Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)

Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
магістр
(ступінь вищої освіти)

на тему Оцінка безпеки протоколів зв'язку в Інтернеті речей
(назва теми)

Виконав(ла): студент(ка) IIм курсу,
групи БКз-813м
Спеціальності 125 Кібербезпека та захист інформації
(код і найменування спеціальності)
Освітня програма (спеціалізація)
Безпека інформаційних і комунікаційних систем
ПЕРВАШОВА Л.А.
(ПРІЗВИЩЕ та ініціали)

Керівник КОРОТУН А.В.
(ПРІЗВИЩЕ та ініціали)

Рецензент ЛИТВИЦЬКИЙ О.П.
(ПРІЗВИЩЕ та ініціали)

2024

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
Кафедра інформаційної безпеки та наноелектроніки
Ступінь вищої освіти магістр
Спеціальність 125 Кібербезпека та захист інформації
(код і найменування)
Освітня програма (спеціалізація) Безпека інформаційних і комунікаційних систем
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ІБтаН
Андрій КОРОТУН
«____» _____2024року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПЕРВАШОВОЇ Лілії Анатоліївни
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Оцінка безпеки протоколів зв'язку в Інтернеті речей
Assessment of communication protocols security in the Internet of Things

керівник проєкту (роботи) к.ф.-м.н., доцент КОРОТУН Андрій Віталійович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від «05» грудня 2024 року №507
2. Строк подання студентом проєкту (роботи) 10.12.2024
3. Вихідні дані до проєкту (роботи) загрози безпеки для користувачів IoT, протоколи безпеки IoT

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Загальні уявлення про IoT; проблеми безпеки та безпечні протоколи зв'язку IoT, аналіз продуктивності пртоколів безпеки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Презентація доповіді (в MS PowerPoint), 12 слайдів.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1 – 3	КОРОТУН А.В., завідувач кафедри ІБтаН	02.09.2024	05.12.2024
Нормоконтроль	КОРОЛЬКОВ Р. Ю., доцент кафедри ІБтаН		09.12.2024

7. Дата видачі завдання «02» вересня 2024року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1.	Збір та аналіз інформації про загрози безпеки для користувачів IoT	02.09.24 – 16.09.24	виконано
2.	Систематизація літературних даних	17.09.24 – 23.09.24	виконано
3.	Складання і затвердження наукового завдання	24.09.24 – 29.09.24	виконано
4.	Формування та уточнення наукового завдання	30.09.24 – 05.10.24	виконано
5.	Планування архітектури VLAN	06.10.24 – 15.10.24	виконано
6.	Складання оглядупротокрлів безпеки IoT	16.10.24 – 31.10.24	виконано
7.	Одержання результатів	01.11.24 – 13.11.24	виконано
8.	Оформлення графічної частини	14.11.24 – 19.11.24	виконано
9.	Оформлення ПЗ	20.11.24 – 30.11.24	виконано

Студент(ка) _____ Лілія ПЕРВАШОВА
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ Андрій КОРОТУН
(підпис) (Ім'я ПРІЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до магістерської роботи: 96 с., 9 табл., 23 рис., 111 джерел.

ЗАГРОЗИ БЕЗПЕКИ, ІНТЕРНЕТ РЕЧЕЙ, КОНФІДЕНЦІЙНІСТЬ, ПРОПУСКНА ЗДАТНІСТЬ, ПРОТОКОЛИ БЕЗПЕКИ, ЦІЛІСНІСТЬ

Мета роботи: дослідити загрози безпеки та можливості запобігання загрозам для користувачів IoT, які можуть бути підключені до критичних систем.

Об’єкт та предмет дослідження: об’єктом дослідження є загрози безпеки в IoT; предметом дослідження є протоколи безпеки IoT.

Методи дослідження: описово-аналітичний.

Результати: результатом дослідження є аналіз алгоритмів, які забезпечують найкращу ефективність передачі даних в IoT.

Рекомендації щодо впровадження: робота носить прикладний характер, деякі її результати можуть бути для розширення можливостей запобігання загрозам безпеки в IoT.

Практична цінність: досліджено підходи до забезпечення безпеки в IoT на основі VLAN.

ABSTRACT

Explanatory note to the master's thesis: 96 p., 9 table, 23 figure, 111 sources.

SECURITY THREATS, INTERNET OF THINGS, CONFIDENTIALITY, BANDWIDTH, SECURITY PROTOCOLS, INTEGRITY

The purpose of the work: to investigate security threats and possibilities for preventing threats for IoT users who may be connected to critical systems.

Object and subject of research: the object of the study is security threats in IoT; the subject of the study is IoT security protocols.

Research methods: descriptive-analytical.

Results: the result of the study is an analysis of algorithms that ensure the best efficiency of data transmission in IoT.

Recommendations for implementation: the work is of an applied nature, some of its results may be for expanding the possibilities for preventing security threats in IoT.

Practical value: approaches to ensuring security in IoT based on VLANs have been investigated.

ЗМІСТ

ВСТУП

Під терміном «Інтернет речей», скорочено від IoT, мають на увазі незліченну кількість матеріальних пристроїв по всьому світу, які можна підключити до Інтернету. Усі ці пристрої збирають і обмінюються даними один з одним, одночасно усуваючи потребу в спілкуванні «людина-людина» або навіть «людина-комп'ютер». Завдяки появі комп'ютерних мікросхем надзвичайно низької вартості, а також тому факту, що бездротові мережі є повсюдним, і, крім того, розвиток обчислювальних технологій, таких як машинне навчання, аналіз великих даних, інтелектуальні датчики та особливо 5G, стало правдоподібним перетворювати будь-що, незалежно від його розміру, до частини IoT.

Незважаючи на те, що багато пристроїв можуть підключатися до Інтернету, ми визначаємо IoT-пристрої як ті, які зазвичай не мають доступу до Інтернету, зокрема побутова техніка, пристрої для моніторингу здоров'я чи будь-яке обладнання, і які, в той же час, мають здатність взаємодіяти один з одним без участі людини. Надалі ані ноутбук, ані смартфон не вважаються пристроями IoT, незалежно від того факту, що вони мають датчики та спілкуються через Інтернет. Однак пристроями IoT можна вважати розумні годинники або фітнес-трекери. Тим не менш, персональний комп'ютер або смартфон можуть взаємодіяти з мережею IoT.

Це поняття збільшило кількість сфер застосування IoT, що, у свою чергу, може покращити загальний добробут, використовуючи вже доступні пристрої способами, про які раніше не думали, і це вважається однією з найважливіших сфер технологій майбутнього. Окрім ефективності та точності, взаємозв'язок пристроїв Інтернету речей створює низку загроз безпеці для користувачів, які можуть бути підключені до критичних систем.

Проблеми безпеки повинні бути пріоритетними, щоб мінімізувати поверхню атаки та запобігти проблемам безпеки, оскільки технологія IoT

призначена для використання в багатьох критичних секторах, зокрема в економіці та національній безпеці, з різними галузевими стандартами та специфікаціями. На додаток до кібератак, створення великомасштабних гетерогенних мереж, що складаються з обмежених вузлів, що працюють у режимі реального часу, має базуватися на архітектурі, яка може впоратися з такими факторами, як надійність, якість обслуговування, модульність, семантична сумісність, управління конфіденційністю та сумісність апаратного та програмного забезпечення.

Тому в роботі розглянуто такі актуальні питання як загальна архітектура IoT, протоколи зв’язку, що використовуються в середовищі IoT, основні загрози доступності, цілісності та конфіденційності.

1 ІНТЕРНЕТ РЕЧЕЙ: КОМУНІКАЦІЙНІ ПРОТОКОЛИ ТА ЗАГРОЗИ БЕЗПЕЦІ

1.1 Загальна архітектура IoT

Теоретично термін IoT зазвичай використовується для опису дизайну та реалізації мережі, яка успішно обробляє інформаційні дані в пристроях, які входять до неї. Однак на практиці, оскільки цією мережею є Інтернет, це є чимось складним, оскільки всі пристрої (розумні датчики, центри обробки даних тощо), які беруть участь, повинні мати можливість безперебійно спілкуватися один з одним, прямо чи опосередковано (тобто через шлюзи), безпечним способом. Як наслідок, для забезпечення сумісності всіх пристроїв Інтернету потрібні спеціальні протоколи зв'язку, стандартна структура, сумісність програм, розширені можливості обробки даних і багато іншого. Незважаючи на їх складність у певних реалізаціях, їх елементарна структура досить проста [1].

Розумний об'єкт передає дані, зібрані його датчиками (фізичним світом), до центру обробки даних (локального чи хмарного) або навіть до іншого розумного об'єкта через проміжне з'єднання (шлюз). Використання шлюзу не є обов'язковим, оскільки потенційно смарт-об'єкт також може працювати як шлюз. Потім дані, отримані «з іншого боку», обробляються, і можна ініціювати кілька дій. Ці дії ускладнюють реалізацію, оскільки потрібна більша сумісність для керування або моніторингу автономного пристрою, наприклад, увімкнення обігрівача при певній температурі.

Хоча технологія IoT застосовується до значної кількості сфер і жодним чином не стандартизована, ми розглянемо простий підхід, розглядаючі базову архітектуру та найпоширеніші протоколи, винайдені для цієї технології [2].

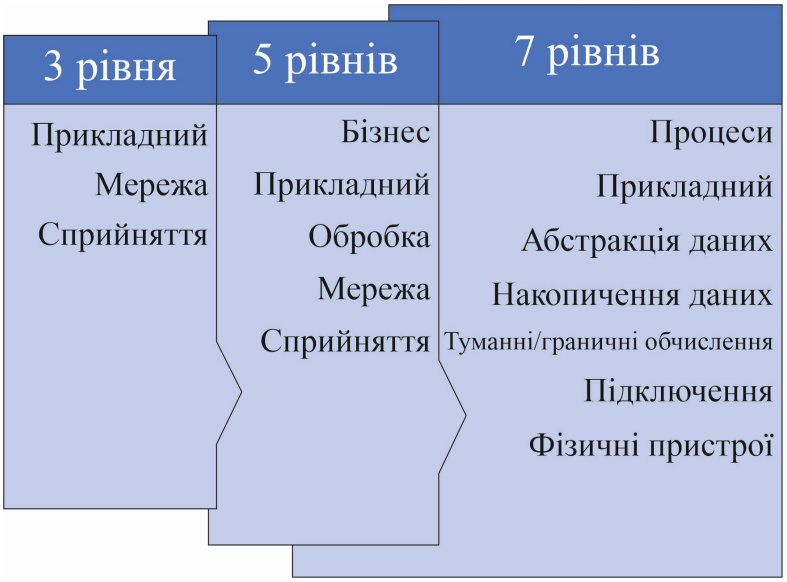


Рисунок 1.1 – Елементарна структура IoT

Щоб визначити еталонну архітектуру, яка підтримує поточні функції та майбутні розширення, масштабованість, сумісність, розподіл даних, обчислювальну потужність і, звичайно, безпеку, необхідно врахувати деякі фундаментальні фактори щодо стандартизації архітектури [3].

Наприклад, в дослідженні [4] класифікація шарів була застосована в трьох-, чотирьох-, п’яти-, шести- або семишарових моделях (див. рис. 1.1).

Ускладнює ситуацію те, що міжнародні організації та великі технологічні компанії, такі як Міжнародний союз електрозв’язку (ITU), Інститут інженерів з електротехніки та електроніки (IEEE), Cisco, Google, Amazon і Європейський інститут стандартів телекомунікацій (ETSI), представили різні фреймворки IoT на основі вимог до додатків, топології мережі, протоколів, бізнес-моделей і моделей обслуговування, оскільки вони охоплюють різноманітні технології [5].

Оскільки досі немає єдиної стандартної еталонної архітектури для IoT, якої можна дотримуватися для всіх можливих реалізацій, обрано 3-рівневу модель, яка складається з рівня сприйняття, мережі/передачі та рівня додатків, у яких рівні у будь-якому випадку не можна розглядати як підрівні і які можуть повністю описати елементарні операції реалізації IoT [6].

1. Рівень сприйняття. Рівень сприйняття або фізичний рівень складається з фізичних пристроїв, які є наріжним каменем технології IoT, метою яких є збір

інформації, перетворення її в цифрові дані та передача на інший рівень, щоб на основі цієї інформації можна було виконувати дії. Діючи як середовище між цифровим і реальним світом, ці фізичні пристрої можуть бути датчиками (температури, вологості, світла тощо), приводами (електричними, механічними, гідравлічними тощо), RFID (мітки RFID) [7], відеотрекери (IP-камери) або будь-що, що може використовувати дані для взаємодії з різними пристроями через мережу.

Однак різниця між традиційними датчиками та інтелектуальними датчиками, які використовуються в IoT, полягає в тому, що розумні датчики включають інтегрований мікропроцесор (DMP), який може обробляти оцифровані дані, отримані датчиком. Ці дані можуть бути нормалізовані, відфільтровані від шуму або перетворені для кондиціювання сигналу перед пересиланням на інші пристрої в мережі.

2. Рівень передачі. Рівень передачі, який також називають транспортним або мережевим рівнем, розташований між рівнем сприйняття та прикладним рівнем. На цьому рівні дані, зібрані інтелектуальними датчиками, перетворюються та пересилаються на прикладний рівень за допомогою відповідних каналів зв’язку та протоколів для подальшої обробки, а саме, аналізу, інтелектуального аналізу даних, агрегації та кодування даних, забезпечуючи при цьому функції керування мережею, а не лише базовою маршрутизацією пакетів, як це робить мережевий рівень моделі ISO/OSI.

У реалізаціях IoT бездротові протоколи використовуються частіше у порівнянні з дротовими, оскільки бездротові датчики можна встановлювати навіть у місцях, де немає основних реквізитів для дротових датчиків, таких як живлення, комунікаційні кабелі тощо. Крім того, у бездротовій сенсорній мережі легше додавати, видаляти або переміщувати вузли без перегляду структури всієї мережі. Вибір протоколів для використання може ґрунтуватися на кількох факторах, таких як неоднорідність апаратного забезпечення, енергоспоживання, швидкість передачі та відстань передачі, необхідна для кожної програми та багато інших.

Проте, в деяких інших реалізаціях дротова сенсорна мережа є кращою, оскільки ці мережі надійніші, безпечніші та пропонують вищу швидкість передачі даних. Наприклад, у впровадженні IoT у лікарні, де надійність і швидкість є основними факторами для порятунку життя пацієнта, дротові датчики є кращими, а необхідні умови для їх встановлення можна спланувати під час початкового проектування лікарні (проводка, кабелі подачі електроенергії тощо).

Загалом, інтелектуальні датчики повинні мати можливість спілкуватися один з одним через Інтернет, щоб обробляти інформацію та взаємодіяти з фізичним світом, при цьому вони мають унікальну ідентифікацію для запобігання конфліктам даних. Залежно від конкретних програм, смарт-об’єкти можуть бути доступними безпосередньо без необхідності проміжного шлюзу, реалізувати інтерфейс користувача, що робить можливим взаємодію з користувачем, і багато іншого.

3. Прикладний рівень. Рівень додатків присутній безпосередньо над рівнем передачі, він заснований на реалізації та може бути організований різними способами. Цей рівень, залежно від реалізації, відповідає за аналіз та обробку інформаційних даних, які надійшли з нижчих рівнів (сприйняття та передачі). Більш конкретно, він обробляє ці дані додаткам для того, щоб використовувати їх для бажаних дій (тобто, виконавчих механізмів керування), діючи як міст для перетворення та пересилання їх іншим вузлам або передачі іншим додаткам для подальшої обробки.

Крім того, це рівень, на якому розміщується користувальницький інтерфейс (якщо такий є), що дає користувачам вибір для взаємодії з системою IoT і виконання різних дій (наприклад, якщо частина технічного обладнання потребує обслуговування, IoT повідомить технічного спеціаліста через інтерфейс, який «структурно» працює на прикладному рівні.

Прикладний рівень, на відміну від рівня передачі та сприйняття, може значно відрізнятися залежно від реалізації. Оскільки він розроблений з урахуванням бажаного застосування, цей рівень формується його функціями.

Наприклад, додатки для моніторингу та прийняття рішень у реальному часі відповідають за виконання дій на основі даних, зібраних із рівня сприйняття, оцифрування інформації відповідає за збір і перетворення аналогових даних у цифрові, аналітика використовується для обробки зібраних даних і створення моделі оцінки, тоді як апаратне керування використовується для перетворення даних у фізичні дії [8].

1.2 Комунікаційні протоколи

Багато протоколів сприяють реалізації IoT, але протоколи зв'язку є обов'язковими для мереж IoT. Вибір найкращого протоколу IoT означає точне зважування критеріїв бажаного діапазону додатків, порогового значення енергоспоживання, пропускної здатності інформації, затримки та якості обслуговування, і все це розглядається через призму безпеки. Як згадувалося раніше, пристрої IoT використовують мережеві стандарти та протоколи для забезпечення зв'язку між фізичними об'єктами, підключеними через хмару. Мережеві протоколи та стандарти – це політики, які містять певні правила, які визначають мову спілкування між різними мережевими пристроями.

Кожен пристрій зазвичай підключається до Інтернету за допомогою Інтернет-протоколу (IP), але також може бути підключений локально через blacktooth, NFC (зв'язок ближнього поля) тощо. Деякі відмінності між обома типами підключень полягають у потужності, радіусі дії та використаній потужності ЦП. IP-з'єднання є складними та вимагають збільшення потужності та пам'яті, але не мають обмежень діапазону. З іншого боку, з'єднання blacktooth прості та потребують менше енергії та пам'яті, але мають обмежений діапазон.

Окремі пристрої, як-от смартфони та персональні комп'ютери, використовують мережеві протоколи для зв'язку, однак загальні протоколи, які

використовуються цими пристроями, можуть не відповідати певним вимогам, таким як пропускна здатність, затримка та відстань покриття рішень на основі IoT. Незважаючи на те, що пристрої IoT легко розгортати, їхні протоколи зв'язку є такими, які повинні подолати брак обчислювальної потужності, радіусу дії та надійності з існуючою інтернет-інфраструктурою. Оскільки існуючі протоколи не відповідають критеріям впровадження IoT (Wi-Fi 802.11 a/b/g/n/ac тощо), розглянемо деякі нові протоколи IoT, створені для вимог до додатків IoT.

Оскільки енергоспоживання є важливим фактором при проектуванні мереж IoT, перевагу надають малопотужним технологіям бездротової мережі. Ці технології зазвичай поділяються на дві групи:

- глобальна мережа з низьким енергоспоживанням (LPWAN), яка забезпечує розширений радіус дії до кількох кілометрів, але з обмеженою швидкістю передачі даних для більшості (наприклад, 6LoWPAN, LoRaWAN, Sigfox, NB-IoT, Wi-Fi HaLowTM);
- технології бездротової персональної мережі (WPAN) із радіусом дії до 100 метрів і швидкістю передачі даних до 250 Кбіт/с для Zigbee та до 3 Мбіт/с для blacktooth Low Energy.

а) LPWAN. LPWAN (глобальні мережі малої потужності) — це категорія протоколів, розроблених для зв'язку малого радіусу дії. Незважаючи на те, що «традиційні» стільникові мережі здатні підтримувати глобальні мережі зв'язку, їхні недоліки, як-от складна інфраструктура (антени, підсилювачі тощо) і вимоги до високого енергоспоживання, роблять їх менш сприятливим рішенням при розгляді додатків IoT. З іншого боку, протоколи LPWAN повинні використовуватися простими, малопотужними та малими можливостями ЦП, дозволяючи розгортати датчики без інвестицій у шлюзи, які базуються на недорогих довго працюючих батареях, що робить його більш вигідним варіантом на відміну від стільникової мережі.

Враховуючи низькі вимоги до обладнання, технологія LPWAN може працювати на відстані понад 10 км залежно від оточення та перешкод і

швидкості передачі даних від 0,3 кбіт/с до 50 кбіт/с на канал. Крім того, хоча енергоспоживання та швидкість передачі даних є серйозними проблемами для LPWAN, якість обслуговування (QoS) і масштабованість є важливими факторами при виборі протоколу LPWAN. Протокол 6LoWPAN є прикладом протоколу LPWAN, який поєднує в собі технології IPv6 і LoWPAN і має багато переваг, а саме, виняткове підключення, сумісність із попередніми архітектурами, низьке споживання енергії та спеціальну самоорганізацію.

б) WPAN. WPAN — це локальна сітчаста мережа пристроїв, організованих у сітчасту топологію, у якій кожен пристрій підключений безпосередньо (без шлюзу) до інших пристроїв мережі, які передають дані один одному, доки вони не досягнуть кінцевого одержувача в цій мережі. Ця структура сприяє відмовостійкості мережі, є простою у впровадженні та обходиться дешевше, ніж інші мережі, особливо на великих територіях через відсутність додаткового обладнання (тобто шлюзів).

ZigBee вважається найпопулярнішим mesh-протоколом, який використовується в IoT. Він має малий радіус дії, але споживає мінімальну потужність, що може розширити зв’язок між кількома пристроями IoT. У порівнянні з протоколами LPWAN, ZigBee може забезпечувати високу швидкість передачі даних за один раз, але з більшою енергоефективністю завдяки сітчастій топології. Проте, через короткий фізичний радіус дії ZigBee та будь-який інший mesh-протокол найкраще підходять для реалізацій малого та середнього радіусу дії, наприклад, розумних домашніх мереж [9].

Комунікація в технологіях IoT охоплює як дротові, так і бездротові з’єднання. Залежно від типу з’єднання протоколи зв’язку в 4-рівневій мережі описані в подальшому для кожного рівня.

