в систему, що може призвести до крадіжки даних, підробки даних або навіть відмови в обслуговуванні.

4. Троянський кінь – це легітимна програма, у якій присутня якась раніше визначена подія чи дата. Ця атака запускається в попередньо встановлених умовах, а потім доставляє корисне навантаження, яке може повністю вимкнути систему.

5. Відмова в обслуговуванні: зловмисник може використовувати розподілені атаки DDoS і DoS, щоб запобігти доступу користувачів до системи або мережевого ресурсу. Ця атака підриває здатність мережі або системи виконувати очікувані функції. Ця атака йде з багатьох точок, і від неї важко захиститися.

6. Атака на розподілену відмову в обслуговуванні з (DDoS): DDoS є однією з форм атаки DDoS. У цій атаці зловмисник зберігає анонімність

через IP-адресу за допомогою третьої сторони, яка називається зомбі, яку не потрібно скомпрометувати. Численні комп'ютери-жертви, які ненавмисно беруть участь у DDoS-атаці на ціль зловмисника, зазвичай використовуються в атаках DDoS. Перевага атаки полягає в тому, що особу зловмисника не буде розкрито, оскільки атаку буде здійснено вузлом зомбі, а не зловмисником, що ускладнює ідентифікацію початкового зловмисника та блокування служби. Брандмауери, системи виявлення вторгнень, машинне навчання використовують для пом'якшення атак DRDoS [11].

7. Викрадення мікропрограми: основним компонентом апаратного забезпечення вузла є мікропрограма. На кожному апаратному забезпеченні комп'ютера попередньо встановлено базове програмне забезпечення. Модифікація мікропрограми або викрадення є однією з найбільш катастрофічних атак, коли зловмисники можуть отримати контроль над усією системою [12].

8. Атака ботнету: ботнет – це мережа роботів. Зазвичай зловмисник отримує контроль, поширюючи вірус або інший шкідливий код, щоб заволодіти мережею пристроїв і створити ботнет. Це не атака зловмисного програмного забезпечення, хоча вона здійснюється через компрометуючі пристрої. Ботнет IoT також є мережею різноманітних заражених шкідливим програмним забезпеченням пристроїв IoT, таких як маршрутизатори, переносні пристрої та вбудовані технології. Це зловмисне програмне забезпечення дозволяє зловмиснику контролювати всі підключені пристрої та, зрештою, мережу [13].

9. Атака на пароль грубою силою – це метод пошуку для отримання привілейованого доступу, коли зловмисник вгадує можливі комбінації цільового пароля, доки не буде виявлено правильний пароль. Залежно від довжини та складності пароля знадобиться як час, так і застосована комбінація.

10. Фішингові атаки: під час цієї атаки зловмисник отримує приватні дані, як-от імена користувачів і паролі, за допомогою веб-сайтів для підробки електронної пошти та фішингових веб-сайтів [14].

2.2 Механізми безпеки Інтернет – речей

Існує програмний і апаратний підходи до захисту пристроїв IoT від потенційних вторгнень. Програмне забезпечення відповідає за захист пристроїв від програмних атак. Важко зламати математичний алгоритм програмного забезпечення за допомогою сучасної комп'ютерної системи. Однак він зможе розгадувати математичні ключі за більш короткий час порівняно з поточним підходом, коли квантові комп'ютери досягнуть відповідної потужності. У рішеннях безпеки на основі програмного забезпечення пристрої схильні до атак, оскільки ключі зберігаються в енергонезалежній пам'яті (NVM) пристроїв. Поява квантових комп'ютерів може зробити програмні рішення безпеки вразливими. Таким чином, апаратне рішення може бути одним із можливих рішень через фактор ризику існуючої програмної безпеки. За існуючими прогнозами більшість асиметричної криптографії буде зламано з появою квантових комп'ютерів та використання алгоритму Шора [15]. Тому необхідно стандартизувати нові алгоритми для збереження цілісності. NIST працює над стандартизацією криптографічних механізмів для протистояння атакам на постквантову криптографію. На рисунку 2.2 приведені апаратні механізми безпеки.

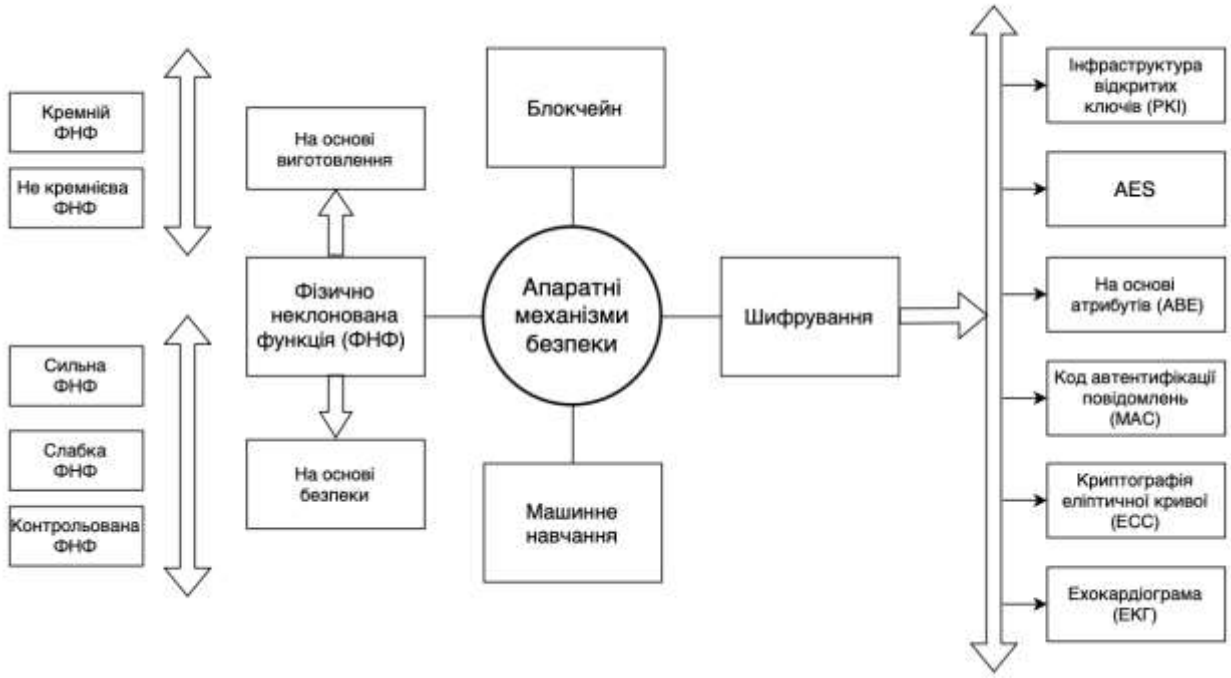


Рисунок 2.2 – Апаратні механізми безпеки

Криптографічні функції можна використовувати в апаратній безпеці за допомогою спеціальної мікросхеми для зберігання ключів. Фізичні обчислювальні пристрої, які називаються апаратними модулями безпеки, можна використовувати для криптообробки та надійної перевірки. Вони можуть шифрувати, розшифровувати, зберігати та обробляти цифрові ключі за допомогою різних алгоритмів шифрування. Так як відкритий ключ передаються кожним вузлом мережі, отже легко запустити атаки типу “людина посередині”. Зловмисники можуть клонувати пристрій після викрадення. Фізичні функції, які не підлягають клонуванню (Physical Unclonable Functions, PUF; фізично неклоновані функції, ФНФ) є рішенням, щоб уникнути атак на IoT [14]. ФНФ може створювати цифрові відбитки пристрою за запитом. ФНФ створює інший ключ (Відповідь), коли інший вхід (Виклик) подається на пристрій, що пов’язано з виробничими особливостями мікросхеми [15, 16]. На рисунку 2.3 приведено характеристики ФНФ [17, 18].

Рисунок 2.3 – Характеристики ФНФ

Унікальність: потрібні різні відповіді на кожний вхід. Це визначається тим, наскільки один чіп відрізняється від іншого з точки зору рядків і, зрештою, варіації процесу ФНФ. Відстань Хемінга (ВХ), розрахована з кожною відповіддю, представляє унікальність. Кількість бітових відмінностей між двома відповідями обчислює ВХ. В ідеалі унікальність повинна бути 50%.

Випадковість: обчислюється на основі загальної кількості «1» і «0» у відповіді. У відповіді очікується наявність рівної кількості «1» і «0». Якщо так, ФГФ матиме 100% випадковість.

Правильність: ФНФ має генерувати однакову пару виклик-відповідь (Challenge-Response Pair, CRP) незалежно від зовнішніх змінних, таких як

температура, тиск, час тощо. У будь-якому робочому стані ідеальним значенням є 100%.

Надійність: CRP, як випливає з назви, має бути надійним на 100%. Щоразу, коли поступає виклик, ідеальний ФНФ дасть однакову відповідь.

Однорідність: вимірює, наскільки випадковим є ФНФ. Імовірність «1» і «0» має бути рівною, щоб будь-які зловмисники не змогли вгадати.

Зміщення бітів: як певний біт відповіді на кількох чіпах вимірюється за допомогою зміщення бітів, і ідеальне значення становить 50%.

Стабільність: усі відповіді мають бути ідентичними, коли ті самі виклики надходять до ФНФ.

2.3 Фізично неклоновані функції

У все більш взаємопов’язаному світі, який значною мірою залежить від електроніки, безпека має першочергове значення. Сучасна електроніка майже повністю покладається на криптографію як основний метод захисту електронних даних. Проте нова сфера досліджень апаратної безпеки довела, що криптографія, яку ми знаємо, не така вже й безпечна.

З цією метою з’явилися фізично неклоновані функції (ФНФ) як апаратна техніка безпеки, яка пропонує все від покращеної криптографії до захисту від підробок на мікросхемах.

У цьому пункті розглянемо концепцією ФНФ, як вони працюють і як вони використовуються для захисту даних.

Фізично неклоновані функції – це техніка в апаратній безпеці, яка використовує властиві варіації пристроїв для створення неклонованої унікальної реакції пристрою на заданий вхід. На більш високому рівні ФНФ можна розглядати як аналог біометрії для людей – вони є невід’ємними та унікальними ідентифікаторами для кожного шматка кремнію.

Через недосконаліть технологій обробки кремнію кожна окрема мікросхема, коли-небудь виготовлена, фізично відрізняється одна від одної. Від інтегральної схеми (IC) до IC ці варіації процесу проявляються такими способами, як різні затримки шляху, порогові напруги транзисторів, посилення напруги та незліченна кількість інших.

Важливо, що хоча ці варіації можуть бути випадковими від IC до IC, вони є детермінованими та повторюваними, коли вони відомі. ФНФ використовує цю властиву різницю в поведінці IC, щоб створити унікальний криптографічний ключ для кожної IC (рисунок 2.4).