в) Рівень програми. П’ять різних протоколів описані нижче для прикладного рівня; MQTT, CoAP, REST, XMPP і AMQP. Також обговорюються властиві функції та проблеми, пов’язані з безпекою.

- MQTT. Протокол телеметрії черги повідомлень (MQTT) – це протокол обміну повідомленнями для публікації та підписки, який працює на дуже

простій моделі клієнт/сервер і працює через TCP/IP або інші протоколи. Він більше підходить для обмежених середовищ, таких як IoT, оскільки він відкритий і легко реалізований. Вимоги безпеки, які повинні бути виконані в реалізаціях MQTT, це автентифікація, авторизація та безпечний зв'язок. У критично важливих інфраструктурах і програмах з конфіденційною інформацією MQTT може працювати та пропонувати розширені служби безпеки з використанням спеціальних рекомендованих функцій.

– CoAP. Протокол обмежених додатків (CoAP) визначено як спеціалізований протокол веб-передачі в RFC 7252. Це легкий протокол із низькою швидкістю передачі, запропонований для використання з обмеженими вузлами та обмеженими мережами. Конструкція підходить для додатків «машина-машина» (M2M), таких як управління ланцюгом поставок і інтелектуальними лічильниками для відстеження споживання енергії. Він може дуже добре взаємодіяти з HTTP, що полегшує інтеграцію з Інтернетом. Але CoAP не є безпечним протоколом, тому він не має широкого використання в IoT.

– REST. Representational State Transfer (REST) – це гібридний архітектурний стиль для розподілених гіпермедійних систем. Він містить набір правил, які описують керівні принципи розробки програмного забезпечення для створення програми з певними обмеженнями. Він використовується для створення веб-сервісів, також званих RESTful. REST включає а) обмеження клієнт-сервер, б) обмеження без стану, яке забезпечує видимість, надійність і масштабованість, в) обмеження кешу, яке покращує ефективність мережі, г) набір із чотирьох обмежень для єдиного інтерфейсу між компонентами, е) багаторівневі системні обмеження та f) необов'язкові обмеження коду на вимогу.

– XMPP. Розширюваний протокол обміну повідомленнями та присутності (XMPP) – це відкрита технологія XML для спілкування в реальному часі. Він використовується для обміну миттєвими повідомленнями, присутності та співпраці. Присутність вказує на те, що сутність готова до

обміну повідомленнями. Обмін повідомленнями використовує ефективний механізм push, який забезпечує можливість роботи в реальному часі. Відкритий дизайн XMPP полегшує зміни та дозволяє його розширювану функцію, яка відповідає реалізації IoT. Нещодавно до баз даних NVD, які підтримує NIST, було додано значну кількість кодів CVE, пов'язаних із відомими вразливими місцями XMPP, які дозволяють здійснити низку атак.

- AMQP. Advanced Message Queuing Protocol (AMQP) – це відкритий стандарт, придатний для обміну бізнес-повідомленнями між програмами, який працює асинхронно в різних організаціях і платформах. Це протокол дротового рівня, який забезпечує надійний обмін бізнес-повідомленнями. Деякі з основних характеристик, включених до конструкції AMQP, спрямовані на забезпечення безпеки, надійності та сумісності. Він був схвалений для випуску як міжнародний стандарт ISO та IEC у 2014 році та складається з кількох рівнів. Найнижчий рівень призначений для транспортування повідомлень між двома процесами, а рівень обміну повідомленнями визначає стандартний формат кодування, який повинне мати кожне повідомлення.

г) Транспортний рівень. На транспортному рівні зазвичай використовується значна кількість протоколів, як описано надалі.

- TCP. Протокол керування передачею (TCP) – це надійний протокол, орієнтований на з'єднання, який працює в три фази. Він належить до набору інтернет-протоколів і широко використовується для підключення між пристроями. Великі накладні витрати на пакети відносять його до важкої категорії протоколів із великим енергоспоживанням.

- UDP. Протокол UDP (User Datagram Protocol) – це легкий протокол без з'єднання, який можна використовувати, коли втрата пакетів є прийнятною під час передачі даних. Це краще для зв'язку в бездротових сенсорних мережах, але не є надійним. Його перевагою є те, що не потрібно встановлювати з'єднання перед передачею даних.

- DCCP. Протокол керування перевантаженням дейтаграм (DCCP) – це транспортний протокол для двонаправлених одноадресних з'єднань. Він

використовується для таких додатків, як потокове медіа та VoIP, де TCP не може контролювати часові затримки та здійснювати надійну доставку у правильному порядку. З іншого боку, програми UDP здатні контролювати затримки, але DCCP має вбудований механізм контролю перевантаження щоб їх уникнути.

- SCTP. Протокол передачі керування потоком (SCTP) є надійним транспортним протоколом для сигналізації PSTN повідомлень, що передаються через IP. Він був розроблений, щоб протистояти різним типам атак.
- RSVP. Протокол резервування ресурсів (RSVP) – це протокол для конкретних запитів QoS, які застосовуються хостами та доставляються роутерами до вузлів, щоб забезпечити та надати запитану послугу. Результатом є резервування ресурсів уздовж шляхів потоків даних.
- TLS. Безпека транспортного рівня (TLS) – це протокол, який використовується через Інтернет для забезпечення безпечного зв'язку між програмами клієнт/сервер. Використання криптографічних алгоритмів запобігає перехопленню даних, підробці та зміні повідомлень. Версія 1.3 цього протоколу дійсна з 2018 року.
- DTLS. Безпека транспортного рівня дейтаграм (DTLS) базується на протоколі TLS, який не можна безпосередньо використовувати в середовищах дейтаграм через проблеми з втратою пакетів і переупорядкуванням пакетів. Таким чином, DTLS – це TLS із необхідними змінами, які вирішують ці проблеми та підвищують надійність.
- RPL. RPL – це протокол маршрутизації IPv6, розроблений для мереж з низьким енергоспоживанням і втратами (LLN), класу мереж з обмеженнями пам'яті, обчислювальної потужності та енергії. Він використовує Destination Oriented Directed Acyclic Graph (DODAG) для маршрутизації даних, і оскільки він базується на стандарті IPv6, він кращий для додатків IoT.
- CARP. Протокол маршрутизації з урахуванням каналів (CARP) – це розподілений міжрівневий протокол, розроблений для підводних бездротових сенсорних мереж для доставки даних до споживача з кількома переходами.

– CORPL. Когнітивний RPL (CORPL) є розширенням протоколу RPL для когнітивних мереж, який також використовує DODAG, належним чином адаптований до когнітивних мереж.

– QUIC. Швидкі підключення до Інтернету UDP (QUIC) – це протокол, орієнтований на підключення між двома кінцевими точками, які обмінюються дейтаграмами UDP. Він забезпечує з'єднання з низькою затримкою та гарантує конфіденційність, цілісність і доступність завдяки впровадженню заходів безпеки. Це робить QUIC таким же безпечним, як і протокол TLS.

– uIP. Стек uIP TCP/IP забезпечує зв'язок за допомогою набору протоколів TCP/IP на дуже малих мікроконтролерах, навіть 8-розрядних. Це дуже мала реалізація стеку TCP/IP, написана максимально просто мовою програмування C. Код потребує кілька КБ, а оперативна пам'ять надзвичайно обмежена. Його дизайн містить мінімальний набір функцій, необхідних для повного стеку TCP/IP, і містить протоколи IP, ICMP, UDP і TCP. Піри uIP також можуть запускати легкий стек.

– Aeron. Aeron – це стек протоколів, розроблений для UDP unicast і UDP multicast і використовується для потокових даних. Він відрізняється двома основними функціями: високою пропускною здатністю та низькою затримкою.

– CCN. Контент-центрична мережа (CNN) або інформаційно-орієнтована мережа (ICN) представляє нову парадигму комунікацій. Згідно з цією архітектурою запити іменованого вмісту замінюють надсилання пакетів. Дві архітектури ICN — мережа іменованих даних (NDN) і мережа, орієнтована на вміст (CCNx).

– NanoIP. NanoIP – це набір протоколів, спеціально розроблений для невеликих пристроїв, таких як датчики та вбудовані пристрої. Транспорт під назвою NanoIP підтримує надійні з'єднання, а інший, nanoUDP, підтримує зв'язок без з'єднання. Жоден із них не стосується безпосередньо стандартних TCP і UDP, вони скоріше стосуються функціональних еквівалентів.

– TSMP. Протокол Time Synchronized Mesh – це стек протоколів для WSN. Він був розроблений відповідно до вимог надійності, безпеки, своєчасної

доставки та низької потужності.

ґ) Мережевий рівень. Для прикладного рівня представлено п’ять мережевих протоколів; Також обговорюються WiFi, Bluetooth, ZigBee, Z-Wave і LoRaWAN, а також функції та проблеми, пов’язані з безпекою.

- WiFi. Wi-Fi є найбільш поширеною та відомою технологією зв’язку, яка базується на стандарті бездротового зв’язку 802.11 Інституту інженерів з електротехніки та електроніки (IEEE). Він проходить постійні вдосконалення, які роблять його швидшим, з меншою затримкою та придатним для кількох різних пристроїв. Залежно від покоління Wi-Fi, безпека посилюється, щоб відповідати вимогам конфіденційності даних автентифікації та доступності, захищаючи з’єднання Wi-Fi. Пристрої підключаються бездротовим шляхом, надсилаючи сигнали в межах 100 метрів.
- Bluetooth. Bluetooth Low Energy (LE) радіо є кращим для реалізації IoT, оскільки воно розроблено для роботи з дуже низьким споживанням енергії. Він здатний передавати дані через велику кількість каналів, пропонуючи необхідну відкритість для реалізації в багатьох різних топологіях зв’язку, від «точка-точка» до широкомовної та сітчастої топологій, а також поруч із великомасштабними мережами бездротових пристроїв. Крім того, він надає послуги позиціонування пристрою з високою точністю. Він широко використовується, тому що ідеально підходить для найсучасніших мобільних пристроїв, таких як переносні пристрої та смартфони, які поширені по всьому світу.
- ZigBee. ZigBee – це протокол із таким самим значним використанням, як і blacktooth в інфраструктурах IoT. Він відповідає розширеним вимогам безпеки з низьким енергоспоживанням, низьким радіусом дії даних і діапазоном зв’язку до 200 метрів, що вдвічі більше порівняно з відповідним blacktooth. Підходить для датчиків і пристроїв з декількома обмеженнями, він полегшує створення великих моделей IoT з великою кількістю вузлів.
- Z-Wave. Z-Wave – це бездротовий протокол, розроблений для домашньої автоматизації. Він працює у власному діапазоні радіочастот, що

зменшує проблеми з перешкодами.

– LoRaWAN. LoRaWAN – це мережевий протокол із низьким енергоспоживанням, широка зона (LPWA), який використовується для бездротового підключення пристроїв на основі батарей у реалізаціях IoT. Він відповідає значним вимогам двонаправленого зв’язку та наскрізної безпеки [10].

Таблиця 1.1 – Протоколи зв’язку для IoT у 4-рівневій архітектурі ISO: переваги та недоліки

Протокол	Переваги	Недоліки
AMQP	Надійність Безпека, можливість розширення з мінімальними зусиллями	Великі вимоги до пам'яті, повільна передача даних
MQTT	Низьке енергоспоживання Низьке використання смуги пропускання	Обмежена сумісність, Внутрішні обмеження безпеки, Погана розширюваність
Zigbee	Висока безпека, низьке енергоспоживання, великий діапазон зв'язку	схильний до перешкод, дорогий
Z-Wave	Низька затримка, низьке енергоспоживання, розумне покриття	Низька швидкість передачі даних, преміальні ціни
Wi-Fi	Зручний і простий в установці, висока швидкість передачі даних	Високе енергоспоживання, важко масштабувати
LoRaWAN	Масштабованість, велике покриття, низьке енергоспоживання	Низька швидкість передачі даних, спеціальний шлюз LoRa

д) Фізичний рівень. IEEE 802.15.4 – це протокол, розроблений для фізичного рівня та рівня MAC, який забезпечує зв’язок між пристроями з обмеженнями живлення та певними вимогами для надання послуг через датчики. Підтримується недорогий зв’язок і зв’язок на короткій відстані, а пристрої співпрацюють, щоб полегшити маршрутизацію з кількома стрибками

та досягти розширення діапазону. Він містить описи низькошвидкісних бездротових персональних мереж (LR-WPAN).

На рис. 1.2 показано протоколи зв’язку, які переважно використовуються в реалізаціях IoT у 4-рівневій архітектурі ISO. У табл. 1.1 висвітлено основні переваги та недоліки основних протоколів.

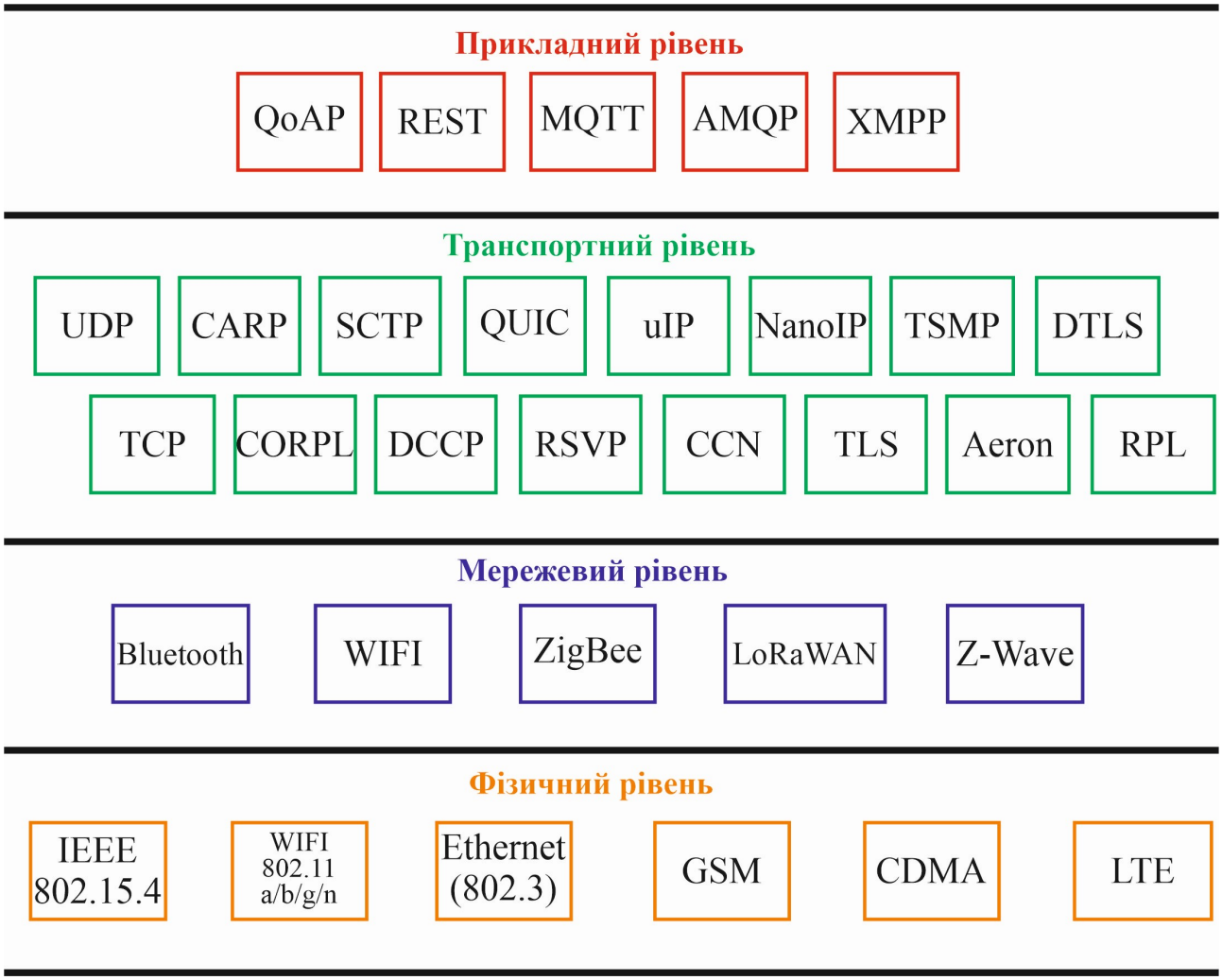


Рисунок 1.2 – Протоколи зв’язку для IoT у 4-рівневій архітектурі ISO

1.3 Питання та проблеми безпеки

Оскільки технологія IoT розроблена для застосування в багатьох секторах, які мають вирішальне значення, особливо для національної безпеки та економіки з різними галузевими стандартами та специфікаціями, питання безпеки вимагають першочергової уваги, щоб мінімізувати поверхню атаки та запобігти проблемам безпеки [11]. Наприклад, 29 квітня 2021 року дослідницька група Microsoft із безпеки IoT виявила критичні вразливості розподілу пам’яті в пристроях IoT, які потенційно можуть бути використані для обходу елементів керування безпекою та виконання шкідливого коду або спричинення збою системи [12].

Окрім кібератак, розробка великомасштабних гетерогенних мереж обмежених вузлів, що працюють у режимі реального часу, має базуватися на архітектурі, яка є стійкою до керування факторами, що випливають із надійності, якості обслуговування, модульності, семантичної сумісності, керування конфіденційністю, обладнання та програмного забезпечення. Сумісність. Базуючись на 3-рівневому протоколі, обговоримо наступні проблеми та проблеми, які стосуються загроз безпеці кожного рівня.

Найціннішу інформацію можна отримати, розглядаючи кожен тип атаки та відповідний великий вплив на конфіденційність, цілісність і доступність. Рис. 1.3 ілюструє атаки, описані вище, на рівні кожного рівня та об’єднує їх, щоб показати ті, які впливають на два чи навіть три ставлення до безпеки, які потрібно зберегти. Виділемо більшість атак, що впливають на всі три характеристики безпеки, велика кількість з яких впливає лише на дві, головним чином на цілісність і доступність, і лише деякі мають серйозний вплив на конфіденційність даних, що зберігаються або передаються. Ці висновки можуть допомогти розробникам IoT створити безпечні реалізації IoT, які захистять їхніх користувачів і полегшать розгортання додатків IoT.

а) Рівень сприйняття. Найважливіші загрози, які загрожують рівню

сприйняття.

– Прослуховування: пристрої IoT вразливі до атак з підслуховуванням, оскільки їм бракує обчислювальної потужності для методів шифрування, на відміну від мережевих пристроїв, які не є IoT. Крім того, якщо пристрої працюють у віддаленому місці з мінімальним фізичним моніторингом або без нього, атаки підслуховування легше здійснити [13].

– Захоплення вузла: оскільки існує величезна кількість пристроїв, які можуть брати участь у мережі IoT, поверхня атаки мережі зростає експоненціально. Зловмисник потенційно може отримати контроль над ключовим вузлом мережі, таким як шлюз, який, у свою чергу, надає йому доступ до всієї інформації, якою обмінюються через мережу [14].

– Шкідливий підроблений вузол: перевага IoT у легкому створенні мережі може стати слабкою стороною. Зловмисник завжди може встановити в мережу вузол, який вводить неправдиві дані, дія може виснажити ресурси з законних вузлів, підриваючи роботу всієї мережі [15].

– Атака повторного відтворення: під час атаки повторного відтворення зловмисник підслуховує автентичну інформацію, що передається по лінії зв’язку між відправником і одержувачем, і захоплює її. Потім він надсилає ту саму автентифіковану інформацію жертві, яку вже отримав у своєму спілкуванні, демонструючи доказ її особи та автентичності. Оскільки повідомлення зашифроване, одержувач може розглядати його як законний запит і відповідним чином відповісти зловмиснику [16].

– Timing Attack: Timing Attack більш ефективний на пристроях з мінімальними обчислювальними можливостями. Ця атака дозволяє зловмиснику виявляти вразливості та витягувати інформацію, що зберігається в безпеці системи, визначаючи час, який потрібно системі, щоб відповісти на різні запити, вхідні дані, криптографічні алгоритми тощо [23].

У табл. 1.2 представлено типи атак, визначені на рівні сприйняття в системах IoT як найбільш значущі. Цілями цих атак є пристрої, вузли, вся мережа або інформація, яка передається під час процедури автентифікації [24].

Слабкі сторони пристроїв, систем або протоколів, які їм сприяють, здебільшого зосереджені в обмеженнях потужності, які мають пристрої, у внутрішніх проблемних питаннях і протоколах або інфраструктурі та конструкції IoT. В останньому стовпці таблиці пропонуються будь-які контрзаходи для запобігання або виявлення таких атак, уникнення наслідків і пом’якшення поширення шкоди.