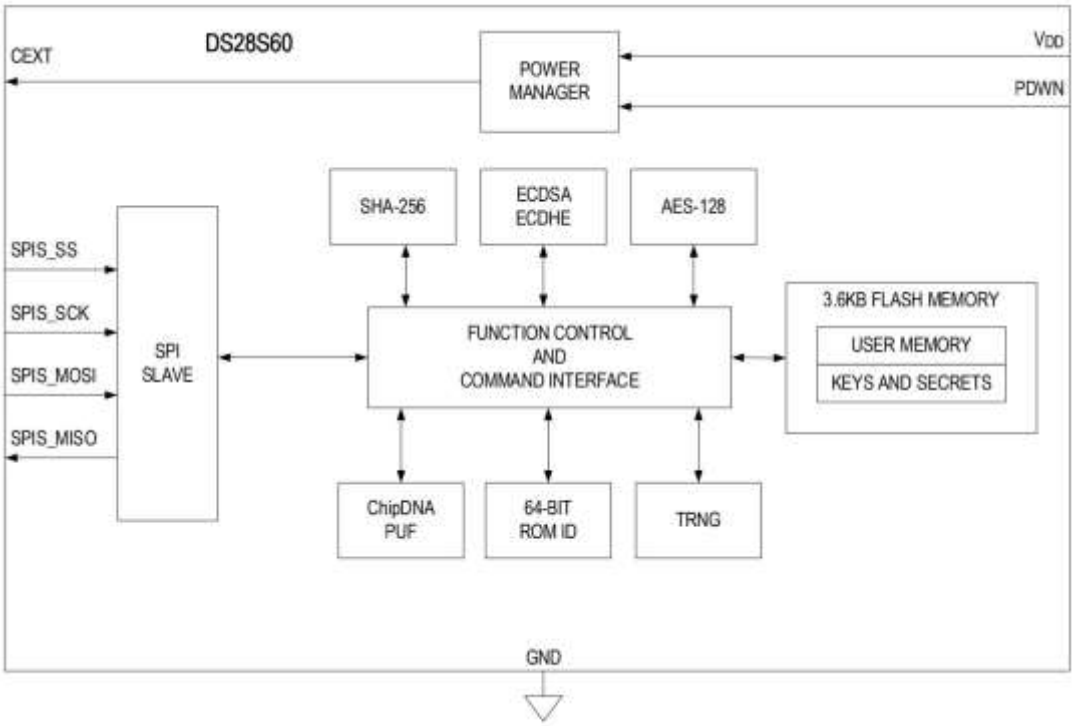


Рисунок 2.4 – Співпроцесор DS28S60 від Maxim Integrated використовує ФНФ для генерації криптографічного ключа [19].

На відміну від традиційного криптографічного підходу, який використовує один збережений ключ, ФНФ працюють шляхом реалізації автентифікації виклик-відповідь. Для певного ФНФ певний вхід, відомий як «виклик», створить вихідну відповідь, яка є унікальною для конкретного ФНФ і тому не клонується.

Після виготовлення ФНФ буде піддаватися серії різних викликів і відповіді на нього будуть записані. Завдяки цій вправі розробники знають

унікальну відповідь кожного ФНФ на певний виклик і можуть використовувати цю інформацію для запобігання підробкам, створення та зберігання криптографічних ключів і багатьох інших заходів безпеки.

Приклад ФНФ. Щоб краще проілюструвати, як працюють ФНФ, ми розглянемо DRAM ФНФ – ФНФ, який використовує варіації процесу DRAM для генерації криптографічних ключів і пропонує автентифікацію пристрою.

Стандартна комірка DRAM працює з одним конденсатором, який утримує збережений заряд у двійковому стані, і транзистором, який контролює потік заряду до конденсатора та від нього. Через неідеальність пристрою, наприклад підпороговий витік транзистора, заряд конденсатора має тенденцію до витоку з часом, що спричиняє втрату стану елемента. Це означає, що повністю заряджена комірка DRAM, яка представляє бітове значення «1», з часом небажано розрядиться до бітового значення «0».

Що важливо для ФНФ, на швидкість виходу кожної окремої комірки DRAM сильно впливають варіації процесу у виробництві комірки (рисунок 2.5).

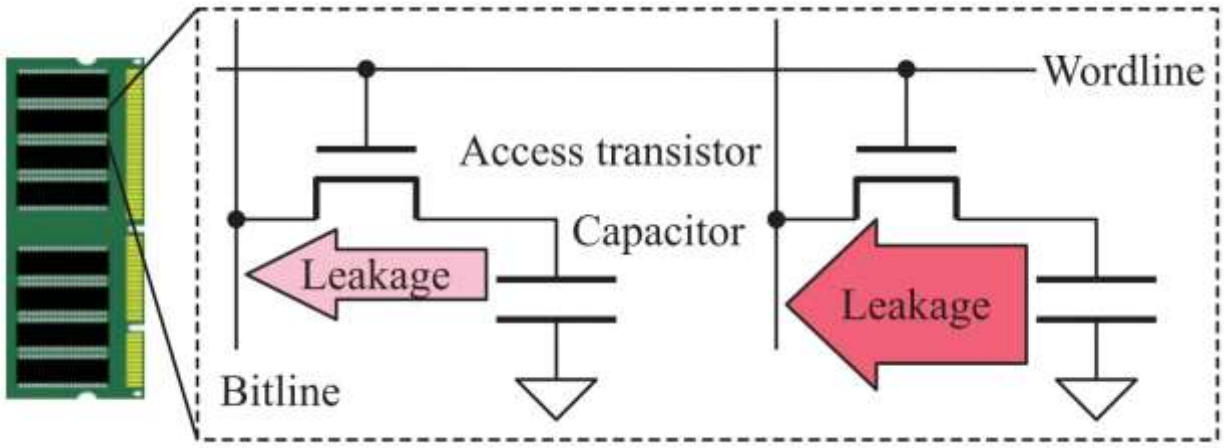


Рисунок 2.5 – Витік заряду в елементах DRAM сильно змінюється залежно від змін процесу [20].

Щоб протидіяти цьому, усі комірки DRAM виконують періодичні команди оновлення, які відновлюють заряд для «оновлення» накопичувального конденсатора. З іншого боку, DRAM ФНФ працює,

призупиняючи це оновлення на довший, ніж зазвичай, заданий проміжок часу та спостерігаючи, як комірка змінили стан через вихід.

Оскільки різні елементи втрачають заряд з різною швидкістю, ми можемо очікувати, що деякі елементи повністю розрядяться та змінять стан протягом певного інтервалу часу, тоді як інші можуть не розрядитися настільки, щоб взагалі змінити стан.

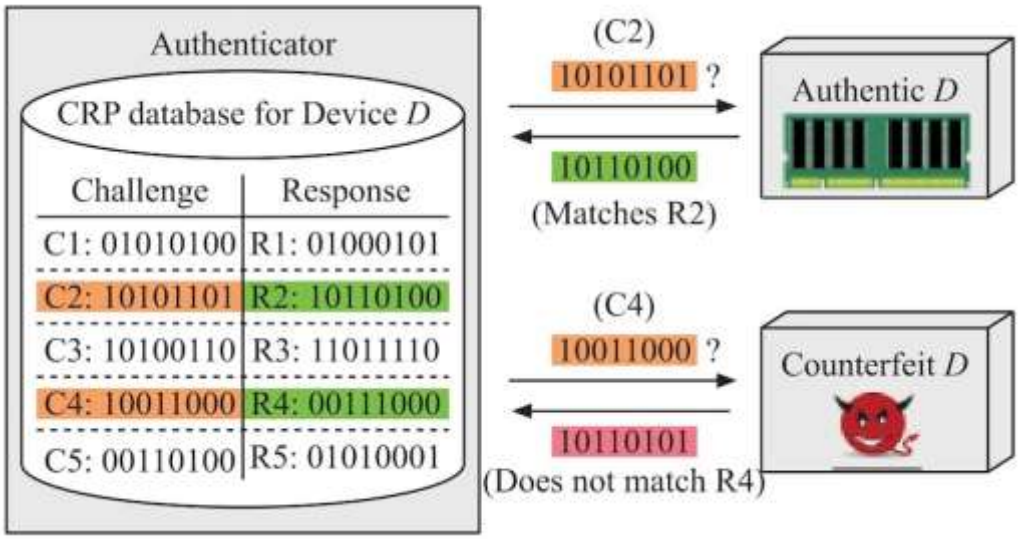


Рисунок 2.6 – Аутентифікатор може використовувати базу даних пари виклик-відповідь для автентифікації пристрою [20].

У цьому випадку «завданням» є вихідне двійкове значення, подане до масиву комірок DRAM, а відповіддю є значення цього масиву після заданого інтервалу часу. Цю техніку можна використовувати для створення справді випадкових чисел для генерації криптографічного ключа або для ідентифікації пристрою для захисту від підробок. В останній програмі автентифікатор може зберігати базу даних пар виклик-відповідь і використовувати ці знання для ідентифікації підроблених пристроїв проти автентичних.

Переваги ФНФ. Переваги використання ФНФ великі, тому ця технологія стає все більш популярною для апаратної безпеки.

Для початку одна з найкращих особливостей ФНФ полягає в тому, що це за своєю суттю енергонезалежна техніка, але вона фізично не зберігає жодних ключів. Зберігання ключів в енергонезалежній пам’яті зазвичай відкриває ІС для апаратних атак, які дозволяють зловмисникам читати вміст пам’яті.

Натомість ФНФ взагалі не зберігає ключ. Вона генерує ключ за потреби у відповідь на запит, після чого ключ миттєво стирається. ФНФ працює за принципом: «завжди є ключ, але ви ніколи не можете на нього дивитися». І навіть якщо ви спробували «подивитися на це», спроби дослідити ФНФ можуть суттєво вплинути на їх реакцію на запит. Загалом це робить «зберігання» ключа надзвичайно безпечним від атак.

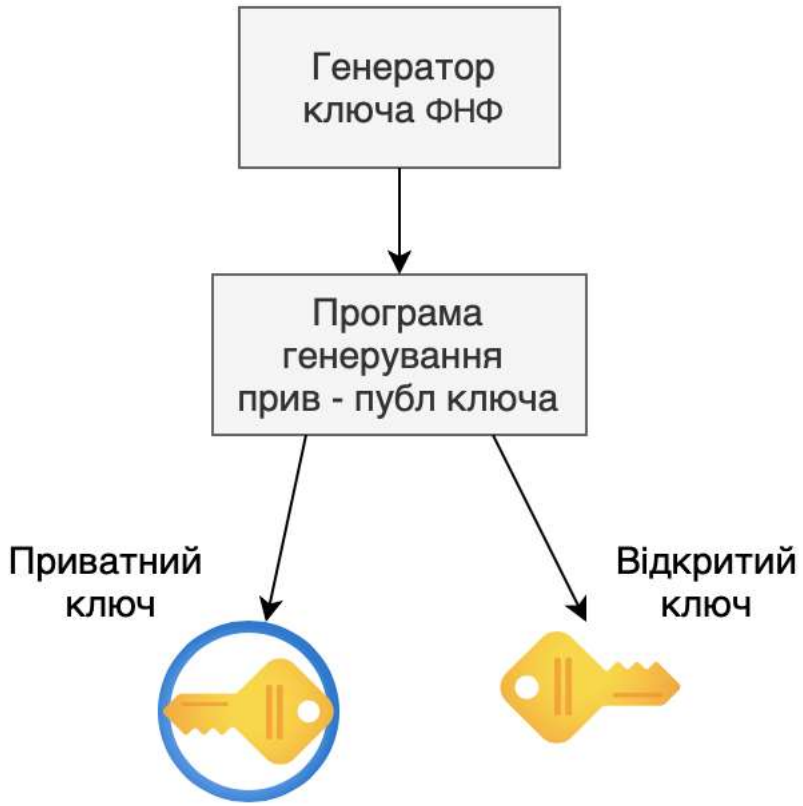
Окрім цього, ФНФ має переваги як справжнє апаратне рішення. Подібно до того, як справжній генератор випадкових чисел може створювати справді непередбачувані бітові послідовності, ФНФ можуть створювати справді непередбачуваний ідентифікатор ІС або криптографічний ключ, використовуючи справжню випадковість у природі. Це підвищує безпеку, оскільки ключі не можна передбачити на основі детермінованого або квазідетермінованого процесу.

Необхідність безпеки. ФНФ є чудовим варіантом для захисту апаратного забезпечення завдяки своїй універсальності – вони корисні для генерації та зберігання випадкових ключів, автентифікації пристроїв, генерації випадкових чисел, захисту від підробок тощо.

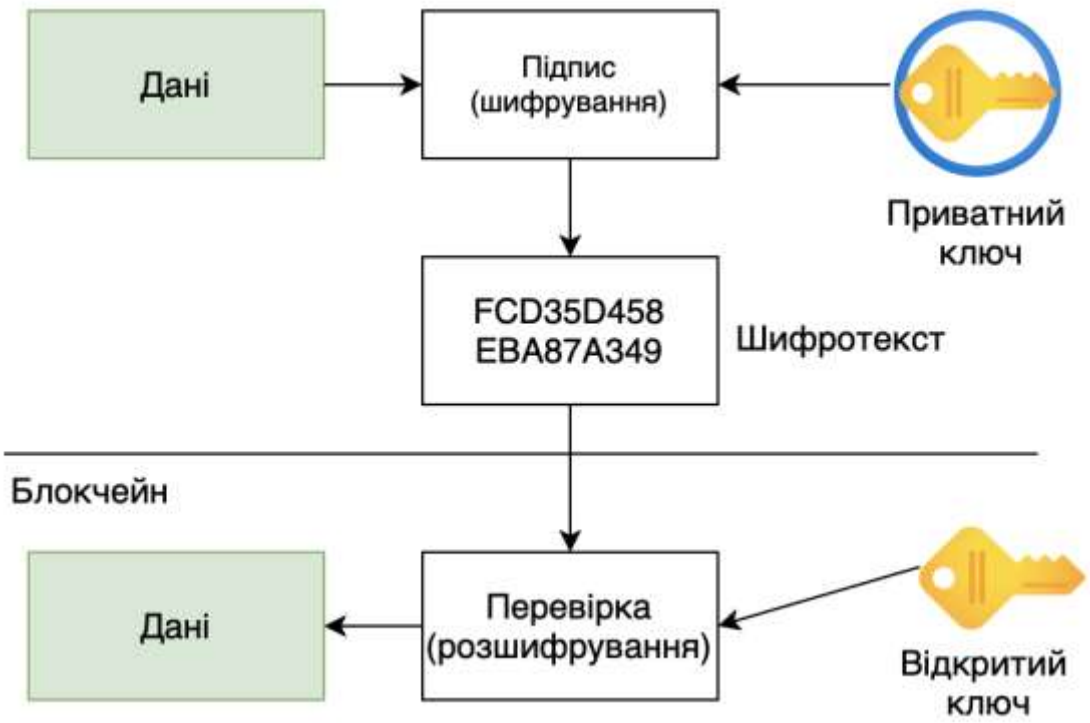
Інфраструктура відкритого ключа на базі ФНФ. В даний час ключ, отриманий з ФНФ, зазвичай представлений симетричним способом, де верифікатор, який реєструє ключ ФНФ, і пристрій ФНФ відновлюють той самий ключ (рисунок 2.7а). Це може бути громіздким, якщо інша сторона, яка не зареєструвала ключ ФНФ, хоче автентифікувати та безпечно спілкуватися з пристроєм ФНФ, тому що цей симетричний ключ потрібно безпечно надати цій стороні заздалегідь.


Генератор ключа ФНФ
Симетричний ключ
Повідомлення
Шифрування
4BE96750
03E08EC6
Шифротекст
Повідомлення
Розшифрування
Симетричний ключ

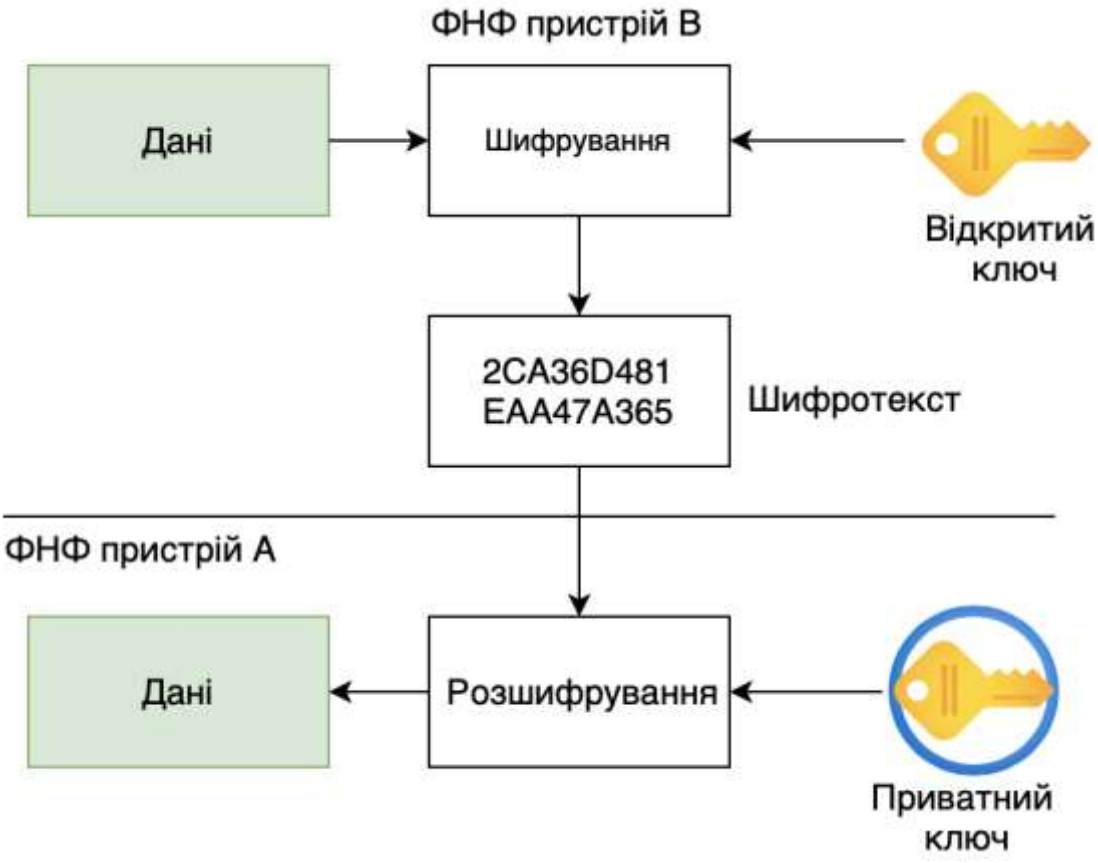
а)


Генератор ключа ФНФ
Програма генерування прив - публ ключа
Приватний ключ
Відкритий ключ

б)



в)



г)

Рисунок 2.7 – Використання ФНФ в інфраструктурі відкритих ключів:

а) Використовує ключ ФНФ як спільний симетричний ключ, який повинен бути відомий приймачу, з яким ФНФ спілкується. Розподіл цього симетричного ключа між сторонами, які мають намір спілкуватися з пристроєм ФНФ, є незручним;

б) Ключ ФНФ використовується для створення пари закритого та відкритого ключів (пара ключів priv-pub), відкритий ключ транслюється на сервер відкритих ключів, доступ до якого має будь-яка сторона. Пристрій ФНФ більше не зберігає в цифровому вигляді пару ключів priv–pub, натомість вона генерується на вимогу та вимкнена за замовчуванням;

в) Дані/транзакція, які збираються/видаються пристроєм ФНФ, підписуються для перевірки;

г) Захищений обмін повідомленнями між пристроями ФНФ, де жоден інший пристрій не може відкрити або отримати доступ до повідомлення, надісланого на пристрій ФНФ А, не маючи закритого ключа А, тоді як повідомлення зашифровано відкритим ключем А.

У цьому контексті застосування ФНФ до інфраструктури відкритих ключів стає привабливим. Зокрема, під час реєстрації ключа пристрій ФНФ генерує ключ, який можна безпосередньо використовувати як приватний ключ, інакше він використовується як випадкове початкове число для отримання приватного ключа. Потім генерується відповідний відкритий ключ на основі закритого ключа, як показано на рисунку 2.7б, і відкритий ключ транслюється на сервер відкритих ключів.

Перевага налаштування цього асиметричного ключа полягає в тому, що пара закритого та відкритого ключів прив’язана до пристрою ФНФ. Це полегшує низку застосувань. По-перше, будь-яка сторона тепер може зручно автентифікувати цей ФНФ-пристрій, надіславши nonce та прийнявши ФНФ-автентифікацію, якщо підпис приватного ключа (тобто підписаний nonce) правильно перевірено за допомогою відкритого ключа. По-друге, подібним чином дані/транзакцію можна підписати, коли вони надсилаються, і одержувач може перевірити, який є заявленим пристроєм ФНФ, як показано

на рисунку 2.7в. Враховуючи такий підпис, пристрій ФНФ не може відхилити видані/підписані дані/транзакцію. Це можна інтегрувати в систему блокчейну, де записані дані/транзакції в книзі гарантуються з авторизованих пристроїв Інтернету речей на основі підпису. Було б більш цінним, щоб відповідь ФНФ, тобто приватний ключ, був прив'язаний не лише до апаратного пристрою, але й до поведінки користувача. Отже, як користувач, так і пристрій відстежуються або автентифікуються.

Захищений зв'язок між пристроями реалізовано таким чином, що лише призначений ФНФ-пристрій здатний згенерувати закритий ключ для відкриття/дешифрування повідомлення, зашифрованого його відкритим ключем (рисунок 2.7 г). У рамках приведеної інфраструктури відкритого ключа на основі ФНФ буде цінним мінімізувати накладні витрати на реалізацію алгоритмів відкритого ключа, щоб відповідати системам з обмеженими ресурсами.

З поширенням пристроїв Інтернету речей у додатках автономних транспортних засобів, розумних будинків тощо забезпечення безпеки та довіри, збереження конфіденційності та боротьба з підробками є головними пріоритетами для виробників і споживачів. Для вирішення вищезазначених проблем були запропоновані різні надійно захищені криптографічні алгоритми, які всі покладаються на те, що ключ потрібно надійно зберігати. У цьому відношенні ФНФ генерують унікальні та непостійні безпечні ключі. Таким чином, у промисловому секторі ФНФ приділили значну увагу для вирішення проблем безпеки пристроїв IoT. Це особливо актуально, коли SRAM ФНФ починає працювати, оскільки вона є внутрішньою, не потребує модифікації апаратного забезпечення, і, отже, може бути застосована до всіх існуючих електронних пристроїв із вбудованою пам'яттю SRAM.