Таблиця 1.2 – Поверхня атаки на рівні сприйняття в системах IoT

Атака	Ціль	Слабкість	Протидія
Підслуховування	Пристрої	Низька потужність (без шифрування), без моніторингу.	Шифрування [17]
Захоплення вузла	Ключовий вузол мережі	Вразливі протоколи.	Механізми виявлення
Шкідливий підроблений вузол	Мережа	Легкість IoT для створення мереж.	Механізми виявлення [18], Довірчі послуги [19]
Повторна атака	Аутентифікація Інформація	Вразливі протоколи.	Ключі сесії, блокчейн [20], Механізми виявлення [21]
Тимінг Атака	Пристрої з обмеженими можливостями	Унікальна поведінка пристрою та час відгуку.	Механізми захисту конфіденційності [22]

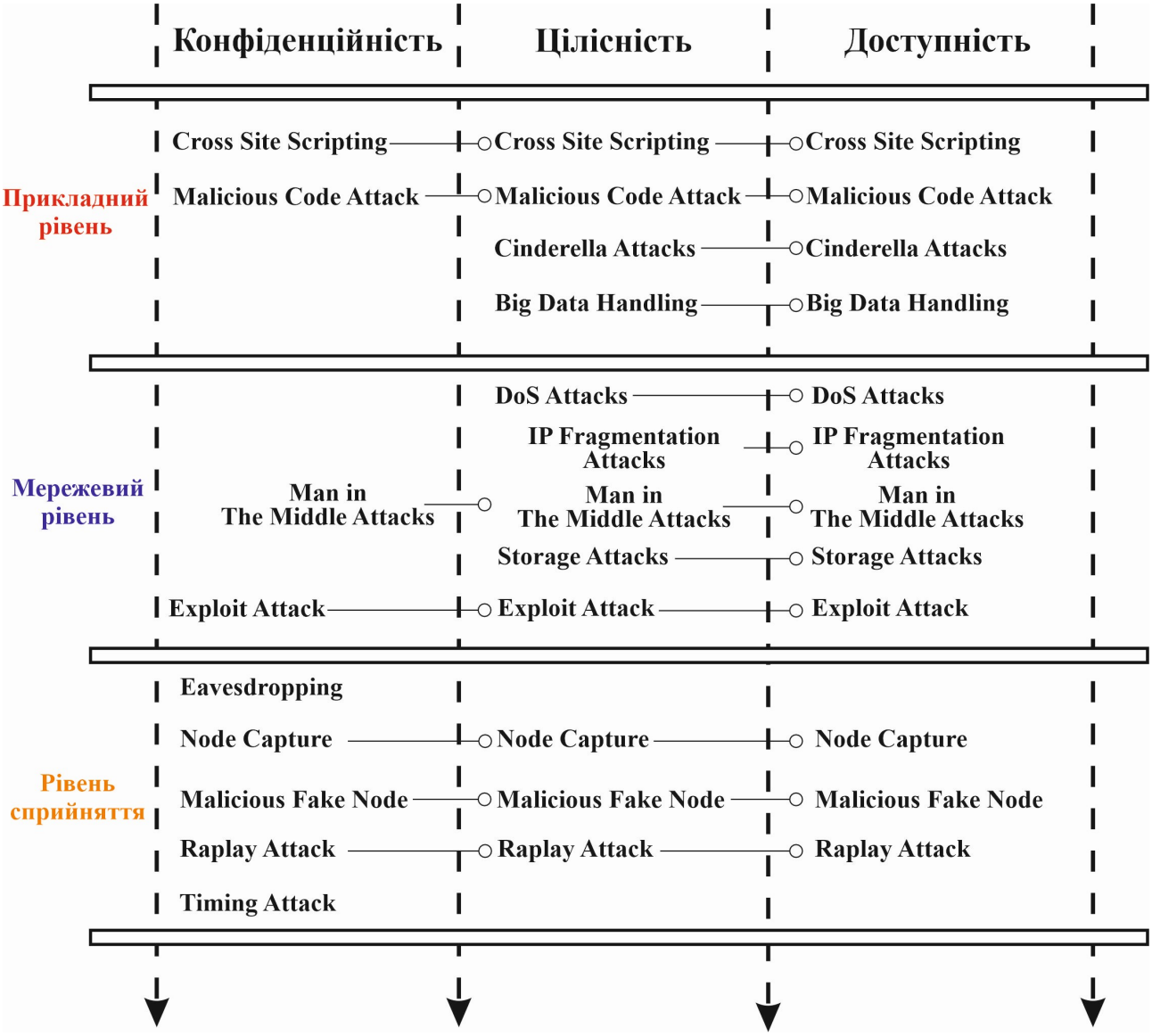


Рисунок 1.3 – Типи атак, які впливають на конфіденційність, цілісність і доступність у 3-рівневій архітектурі IoT

б) Мережевий рівень. Мережевий рівень дуже чутливий до атак із проблемами безпеки головним чином щодо цілісності та доступності інформації, якою обмінюються в мережі. Вибрані загрози безпеці мережевого рівня підсумовуються далі.

- Атаки на відмову в обслуговуванні (DoS): за допомогою DoS-атаки користувачі не можуть отримати доступ до пристроїв або інших мережевих ресурсів. Ця дія виконується шляхом заповнення цільових пристроїв або мережевих ресурсів зайвими запитами, що унеможливлює або ускладнює зв’язок між іншими користувачами [25].

– Атаки на фрагментацію IP: це атака категорії DoS, коли зловмисник використовує максимальну одиницю передачі (MTU) мережі. Коли IP-пакети збираються повторно після передачі, їх розмір перевищує максимальну одиницю передачі, яку може обслуговувати мережа, і тому це згортає мережу [26].

– Атака Man in The Middle: під час атаки MiTM зловмисник, поки його не спостерігають, перехоплює та змінює дані зв’язку між двома сторонами. Оскільки вони обидва не знають про перехоплення, зловмисник може контролювати їх спілкування, змінюючи повідомлення відповідно до своїх потреб. Це вважається серйозною загрозою безпеці мережі, оскільки зловмисник може захоплювати та маніпулювати інформацією в режимі реального часу, перш ніж його викриють [27].

– Атаки на сховище: оскільки всі дані зберігаються на пристроях зберігання (локально чи в хмарі), їх можна атакувати, змінивши законні дані на неправильні або навіть видаливши їх назавжди. Таким чином, якщо багато груп користувачів мають доступ до сховища, тим більше ймовірність таких типів атак, навіть якщо процес базується на технології блокчейн. [28].

– Експлойт-атаки: експлойт-атаки – це атаки, які використовують вразливі місця безпеки в програмах, системах [29], або обладнання [30]. Їхня мета — отримати частковий або повний контроль над системою та викрасти чи змінити збережену інформацію. Хоча системний адміністратор може виправити вразливість безпеки, кожна окрема зміна в програмі чи апаратному забезпеченні може створити для зловмисника нові вразливості.

У табл. 1.3 представлено типи атак, визначені на мережевому рівні в системах IoT як найбільш значущі. Цілями цих атак є пристрої, мережеві ресурси, комунікаційні дані або збережені дані. Слабкі місця тепер знаходяться в протоколах, а також у програмах або навіть апаратному забезпеченні. В останньому стовпці таблиці пропонуються деякі контрзаходи для запобігання або виявлення цих атак і підвищення безпеки.

Таблиця 1.3 – Поверхня атаки на мережевому рівні в системах IoT

Атака	Ціль	Слабкість	Протидія
DoS	Пристрої або мережеві ресурси	Вразливі протоколи.	Механізми виявлення [32]
IP Fragmentation	MTU мережі	Вразливі протоколи.	Механізми виявлення [33]
Man in The Middle	Зв'язок, дані	Вразливі протоколи.	Шифрування E2E [17]
Storage	Дані, що зберігаються на пристроях зберігання	Відсутність шифрування.	Полегшені алгоритми шифрування [34]
Exploit	Система та збережена інформація	Вразливості програми, системи та апаратного забезпечення.	Оновлення програм і системи, заміна обладнання [35]

в) Рівень програми. Рівень програми більш схильний до проблем безпеки порівняно з двома іншими рівнями через свою різноманітність. Рівень додатків складається з додатків і програмного забезпечення, створених для реалізації IoT, і оскільки їх незліченна кількість, то і додатків, створених для них, також. Наприклад, коли IoT використовується для додатків «Розумний дім», загрози та вразливості можуть надходити від кожної програми, яка має доступ до апаратного забезпечення, що використовується як зсередини (центр керування або навіть наш мобільний додаток), так і зовні (віддалені програми).

Деякі з найпоширеніших загроз безпеці прикладного рівня в IoT:

- Cross Site Scripting: під час атак Cross Site Scripting зловмисник впроваджує сценарії зловмисного коду, а саме сценарії Java, на сайт довіреного домену, який переглядає багато інших користувачів. За допомогою цієї дії зловмисник може змінити вміст програми відповідно до своїх цілей і використати оригінальну інформацію у зловмисний спосіб [31].
- Атака зловмисним кодом: кожне програмне забезпечення, в тому числі і шкідливе, створено за допомогою коду. Троян, вірус, хробаки чи бекдори є зловмисним кодом, який має на меті спричинити небажаний вплив на роботу

системи [36]. Зазвичай ці типи атак неможливо заблокувати або викрити за допомогою антивірусного програмного забезпечення, і вони можуть активуватися самі по собі, коли виконуються певні критерії або після взаємодії користувача (наприклад, відкриття файлу).

– Атаки Cinderella: ці атаки можуть статися, коли зловмисник отримує доступ до системи та змінює внутрішній годинник мережі. Ця дія призводить до хибного передчасного завершення терміну дії програмного забезпечення безпеки (тобто антивірусу), що робить його марним, що збільшує вразливість мережі [37].

– Обробка великих даних: великі мережі IoT із багатьма взаємодіючими пристроями створюють величезну кількість даних. Якщо апаратне забезпечення, що використовується в мережі, не може обробляти дані відповідно до поточних або майбутніх вимог, це може призвести до збоїв у мережі та втрати даних [38].

У табл. 1.4 представлено типи атак, визначені на прикладному рівні в системах IoT як найбільш важливі. Цілями цих атак є програми та програмне забезпечення загалом. Слабкі місця знаходяться в програмах і системі. Останній стовпець таблиці, в якому пропонуються деякі контрзаходи, спрямовані на виявлення цих атак, оскільки механізми запобігання не змогли їх зупинити, і тому вони відбуваються [39].

Таблиця 1.4 – Поверхня атаки на прикладному рівні в системах IoT

Атака	Ціль	Слабкість	Протидія
Cross Site Scripting	Застосунок	Уразливості програми та системи.	Механізми виявлення [40]
Malicious Code	Застосунок та система	Уразливості програми та системи.	Механізми виявлення [41]
Cinderella	Програмне забезпечення безпеки	Вразливості системи.	Механізми виявлення [32]
Big Data Handling	Система	Вразливості системи.	Механізми виявлення [32]

г) Міжрівневі атаки. Окрім вищезгаданого, міжрівневі атаки також становлять загрозу для систем IoT. Як зазначено в роботі [42] міжрівнева атака, яка поєднує вразливості на кількох рівнях мережевого протоколу, може завдати більшої шкоди порівняно з однорівневою. В роботі [43] запроваджено атаки DoS (Denial of Service) у бездротових ad hoc мережах, які поширюються від MAC до мережевого рівня, викликаючи переривання критичних маршрутів. В роботі [44] вивчалися скоординовані атаки, повідомляючи про атаки з помилковими датчиками даних (RFSD) на рівні PHY. В роботі [42] запропоновано перехресну атаку за допомогою маніпуляції рангом і затримки скидання (RMDD) у IoT і досліджено, як атака низької інтенсивності на протокол маршрутизації для мереж з низькими втратами електроенергії (RPL) знижує пропускну здатність програми.

г) Заходи протидії. Вище було представлено безліч атак, які можуть бути матеріалізовані на одному або кількох рівнях, що впливає на належну роботу додатків, які підтримуються IoT. Ці програми охоплюють усі важливі та повсякденні аспекти життя громадян у сучасному місті та вимагають рішень із кібербезпеки, які можуть зробити ці програми надійними, стабільними та безпечними. Рішення безпеки можна розділити на три основні категорії: програмне забезпечення, апаратне забезпечення та організаційні/процедурні заходи.

Кожна архітектура, яка містить рішення IoT, повинна починатися з прийняття міжнародно прийнятих стандартів безпеки в організаціях, особливо тих, які мають справу з критично важливими операціями, такими як охорона здоров’я чи енергетика. За потреби потрібно також включити використання інструментів безпеки як для запобігання, так і для розслідування, таких як брандмауери, системи запобігання вторгненням (IPS), системи виявлення вторгнень (IDS), антивірусні та шкідливі програми. Реалізація заходів для судової експертизи, виправлення та оновлення, фізичної безпеки, контролю доступу та автентифікації також є важливими. Нарешті, вдосконалення можливостей реагування на інциденти завжди має бути пріоритетом для всіх

сучасних цифрових систем.

Спеціально для IoT рішення повинні включати полегшені алгоритми шифрування, розподілені механізми виявлення, федеративне навчання, змагальні методи навчання та розширену автентифікацію як пристроїв, так і користувачів [45]. Як зазначено в роботі [46] через неоднорідність, масштабованість і динамічний характер Інтернету речей, звичайну криптографію кібербезпеки, таку як AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), DES (Data Encryption Standard), Blowfish і RC6 не можна відразу використовувати в цих доменах. Рішення, подібні до запропонованих у роботах [47], [34] є хорошими прикладами таких рішень.

Щодо механізмів виявлення, які можна використовувати для повідомлення про ненормальну роботу системи IoT, нещодавно було представлено кілька рішень. В роботі [32] запропоновано об’єднану систему виявлення вторгнень на основі навчання для захисту інфраструктур сільськогосподарського Інтернету речей під назвою FELIDS, яка може одночасно захистити конфіденційність даних пристроїв Інтернету речей і досягти високої точності проти кількох атак.

2 ОГЛЯД БЕЗПЕЧНИХ ПРОТОКОЛІВ ЗВ'ЯЗКУ ДЛЯ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Огляд безпеки IoT

IoT пропонує підключення як для зв'язку «людина-машина», так і для зв'язку «машина-машина». Найближчим часом *все* , ймовірно, буде оснащено невеликими вбудованими пристроями, здатними підключатися до Інтернету. Така здатність корисна для різних сфер нашого повсякденного життя: від автоматизації будівель, розумних міст і систем спостереження до всіх носимих розумних пристроїв. Проте, чим більше пристроїв IoT розгортається, тим більший ризик зазнає наша інформаційна система. Дійсно, незначна кількість пристроїв в Інтернеті речей вразлива до атак на безпеку, наприклад, атак на відмову в обслуговуванні та повторних атак через їх обмежені ресурси та відсутність методів захисту. Такого роду атаки призводять до розрядження батареї датчика та погіршують роботу сенсорних програм. У більш серйозних випадках витік інформації з таких крихітних пристроїв може призвести до оприлюднення конфіденційних даних. У цьому розділі спочатку представлено основні властивості безпеки для IoT. Потім підсумовуються проблеми, які необхідно вирішити в безпеці IoT.

Властивості безпеки. Для захисту IoT може знадобитися виконати кілька параметрів безпеки. Ці загальні властивості безпеки також були визначені в [48, 49]. Як правило, послуги безпеки, які повинні надаватися, включають конфіденційність, цілісність, автентифікацію, авторизацію та свіжість. Вимоги безпеки зосереджені на даних, якщо конфіденційні дані, виміряні або надані пристроями Інтернету речей, можуть потребувати захисту. Вимоги до безпеки також можуть передбачати контрольований доступ до інших ресурсів, наприклад, рівня мережі IoT. Табл. 2.1 визначає властивості безпеки, які надалі будуть обговорюватися у зв'язку з протоколами безпеки та рішеннями, запропонованими для IoT.

Таблиця 2.1 – Властивості безпеки для протоколів безпеки в IoT.

Конфіденційність	Повідомлення, якими обмінюються в IoT, може знадобитися захистити. Зловмисник не повинен отримати інформацію про повідомлення, якими обмінюються сенсорний вузол і будь-який інший об’єкт Інтернету
Цілісність	Зміна повідомлень повинна бути виявлена одержувачем
Автентифікація	Одержувач також повинен мати можливість перевірити походження повідомлень, якими обмінюється
Авторизація	Пристрої Інтернету речей повинні мати можливість перевіряти, чи мають певні об’єкти доступ до їхніх виміряних даних. На мережевому рівні лише авторизовані пристрої повинні мати доступ до мережі IoT. Неавторизовані пристрої не повинні мати можливість маршрутизувати свої повідомлення через пристрої IoT, оскільки це може виснажити їх енергію
Свіжість	Ця властивість гарантує, що старі повідомлення не відтворюватимуться повторно. Це важливо, щоб захистити канал зв’язку від атак повтору

Таблиця 2.2 – Проблеми дослідження в IoT

Сумісність	Розгортання рішень безпеки в Інтернеті речей не повинно перешкоджати функціональній роботі взаємопов’язаних різнорідних пристроїв. Більшість пристроїв Інтернету речей обмежені щодо процесора, об’єму пам’яті та живлення акумулятора. Вони часто працюють на каналах зв'язку з втратами та низькою пропускною здатністю. Видається неможливим застосувати безпосередньо стандартні звичайні протоколи безпеки Інтернету в контексті IoT. Як приклад, використання малих пакетів (тобто IEEE 802.15.4 підтримує лише 127-байтові пакети [50]) може призвести до фрагментації більших пакетів при використанні стандартних протоколів. Це вичерпає термін служби сенсорних вузлів і відкриє нові можливості DoS-атак. Отже, стандартні протоколи безпеки повинні бути перероблені, щоб адаптувати такий складний сценарій, щоб запропонувати еквівалентні рівні безпеки, але більш ефективну продуктивність для IoT
Ресурсні обмеження	Сенсорні вузли повинні бути доступні, коли це необхідно. Мережа високої доступності повинна залишатися функціональною, особливо проти атак на відмову в обслуговуванні, таких як перелив вхідних повідомлень на цільові вузли, що змушує їх вимикатися
Стійкість до атак	Система повинна уникати окремих точок відмови, щоб скомпрометований вузол не вплинув на всю систему. Крім того, захищена мережа також повинна уникати атак із виснаженням ресурсів, запущених проти пристроїв з обмеженими ресурсами
Захист	Популярність тегів RFID викликала занепокоєння щодо конфіденційності, оскільки будь-хто може відстежувати мітки та ідентифікувати об’єкти, які їх містять. Крім

конфіденційності	того, у міру того, як технологія носіння набирає обертів, незабаром люди зможуть підключати свої тіла до Інтернету, «встановлюючи» крихітні апаратні пристрої (наприклад, мікросхеми імплантатів усередині нашого тіла). Отже, наша особиста інформація (тобто записи про медичне обслуговування) має залишатися в безпеці, її не можна відстежувати, зв'язувати та ідентифікувати.
Масштабованість	Мережа IoT, наприклад WSN, зазвичай складається з великої кількості пристроїв. Пропонований протокол безпеки повинен мати можливість масштабування. Ця властивість тісно пов'язана з обсягом інформації, яку кожен пристрій має зберігати в пам'яті, щоб захищений канал узгоджувався з якомога більшою кількістю об'єктів (інших вузлів датчиків або об'єктів Інтернету).

Виклики. Гетерогенна природа IoT створює різні проблеми з точки зору безпеки даних і функціональності мережі. Захищений і працездатний IoT повинен подолати виклики, наведені в табл. 2.2, щоб відповідати наведеним вище вимогам безпеки.

2.2 Таксономія протоколів безпеки для IoT

Життєвий цикл «речі» складається з трьох фаз: фази завантаження, експлуатації та обслуговування. Фаза початкового завантаження стосується будь-яких завдань обробки, необхідних перед тим, як мережа почне працювати. В роботі [49] визначено, що цей процес включає низку налаштувань, які передаються між вузлами, які не мають жодних попередніх знань один про одного. Етап завантаження пристрою завершується, коли всі параметри безпеки (наприклад, секретні ключі) безпечно передано на пристрій. Розглянемо рішення безпеки, які запропоновано для безпечного процесу завантаження. Терміни та визначення, які далі використовуються, представлено в табл. 2.3.

Таблиця 2.3. – Скорочення та позначення.

Абревіатура	Визначення
WSN	Бездротова сенсорна мережа
PKC	Криптографія з відкритим ключем
KDC	Центр розподілу ключів
6LBR	Прикордонний маршрутизатор 6LoWPAN
PKG	Генератор закритих ключів
DH	Обмін Діффі-Хеллмана
IBE	Шифрування на основі ідентифікації
ECC	Криптографія еліптичної кривої
ECDH	Еліптична крива обміну Діффі–Хеллмана

Спочатку опишемо еталонну модель, яка ілюструє сценарій, у якому розглянуті протоколи безпеки можуть бути розгорнуті. Потім представимо в класифікацію протоколів безпеки, засновану на механізмі завантаження ключа, і порівняємо представлену класифікацію з відповідними роботами.

Розгляд сценарієв. Протоколи безпеки, як показано в рис. 2.1, включають два об’єкта. Принаймні один із них є пристроєм із обмеженими ресурсами, тоді як другий об’єкт можна розглядати як інший обмежений пристрій або зовнішній Інтернет-сервер (тобто з багатими ресурсами). Розглянута мережа «речей» складається з кількох крихітних вузлів, які спілкуються один з одним і з необмеженим маршрутизатором кордону ресурсів (6LBR). 6LBR є мостом між сенсорним вузлом і зовнішнім світом. 6LBR може брати участь у комунікації між двома об’єктами пасивним (прозорим для сторін, що спілкуються) або активним (як посередник у процесі комунікації) способом. Опишемо забезпечення одноадресного зв’язку між двома об’єктами. Зауважимо, що групове спілкування не розглядається.

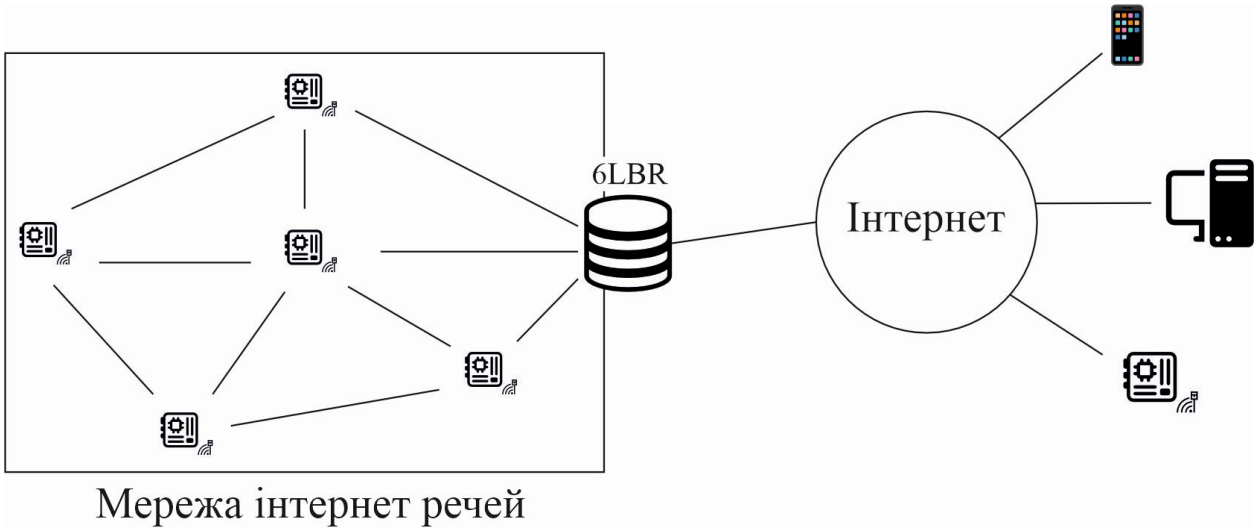


Рисунок 2.1 – Архітектура мережі розглянутого сценарію

Класифікація. Існуючі рішення безпеки для IoT класифікуються на два основні типи: рішення, які покладаються на асиметричні схеми ключів, і рішення, які попередньо розподіляють симетричні ключі для завантаження безпечного зв'язку. Опишемо два перші рівні запропонованої таксономії.

Схеми асиметричного ключа (AKS): схеми ключа, засновані на асиметричній криптографії, також відомі як криптографія з відкритим ключем (PKC), вважаються дуже поширеним підходом для встановлення безпечного зв'язку між двома (або більше) сторонами. Вони використовують асиметричні алгоритми і широко розгорнуті в традиційному Інтернеті. Застосовність AKS в Інтернеті речей має одну важливу незручність, яка полягає у вартості обчислень і споживанні енергії. Незважаючи на дорогі операції, багато дослідників все ще прагнуть застосувати AKS в контексті IoT. Запропоновані підходи можна розділити на дві категорії: транспортування ключів на основі шифрування з відкритим ключем і узгодження ключів на основі асиметричних методів.

Передача ключів на основі шифрування з відкритим ключем: Подібно до традиційного механізму транспортування ключів, перша категорія вимагає від відкритого ключа безпечного транспортування інформації. Для IoT були запропоновані різні методи встановлення ключів, починаючи від використання необробленого відкритого ключа до складних реалізацій у стандарті X.509.

Угода ключів на основі асиметричних методів: Друга категорія заснована на асиметричних примітивах, у яких спільний секрет виводиться двома або більше сторонами. У цій категорії важливим є протокол DH [51] та його варіанти, про які буде згадано пізніше.

Схеми попереднього розподілу симетричних ключів: на додаток до асиметричних підходів, дослідники пропонують також кілька методів, що використовують механізми встановлення симетричних ключів для завантаження безпечного зв'язку в IoT. Симетричні підходи часто припускають, що вузли, залучені до встановлення ключа, мають спільні облікові дані. Попередні спільні облікові дані можуть бути симетричним ключем або деякими випадковими байтами, запрограмованими в датчик перед його розгортанням. Цю категорію можна розділити на дві основні підкатегорії:

Розподіл імовірнісних ключів: ця підкатегорія стосується механізмів, які розподіляють облікові дані безпеки (ключі, випадкові байти), вибрані випадковим чином із пулу ключів, до обмежених вузлів. Під час початкового зв'язку кожні два вузли можуть з певною ймовірністю виявити загальний ключ для встановлення безпечного зв'язку.

Детермінований розподіл ключів: у цій підкатегорії застосовано детермінований дизайн для створення пулу ключів і рівномірного розподілу ключів таким чином, щоб кожні два вузли мали спільний ключ.

На рис. 2.2 узагальнено розглянуту таксономію. Кожний клас рішень безпеки має свої власні переваги та недоліки, як це буде розглянуто далі.

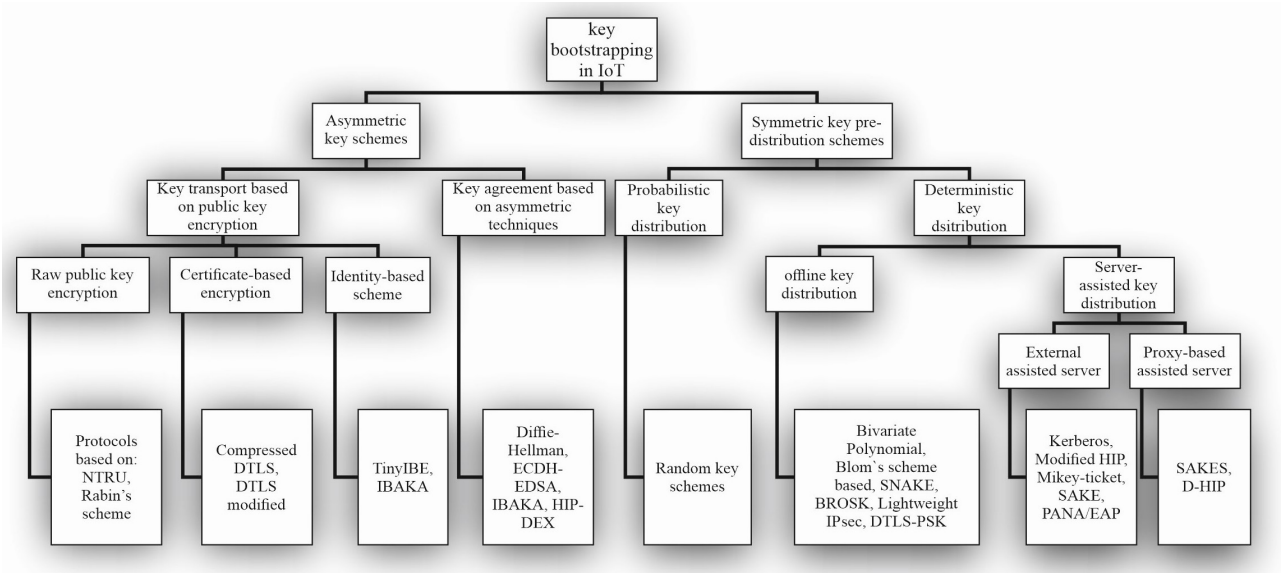


Рисунок 2.2 – Класифікація механізмів завантаження ключів в IoT

Підходи до класифікації було запропоновано в роботах [52-55]. В [52], автори пропонують кілька способів класифікації підходів до встановлення ключів, наприклад, на основі використовуваного методу автентифікації або основного криптографічного примітиву. В [53] надано детальну класифікацію симетричних протоколів розподілу ключів для двох різних сценаріїв: розподілених та ієрархічних WSN. У кожному сценарії автори аналізують різні механізми встановлення парних і групових ключів між сенсорними вузлами. Подібним чином в [55] пропонують класифікацію симетричних протоколів керування ключами в WSN, але на основі структури мережі та ймовірності спільного користування ключами між парою сенсорних вузлів. В цих роботах на самому першому рівні розрізняють централізовані схеми розподілу ключів. На другому рівні вони забезпечують іншу диференціацію на основі імовірнісних і детермінованих механізмів встановлення ключів. В [54] надано високорівневу класифікацію на основі систем управління ключами (KMS), а саме: структура пулу ключів, математична структура, структура переговорів і структура відкритого ключа. Зроблено висновок, що криптографія з відкритим ключем може бути життєздатним рішенням для сенсорних вузлів, які працюють як клієнтські вузли (в моделі клієнт-сервер). Для серверних вузлів KMS поліноміальна схема забезпечує кращу продуктивність. Вищезазначені підходи

недостатньо висвітлюють можливі механізми розподілу ключів (асиметричні та симетричні методи), наприклад, досліджуються лише симетричні підходи [53, 55]. Крім того, вони забезпечують неоднорідні класифікації через незв'язані різні критерії, як у [54, 55].

Беручи до уваги описані вище класифікації, розглянута таксономія охоплює асиметричні механізми розподілу ключів для IoT на додаток до симетричних підходів. Таксономія розмічає різні протоколи за схемою встановлення ключа, яка використовується для встановлення секретного ключа сеансу: асиметричні або симетричні методи. Як згадувалося раніше, не розглядаються протоколи, які встановлюють групові ключі між сенсорними вузлами, які описано в [53], а розглядаються лише попарні розподіли ключів. Розглянута таксономія має високий ступінь класифікації, що веде до більш глибокої оцінки протоколу. Наприклад, в асиметричному підході не лише обговорюється застосовність криптографії з відкритим ключем у контексті IoT, як описано в [54], але також розрізняються різні асиметричні схеми ключів на основі схеми доставки ключів (транспортування ключів або угода ключів). У симетричних схемах попереднього розподілу ключів протоколи діляться на дві категорії: імовірнісний і детермінований розподіли ключів. Проте, у детерміністському підході розрізняють протоколи, у яких сервер(и) беруть участь у процесі узгодження ключа і протоколи, які не залежать від будь-якої третьої сторони на етапі встановлення ключа.

2.3 Схеми асиметричних ключів

Позиція асиметричної криптографії або PKC зрозуміла в звичайному Інтернеті. Однак це не так у контексті IoT через дорогі операції шифрування та перевірки. Проте розробка та впровадження PKC в IoT ніколи не припинялися. Насправді нові вдосконалення кількох примітивів (наприклад, ECC, NTRU)

продовжують знижувати вартість криптографічних операцій, тому підхід PKC викликає зростаючий інтерес для обмежених середовищ. Далі демонструються різні можливі форми асиметричних схем ключів в IoT.

Передача ключів на основі шифрування з відкритим ключем. Ця підкатегорія розглядає схеми встановлення ключів, де відкритий ключ використовується для передачі секретних даних або узгодження ключа сеансу. Для створення пари відкритих і закритих ключів використовується кілька методів. У цій підкатегорії класифікуємо ці механізми на основі методів генерації відкритих/закритих ключів.

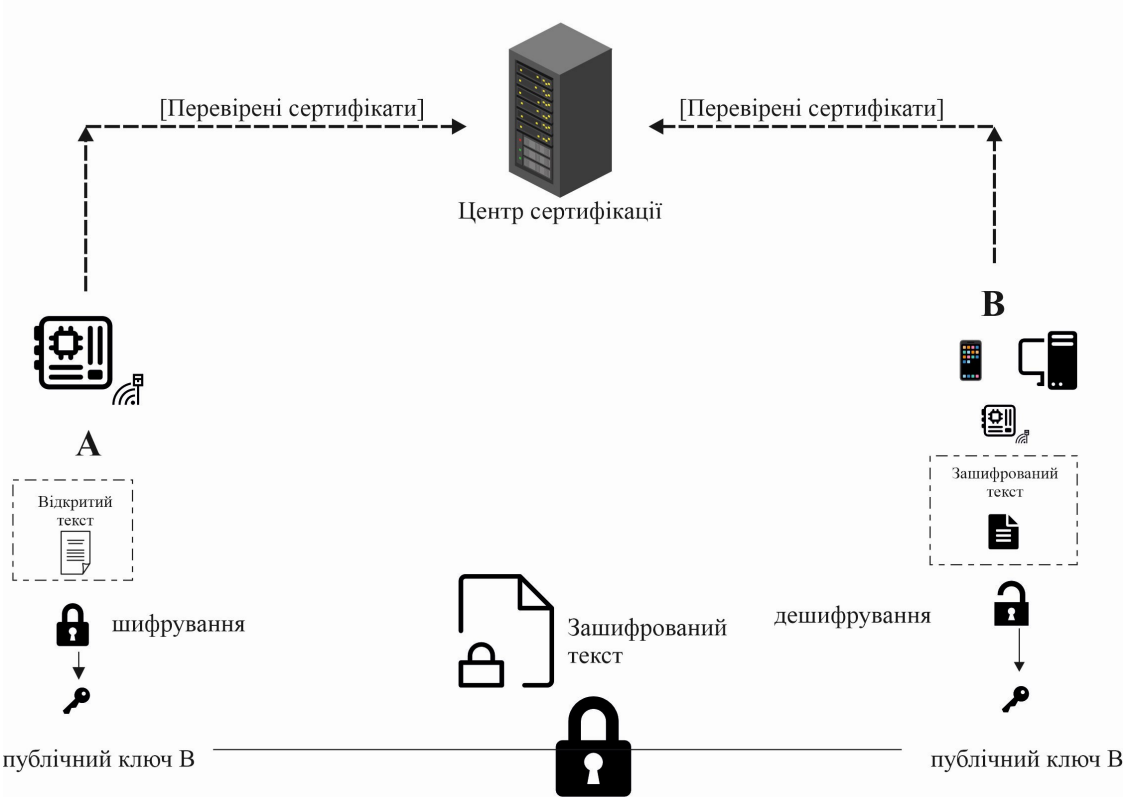


Рисунок 2.3 – Механізм транспортування відкритих ключів

На рис. 2.3 наведено приклад сценарію зв’язку між двома об’єктами A та B. У цьому сценарії A та B можуть безпосередньо використовувати відкриті ключі для створення зашифрованого каналу. Центр сертифікації (CA) може брати участь у перевірці ідентичності передавача повідомлень, якщо підтримуються сертифікати. Цей метод може бути дорогим для вузлів датчиків з обмеженими ресурсами, зокрема при використанні традиційного алгоритму,

такого як RSA. Без перевіреного зв’язку між відкритим ключем та ідентифікатором (тобто криптографії на основі ідентифікатора, ідентифікатора на основі криптографії або з посередництвом ЦС) цей підхід стає вразливим до атаки «людина посередині». Дійсно, як A, так і B не можуть підтвердити автентичність один одного. Зловмисник може генерувати будь-які відкриті/закриті ключі та видавати себе за A під час спілкування з B.

Необроблене шифрування з відкритим ключем. Деякі механізми припускають, що відкритий ключ було розповсюджено заздалегідь або за допомогою позасмугового зв’язку. Ці механізми пропонують невелику кількість обмінів повідомленнями, але вони не масштабовані, оскільки відкриті ключі всіх пристроїв повинні бути відомі кожному пристрою.

Деякі механізми «необробленого шифрування відкритим ключем», тобто схема Рабіна [56] або NtruEncrypt [57] були рекомендовані для WSN.

Схема Рабіна дуже схожа на алгоритм RSA (широко використовувана криптосистема), яка також базується на складності проблеми факторизації. Насправді схема вимагає такого ж споживання енергії для операцій дешифрування, як RSA з таким самим рівнем безпеки. Проте, він пропонує набагато швидший механізм для операцій шифрування, оскільки для шифрування повідомлення потрібен лише один квадрат.

NtruEncrypt — це криптосистема, яка є альтернативою примітивів RSA та ECC (Криптографія з еліптичною кривою) на основі решітки. Механізм є високоефективним і підходить для пристроїв з найбільш обмеженими ресурсами, таких як смарт-картки та RFID-мітки. В [57] автори порівнюють три запропоновані механізми PKC обмежені пристрої: схема Рабіна, NtruEncrypt і ECC. Результати показують, що NtruEncrypt забезпечує найменше середнє енергоспоживання. Проте, ця криптосистема часто вимагає повідомлень великого розміру, що може призвести до фрагментації пакетів на нижчих рівнях і багатьох повторних передач за наявності помилок зв’язку.

Протоколи, які базуються на «необробленому шифруванні з відкритим ключем» вимагають невеликої кількості обмінених повідомлень; це насправді вигідно, якщо потужність передачі є найважливішим і обмежуючим фактором.

Шифрування на основі сертифіката. Протоколи на основі сертифікатів є популярним вибором для встановлення безпечного зв'язку між двома об'єктами через Інтернет. Довірчі відносини між двома об'єктами гарантуються відомою третьою стороною (ЦС) за допомогою стандартного сертифіката X.509, який підтверджує особу об'єкта, як показано на рис. 2.3. Дійсно, кожен сенсорний вузол має сертифікат, підписаний довіреним ЦС. Останній може бути завантажений у вузол перед розгортанням або може бути отриманий безпосередньо за запитом від довіреної сторони.

TLS [58] рекомендовано багатьма стандартами, визначеними IETF (Internet Engineering Task Force) для послуг безпеки. Проте, TLS не є гарним вибором щодо найкращих практик безпеки в IoT. Насправді TLS нормально працює в надійному транспортному протоколі, такому як TCP, який не підходить для пристроїв з обмеженими ресурсами через його алгоритм контролю перевантаження. На заміну TLS у жорстко обмежених середовищах нещодавно було запропоновано протокол DTLS (Datagram Transport Layer Security). Він працює через ненадійний транспортний протокол, тобто UDP, і забезпечує такий самий високий рівень безпеки, як і TLS.

Використання сертифіката в основному є дороговартісним. Щоб зменшити енергоспоживання, дослідники розглядають вдосконалення апаратного та програмного забезпечення.

Використання криптографічних апаратних прискорювачів: апаратні прискорювачі відповідають за всі криптографічні обчислення. В [59] запропоновано метод реалізації DTLS з використанням апаратної допомоги на сенсорних вузлах. Рішення передбачає, що кожен датчик оснащений TPM (Trusted Platform Module). TPM – це вбудований чіп, який забезпечує безпечне створення криптографічних ключів і запечатане зберігання, а також апаратну підтримку криптографічних алгоритмів. Повністю автентифіковане

рукостискання можна здійснити між датчиком (обладнаним TPM) і абонентом (іншим датчиком або зовнішнім об’єктом). І датчик, і абонент передають свій сертифікат X.509 для початку фази автентифікації. Ці сертифікати підписані довіреним центром сертифікації та включені в повністю автентифіковане рукостискання DTLS. Це рішення не тільки має високий рівень безпеки, встановлюючи довірчі відносини за допомогою схваленої третьої сторони, але також забезпечує цілісність, конфіденційність і автентичність повідомлень із доступним енергоспоживанням, наскрізною затримкою та накладними витратами пам’яті.

Проте, такий підхід є дорогим і складним щодо розгортання апаратного прискорювача біля кожного датчика, особливо для великої кількості датчиків.

Оптимізація існуючих протоколів (реалізація програмного забезпечення): протокол безпеки, що використовує сертифікати, створено для забезпечення вищої продуктивності без впливу на надійність протоколу. В [60] запропоновано модифікацію DTLS з використанням механізму стиснення 6LoWPAN [61]. Модифікований протокол зменшує розмір деяких заголовків (тобто заголовок запису DTLS, заголовок рукостискання, повідомлення рукостискання). Ці зміни покращують продуктивність DTLS з точки зору розміру пакета, споживання енергії, часу обробки та часу відповіді мережі. Однак запропоноване рішення не передбачає зворотної сумісності з фактичним стандартом DTLS, зокрема щодо стиснення заголовка.

В роботі [62] запропоновано ідею дизайну для ефективного зменшення накладних витрат на квиткування DTLS. Процедура повного рукостискання вимагає 15 обмінів повідомленнями, високої динамічної пам’яті (RAM) під час спілкування та тривалого часу обробки для криптографічних задач. Щоб пом’якшити незручність повного рукостискання, пропонується делегувати процедуру рукостискання rich-resource, наприклад, шлюз або власника пристрою. Усі задачі, пов’язані із сертифікатом, виконуються в сутності багатих ресурсів, і лише повідомлення про стан сеансу надсилається на обмежений пристрій. Після цього сеанс можна встановити за допомогою цього

повідомлення без додаткових розрахунків. Цей модифікований DTLS може значно зменшити витрати зв'язку за умови, що сервер із багатими ресурсами є довіреним.

В роботі [63] представлено модифікації, подібні до DTLS, але рукостискання DTLS здійснюється через прикордонний маршрутизатор 6LoWPAN (6LBR). 6LBR бере участь у безпечному зв'язку, але є прозорим для сенсорних пристроїв та Інтернет-хосту. Прикордонний маршрутизатор перехоплює та пересилає пакети на транспортному рівні. З точки зору Інтернет-хосту, він спілкується з 6LBR за допомогою традиційного протоколу DTLS, де автентифікація підтримується сертифікатом на основі ECC. З іншого боку, 6LBR працює в режимі безпеки попереднього спільного ключа для зв'язку з обмеженими сенсорними пристроями. Крім того, 6LBR автентифікує вузли за допомогою механізму, описаного в [64]. Якщо автентифікація пройшла успішно, генерується секретний ключ сеансу для захисту зв'язку між сенсорними пристроями та 6LBR. Насправді він використовується для шифрування попереднього головного ключа в повідомленні *ClientKeyExchange* , яке Інтернет-хост надсилає до 6LBR. Коли попередній головний секретний ключ успішно обчислюється в Інтернет-хості та сенсорному пристрої, вмикається наскрізна безпека DTLS. Запропонована архітектура делегує всі дорогі операції (обчислення ECC, узгодження ключів) прикордонному маршрутизатору, щоб забезпечити кращий термін служби сенсорних пристроїв. Проте, 6LBR вважається єдиною точкою відмови.