3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ АЛГОРИТМІВ АВТЕНТИФІКАЦІЇ ПРИСТРОЇВ ІНТЕРНЕТ РЕЧЕЙ

3.1 Схема автентифікації пристрою на основі фізично неклонованої функції

Впровадження належних методів автентифікації та безпечного зв’язку, включаючи безпечне керування ключами в пристроях з обмеженими апаратними ресурсами, має велике значення, особливо з розвитком пристроїв IoT. Коректне використання криптографії та відповідних ключів є однією з важливих задач, коли йдеться про пристрої з обмеженими обчислювальними ресурсами та низьким енергоспоживанням.

На даний час запропоновано різноманітні протоколи зв’язку для безпечної автентифікації та зв’язку для IoT, наприклад, протокол зв’язку між машинами/Інтернету речей (MQTT), протокол обмежених додатків (CoAP) або безпеку транспортного рівня датаграм (DTLS), які можуть бути інтегровані з CoAP. Однак це все ще досить важкі та обчислювально дорогі протоколи, якщо розглядати прості та обмежені пристрої. Крім того, ці протоколи не стосуються безпечної генерації та зберігання криптографічних ключів, що є скоріше необхідною умовою для їх використання.

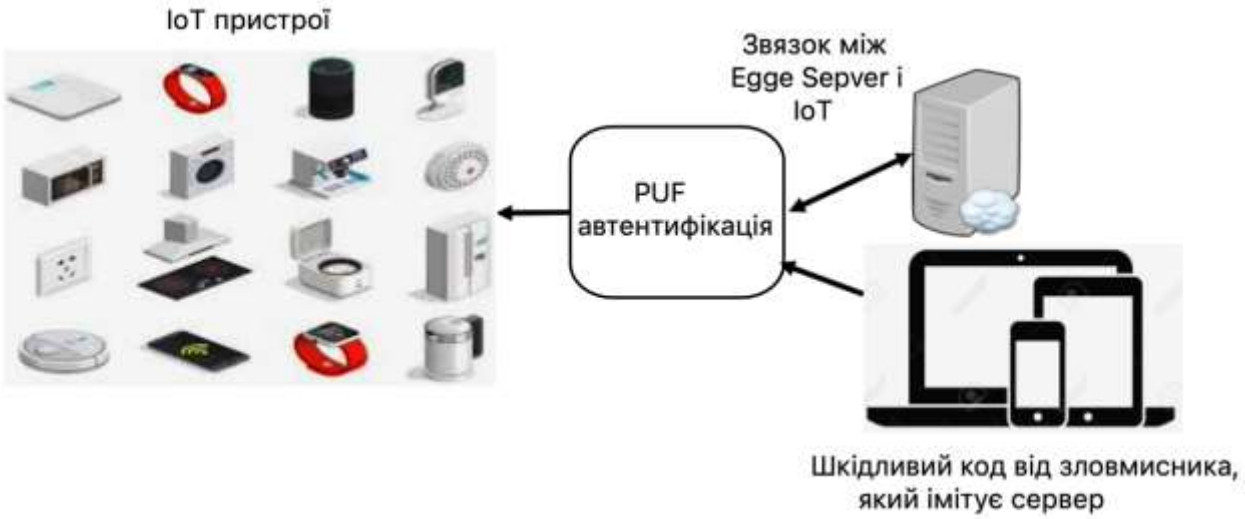
Зараз генератори випадкових чисел (ГВЧ) переважно використовуються для генерації ключів, які далі використовуються в криптографічних протоколах для автентифікації та безпечної передачі даних. Для вбудованих пристроїв зазвичай реалізуються справжні генератори випадкових чисел (True Random Number Generators, TRNG), які використовують недетерміновані ефекти в аналогових або цифрових схемах, так як це ресурсоефективний спосіб. Таким чином, якість ГВЧ має значний вплив на безпеку всієї системи. Неправильно реалізований ГВЧ часто призводить до компрометації всієї системи або знижує складність атаки.

Крім того, коли ключ згенеровано, його потрібно надійно зберігати в пристрої, наприклад використання сховища з реалізованими методами

захисту від несанкціонованого доступу. Однак реалізація таких заходів є складним і економічно неефективним завданням, тому часто нею нехтують у практичних застосуваннях. Таким чином, правильно визначене та реалізоване керування ключами, включаючи створення ключів, зберігання ключів і використання ключів для різних програм (автентифікація, контроль доступу, шифрування) у взаємопов'язаних IoT та вбудованих системах, все ще є складним завданням. Необхідний послідовний спосіб обробки різних криптографічних ключів, можливості повторного використання традиційних механізмів безпеки та забезпечення механізмів наскрізної перевірки цілісності [15]. Усі протоколи безпеки вимагають облікових даних, тому для зберігання та розповсюдження цих облікових даних повинні бути реалізовані оптимальні системи керування ключами [16].

Пристрої IoT здатні збирати дані для обробки та аналізу. Зловмисник може скористатися різними вразливими місцями для атаки на пристрої IoT, як показано на рисунку 3.1. Деякі пристрої IoT можна налаштувати за допомогою дистанційного керування або власного контролера, розробленого виробником.

а)



б)

Рисунок 3.1 – Модель загрози та рішення для пристроїв IoT:
а) модель загрози; б) рішення з ФНФ.

Зловмисник може видати себе за пульт дистанційного керування, щоб надіслати шкідливі інструкції на пристрій IoT. Дані з деяких пристроїв IoT передаються на периферійний сервер для обробки. У таких випадках зловмисник виконує атаки зворотного проектування та радіоатаки, щоб отримати доступ до даних і видавати себе за крайовий сервер і отримати неавторизований доступ до пристрою IoT.

Запропоновано схему автентифікації пристрою, яка здатна автентифікувати пристрій до того, як з нього будуть зчитані будь-які дані. Однією з головних проблем пристроїв IoT є їх низька обчислювальна потужність і обмежений обсяг пам’яті.

Модель загрози та рішення ФНФ для пристроїв IoT. Пристрої IoT мають низьку обчислювальну потужність, що робить їх нездатними запускати інтенсивні криптографічні програми. Тому розроблена схема автентифікації на основі фізично неклонованої функції. Дана конструкція здатна автентифікувати пристрої без будь-якого навантаження на процесор. Це робить її придатною для різних середовищ, включаючи медичні пристрої.

Завдяки інтеграції архітектури ФНФ з низьким енергоспоживанням енергоспоживання всієї системи можна знизити до мінімуму, що є основною необхідністю будь-якого пристрою IoT. Завдяки реалізації ФНФ у механізмі автентифікації ключі не зберігаються в пам'яті, що також зменшує вимоги до пам'яті системи. Ключ може бути згенерований за допомогою відповідного запиту в модуль ФНФ, коли це необхідно. Модуль ФНФ реалізований на FPGA і підключений до периферійного сервера та пристрою IoT для процесу автентифікації.

Розглядається сценарій, коли дані обмінюються між клієнтом IoT і сервером, а клієнт передає дані на крайовий сервер. У схемі автентифікації є два етапи: етап реєстрації та етап автентифікації. Коли пристрій спочатку вводиться в мережу, він проходить реєстрацію, і коли пристрій зареєстровано на сервері, його можна розгорнути в програмі. На етапі автентифікації пристрій перевіряється на автентичність і дані отримуються від клієнта. Запропонована схема автентифікації підходить для різних сценаріїв і додатків IoT, де дані потрібно передавати між двома різними пристроями, і вона не обмежена протоколом зв'язку, який використовується між двома пристроями. Розглянемо два етапи схеми автентифікації.

Концепція безпеки у периферійних обчисленнях на основі ФНФ показана на рисунку 3.2. Як видно з рисунку 3.2, кінцевими пристроями є електронні пристрої, які підключаються до периферійних маршрутизаторів, серверів та шлюзів залежно від потреб. Оператор матиме доступ до даних через локальну мережу, а дані з периферійних пристроїв надсилатимуться до хмарних сервісів через Інтернет.

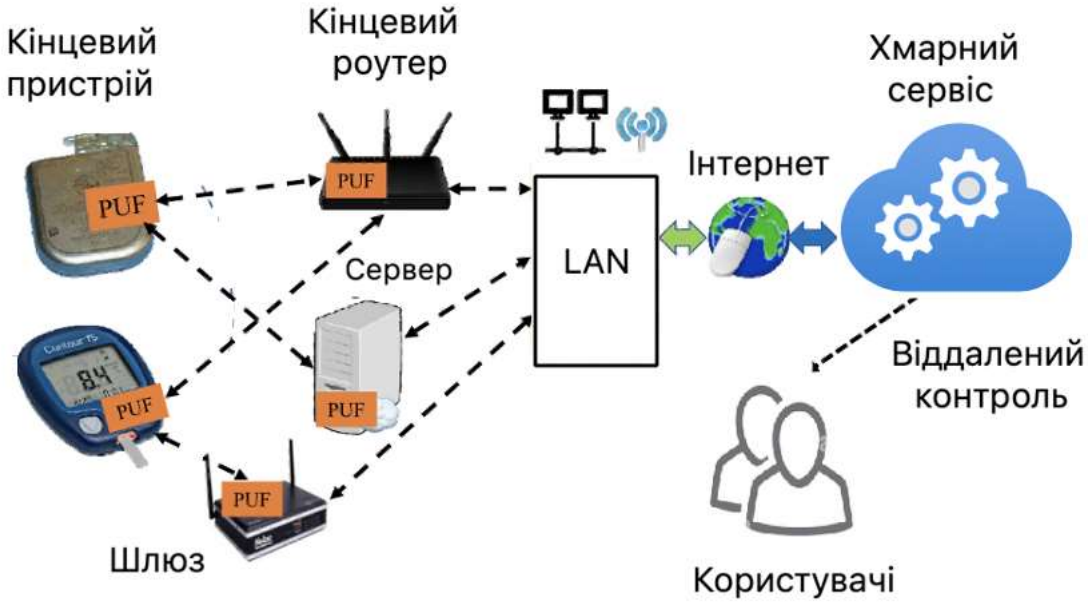


Рисунок 3.2 – Система безпека у периферійних обчисленнях та IoT на основі ФНФ

Граничні пристрої, такі як крайовий сервер, периферійний маршрутизатор і шлюз, мають вбудований модуль ФНФ, а також кінцеві пристрої.

3.2 Процедура реєстрації пристроїв на основі фізично неклонованої функції

Коли потрібно додати новий пристрій до мережі, пристрій IoT і сервер проходять етап реєстрації. Модуль ФНФ вбудовано в кожен пристрій, який додається до мережі. Передбачається, що вхідні запити безпечні, задовольняють необхідні характеристики ФНФ і доступні на етапі реєстрації.