Протокол IKE [65] зазвичай працює спільно з IPsec, щоб забезпечити асоціації безпеки (SA) між двома об'єктами. У цього протоколу є варіант, у якому активована взаємна автентифікація за допомогою сертифікатів на основі RSA. В [65] пропонують інший варіант для IKE, який базується на сертифікаті відкритого ключа на основі ECC для автентифікації та ECDH для узгодження ключів замість протоколу RSA та DH. Пропозиція зменшує вартість обчислень, оскільки вона в основному обмежена операціями множення точок і вимагає меншого розміру ключа, ніж RSA для того самого рівня [66].

Схеми на основі ідентифікації (IBS). Перша реалізація криптографії на основі ідентифікації була розроблена Шаміром [67]. Цей тип криптографії визначає добре відомий рядок (ідентифікатор), що представляє особу чи організацію, який використовується як відкритий ключ. Відкритий ключ кожної сутності генерується з її відкритого ключа довіреною стороною (рис. 2.4), яка називається генератором відкритих ключів (PKG). Це рішення усуває потребу в сертифікатах, що робить рішення вигідним, особливо для WSN. Дійсно, будь-які сенсорні вузли можуть просто згенерувати відкритий ключ інших вузлів, коли це необхідно для встановлення безпечного зв'язку, використовуючи їхні ідентифікатори. Крім того, механізм відкликання підтримується переглядом списку дійсних ідентифікаторів датчиків. Однак схеми на основі ідентифікатора вразливі до атак депонування ключів, оскільки PKG знає закриті ключі всіх вузлів у мережі. Він може імітувати будь-який вузол і, отже, перехоплювати весь трафік у системі. Таким чином, PKG завжди вважається добре захищеним і надійним для всіх вузлів мережі.

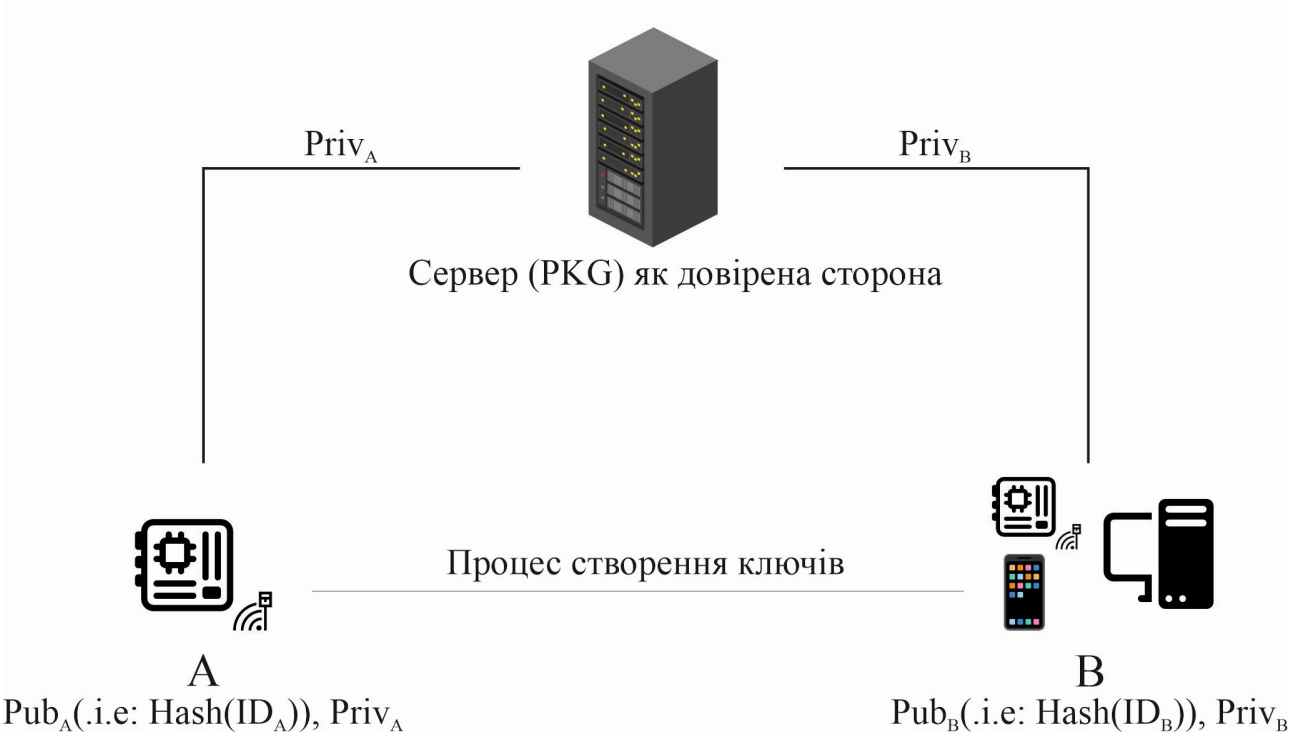


Рисунок 2.4 – Криптографічна інфраструктура на основі ідентифікації

У обмеженому середовищі парадигма IBE здебільшого реалізується за допомогою примітиву ECC [63,68]. Існують реалізації на інших примітивах, наприклад, RSA або ElGa - IBE mal-типу [69]. Проте, вони занадто дорогі для обмежених вузлів, тому що вони засновані на операціях піднесення до степеня з великим показником степеня. В [68] запропоновано IBAKA – схему IBE, засновану на результатах [70]. Однак вони адаптують метод IBE до ECDH [71], здійснюючи обмін ключами для встановлення сеансового ключа. Їхня пропозиція все ще вимагає 2 білінійних пар і 3 скалярних множень кожного разу, коли секретний ключ завантажується.

В [72] запропоновано TinyIBE – дуже простий розподіл автентифікованих ключів на основі IBE для гетерогенних сенсорних мереж. Схема не вимагає розрахунку пар. Він здатний отримати ключ сеансу для двох вузлів лише після 2 обмінів повідомленнями.

Узгодження ключів на основі асиметричних прийомів. Ця підкатегорія стосується протоколів узгодження ключів на основі асиметричних примітивів в IoT. Як згадувалося в різних дослідницьких роботах, протокол угоди обміну ключами — це механізм, за якого дві (або більше) сторони отримують спільний секрет, і жодна інша сторона не може заздалегідь визначити секретне значення. Рис. 2.5 ілюструє процес типового узгодження асиметричного ключа. Km – це секрет, згенерований після процедури узгодження. Потім цей симетричний ключ використовується для захисту зв’язку.

Протокол Діффі-Хеллмана (DH). [51] та його варіанти є класичними прикладами узгодження симетричних ключів. Проте протоколи DH вважаються дорогими та непридатними для обмежених вузлів, зокрема, для класів 0 та 1 відповідно до класифікації вузлів з точки зору ресурсної потужності в lwig – термінології [73].

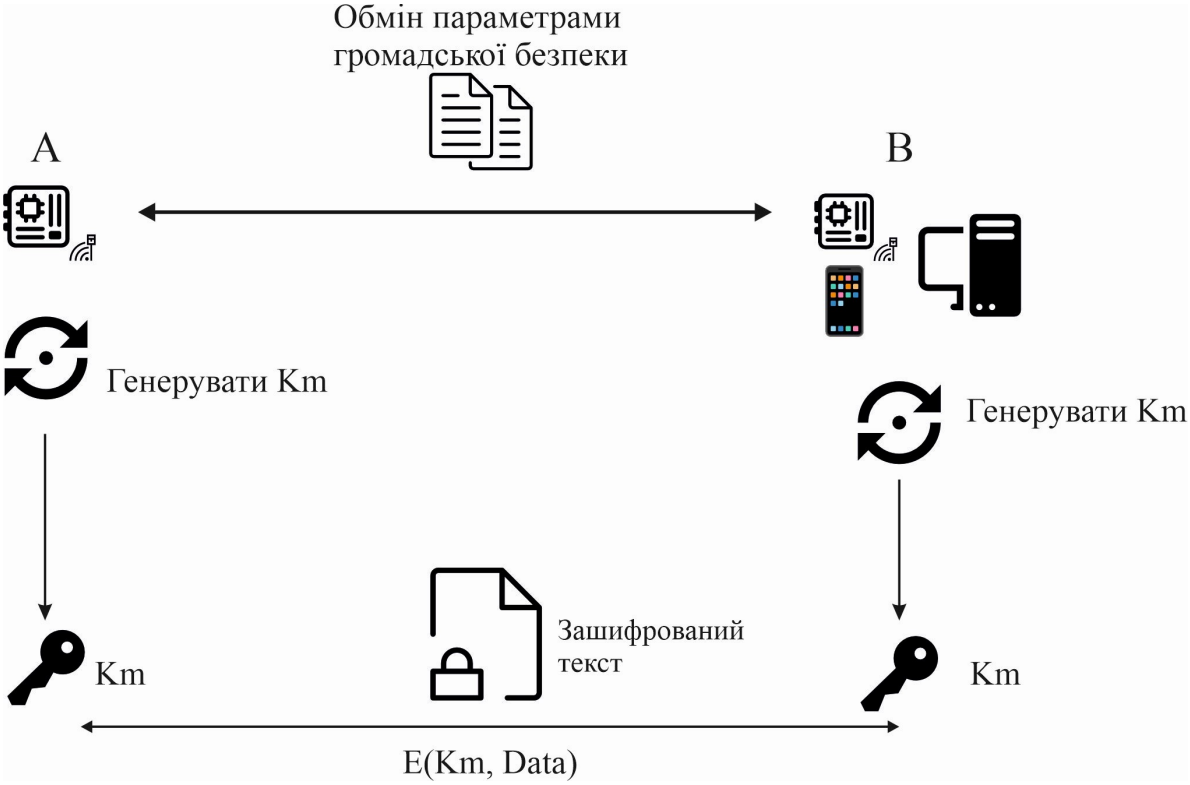


Рисунок 2.5 – Угода обміну ключами на основі асиметричних механізмів

Деякі варіанти протоколу DH розглядаються в обмежених середовищах з використанням ECC, тобто ECDH. Криптографічний примітив ECDH пропонує менший розмір ключа, ніж RSA. Дійсно, у [66] показано, що для досягнення рівня безпеки 128-бітного розміру ключа AES можна віддати перевагу 256-бітному розміру ключа з використанням еліптичної кривої замість 3072-бітних параметрів у протоколі RSA та DH. Як приклад, в [71] було реалізувано протокол узгодження ключів на основі ECDH, що забезпечує автентифікацію за допомогою алгоритму цифрового підпису еліптичної кривої (ECDH-ECDSA).

Практичні вимірювання на датчиках MICAz і TelosB показали, що ECDH-ECDSA доступний з точки зору складності обчислень.

IBAKA [68] пропонує комбінацію ECDH та IBE для сенсорних мереж. Схема базується на протоколі ECDH і додатково забезпечує конфіденційність обміну повідомленнями за допомогою схеми на основі ідентичності [70].

HIP-DEX (Host Identity Protocol Diet Exchange) [74] також застосовує протокол DH для генерації ключа сеансу між двома об’єктами лише після

обміну 4 повідомленнями. Цей протокол є варіантом HIP Base Exchange [75] і спеціально розроблений для зменшення складності криптографічних обчислень. Він використовує найменший можливий набір криптографічних примітивів (наприклад, AES-CBC замість криптографічних хеш-функцій), видаляє цифрові підписи та реалізує статичний ECDH для шифрування ключа сеансу тощо. Цей протокол значною мірою враховується в контексті IoT багатьма останніми роботами [75, 76]. Наприклад, в [76] пропонують ефективний механізм доступу до мережі на основі HIP-DEX для мобільних вузлів, які приєднуються до локальної сенсорної мережі. Крім того, в [75] адаптувано HIP-DEX до IoT, зокрема, механізм відновлення сесії, як у TLS [77]. Таким чином, обмежений вузол виконує дорогі операції один раз і підтримує стан сеансу для повторної автентифікації та відновлення безпечного каналу.

Протоколи узгодження ключів на основі DH вимагають менше повідомлень для встановлення ключа сеансу, але обчислювальні задачі на сенсорних вузлах зазвичай складні.

2.4 Симетричні схеми попереднього розподілу ключів

У цій підкатегорії сторони, що спілкуються, часто спочатку спільно використовують деякі облікові дані перед завантаженням зв’язку. Механізми попереднього розподілу ключів можуть відрізнятися, як це описано в наступних розділах.

Імовірнісний розподіл ключів. Механізм попереднього розподілу випадкових ключів (RKP) був вперше запропонований в [78]. Типовий RKP складається з трьох етапів: попереднє розповсюдження ключа, виявлення спільного ключа та встановлення ключа шляху. У схемі створюється великий пул ключів. Потім ключі випадковим чином вибираються з пулу ключів і розподіляються між сенсорними вузлами. Будь-які два вузли можуть мати

загальний ключ з певною ймовірністю. Третя фаза запускається, коли два вузли не мають спільного ключа. У цьому процесі один вузол спочатку генерує випадковий ключ K. Потім він надсилає ключ своїм сусідам, використовуючи попередньо встановлений безпечний канал. Процес триває до тих пір, поки ключ K не надійде на інший вузол. Згодом K розглядається як попарний ключ між обома вузлами.

Кілька рішень засновано на схемах, описаних в [79-82]. Ці пропозиції особливо покращують фазу попереднього розповсюдження, щоб покращити зв'язок ключів між вузлами та зменшити простір пам'яті, необхідний для зберігання ключів. Фактично, в [79] запропоновано схему попереднього розподілу ключів, яка спирається на знання про розгортання та уникає непотрібних призначень ключів. В [81] розроблено схему на основі підходу з роботи [79], але ключі відображаються на двовимірних позиціях. Запропоновано функцію густини ймовірності, яка забезпечує кращу зв'язність ключів. В [80] також представлено механізм для посилення фази встановлення ключа шляху. Основна ідея полягає в тому, що вузол A знаходить усі можливі посилання на вузол B. Він генерує для кожного посилання випадкове значення та направляє ці значення до B. Загальні ключі між A і B захищені цими випадковими значеннями. Згенерований ключ буде спільний для обох вузлів, якщо зловмиснику не вдасться підслухати всі шляхи між ними.

Імовірнісний розподіл ключів зазвичай не гарантує встановлення ключа сеансу між усіма вузлами навіть на етапі встановлення ключа шляху. З певною ймовірністю два вузли можуть не мати спільних ключів.

Детермінований розподіл ключів. У цій підкатегорії описані схеми ключів покладаються на детермінований процес для створення пулу ключів і розповсюдження ключів між вузлами, щоб гарантувати безпечне повне підключення в мережі. У детермінованих рішеннях схеми ключів розрізняються за наявністю чи ні довіреної третьої сторони під час початкового завантаження ключа.

Офлайн роздача ключів. Офлайн-метод розподілу ключів широко використовується в WSN через його простоту. Залежно від протоколу, який використовується, кожен вузол в одній мережі може мати спільний мережевий ключ або кожні два вузли можуть мати загальний попарний ключ. Потім ключ сеансу генерується після кількох обмінів даними без присутності третьої сторони. Офлайн-розподіл ключів забезпечує ефективність з точки зору споживання енергії, оскільки не вимагає дорогих криптографічних обчислень, таких як асиметричні підходи. Проте, коли сенсорний вузол фізично атакують, секретні дані, що зберігаються всередині вузла, можуть бути розкриті. Таким чином, зловмисник може отримати доступ до кількох вузлів, які мають спільний секретний ключ з атакованим вузлом, або в гіршому випадку він може отримати доступ до всієї мережі.

У деяких роботах математичні властивості були застосовані для створення моделі для забезпечення обміну ключами між сенсорними вузлами. Ці механізми все ще застосовуються в контексті IoT. Найбільш відомі схеми засновані на двовимірних поліномах [83, 84]. У цих схемах вузол A ділиться з іншими вузлами біваріантним поліномом n-ступеню $f(x, y)$. A може отримати попарний ключ з іншим вузлом B шляхом обчислення значення $f(Id_A, Id_B)$, де Id_A та Id_B є відповідними ідентифікаторами A та B. Таким же чином B може отримати такий саме парний ключ, оскільки $f(Id_A , Id_B)$ дорівнює $f(Id_B, Id_A)$. В іншій схемі, яка називається схемою Блума [85], секретна симетрична матриця D генерується зі спільного секретного ключа між двома вузлами A і B. Кожен з них генерує відкриту матрицю I_A і I_B відповідно для A і B. Приватні ключі є відповідно $priv_A = D \times I_A$ та $priv_B = D \times I_B$ для A і B. Нарешті, парний ключ обчислюється шляхом розвязання рівняння ($priv_A \times I_B$) або ($priv_B \times I_A$). Проблема з цими двома останніми схемами полягає в тому, що ключ сеансу залишатиметься незмінним для кожних двох вузлів.

SNAKE [86, 87] і BROSK [88] – це дві схеми встановлення ключів, де ключ сеансу генерується без необхідності використання сервера ключів для керування ключами. Ці два протоколи припускають, що всі вузли в одній

мережі спільно використовують головний секретний ключ. У SNAKE ключ сеансу отримується шляхом хешування двох випадкових одноразових номерів, згенерованих кожною стороною, що спілкується, за допомогою попереднього спільного ключа. BROSK транслює повідомлення узгодження ключа, що містить nonce. Після того, як вузол отримує повідомлення від своїх сусідів, він може створити ключ сеансу, обчисливши код автентифікації повідомлення (MAC) з двох одноразових номерів.

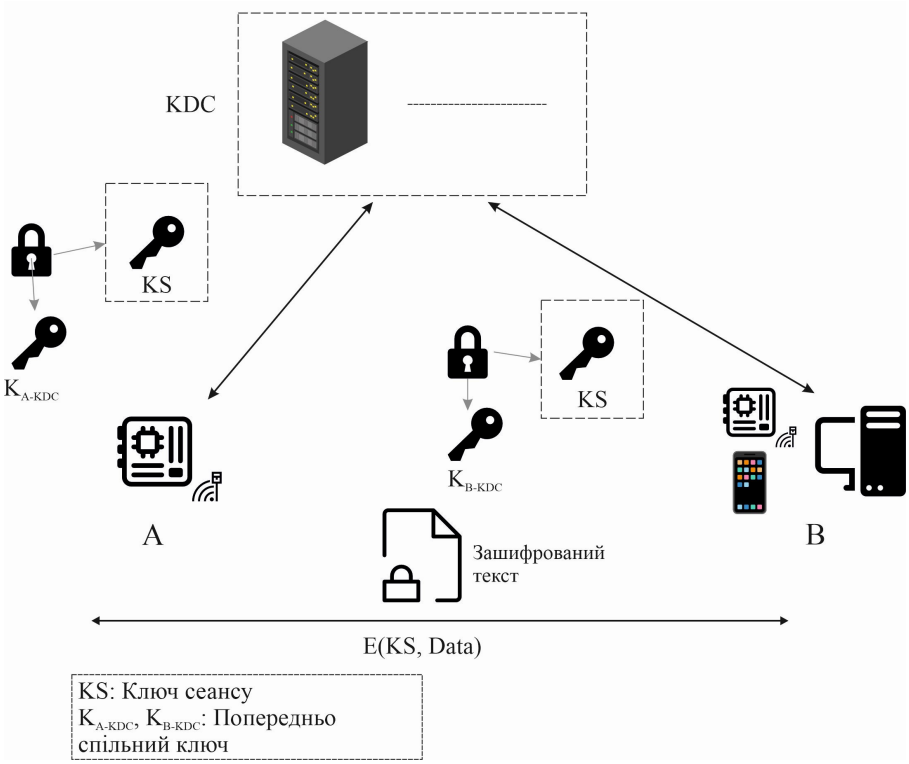


Рисунок 2.6 – Серверний механізм

В [50] реалізують стандартний протокол безпеки Інтернету IPsec у WSN на основі IP (з використанням 6LoWPAN). Запропоновано механізми стиснення заголовків AH і ESP для інтеграції IPsec з рівнем 6LoW-PAN, але вони зберігають прийнятний розмір пакета. Механізми AH і ESP забезпечують автентичність походження, цілісність повідомлень і захист конфіденційності IP-пакетів, але вони не обробляють обмін ключами. Асоціації безпеки встановлюються вручну за допомогою спільного ключа.

Офлайн-розповсюдження ключів не забезпечує операції зміни ключів. Коли система змінює на інші секретні ключі, усі об’єкти в мережі потрібно оновити, щоб встановити безпечний зв’язок за допомогою нових ключів.

Розподіл ключів за допомогою сервера. Через обмеження ресурсів обмежених пристроїв криптографічне обчислення та інші дорогі завдання (наприклад, керування ідентифікацією, генерація ключів) можна виконувати на серверах із багатими ресурсами. У цьому відношенні в IoT були запропоновані підходи з підтримкою сервера для ключових протоколів встановлення. У таких протоколах в обміні повідомленнями беруть участь два об’єкти та один (чи більше) довірений сервер. Сервер ділиться довгостроковим ключем *anpiopi* з кожною комунікаційною сутністю. Він часто відіграє роль центру розповсюдження ключів (KDC), а потім надає ключ сеансу кожній стороні, повторно шифруючи його за допомогою спільних ключів, як показано на рис. 2.6.

1. Зовнішній підтримуваний сервер. У цій підкатегорії допоміжні об’єкти – це зовнішні сервери з багатими ресурсами, розташовані за межами WSN. У результаті вони можуть обробляти розподіл ключів одного або кількох WSN.

Другий підхід запропоновано в [59] з урахуванням пакету шифрів TLS Pre-Shared Key [77]. Перед розгортанням кожен датчик має попередньо встановити кілька випадкових байтів, які називаються протокеями. Ці випадкові байти використовуються для отримання ключа PSK (попередньо спільного ключа) для кожного сеансу. Замість використання TPM для створення асоціації безпеки між сенсорним вузлом і абонентом використовується центральний сервер із багатими ресурсами. Протокові ключі також відомі довіреному серверу. Потім сервер генерує той самий ключ сеансу для абонента з протокових ключів.

MIKEY-Ticket [89] – це додатковий режим до основного протоколу MIKEY [90], у якому KDC бере участь у процесі встановлення асоціації безпеки між двома сторонами. MIKEY-Ticket виник із концепції квитка [64]. KDC

безпечно зв’язується з вузлом, який ініціює протокол (ініціатор), і вузлом-відповідачем (відповідач), шифруючи важливі дані за допомогою попереднього спільного головного ключа, спільного з кожним вузлом. Проте, цей протокол вразливий до атак на відмову в обслуговуванні (DoS), зокрема відтворення повідомлень відповідачу. Щоб запобігти цим атакам, в [91] пропонують нове встановлення ключа під назвою SAKE (Sever Assisted Key Establishment) на основі режиму MIKEY-Ticket, що усуває загрозу атак DoS. SAKE дозволяє встановлювати асоціації безпеки між двома сторонами лише після п’яти обмінів повідомленнями, порівняно з шістьма повідомленнями в оригінальному квитку MIKEY. Дійсно, після отримання першого повідомлення від ініціатора KDC генерує ключ сеансу та зв’язується безпосередньо з відповідачем. Ця зміна скорочує один обмін повідомленнями порівняно з MIKEY-квитком. Крім того, оскільки кожне повідомлення SAKE містить MAC, обчислений за допомогою ключа, спільного з одержувачем, DoS-атака пом’якшується.

Інші IoT-рішення розподілу ключів на основі зовнішнього сервера включають рішення, які реалізують протокол PANA (Protocol for Carrying for Network access) [92]. PANA працює через UDP і використовує EAP [93] (Extensible Authentication Protocol) для автентифікації, яка підтримує кілька методів автентифікації, включаючи розповсюдження попереднього спільного ключа. В [94] запропоновано удосконалення PANA для адаптації ресурсних обмежень. Основні модифікації полягають у зменшенні кількості обмінів повідомленнями (наприклад, вибір EAP-PSK як єдиного методу автентифікації), видаленні невикористаних полів заголовка PANA, мінімізації збору криптографічних примітивів на пристрої з обмеженнями. Ці пропозиції можуть ефективно зменшити розмір коду впровадження PANA на пристрої, але автори не дають оцінки виграшу, який можна отримати, наприклад, з точки зору споживання енергії або часу відгуку мережі.

2. Допоміжний сервер на основі проксі. Для цієї підкатегорії не потрібен зовнішній сервер, а проксі-сервер (PBS), розташований у WSN, як показано в рис. 2.7. Цей сервер оснащений достатніми ресурсами та ємністю

для зберігання всіх дорогих завдань для обмежених вузлів. Він часто відіграє роль посередника для асоціації сенсорних вузлів та інших об'єктів. Крім того, PBS зазвичай спільно використовують симетричний секретний ключ із обмеженими вузлами та маршрутизатором 6LBR.

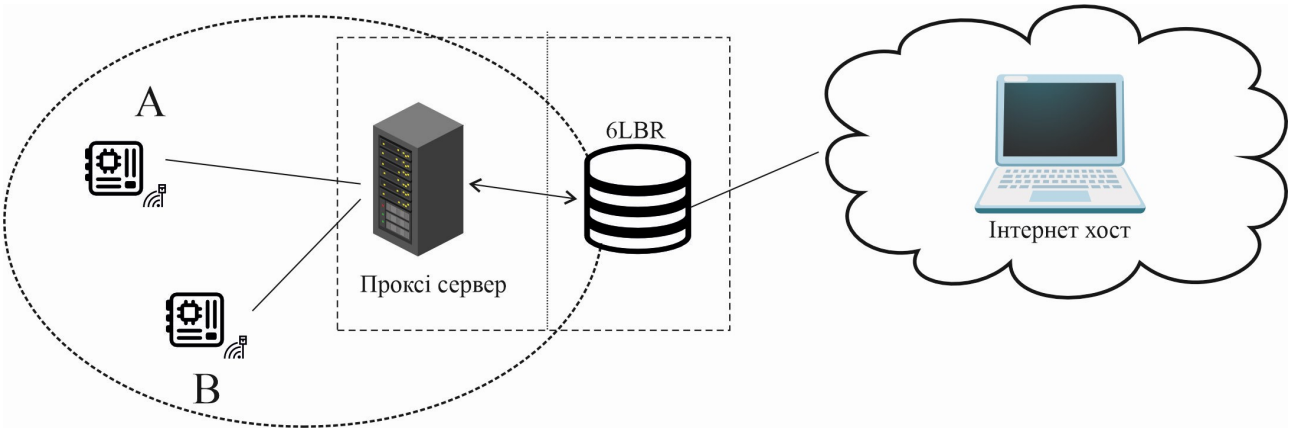


Рисунок 2.7 – Серверна інфраструктура на основі проксі

Використовуючи ті самі міркування, в [95] запропоновано SAKES, що забезпечує безпечну автентифікацію та встановлення ключа між сенсорним вузлом і зовнішнім хостом Інтернету. Після отримання запиту на вузол датчика PBS аутентифікує вузол датчика за допомогою 6LBR. Потім він застосовує механізм узгодження ключа DH з віддаленим сервером і обчислює ключ сеансу (SK) від імені вузла датчика, оскільки вузол датчика має обмежені ресурси. Нарешті, сенсорний вузол може безпечно спілкуватися з віддаленим сервером за допомогою SK, отриманого від PBS.

У цій самій підкатегорії в [96] представлено протокол розподіленого обміну HIP (D-HIP), за мотивами HIP-BEX [97] з використанням тієї самої моделі мережі, що наведена на рис. 2.7. Під час етапу узгодження ключа обмежений вузол встановлює ключ сеансу з сервером за допомогою протоколу DH, делегуючи 2 модульні операції піднесення до степеня проксі-вузлам. Спочатку він розбиває свій секретний експонент a на n частин a_1, a_2,...a_n, де n – кількість менш обмежених вузлів. Потім він надсилає кожну частину a_i до сусіднього вузла (проксі) PBS_i. Вузол PBS_i обчислює свою частину остаточного ключа сеансу DH: $SK_i = (g^b \ mod \ p)^{ai}$ де значення $(g^b \ mod \ p)$ отримано з

віддаленого сервера (або Інтернет-хосту). PBS_i надсилає SK_i назад до обмеженого вузла. З цих значень обмежений вузол отримує той самий остаточний ключ сеансу DH, що й сервер (шляхом множення n отриманих значень). Цей підхід має велику перевагу в тому, що всі дорогі обчислювальні завдання виконуються вузлами PBS. Однак кількість обмінів повідомленнями може бути великою залежно від кількості вузлів PBS. Як відомо, вартість передачі є незначною, і втрата пакета під час зв'язку може статися будь-коли.

Табл. 2.4 ілюструє приклади рішень протоколів безпеки, реалізованих у WSN та IoT. В ніній порівнюються ці рішення за допомогою визначених вище критеріїв.

Таблиця 2.4 – Резюме пропонованих рішень безпеки для IoT. Рішення згруповані на основі зазначеної на рис. 2.1 класифікації.

					К.	Ц.	Аут.	Авт.	Св.	Ст.	ОС	КС	П.	М	ЗК
Ключове заванта ження в IoT	Асиметричні ключові схеми	Передача ключів на основі шифрування з відкритим ключем	RPKE	Protocols based on: NTRU [49]	●	n/a	n/a	n/a	n/a	●	○	○	●	●	n/a
				Rabin's scheme Moustaine and Laurent [98]	n/a	●	●	○	○	●	●	●	●	●	●
				ZKP based on ECDLP [99]	n/a	●	●	○	●	●	○	●	●	●	●
			CBE	DTLS modified [60,62]	●	●	●	○	●	●	○	○	●	●	●
				IKEv2-ECC based [65]	●	●	●	○	●	●	○	○	●	●	●
			IBS	TinyIBE [72]	●	○	●	○	○	●	●	○	●	○	○
				IBAKA [68]	●	●	●	○	●	●	○	○	●	●	●
		Узгодження ключів на основі асиметричних прийомів	ECDH-ECDSA [71]		●	●	●	○	●	●	○	●	●	●	○
			HIP-DEX [74]		●	●	●	○	●	●	●	●	●	●	○
	Симетричні схеми попереднього розподілу ключів	Імовірнісний розподіл ключів	E-G [78]		●	○	○	○	○	○	●	○	○	○	○
			Du et al. [79]		●	○	○	○	●	○	●	○	○	○	○
			Chan et al. [80]		●	○	○	○	○	○	●	○	○	○	○
			Ito et al. [81]		●	○	○	○	○	○	●	○	○	○	○
		Детермінований розподіл ключів	OKD	Blom's scheme based [79,85]	●	○	○	○	○	○	●	●	●	○	○
				SNAKE [86]	●	●	●	○	●	○	●	●	●	○	○
				BROSK [88]	●	●	●	○	●	○	●	●	●	○	○
				Lightweight IPsec [50]	●	●	●	○	●	○	○	○	●	●	●
				DTLS-PSK [63]	●	●	●	○	●	●	○	○	●	●	●
				Diet-ESP [100]	●	●	●	○	●	●	●	○	n/a	○	●
			EAS	Mikey-ticket [89]	●	●	●	●	●	○	●	●	●	○	○
				SAKE [91]	●	●	●	○	●	●	●	●	●	○	○
				PANA/EAP-PSK [92,94]	●	●	●	○	●	●	●	●	●	○	○
			PBAS	SAKES [95]	●	●	●	○	●	●	○	●	○	○	○
				D-HIP [96]	●	●	●	○	●	○	○	○	○	○	○

Скорочення: К – конфіденційність, Ц – цілісність, Аут – аутентифікація, Авт – авторизація, Св – свіжість, Ст – стійкість, ОС – обчислювальна складність, КС – комунікаційна складність, П – пам’ять, М – масштабованість, ЗК – захист конфіденційності.

На перший погляд можна легко визначити, що більшість загальних послуг безпеки добре забезпечуються запропонованими протоколами. Проте, кілька протоколів підтримують властивості контролю доступу (АС) і захисту конфіденційності (РР). Сервіс АС є дуже важливим і необхідним у такій перспективі, коли Інтернет-хост може отримати доступ до сенсорного вузла лише для виконання дій або отримання даних відповідно до своїх привілеїв доступу. Серверні протоколи зазвичай пропонують цю вимогу, наприклад, за допомогою сервера авторизації. З іншого боку, ПП посилює анонімність комунікацій. Ця властивість стає дуже важливою в сучасній перспективі, оскільки особисті дані на сенсорних вузлах повинні залишатися недоступними для будь-яких зловмисників.

У синтетичній картині високого рівня таблиця показує, що асиметричні рішення зазвичай вимагають високої складності обчислень на вузлах датчиків. Однак ці підходи мають високу стійкість проти атак захоплення вузлів, низькі вимоги до пам’яті для матеріалів ключів, невелику кількість обмінів повідомленнями та високу масштабованість для великих мереж. З іншого боку, схеми попереднього розподілу ключів пропонують низьку складність обчислень, що дійсно вигідно для обмежених вузлів, але вони мають свої власні незручності, такі як висока складність зв’язку, великий простір пам’яті для ключів, низький рівень масштабованості для великих мереж і вразливість проти атак захоплення вузлів.

Використовуються деякі абревіатури: (RPKE) – шифрування з необробленим відкритим ключем, (CBE) – шифрування на основі сертифіката, (IBS) – схеми на основі ідентифікації, (OKD) – розповсюдження ключів в автономному режимі, (EAS) за допомогою зовнішнього сервера, (PBAS) –на

основі допоміжного проксі серверу. Для оцінки рішень передбачено одинадцять показників: конфіденційність, цілісність, автентифікація, авторизація, свіжість, відмовостійкість, складність обчислень, складність зв’язку, пам’ять або простір для зберігання, необхідний для ключів, масштабованість і захист конфіденційності. Стовпці «Стійкість», «Складність обчислень», «Складність зв’язку» та «Пам’ять» можуть приймати два різні значення: ● (хороший або середній рівень продуктивності) і ○ (низький рівень продуктивності), які вказують на рівень певного протоколу для підтримки властивості. Складність зв’язку стосується загальної кількості обмінів повідомленнями до моменту узгодження секретного ключа. Помітка (n/a) означає "не застосовується". Можна визначити прості позначення для оцінки служб безпеки: ● - підтримується, ○ - не підтримується. Оцінка RPKE передбачає протоколи, які використовували згадані примітиви (без справжнього посилання на протокол).

2.5 Огляд останніх тенденцій щодо протоколів безпеки IoT

Дослідники висувають також і нові підходи. Вони завжди цікавляться як асиметричними, так і симетричними підходами; навіть якщо симетрична парадигма вважається більш енергоефективною. Асиметричні рішення все ще є кращими через їхню можливість розгортання, гнучкість і масштабованість з точки зору управління ключами. Крім того, парадигма відкритого ключа дозволяє двом об’єктам без будь-яких попередніх довірчих відносин один з одним встановлювати безпечний канал, що, як правило, є важливою функцією в сценаріях реального часу.

Перш ніж розробляти будь-які ефективні протоколи безпеки для обмежених пристроїв в IoT, необхідно виділити наступні моменти:

Оптимізація асиметричних рішень: асиметричні підходи, як правило, споживають енергію. Перше завдання полягає в тому, щоб скоротити

необхідний час обчислень, щоб заощадити енергію для сенсорних вузлів. Можна подумати про адаптацію безпосередньо NTRU до стандартних протоколів, оскільки на даний момент це найбільш енергоефективний примітив. Однак цей примітив потребує більше пам'яті для введення матеріалів, ніж інші асиметричні примітиви. Деякі дослідники працюють над оптимізацією математичних механізмів, що використовуються в криптографічних алгоритмах, наприклад в [101] обговорюється рішення для оптимізації примітивів ECC. Пропонується оптимізація для операції модульного множення. Рішення оцінено в широко поширеному мікропроцесорі MSP430. Стверджується, що оптимізація представляє найменший час і кількість необхідних операцій для множення ECC. Інший метод зменшення споживання енергії на сенсорних вузлах покладається на методи попереднього обчислення. Це допомагає зменшити вартість модульного піднесення до степеня в кількох схемах керування підписами та ключами, наприклад ECDSA або обмін ключами Діффі-Хеллмана. Ідея полягає в тому, щоб зберегти набір із n пар дискретних журналів у формі (a_i, $g^a \ mod \ q$). Потім «випадкова» пара (r, $g^r \ mod \ q$) генерується з підмножини k пар, вибраних випадковим чином у пам'яті. Техніка виглядає простою, але вона вимагає, щоб значення n було достатньо великим, щоб забезпечити випадковість згенерованих пар $(r, g^r \ mod \ q)$. В [102] вдосконалено наведені вище методи попереднього обчислення та застосовано їх до ECDSA. Показано, що майже 50% енергії зберігається за допомогою ECDSA з попереднім обчисленням порівняно з оригінальною схемою підпису, а також зі схемою підпису *NTRUsign* (яка вважається природним кандидатом у пристроях з низьким споживанням енергії).

З іншого боку, кілька досліджень адаптують властивості асиметричних примітивів оптимізованим чином, щоб відповідати найбільш обмеженому середовищу IoT. Автори [98] пропонують ефективний протокол автентифікації для недорогих систем RFID на основі адаптації NTRU. Ця адаптація спочатку делегує складні операції NTRU (тобто модульну арифметику, поліноміальне множення) серверу. По-друге, теги вимагають лише додавання та циклічних

зсувів для шифрування викликів на етапі автентифікації. Крім того, протокол стійкий до класичних атак, включаючи повтори, відстеження та атаки людини посередині з дуже низькими вимогами до обчислень.

Як ще одна асиметрична техніка, докази з нульовим знанням (ZKP) [99,103] також є кандидатом на майбутні пропозиції в IoT. ZKP – це інтерактивні системи перевірки, що включають дві сутності: перевірку та верифікатор. Пристрій, що перевіряє, демонструє знання секрету верифікатору, не розкриваючи жодної деталі секрету. ZKP спирається на деякі важкі математичні проблеми, такі як розкладання цілих чисел на множники [103] або проблема дискретного логарифмування (DLP) [99]. Цей механізм зазвичай використовується в WSN для автентифікації вузла. Наприклад, автори в [99] забезпечують ефективну схему автентифікації на основі DLP над групами еліптичних кривих. Схема вимагає лише трьох повідомлень між сервером і верифікатором. ZKP має переваги з точки зору кількості повідомлень, що надсилаються, і використання пам’яті на вузлах, як також зазначено в [99,103]. ZKP може принести користь, запропонувавши ефективний протокол початкового завантаження ключа в IoT з автентифікацією вузла, наданою ZKP.

Адаптація існуючих стандартних протоколів до IoT: стандартні протоколи безпеки можна адаптувати для роботи в обмежених і неоднорідних середовищах IoT. Було зроблено багато спроб адаптувати та застосувати стандартні протоколи в контексті IoT, наприклад, DTLS [60, 62], IPsec [50], IKEv2 [65], HIP-DEX [74-76]. Як інший приклад, в [104] пропонують мінімальну реалізацію стандарту IKE [105] шляхом скасування вимоги щодо сертифікатів. Цей мінімальний варіант визначає лише два обміни повідомленнями для узгодження ключів і забезпечує автентифікацію об’єктів за допомогою підходу попереднього спільного ключа. З іншого боку, в [100] припущено, що асоціації безпеки між об’єктами встановлюються за допомогою існуючого механізму, такого як IKEv2. Вони зацікавлені в безпеці передачі пакетів, пропонуючи Diet-ESP — адаптацію ESP (Encapsulation Security Protocol) до IoT з метою стиснення та зменшення накладних витрат ESP.

Автори визначають механізми видалення або зменшення деяких «непотрібних» або «більших, ніж потрібно» полів ESP для конкретних потреб або застосувань пристроїв IoT. Однак розгортання Diet-ESP має підтримувати компроміс між вимогами безпеки та часом автономної роботи обмежених пристроїв. Дійсно, як показано, малий розмір SPI (індекс параметрів безпеки), малий розмір ICV (значення перевірки цілісності) і видалення SN (номер послідовності) наражають пристрої на атаки типу «відмова в обслуговуванні», спуфінгу та повторного відтворення.

Використання гібридних підходів: Інша тенденція полягає в поєднанні переваг як симетричних, так і асиметричних рішень. В [76] вибрано HIP-DEX (асиметрична техніка) [74] для забезпечення доступу до локальної сенсорної мережі. Аутентифікація мобільного вузла здійснюється за допомогою центрального сервера. Якщо автентифікація пройшла успішно, сервер безпечно надсилає необхідні параметри для мобільного вузла, шифруючи дані за допомогою ключа сеансу, згенерованого після обміну DH. Ці параметри фактично є двовимірним поліномом, який використовується для завантаження безпечного зв'язку з локальним вузлом (симетричний метод). Попарний ключ, згенерований спільним поліномом, використовується як головний ключ для створення кількох сеансових ключів для певних цілей.

Присутність третьої сторони в такому гібридному підході стає важливою в IoT. По-перше, очікується, що сервер із багатими ресурсами підтримуватиме майже всі важкі обчислення. Таким чином, сенсорні вузли з обмеженою енергією та можливостями більше не беруть участь у цьому дорогому процесі, як описано в [95,96]. Обмежений вузол може встановити зв'язок із зовнішніми хостами без впровадження повного асиметричного процесу. Крім того, допоміжні сервери здатні забезпечувати точне керування доступом, щоб на вузлах датчиків виконувались лише авторизовані дії.

3 АНАЛІЗ ПРОДУКТИВНОСТІ ПРОТОКОЛІВ БЕЗПЕКИ ДЛЯ РОЗПОДІЛЕНИХ СИСТЕМ ВИМІРЮВАННЯ НА ОСНОВІ ІНТЕРНЕТУ РЕЧЕЙ З ОБМЕЖЕНИМ АПАРАТНИМ ЗАБЕЗПЕЧЕННЯМ ТА ІНФРАСТРУКТУРОЮ З ВІДКРИТИМ КОДОМ

3.1 Архітектура та її розгортання

Розглянемо комплексну пропозицію, яка розглядає орієнтовану на безпеку стратегію віртуальної локальної мережі (VLAN) і рішення для апаратних обмежень. Звертаючись до найгіршого сценарію, будемо намагатися узагальнити отримані результати, забезпечуючи застосовність у всьому спектрі практичних сценаріїв. Основною метою дослідження є ретельна перевірка всіх наборів шифрів OpenSSL на предмет сумісності з сервером Mosquitto .

Проведемо аналіз ключових показників продуктивності, включаючи (але не обмежуючись) загальний коефіцієнт, загальний час виконання, середній час виконання, час повідомлення, середню пропускну здатність і загальну пропускну здатність. Ця оцінка проводиться для кожного набору шифрів і рівня якості обслуговування (QoS), забезпечуючи детальне розуміння складної взаємодії між заходами безпеки та загальною ефективністю системи.

Для досягнення першої мети дослідження розглянемо визначення алгоритмів, які можуть гарантувати оптимальне співвідношення передачі даних/шифрування. Це передбачає ретельне вивчення криптографічних методів і протоколів передачі, щоб знайти детальний баланс між безпекою даних і ефективним зв’язком у середовищах з обмеженими ресурсами.

Друга мета передбачає комплексне дослідження алгоритмів, що забезпечують сумісність з різноманітними інфраструктурами MQTT. Дослідження визнає проблеми, які створюють географічно розкидані мережі Інтернету речей, особливо у важкокерованих приміських або сільських середовищах. Наголос робиться на створенні безпечної системи з’єднання, яка враховує різноманітні інфраструктурні нюанси, притаманні таким ландшафтам.