Процес реєстрації схеми автентифікації показаний на рисунку 3.3. На сервері та в пристрої IoT є модулі ФНФ. Спочатку запит вибирається для введення в модуль ФНФ на сервері. Нехай цей вхід буде «Завдання 1 (C1)». Для цього запиту отримується відповідь, яка називається «Відповіддю 1 (R1)», як показано на рисунку 3.3.

Процес генерації відповідей на виклики в ФНФ позначається ⟩⟩. C1 і R1 отримуються на модулі ФНФ на сервері. Після генерації відповіді 1 її потрібно перевірити, чи вона задовольняє вимогам для модуля ФНФ, який присутній у пристрої IoT.

Потім отриманий R1 передається на пристрій IoT. На пристрої IoT це стає входом запиту для модуля ФНФ. Вхідні дані представлені як «Завдання» (рисунок 3.3). Для «Завдання (C)» «Відповідь (R)» отримано на пристрої IoT, тобто C ⟩⟩ R. Це діє як унікальний відбиток для пристрою IoT через характеристики ФНФ.

Потім ця відповідь від пристрою IoT передається на сервер, де вона передається як запит модулю ФНФ. Завдання представлено як «Виклик 2 (C2)» на рисунку 3.3.

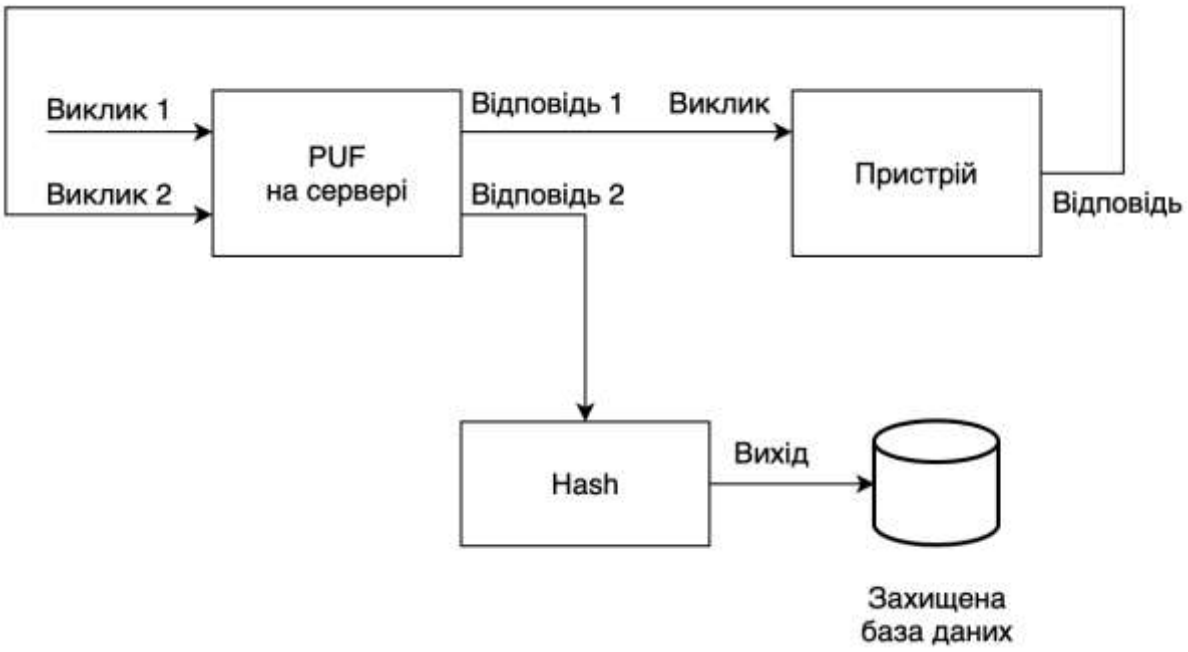


Рисунок 3.3 – Етап реєстрації

Цей запит призведе до результату, представленого як «Відповідь 2 (R2)», тобто C2 ⟩⟩ R2. Після отримання R2 обчислюється його хеш, наприклад X = H(R2). Остаточний хеш-вихід і початковий виклик (C1) зберігаються в базі даних. Процес повторюється для кількох ключів у формі

викликів і генеруються відповідні хеш-значення. Послідовність керування етапами реєстрації показано на рисунку 3.4.

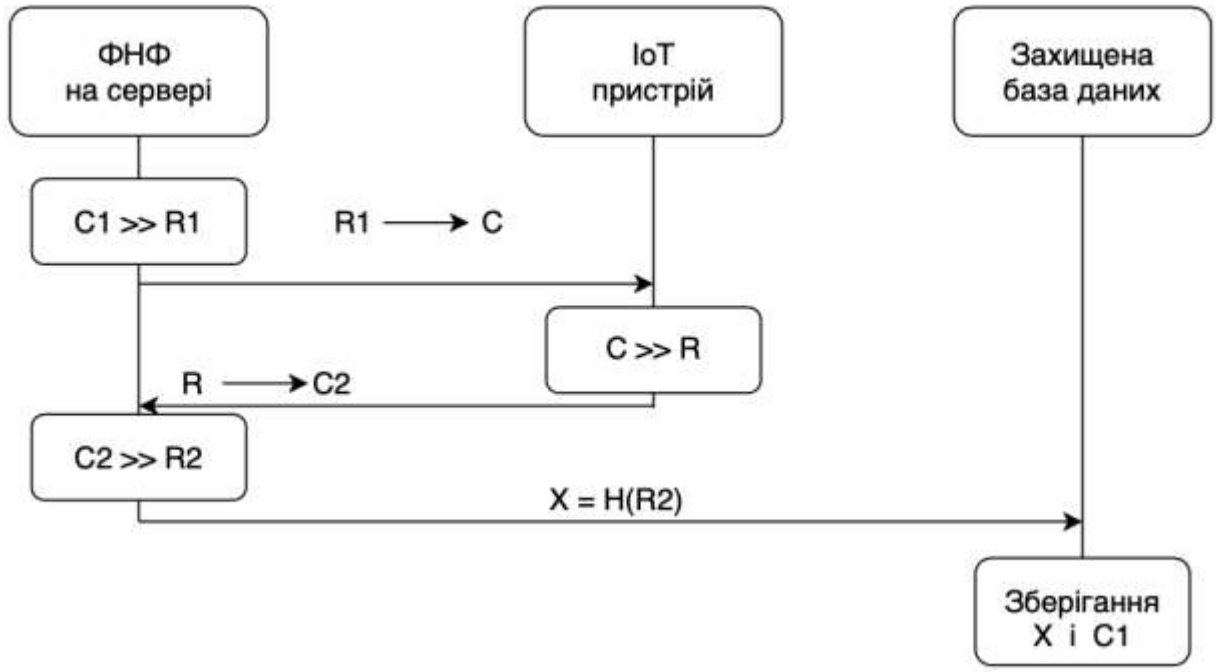


Рисунок 3.4 – Процедура реєстрації пристрою

Процес автентифікації на основі ФНФ. Коли пристрій зареєстровано на сервері, пристрій IoT можна в будь-який час автентифікувати, щоб перевірити його надійність.

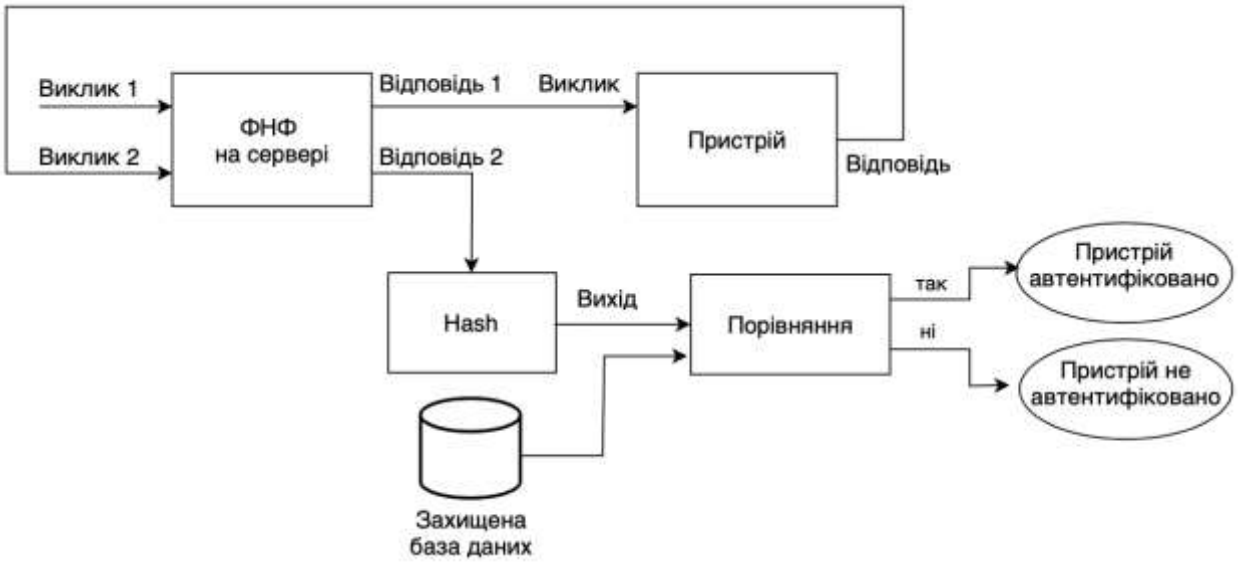


Рисунок 3.5 – Процедура автентифікації

Коли пристрій потрібно автентифікувати, база даних перевіряється, і вхідний запит (C1) передається модулю ФНФ на сервері (рисунок 3.5).

Відповідь передається з модуля ФНФ, відповідь 1 (R1′). Це надсилається на клієнтський пристрій, який потрібно автентифікувати. R1′ стає вхідним сигналом для модуля ФНФ на клієнтському пристрої. Відповідь передається з модуля ФНФ, а потім надсилається на сервер для подальшої обробки та автентифікації, як на етапі реєстрації. Це гарантує, що на клієнті не буде генерації запитів. Тоді відповідь стає запитом, введеним у модуль ФНФ на сервері. Це гарантує, що дані, які надходять від клієнта, не зберігаються безпосередньо в пам’яті або не використовуються безпосередньо для подальшої обробки. Відповідь від ФНФ надсилається до хеш-функції, а хеш обчислюється як (X′ = H(R2′)). Це порівнюється з хеш-значенням, яке зберігається в базі даних (X). Якщо X і X′ однакові, пристрій є справжнім, а якщо вони не збігаються, пристрій шкідливий. Етап автентифікації показаний у вигляді послідовності дій на діаграмі (рисунок 3.6). Повна процедура реєстрації, автентифікації та обміну ключами представлена в алгоритмі 1.