Одночасно третя мета зосереджена на реалізації відкритого мікропрограмного забезпечення на обмежених пристроях, сприяючи сумісності з різними протоколами MQTT. Ця ініціатива розроблена для підвищення адаптивності та сумісності пристроїв Інтернету речей, таким чином сприяючи створенню безпечної та стандартизованої комунікаційної структури. Вивчення відкритого мікропрограмного забезпечення є невід'ємною частиною вирішення динамічної природи протоколів MQTT і забезпечення бездоганної інтеграції з різними пристроями в екосистемі IoT.

Ці три взаємопов'язані цілі разом складають основу дослідження, спрямованого на покращення розуміння безпечних і ефективних комунікаційних протоколів Інтернету речей, одночасно вирішуючи практичні проблеми, пов'язані з обмеженнями обладнання та різноманітними мережевими середовищами.

Представимо архітектуру та її розгортання.

Пропозиція базується на інтеграції належних компонентів і протоколів, що працюють на наступній архітектурі з відкритим кодом.

MQTT (Mosquitto) відноситься до протоколу керування передачею/Інтернет-протоколу (TCP/IP), який базується на моделі публікації-підписки, що працює через спеціалізований брокер повідомлень. Це один із найпоширеніших протоколів у сфері IoT. Комплекти шифрів – це комбінації криптографічних алгоритмів, протоколів і параметрів безпеки, які визначають спосіб шифрування та дешифрування даних під час безпечного зв'язку через Інтернет. Ці пакети визначають методи шифрування та автентифікації, які використовуються для встановлення безпечного з'єднання між клієнтом і сервером, а саме у безпечному протоколі передачі гіпертексту (HTTPS) для безпечних мережевих протоколів. Набір шифрів зазвичай включає наступні компоненти.

– Алгоритм обміну ключами: це безпечний обмін ключами шифрування між клієнтом і сервером.

– Алгоритм шифрування: цей алгоритм шифрує дані, щоб неавторизовані сторони не могли їх прочитати.

– Хеш-функція: криптографічна хеш-функція забезпечує цілісність і автентичність даних.

– Алгоритм коду автентифікації повідомлення (MAC): MAC забезпечує цілісність даних, дозволяючи обом сторонам виявити, чи були дані підроблені під час передачі.

VLAN є важливим інструментом кібербезпеки в IoT та DMS. Пропонуючи сегментацію мережі, детальний контроль доступу та ізоляцію критично важливих систем, VLAN ефективно зменшують поверхню атаки та обмежують шляхи для потенційних зловмисників. Вони сприяють ефективному моніторингу трафіку, полегшують стримування вразливостей і оптимізують розподіл ресурсів, що особливо важливо в управлінні значними обсягами даних пристроїв IoT. Крім того, VLAN спрощують керування мережею, покращують контроль відповідності та забезпечують масштабоване рішення безпеки, здатне адаптуватися до мінливих потреб розширеного середовища IoT. Загалом VLAN відіграють важливу роль у зниженні кіберризиків, підвищенні стійкості мережі та забезпеченні безпечної роботи DMS на основі IoT.

Комплекти шифрів доступні в різних комбінаціях, і їх міцність і рівень безпеки можуть відрізнятися. Вибір набору шифрів залежить від конкретних вимог безпеки та можливостей сторін, що спілкуються. Більш безпечні набори шифрів використовують сильніші алгоритми шифрування та методи обміну ключами, тоді як менш безпечні можуть використовувати слабше шифрування, яке може бути чутливим до атак [106].

Представляючи набори шифрів, важливо підкреслити їхню стійкість до атак із бічних каналів (SCA) разом із потужністю алгоритмів шифрування. Комплекти шифрів різняться за сприйнятливістю до SCA, які використовують витік інформації з реалізацій криптографічного алгоритму. Оцінки наборів шифрів повинні враховувати їхню стійкість проти різних SCA, причому деякі реалізації включають контрзаходи, такі як алгоритми постійного часу або

рандомізація. Цей комплексний підхід забезпечує надійний захист від теоретичних криптографічних уразливостей і недоліків практичного рівня реалізації [107,108].

Щоб оцінити досяжну пропускну здатність мережі DMS щодо частково вибраних протоколів QoS і безпеки, запропонований метод вимірювання розглядає загальну архітектуру IoT, що складається з наступних компонентів.

– Місцеві брокери MQTT: кожен брокер, будь то житловий чи діловий, має власний спеціальний сайт

Брокер MQTT, відповідальний за збір і керування даними з локальних пристроїв IoT.

– Локальні тунелі TLS: дані з датчиків на кожному сайті шифруються за допомогою локальних тунелів TLS перед надсиланням до локального брокера MQTT.

– Головний брокер із тунелю до хмари: тунель SSL використовується для безпечної передачі даних від локальних брокерів до центрального хмарного брокера, відповідального за агрегацію та аналітику даних.

На рис. 3.1 показано розгортання запропонованої архітектури порівняльного аналізу MQTT. Реалізація включає наступне:

– інтеграція пакетів шифрування MQTT і OpenSSL в пристрої Raspberry Pi;

– синхронізована настройка з використанням мережевого протоколу часу (NTP) для надійних вимірювань;

– автоматизація процедур (відправка, запис тощо) для генерації слідів і збору результатів;

– виконання тестів на різних канальних технологіях, з використанням пакету mosquitto -client, доступного для утиліти Linux, для вимірювання пропускної здатності та різних ймовірностей втрат.

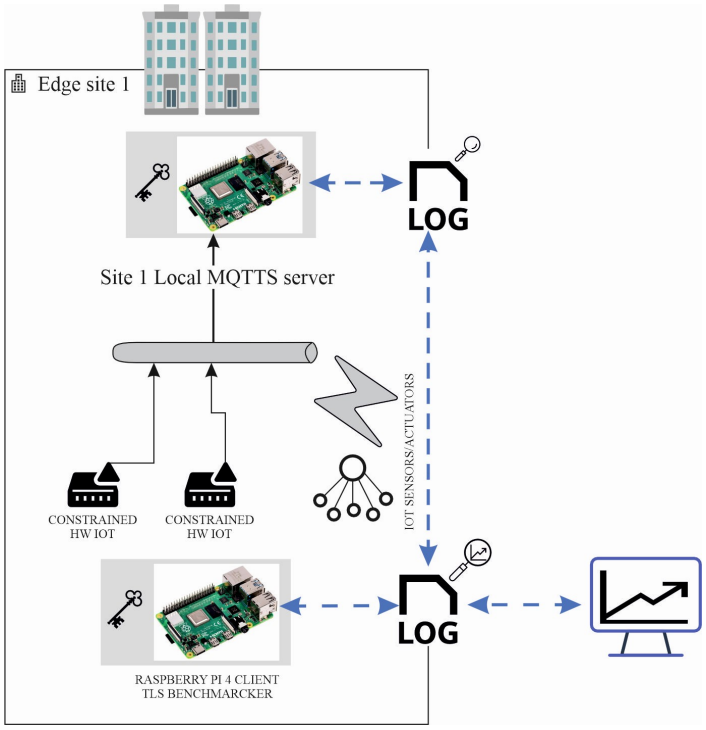


Рисунок 3.1 – Архітектура порівняльного аналізу MQTT

3.2 Метод вимірювання для оцінки пропускної здатності

Для комплексної оцінки пропускної здатності мережі необхідно враховувати різні фактори. Необхідно оцінити сумісність усіх доступних наборів шифрів OpenSSL із сервером MQTT, переконавшись, що шифрування даних не перешкоджає зв'язку між пристроями IoT і брокерами. Наступний псевдокод (у сценаріях bash) представляє кроки, за якими слідує запропонований тут метод вимірювання. Зокрема, посилаючись на рис. 3.1, оцінюється пропускна здатність між будь-якими двома вузлами мережі. Потрібне лише посилання типу MQTT-BRIDGE із відповідними сертифікатами та обліковими даними автентифікації для клієнтів/серверів у різних мережах. Маршрутизатор OpenWrt діє як NTP-клієнт і як локальний NTP-сервер і синхронізується з серверами часу. Це гарантує, що пристрої в мережі мають одне джерело часу. Щоб проаналізувати продуктивність кожного алгоритму

шифрування, використовуємо програмний пакет mosquitto-clients, оскільки він дозволяє більш вичерпний аналіз щодо своїх конкурентів (mqtt-бенчмаркер [109]; mqttx [110]; mqtt -cli [111]). Потім порівнюємо результати 1000 повідомлень, що відрізняються за типом QoS, алгоритмом шифрування та версією TLS. На рис. 3.2 показано фізичне розгортання тестових стендів 4G/LTE, Ethernet і WiFi.

Експерименти проводилися наступним чином. Для мережі IoT використовувалася виділена VLAN. Годинником мережі керує маршрутизатор. Проте, головною дійовою особою є клієнт. Брокер і клієнт взаємодіють через обмін ключами SSH, і обидва вони мають привілеї root. Клієнт ініціює основну програму, яка запускає 30 послідовних прогонів порівняльних тестів.

Під час кожного запуску інструмент тестування

– зупиняє екземпляр Mosquitto брокера;

– запитує версію OpenSSL сервера та запитує список наборів шифрів перед TLSv1.2, а потім TLSv1.3 через SSH;

– створює кілька екземплярів файлу конфігурації брокера з версією TLS і набором шифрів і виконує їх один за одним.

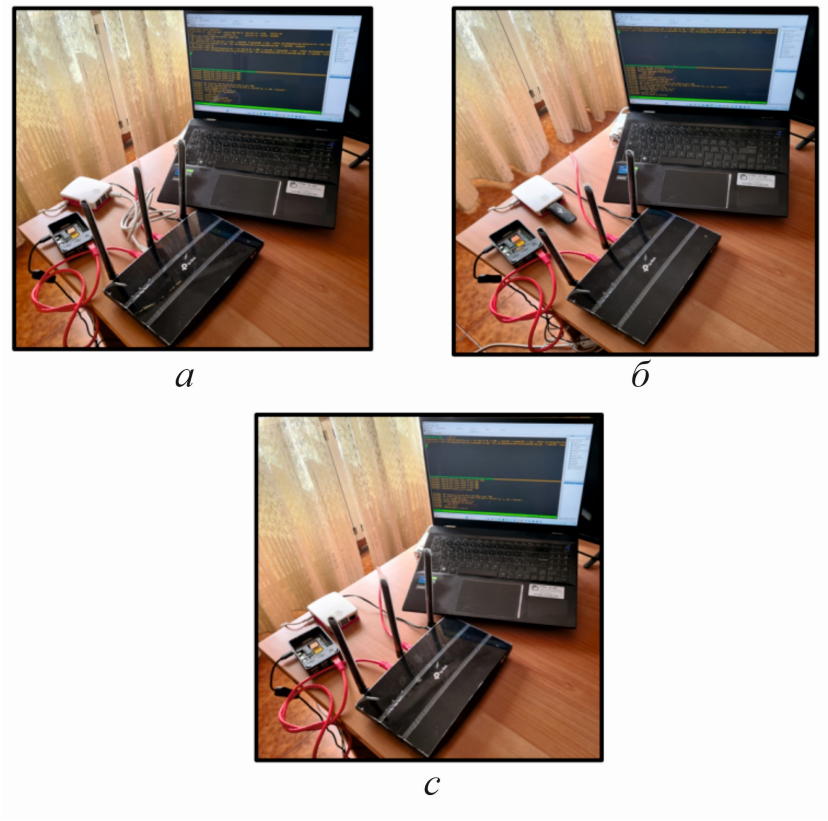
Для кожного екземпляра інструмент порівняння

– використовує mosquitto_pub / mosquitto_sub для публікації на певну тему;

– аналізує логи сервера та клієнта;

– витягує значення, пов’язані з показниками, описане нижче (бенчмаркінг);

– генерує файли CSV.



а *б*

с

а – розгортання Ethernet, *б* – розгортання 4G/LTE, *в* – розгортання WiFi

Рисунок 3.2 – Фізичне розгортання тестових стендів

Бенчмаркінг за допомогою клієнтів і інструментів MQTT. Програмне забезпечення з попереднього підрозділу використовується для оцінки продуктивності мережі IoT. Оцінимо наступні ключові показники під час надсилання корисного навантаження розміром 1 Мбайт для кожного повідомлення:

– загальне співвідношення: загальна кількість надісланих повідомлень відносно отриманих повідомлень, що вказує на загальну ефективність системи;

– загальний час виконання (s): тривалість процесу порівняльного аналізу;

– середній час виконання (s): середня тривалість кожного тестування;

– показники часу повідомлення (min, max, mean та std);

– Msg time min (ms): найкоротший час проходження повідомлення;

– максимальний час повідомлення (ms): найдовший час проходження повідомлення;

– середній час повідомлення (ms): середнє значення часу проходження повідомлення;

– стандартне значення середнього часу повідомлення (ms): стандартне відхилення середнього часу проходження повідомлення;

– середня пропускна здатність (msg/s): середня кількість повідомлень, що передаються за секунду;

– загальна пропускна здатність (мсг/с): загальна кількість повідомлень, переданих за секунду під час процесу порівняльного аналізу.

Одна частина показників аналізує затримку мережі, втрату пакетів і цілісність даних, щоб забезпечити точність і надійність даних датчика. Друга частина аналізує конфіденційність даних: наголошення на конфіденційності отриманих вимірювань має вирішальне значення, особливо в чутливих сферах, таких як охорона здоров'я, фінансові установи та моніторинг критичних структур. Третя частина обговорює методи шифрування, засоби контролю доступу та дотримання правил захисту даних, щоб забезпечити безпеку даних.

3.3 Обговорення результатів

Щоб оцінити продуктивність запропонованого методу вимірювання в реальному сценарії, розроблено спеціальний тестовий стенд, показаний на рис. 3.2.

Зокрема, два пристрої Raspberry Pi 4 (8 ГБ оперативної пам'яті) використовуються для впровадження, відповідно, клієнта MQTT для порівняльного аналізу (білий Raspberry) і сервера MQTT (чорний Raspberry) з порівняльної логічної архітектури MQTT на рис. 3.1.

Рис. 3.3 ілюструє етапи алгоритму, реалізованого двома Raspberries. Окремі функції «mqtt_pub_base-qosX-vY-tlsZ», що використовуються для виконання тестів, викликають програмне забезпечення « mosquitto_pub /

mosquitto_sub », реалізоване у використовуваній операційній системі Debian 12 з Mosquitto версії 2.0.11-1.

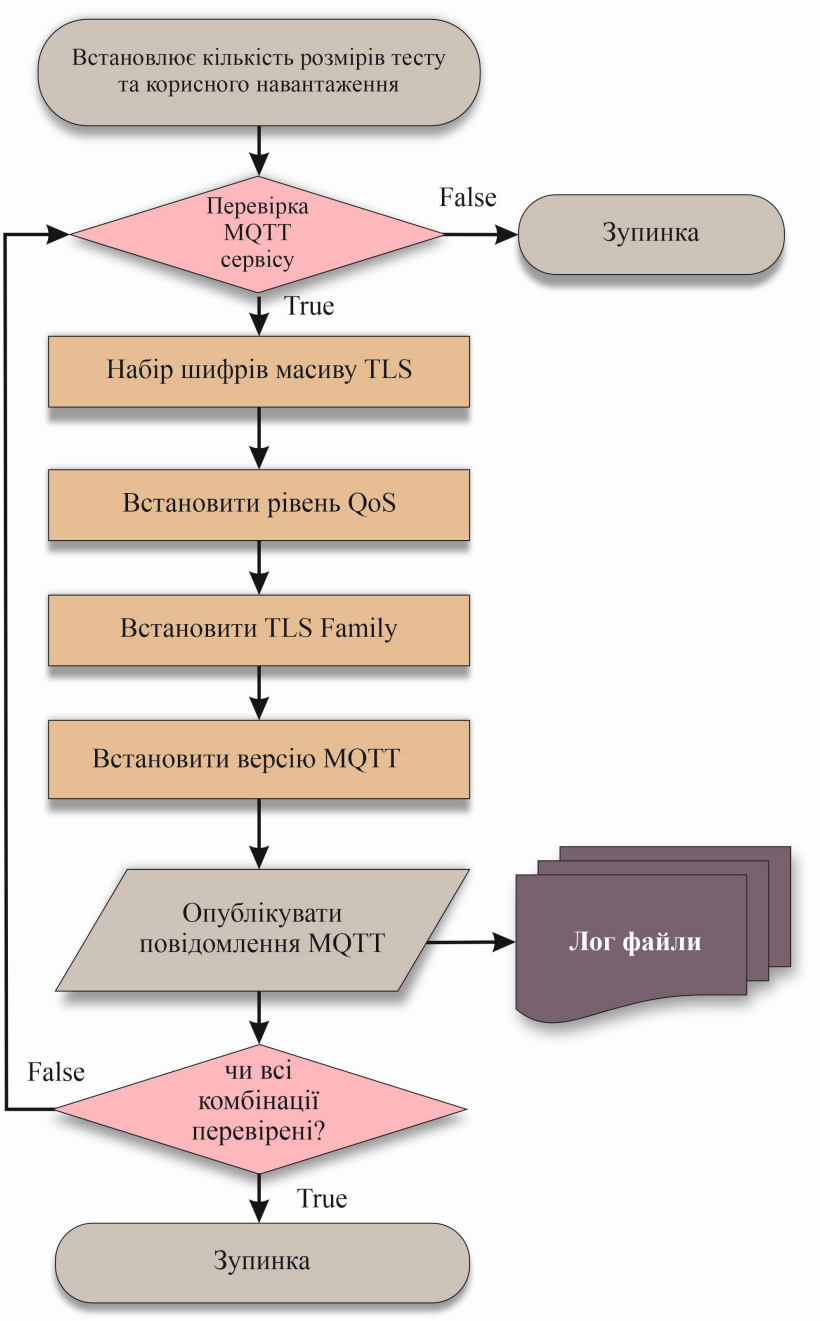


Рисунок 3.3 – Блок-схема алгоритму

Алгоритм складається з чотирьох частин. Перша частина готує середовище порівняльного аналізу. Другий запускає віддалений MQTT через екземпляри TLS на брокері та скидає TLS. Третій запускає виконання клієнтом для надсилання даних. Четвертий підсумовує збір даних наприкінці сеансу. Під час пересилання кожного повідомлення функція mqtt_pub_base-qos збирає дані

про пересилання та отримання у файлах журналу за допомогою інструменту вимірювання, який використовується, і підключається як віддалений системний журнал до файлу журналу mosquitto , який присутній у посереднику.

Еталонним маршрутизатором є TP-Link Archer C7 v5 з архітектурою Qualcomm Atheros QCA956X ver 1 rev 0 і OpenWrt версії 22.03.5.

Випробування проводилися на трьох типах з'єднань для передачі даних:

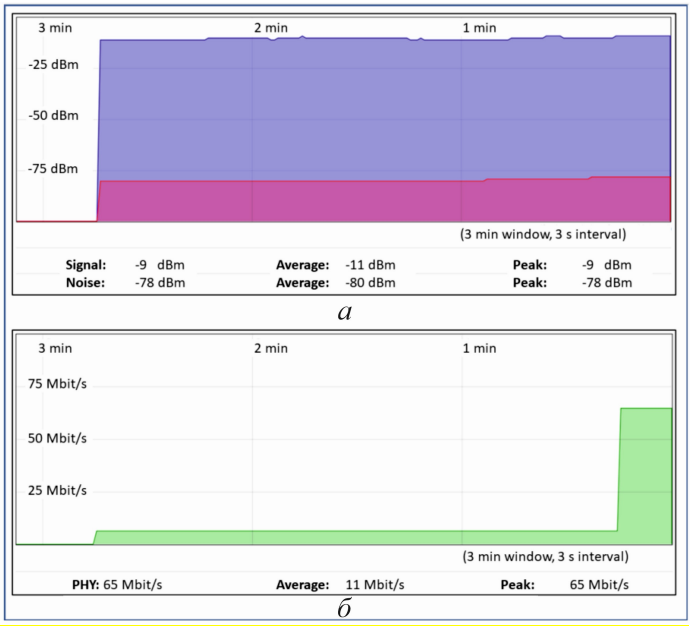
– дротовий кабель (1 Гбіт/с);

– 4G/LTE (для підключень із сільської місцевості);

– бездротовий зв'язок 2,4 ГГц (для максимальної зворотної сумісності сенсора).

Для кожного доступного набору шифрів (стосовно бібліотек OpenSSL 1.x) через SSH (Secure Shell) надсилається команда для запуску віддаленого брокера від клієнта до сервера. На цьому етапі сервер буде слухати призначений порт (у розглянутому випадку 1866/TCP) з вибраним набором шифрів по одному. Клієнт перевірить, чи відповідає віддалений порт, і встановить з'єднання. Програмне забезпечення « mqtt -benchmark», написане мовою Golang, запускається для виконання 10 передач із 10 наборів даних для 10 фіктивних клієнтів, що становить 1000 передач. Через обмеження безпеки у версії TLS розглядаються набори шифрів 1.2 і 1.3.

Роутер з прошивкою OpenWrt дозволяє легко налаштовувати складні мережі з підтримкою VLAN, TRUNK і моніторингу мережевих ресурсів. Щоб оцінити обмеження фізичної продуктивності тестового стенда, рис. 3.4 відображає PhyRate мережі WiFi, пов'язаної з IoT VLAN. Відображене пікове значення дорівнює 65 Мбіт/с, що є обмеженням фізичної продуктивності, досягнутим за допомогою тестів завдяки спеціальному апаратному забезпеченню. Крім того, на цьому ж рисунку показано відсутність перешкод на каналі.