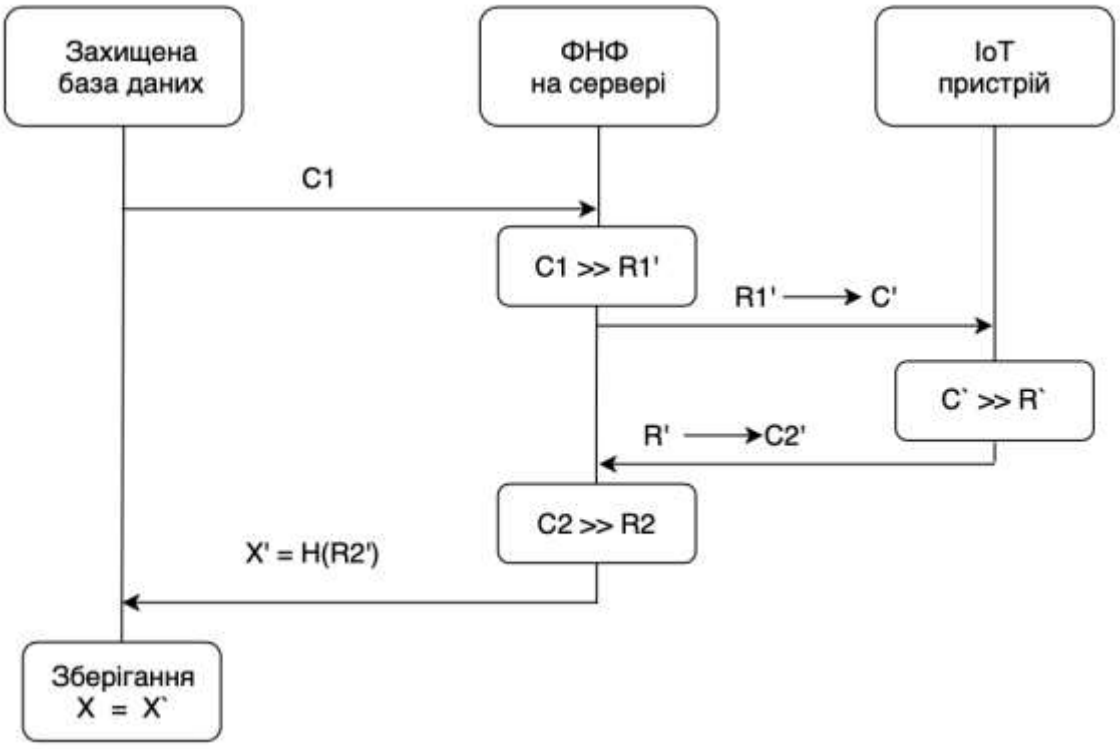


Рисунок 3.6 – Перевірка автентифікації

Алгоритм 1. Безпечний процес автентифікації.

Вхід: Виклик 1 для ФНФ на сервері (ФНФ-С) на етапі 1 та для безпечної бази даних (ЗБД) на етапі 2.

Фаза-1 (Реєстрація)

ФНФ-С → IoT {R1, тобто С} С1 ⟩⟩ R1

ФНФ-С ← IoT {R1 тобто С2} С ⟩⟩ R

ФНФ-С → ЗБД {X} X=H(R2)

Фаза-2 (Аутентифікація)

ЗБД → ФНФ-С {С1} С1 ⟩⟩ $R1'$

ФНФ-С → IoT {$R1'$ тобто C'} C' ⟩⟩ R'

ФНФ-С ← IoT {R' тобто $C2'$} $C2'$ ⟩⟩ $R2'$

ФНФ-С → ЗБД {X'} X' =H($R2'$)

якщо (X=X') тоді автентифікований

інше

Зловмисника знайдено.

3.3 Вбудований модуль для безпечної автентифікації

При розробці вбудованого модуля для безпечної автентифікації та зв’язку основною метою є спрощення керування ключами на самому вбудованому пристрої кінцевої точки. Таким чином, щоб не потрібно було зберігати секрети на апаратному пристрої пропонується використовувати як базовий будівельний блок модуля єдину схему для генерації ключів, на основі ФНФ і ГВЧ.

Загальний модуль, зображений на рисунку 3.7, забезпечує аутентифікацію ФНФ і генерацію ключів на основі ФНФ/ГВЧ. Для автентифікації використовується ФНФ, оскільки вона забезпечує випадковість, присутню в пристрої, і використовує той факт, що згенерована відповідь є унікальною для кожного пристрою. Так як існує потреба як у постійному, так і в тимчасовому ключі, у цьому випадку використовується комбінація ФНФ і ГВЧ– ФНФ використовується для генерації приватного ключа, який ніколи не залишає пристрій, таким чином використовуючи всі переваги ФНФ, тоді як ГВЧ використовується для генерації одноразових ключів, які спільно використовуються іншими сторонами, що спілкуються.

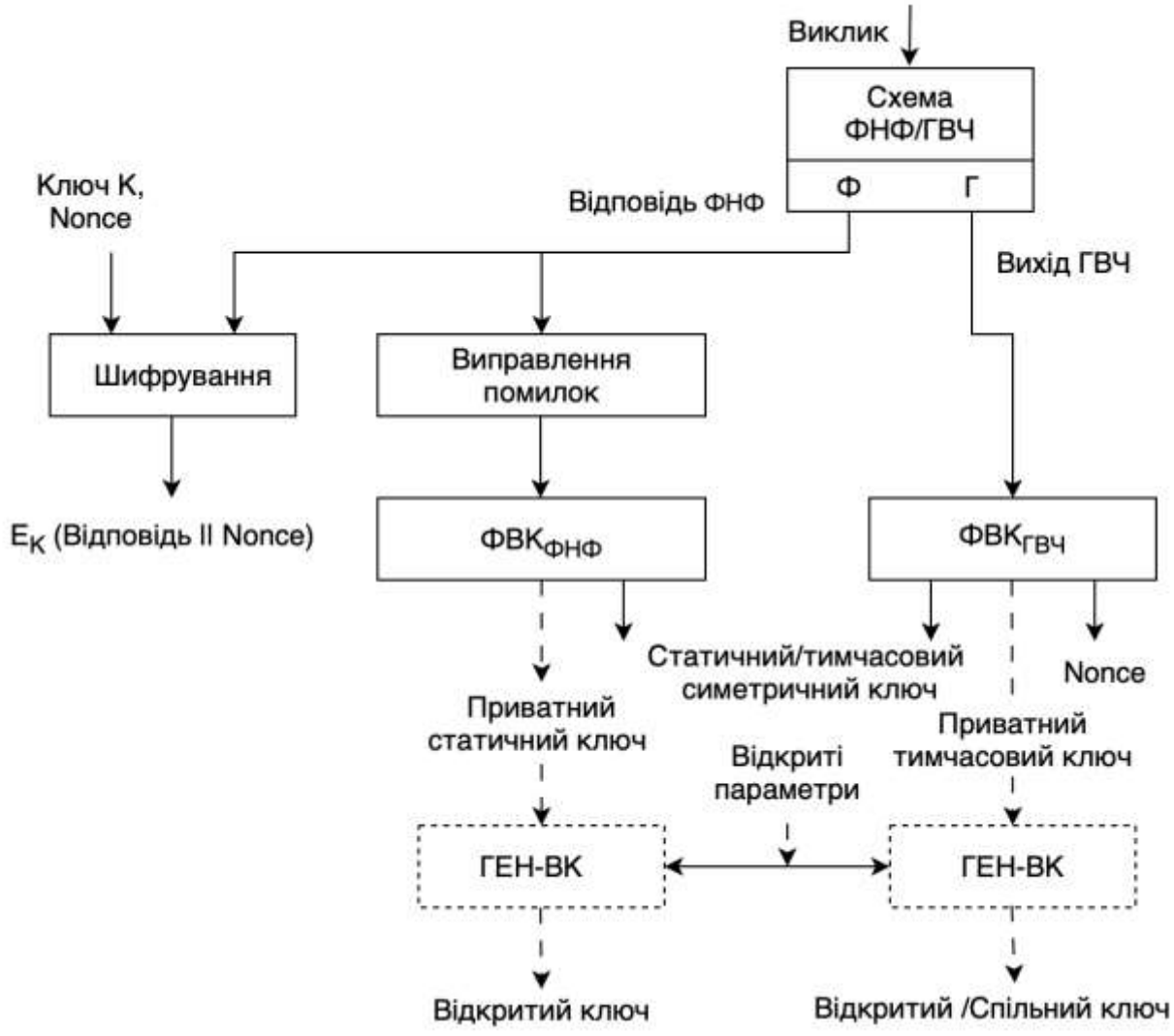


Рисунок 3.7 – Вбудований модуль для безпечної автентифікації та зв’язку: ФВК – це функція виведення ключа, ГЕН-ВК генерує відкритий ключ із закритого ключа та відкритих параметрів.

Виправлення помилок використовується для отримання стабільних ключів матеріалу з відповіді ФНФ Асиметричні схеми підходять, якщо закритий ключ легко генерується з випадкової послідовності за допомогою функції виведення ключа (ФВК). Наприклад, можна використовувати схеми шифрування ElGamal і схеми підпису DSA/ECDSA, якщо вибрано публічні параметри хорошої якості (генерація відкритого ключа з закритого ключа, позначеного як ГЕН-ВК на рисунку 3.7).

Навпаки, ключ RSA потребує більш складної обробки, включаючи безпечну генерацію простого коду. Вихід ГВЧ також використовується для генерації випадкових одноразових даних і даних заповнення.

3.3.1 Автентифікація з використанням центру автентифікації

Перед початком будь-якого зв'язку, підключений пристрій має бути автентифікований. Так як відповіді ФНФ є унікальними для кожного пристрою та є по суті випадковими, і робить ФНФ ідеальним криптографічним примітивом для автентифікації пристрою.

В роботі розроблено протокол автентифікації з використанням попередньо згенерованих пар виклик-відповідь, які можна легко реалізувати в апаратних пристроях. Цей протокол не вимагає від ФНФ великої кількості пар виклик-відповідь (його можна використовувати навіть для однієї пари виклик-відповідь).

Протокол автентифікації складається з двох етапів – етапу безпечної реєстрації та самого етапу автентифікації, і приведено в алгоритмі 1.

Етап реєстрації є критично важливим для безпеки всіх протоколів, заснованих на ФНФ, і має виконуватися в безпечному середовищі (аналогічно методам біометричної автентифікації).

Під час етапу реєстрації алгоритму 1 пара виклик/відповідь (C, R) вимірюється з цільового пристрою та надійно зберігається в центрі автентифікації (ЦА), який може бути інтегрований у шлюз або

представлений окремим пристроєм, який шлюз запитує під час процесу автентифікації. База даних DBDi пар (C, R) створюється для кожного пристрою Di. Крім того, відкритий ключ центру автентифікації (ВКЦА) попередньо встановлено на пристрої, щоб дані автентифікації могли безпечно передаватись.

Для цієї мети можна використовувати асиметричну схему Ель-Гамаля). Припускаємо, що ВКЦА захищено від несанкціонованих змін (властивістю ФНФ, що свідчитиме про втручання).

Перші 4 кроки етапу реєстрації є загальними для всіх представлених алгоритмів. База даних DBDi також використовується на етапах автентифікації алгоритмів 2 і 3.

Етап реєстрації загальний для алгоритмів 1 – 3.

Алгоритм 1. Неєстрація та автентифікація в головному центрі автентифікації

1. ЦА → D1: Запит (C1 , C2 , ...)
2. D1: R1 = ФНФ(C1), R2 = ФНФ(C2) ...
3. D1 → AA: Відповіді (R1, R2, ...)
4. ЦА: Зберігати (Ci , Ri) to DB_{D1}

Специфічний тільки для алгоритму 1

5. ЦА→ D1: Відкритий ключ $ВК_{AA}$
6. D1: Зберігати ($ВК_{AA}$)

Етап автентифікації для D1

1. ЦА: Виберіть (C, R) з DB_{D1}
2. ЦА → D1: Виклик C, одноразовий N
3. D1: R’ = ФНФ(C)
4. D1 → AA: $CR = E_{PK\_AA}(R' \| N)$
5. ЦА: (R’, N’) = D SK_ЦА(CR)

Порівняти ($R \cong R'$),

Порівняти (N = N’)

Етап автентифікації згідно алгоритму 1 виконується кожного разу, коли пристрій підключається до мережі та потребує автентифікації. ЦА випадковим чином вибирає один із викликів C і надсилає його разом із значенням nonce N на пристрій для автентифікації. Значення nonce використовується для запобігання простим повторним атакам і дозволяє повторно використовувати кожну пару виклик-відповідь. На пристрої, який проходить автентифікацію, генерується відповідна відповідь ФНФ, об'єднана зі значенням nonce і зашифрована відкритим ключем центру автентифікації.

Потім орган автентифікації порівнює (строго), чи розшифроване значення nonce N' = N.

Оскільки відповідь ФНФ може незначно відрізнятися в різних вимірюваннях, допускається попередньо визначена кількість помилкових бітів у R'. Якщо обидва збігаються, пристрій успішно автентифіковано.

Недоліком алгоритму 1 є те, що він виконує лише автентифікацію та не надає криптографічного ключа для майбутнього зв'язку.

Автентифікація одного пристрою (D1) для ЦА без асиметричної криптографії приведена в алгоритмі 2. Фаза реєстрації така сама, як і в алгоритмі 1, (кроки 1 – 4).

Цей метод включає генерацію спільного симетричного ключа K, для якого потрібен стабільний вихід ФНФ без помилок. Це досягається за допомогою коду з виправленням помилок (ECC), позначеного в алгоритмі його функціями EnC і DeC.

Цей код повинен мати достатню надлишковість і структуру, щоб виправити максимальну кількість помилок, які допускаються в ФНФ під час експлуатації в різних умовах (напруга, температура тощо).

Вибір відповідного коду ECC залежить від частоти бітових помилок і довжини відповіді ФНФ при дотриманні необхідної довжини вихідного сигналу. Обчислювальна потужність пристрою також є обмежуючим фактором. У випадку пристроїв з низькою швидкодією рекомендується використовувати прості коди (наприклад, код повторення).

Етап автентифікації – з використанням симетричного шифру.

Алгоритм 2. Аутентифікація пристрою D1 на ЦА

1. D1 → АА: Call(D1)
2. АА: r = ГВЧ()
3. Виберіть (C, R) з DBD1
4. H = R ⊕ EnCo(r)
5. K = ФВК(r)
6. АА → D1: Виклик C, допоміжний рядок H
7. D1: R’ = ФНФ(C)
8. r = DeC(R’ ⊕ H)
9. K = ФВК(r)
10. D1 ↔ АА: Автентифікація + Шифрування з K

Допоміжний рядок H – це відстань від необробленої відповіді ФНФ R до випадкового кодового слова EnC(r).

Його обчислює АА (крок 4 алгоритму 2). Потім пристрій використовує його для відновлення ключа (крок 8), а потім виводить ключ K.

Спільний ключ K можна використовувати для автентифікації та зашифрованого зв’язку, на відміну від алгоритму 1, який охоплює лише автентифікацію. З іншого боку, алгоритм 1 не вимагає генерації допоміжного рядка, а також не потребує кодів виправлення помилок.

3.3.2 Взаємна автентифікація пристрою

При підключенні до мережі пристрій має пройти не тільки автентифікацію в центральному органі управління, але й взаємну автентифікацію пристроїв, перш ніж вони почнуть обмінюватися даними. Так само, як і в попередньому випадку, центральний орган автентифікації зберігає попередньо згенеровану пару виклик-відповідь і діє як довірена третя сторона.

Однак цього разу між двома пристроями встановлюється загальний симетричний ключ, а потім відбувається звичайний симетричний автентифікований і зашифрований сеанс. Мета полягає в тому, щоб використовувати ФНФ в обох пристроях D1 і D2, але не передавати жодної відповіді ФНФ через мережу.

Використовуючи односпрямованість застосованих хеш-функцій, жоден пристрій не дізнається відповідь ФНФ іншого пристрою, навіть якщо він відстежує весь зв'язок. Для забезпечення стабільних виходів ФНФ використовується код виправлення помилок. Кодові слова вибираються випадковим чином з кодового простору ЦА. Загальний процес описано в алгоритмі 3.

Взаємна автентифікація D1 і D2 за допомогою ЦА. Взаємна автентифікація пристрою та захист повідомлень (алгоритм 3):

1. D1 → AA: Запит (D1, D2)
2. ЦА: rD1 = ГВЧ()
3. rD2 = ГВЧ()
4. Вибрати (CD1 , RD1) з DBD1
5. Вибрати (CD2 , RD2) з DBD2
6. HD1 = RD1 ⊕ EnC (rD1)
7. HD2 = RD2 ⊕ EnC (rD2)
8. r = Hash (rD1) ⊕ Hash (rD2)
9. ЦА → D1: (CD1 , HD1, r)
10. ЦА → D2: Запит (D1, D2) , (CD2 , HD2 , r)
11. D1: R'D1 = ФНФ(CD1)
12. rD1 = DeC (R'D1 ⊕ HD1)
13. Hash (rD2) = Hash (rD1) ⊕ r
14. K = ФВК (Hash (rD1) || Hash (rD2))
15. D2: R'D2 = ФНФ(CD2)
16. rD2 = DeC (R' D2 ⊕ HD2)
17. Hash (rD1) = Hash (rD2) ⊕ r

18. K = ФВК(Hash (rD1) || Hash (rD2))

19. D1 ↔ D2: Автентифікація + Шифрування з K

Припустимо, що D1 хоче пройти автентифікацію за допомогою D2 і встановити безпечний канал зв'язку. D1 ініціює процес, викликаючи ЦА з ідентифікацією D1 і D2 (Виклик (D1, D2)). ЦА містить повну таблицю викликів і відповідей (CD1, RD1 тощо). Вибирається код виправлення помилок, який може виправити достатню кількість помилок, щоб зробити відповідь ФНФ стабільною, за допомогою відповідних функцій EnC та DeC. ЦА генерує два випадкових компоненти rD1, rD2 з набору прообразів і кодує їх, утворюючи таким чином випадково вибрані кодові слова. Довжина коду повинна відповідати довжині відповіді ФНФ. Допоміжні рядки HD1 і HD2 створюються методом XOR очікуваної відповіді ФНФ (RD1, RD2) до відповідного кодового слова. Два випадкових компоненти хешуються, а хеші обробляються XOR, щоб сформувати r.

На кожен із пристроїв надсилається триплет (CDi, HDi, r) із викликом, допоміжним рядком і r. Крім того, на кроці 10 ЦА ретранслює запит на зв'язок від D1 до D2. Кожен із пристроїв викликає власну ФНФ, щоб отримати відповідь (R'D1, R'D2). Шляхом XOR відповіді з відповідним допоміжним рядком (HD1, HD2), що призводить до кодового слова з помилками, яке потім виправляється функцією декодування. Таким чином, кожен пристрій відновлює свій компонент (rD1, rD2). D1 відновлює значення Hash(rD2) шляхом XOR r з хешем його rD1 і навпаки. Крім того, обидва пристрої знають хеші rD1 irD2 і можуть отримати спільний ключ K шляхом застосування функції виведення ключа ФВК до об'єднання хешів.

Хешування rD1, rD2 виконується, щоб приховати відповіді ФНФ від іншого пристрою. Якщо D1 стежить за зв'язком, він знатиме (CD1, CD2, HD1, HD2, r).

Він може відновити rD1, і якби хешування не було зроблено, і r дорівнював би rD1 ⊕ rD2 напряму, D1 обчислив би rD2 і використовуючи

допоміжний рядок HD2, він міг би виявити відповідь ФНФ RD2. Нам доведеться або довіряти всім пристроям у мережі, або використати всі завдання лише один раз і відкинути їх. Оскільки в даному пристрої використовується хешування rD1, rD2, D1 отримує лише Hash(rD2), а односторонність хеш-функції не дозволяє їй виявити RD2. Таким чином, можна повторно використовувати виклики для наступних автентифікацій.

Вибір коду виправлення помилок у відповіді ФНФ залежить від кількості бітів отриманих від операції ФНФ. Довжина коду та відстань кодового слова визначають кількість інформаційних бітів, отже, довжину rD1, rD2 та обмежують ентропію, що міститься в r. Використовуючи той самий виклик із кількома випадковими rDi, ми можемо отримати більше бітів ентропії з ФНФ. Ентропія результуючого спільного ключа K визначається властивостями використаних хеш-функцій і ФВК, а також вхідними даними. Якщо вибрано правильно, воно дорівнює ентропії rD1, rD2. Ключ K завжди отримується з випадково вибраних кодових слів, і тому для тих самих викликів ФНФ (CD1, CD2) виходить інший K.

Захищений зв’язок. Після процесу автентифікації, описаного в попередньому пункті, встановлюється спільний ключ. На цьому етапі здійснюється звичайна симетрична автентифікація та процес отримання сеансового ключа з використанням блокового шифру AES. В [21] було запропоновано кілька легких блочних шифрів, придатних для вбудованих систем або сенсорних мереж, таких як PRESENT () з 80-бітним ключем. Це дозволяє згенерувати ключ за один цикл схеми ФНФ для більшості конструкцій і реалізацій ФНФ.

Усі представлені алгоритми використовують лише ФНФ на стороні пристроїв, а ГВЧ використовується на АА. Функція ГВЧ на пристроях використовується після встановлення захищеного каналу (кроки 10 – 19) в залежності від протоколів зв'язку.

Як випливає з попереднього пункту, функції ГВЧ і ФНФ мають різні характеристики, які мають перевагами в різних застосуваннях. Таким чином,

різні реалізації криптографічних систем можуть використовувати переваги універсальної схеми для генерації ФНФ і ГВЧ одночасно, що дозволяє безпечно генерувати симетричні (сеансові) ключі (і потенційно також приватні ключі). Щоб перевірити запропонований процес автентифікації, проведено моделювання на емуляторі пристрою що містить конструкцію РОФНФ. Для цього використано конструкцію РОФНФ, яка складалася з 2 груп кільцевих осциляторів (RO), кожна група містила 150 RO. Тільки RO з різних груп були обрані для формування пари, яка потім використовувалася для генерації частини відповіді ФНФ (рисунок 3.8). Отримали 3 біти з кожної пари RO та підвищили стабільність виходу ФНФ шляхом застосування коду Грея до цих бітів.

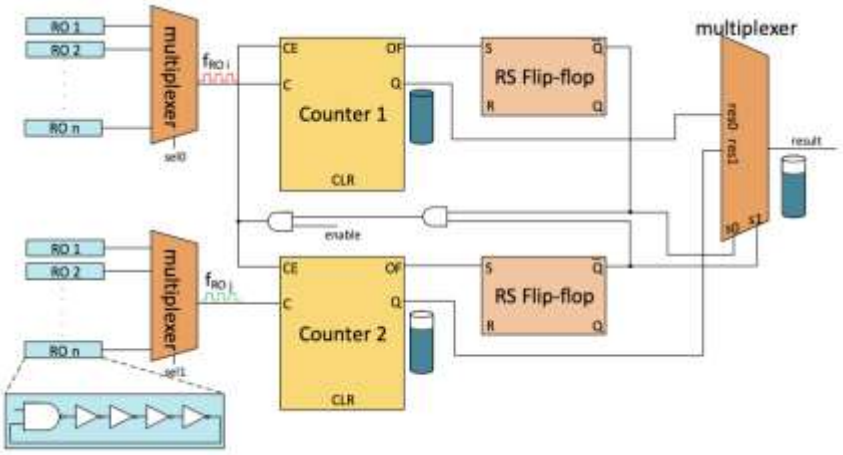


Рисунок 3.8 – Схема ФНФ на основі кільцевих осциляторів, яка служить базовим блоком для схеми автентифікації та безпечного зв'язку [30].

Нарешті, щоб створити відповідь ФНФ, вибрані біти з усіх пар RO об'єднуються. У першому випадку генерували відповіді ФНФ від 120 пар RO (кожен RO з кожної групи використовувався лише один раз), в іншому випадку кожен RO використовувався п'ять разів (один RO з першої групи поєднується з 5 RO з інша група), що призводить до 600 пар RO.

Ці дві установки досягли 450 і 2250 біт відповіді ФНФ відповідно. В обох випадках проведено 100 вимірювань, з яких отримано відповідь

більшості ФНФ – RDi. У даному експерименті довжина блоку в 9 біт виявилася достатньою для коду повторення. Для того, щоб створити допоміжний рядок HDi, потрібно згенерувати 50 або 250 випадкових бітів (rDi), які потім кодуються кодом повторення та XOR з основним виходом ФНФ, утворюючи допоміжний рядок HDi. Цей процес пов'язаний з кроками 2 і 4 в алгоритмі 2.

Щоб збільшити кількість бітів після виправлення, ми можемо або використати більш ефективний код для виправлення помилок, або ми можемо повторно використовувати те саме завдання кілька разів із новим випадковим кодовим словом щоразу. Експеримент показав, що можна генерувати ключі для протоколів передачі даних, використовуючи конструкції ФНФ/ГВЧ, достатньої довжини.

На пристрої ФНФ генерує відповідь R'Di, яка коригується допоміжним рядком HDi, що відповідає крокам 7 і 8 в алгоритмі 2. Після корекції отримали 50 і 250 біт відповідно. Ці біти можна використовувати для створення криптографічного ключа. Для алгоритму 2 можна представити ФВК як вибір перших 128 бітів (з rDi) для симетричного шифру AES. Те саме можна застосувати до алгоритму 3, де два пристрої аутентифікують один одного. Однак цей алгоритм більш складний, оскільки вимагає реалізації відповідної хеш-функції. В цьому випадку ФВК не потрібна Алгоритму 1, оскільки відкритий ключ АА зберігається на пристрої, а ФНФ не використовується для отримання будь-якого криптографічного ключа.

ВИСНОВКИ

В кваліфікаційній роботі розв’язано актуальну задачу підвищення ефективності алгоритмів захисту даних в мережах Інтернет- речей. При цьому отримано наступні результати.

1. Проведено аналіз стану конфіденційності та безпеки даних IoT. Визначено обмеження безпеки пристроїв IoT, зокрема, апаратні обмеження, обмеження програмного забезпечення, обмеження мережі. Сформовано вимоги до безпеки IoT.

2. Розроблено класифікацію атак безпеки на різних рівнях мереж Інтернет-рече. Виділено чотири категорії атаки на IoT: фізичні атаки або атаки сприйняття; мережеві атаки, атаки на програмне забезпечення або програми та атаки на шифрування.

3. Показано можливість використання фізично неклонованої функції в інфраструктурі відкритих ключів. Зокрема, під час реєстрації ключа пристрій ФНФ генерує ключ, який безпосередньо використовується як приватний ключ, в іншому випадку він використовується як випадкове початкове число для отримання приватного ключа.

4. Розроблено алгоритм реєстрації пристроїв на основі фізичної неклонованої функції. Даний алгоритм складається з двох етапів: етапу реєстрації та етапу автентифікації.

5. Розроблено модуль для безпечної автентифікації пристроїв Інтернет-речей, який забезпечує аутентифікацію і генерацію ключів на основі ФНФ. Для автентифікації використовується ФНФ, оскільки вона забезпечує випадковість, присутню в пристрої, і використовує той факт, що згенерована відповідь є унікальною для кожного пристрою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Asghari, P., Rahmani, A. M., & Javadi, H. H. S. Internet of Things applications: A systematic review. Computer Networks, 2019, 148, 2, pp. 41-261.

2. Bandyopadhyay, D., & Sen, J. Internet of things: Applications and challenges in technology and standardization. Wireless personal communications, 2011, 58, pp. 49-69.

3. Zhang, L., Dabipi, I. K., & Brown Jr, W. L. Internet of Things applications for agriculture. Internet of things A to Z: technologies and applications, 2018, pp. 507– 528.

4. Li, X., Lu, R., Liang, X., Shen, X., Chen, J., & Lin, X.. Smart community: an internet of things application. *IEEE Communications magazine*, *49*(11), 2011, pp. 68-75.

5. Kaur, S., & Singh, I. (2016). A survey report on Internet of Things applications. *International Journal of Computer Science Trends and Technology*, *4*(2), 2016, pp. 330-335.

6. Kabalci, Y., Kabalci, E., Padmanaban, S., Holm-Nielsen, J. B., & Blaabjerg, F. Internet of things applications as energy internet in smart grids and smart environments. Electronics,2019, 8(9), 972.

7. Ni, J., Zhang, K., Lin, X., & Shen, X. Securing fog computing for internet of things applications: Challenges and solutions. IEEE Communications Surveys & Tutorials, 20(1), 2017, 601-628.

8. Soldatos, J., Gusmeroli, S., Malo, P., & Di Orio, G. Internet of things applications in future manufacturing. In Digitising the Industry Internet of Things Connecting the Physical, Digital and VirtualWorlds, 2022, pp. 153-183.

9. Nalbandian, S. A survey on Internet of Things: Applications and challenges. In 2015 International Congress on Technology, Communication and Knowledge (ICTCK), 2015, pp. 165-169.

10. Mezzanotte, P., Palazzi, V., Alimenti, F., & Roselli, L. Innovative RFID sensors for Internet of Things applications. IEEE Journal of Microwaves, 1(1), 2021, pp.55-65.

11. Butun, I., Österberg, P., & Song, H. Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1), 2019, pp. 616-644.

12. Li, X., Xu, M., Vijayakumar, P., Kumar, N., & Liu, X. (2020). Detection of low-frequency and multi-stage attacks in industrial internet of things. IEEE Transactions on Vehicular Technology, 69(8), 8820-8831.

13. Abosata, N., Al-Rubaye, S., Inalhan, G., & Emmanouilidis, C. (2021). Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. Sensors, 21(11), 3654.

14. Khanam, S., Ahmedy, I. B., Idris, M. Y. I., Jaward, M. H., & Sabri, A. Q. B. M. (2020). A survey of security challenges, attacks taxonomy and advanced countermeasures in the internet of things. IEEE access, 8, 219709-219743.

15. Ghazal, T. M., Afifi, M. A. M., Kalra, D. Security vulnerabilities, attacks, threats and the proposed countermeasures for the Internet of Things applications. Solid State Technology, 2020, 63(1s).

16. Smys, S., Basar, A., & Wang, H. (2020). Hybrid intrusion detection system for internet of things (IoT). Journal of ISMAC, 2(04), pp. 190-199.

17. Rajendran, G., Nivash, R. R., Parthy, P. P., & Balamurugan, S. (2019, October). Modern security threats in the Internet of Things (IoT): Attacks and Countermeasures. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1-6.

18. Siddiqui, S. T., Alam, S., Ahmad, R., & Shuaib, M. Security threats, attacks, and possible countermeasures in internet of things. In *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*, 2020, pp. 35-46

19. DeepCover Cryptographic Coprocessor with ChipDNA&. Електронний ресурс]. - Режим доступу: https://www.analog.com/media/en/technical-documentation/data-sheets/ds28s60.pdf

20. Sutar, S., Raha, A., Kulkarni, D., Shorey, R., Tew, J., & Raghunathan, V. (2017). D-PUF: An intrinsically reconfigurable DRAM PUF for device authentication and random number generation. ACM Transactions on Embedded Computing Systems (TECS), 17(1), pp.1-31.

21. Ning, H., Farha, F., Ullah, A., & Mao, L. (2020). Physical unclonable function: Architectures, applications and challenges for dependable security. *IET Circuits, Devices & Systems*, *14*(4), 407-424.

22. Gao, Y., Al-Sarawi, S. F., & Abbott, D. (2020). Physical unclonable functions. *Nature Electronics*, *3*(2), pp.81-91.

23. Zhang, J., & Qu, G. (2019). Physical unclonable function-based key sharing via machine learning for IoT security. *IEEE Transactions on Industrial Electronics*, *67*(8), pp.7025-7033.

24. Banerjee, S., Odelu, V., Das, A. K., Chattopadhyay, S., Rodrigues, J. J., & Park, Y. (2019). Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable functions. IEEE Access, 7, pp.85627-85644.

25. Vijay, V., Chaitanya, K., Pittala, C. S., Susmitha, S. S., Tanusha, J., Venkateshwarlu, S. C., & Vallabhuni, R. R. (2022). Physically unclonable functions using two-level finite state machine. Journal of VLSI circuits and systems, 4(01), 33-41.

26. Alkatheiri, M. S., Sangi, A. R., & Anamalamudi, S. (2020). Physical unclonable function (PUF)-based security in Internet of Things (IoT): Key challenges and solutions. Handbook of Computer Networks and Cyber Security: Principles and Paradigms, pp.461-473.

27. Yu, S., & Park, Y. (2022). A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. IEEE Internet of Things Journal, 9(20), pp.20214-20228.

28. Jiang, Q., Zhang, X., Zhang, N., Tian, Y., Ma, X., & Ma, J. (2021). Three-factor authentication protocol using physical unclonable function for IoV. Computer Communications, 173, pp.45-55.

29. Byun, J. W. (2019). End-to-end authenticated key exchange based on different physical unclonable functions. IEEE Access, 7, 102951-102965.

30. Buchovecká, S., Lórencz, R., Kodýtek, F., & Buček, J. (2017). True random number generator based on ring oscillator PUF circuit. Microprocessors and Microsystems, 53, pp. 33-41.

31. Голод Ю.В., Гарматюк В.Р., Волос І.П. Мережеві атаки на інтернет-речей. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 39-42.

32. Дзівак О.А., Мачуляк М.В., Волос І.П. Фізичні атаки на мережі інтернет-речей. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп’ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С. 100-102.

ДОДАТОК А

Копії публікацій