а – сигнал Wi-Fi і потужність шуму, *б* – Wi-Fi PHY Rate

Рисунок 3.4 – Інформація WiFi для виділеного VLAN BRIDGE IoT

Відповідно до процедури, описаної вище, на рис. 3.5 представлено статистичний розподіл затримок для наборів шифрів TLSv1.2 і рівня QoS0 на каналі Ethernet і MQTT V3.11, порівнюючи Qos0, QoS1 і Qos2 з використанням прямокутного представлення.

Кожна прямокутна графіка відображає нижню та верхню межі прямокутників, що представляють 25-й та 75-й процентилі. Крім того, нижні та верхні вуса відповідають 5-му та 95-му процентилям, тоді як середнє значення, або 50-й процентиль, позначено «X».

Щоб зафіксувати ступінь варіації затримки та надати розуміння попередньо представлених середніх значень, включено викидні значення за межами згаданих процентилів. Важливо відзначити, що вісь ординат розділена, щоб краще проілюструвати діапазон викидних значень під час представлення діапазону прямокутної діаграми.

На рис. 3.5 показано стабільний діапазон, куди потрапляє більшість вибірок затримки. У цьому сенсі викиди пояснюють спостережувану варіацію середньої затримки. Як видно, існує кілька наборів шифрів, затримка яких перевищує 95-й процентиль. Проте, на рис. 3.5 чітко видно, що більші затримки

виникають у випадку наборів шифрів: AES256-SHA, DHE-RSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA. Виявляється, що всі SHA з базою AES128/256 демонструють більшу затримку, ймовірно, через те, що вони старіші та мають довший час узгодження порівняно з іншими. Іншим цікавим випадком є набір шифрів ECDHE-RSA-CHACHA20-POLY1305, який демонструє вищу варіабельність з точки зору інтерквартильного діапазону, хоча це новіший набір шифрів із меншим часом узгодження.

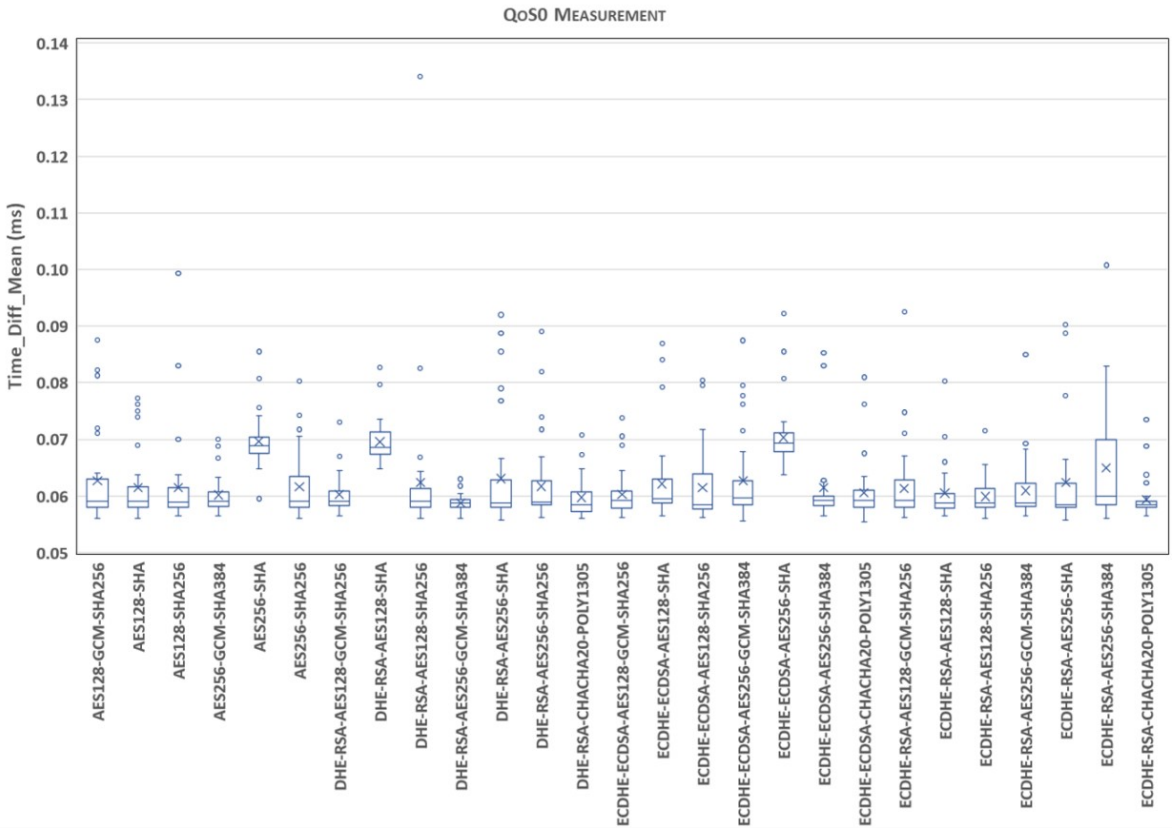


Рисунок 3.5 – Статистичний розподіл затримок для наборів шифрів TLSv1.2 і рівня QoS0 на каналі Ethernet і MQTT V3.11

Рис. 3.6 і 3.7, відповідно, показують порівняння з точки зору пропускної здатності та середнього часу проаналізованих сценаріїв дорогого набору шифрів, розглядаючи випадок обмежень безпеки, пов’язаних із версією TLS, і враховуючи тип з’єднання Ethernet. Як видно з рис. 3.6, ECDHE-RSA-CHACHA20-POLY1305 як набір шифрів видається найкращим, а ECDHE-ECDSA-AES256-SHA – найгіршим.

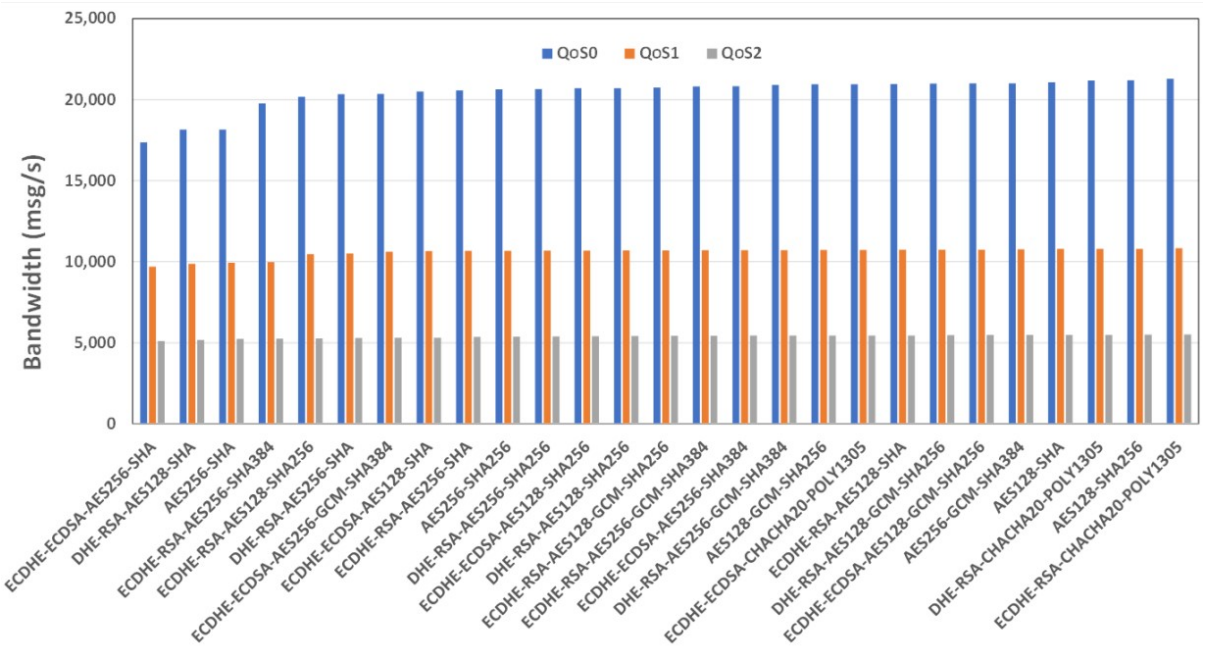


Рисунок 3.6 – Порівняльний показник пропускної здатності для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі Ethernet і MQTT V3.11

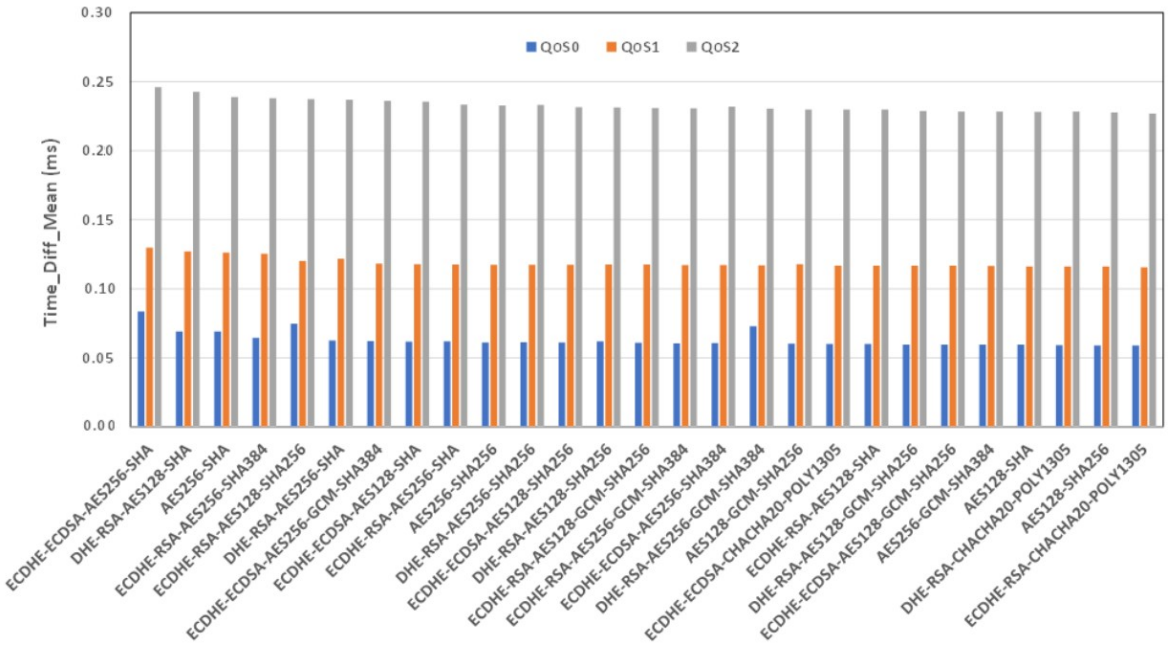


Рисунок 3.7 – Порівняльний показник середнього часу (у мілісекундах) для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі Ethernet і MQTT V3.11

Розглянемо випадок з’єднання Ethernet, яке розгортає з’єднання MQTT 3.x через TLS 1.2. Зосередимося на більш значущих мобільних пристроях IoT.

Результати, отримані з урахуванням різних комбінацій версій протоколу MQTT і TLS, наведені нижче.

Рис. 3.8 і 3.9 показують середню пропускну здатність, використану під час тестування, з наборами шифрів, що належать до сімейства TLS 1.2 і MQTT версій 3.x і 5.0 з посиланнями WiFi , відповідно. Бачимо, що існують відмінності в продуктивності між двома версіями протоколу. У деяких випадках запровадження нових функцій може призвести до збільшення накладних витрат на транспортування повідомлень, що може негативно вплинути на продуктивність пропускної здатності порівняно з попередньою версією протоколу. Фактична продуктивність також залежатиме від конкретної реалізації протоколу MQTT, який використовується клієнтами та брокерами. Деякі реалізації можуть додатково оптимізувати використання пропускної здатності, тоді як інші можуть працювати менш ефективно. Якщо спостерігати, наприклад, за продуктивністю протоколу AE128-SHA у версії MQTT 3.11, виявляється, що він має дуже низьку продуктивність, тоді як версія протоколу MQTT 5.0, здається, має найкращу продуктивність.

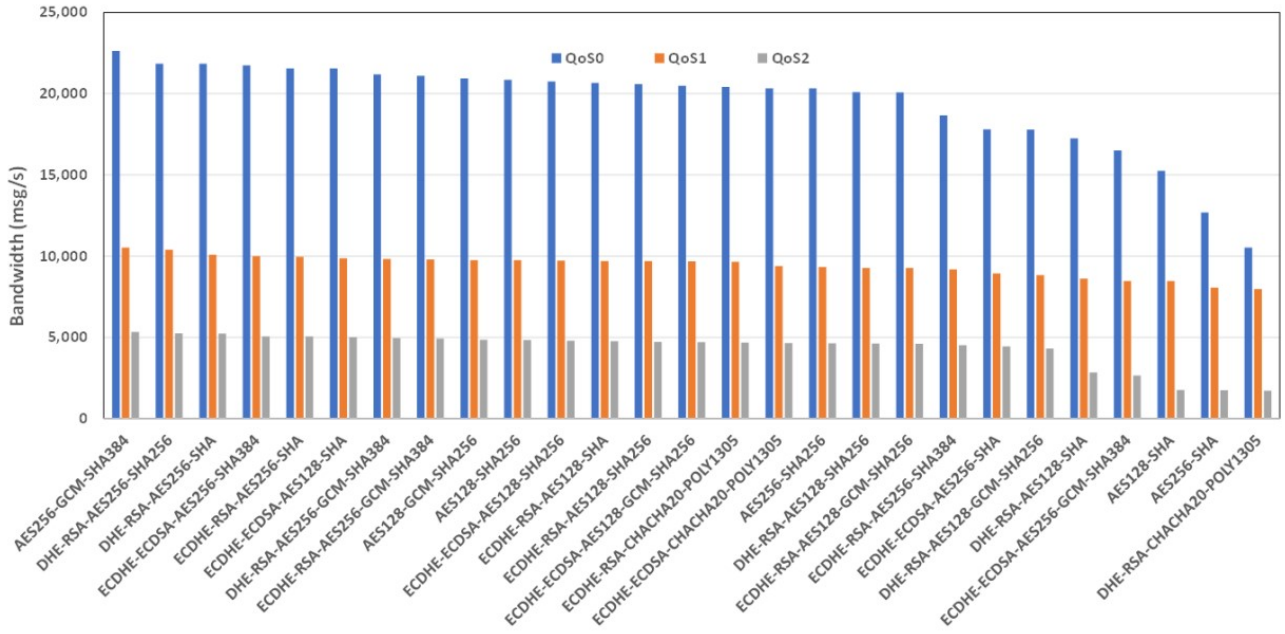


Рисунок 3.8 – Порівняльний тест пропускної здатності для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі WiFi та MQTT V3.11

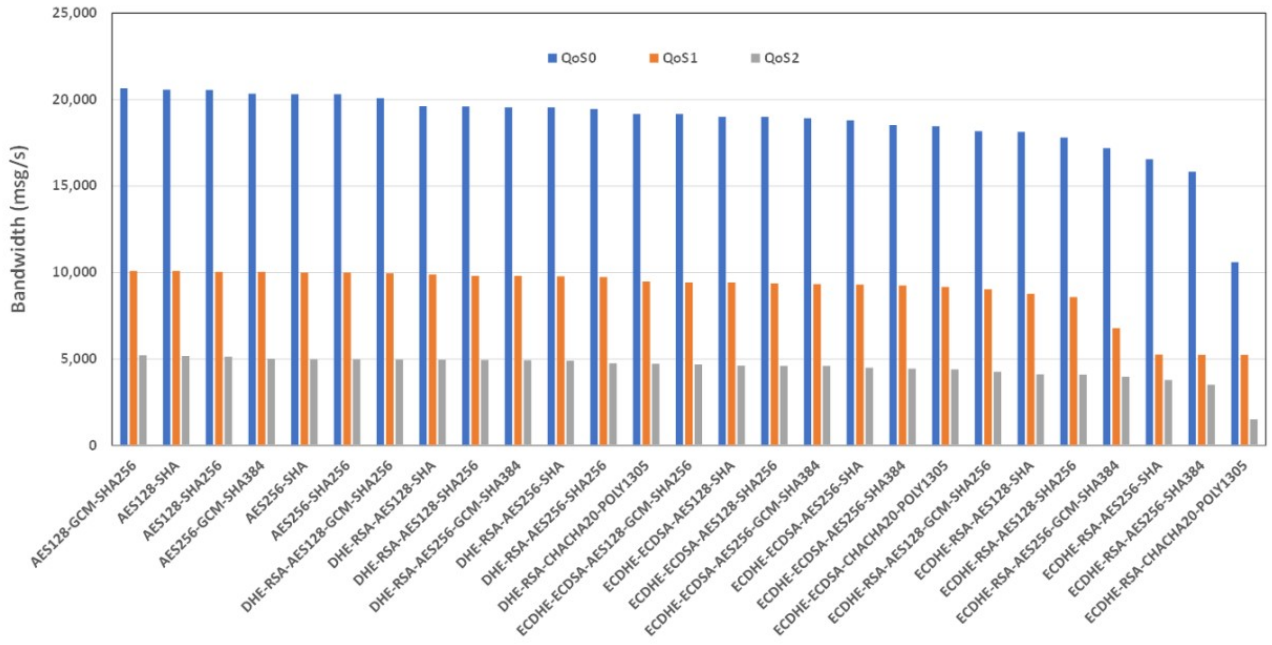


Рисунок 3.9 – Порівняльний тест пропускної здатності для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі WiFi та MQTT V5.0

Рис. 3.10 і 3.11 показують середню пропускну здатність, яка використовується під час тестування, з наборами шифрів, що належать до сімейства TLS 1.2 і MQTT версій 3.x і 5.0 з посиланнями 4G/LTE відповідно. Поведінку, подібну до результатів, отриманих для з'єднання WiFi , можна знайти для з'єднання 4G. Тут також продуктивність AE128-SHA значно відрізняється в залежності від версії протоколу MQTT, який використовується. Продуктивність пропускної здатності також залежатиме від умов мережевого середовища, включаючи затримку, доступну пропускну здатність, перевантаження мережі тощо. Ці фактори можуть суттєво відрізнятися та впливати на продуктивність пропускної здатності MQTT, особливо враховуючи підключення 4G.

Рис. 3.12 показує статистичний розподіл затримок для наборів шифрів TLSv1.2 і рівня QoS0 на каналі 4G і MQTT V3.

Якщо сімейство наборів шифрів змінюється з алгоритмів TLSv1.2 на сімейство TLSv1.3, помічаємо, що коли змінюється версія протоколу MQTT, не

спостерігається значних відмінностей у продуктивності, на відміну від результатів з TLSv1.2 сімейство наборів шифрів.

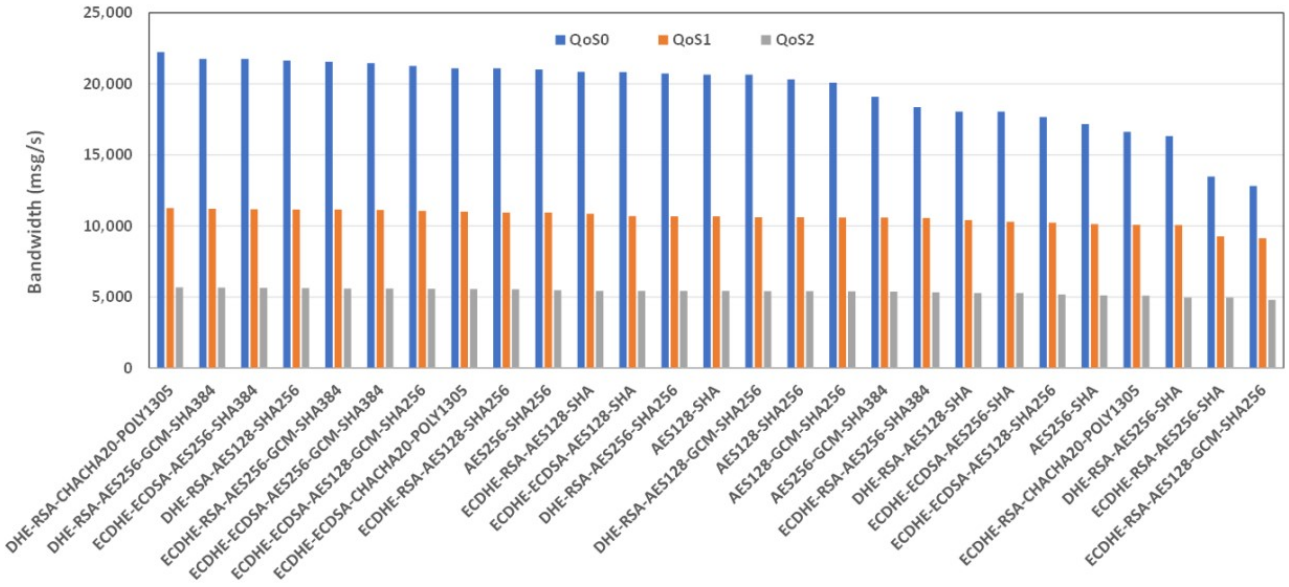


Рисунок 3.10 – Порівняльний показник пропускної здатності для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі 4G і MQTT V3.11

Рис. 3.13 показує статистичний розподіл затримок для наборів шифрів TLSv1.3 і рівня QoS0 на каналі 4G і MQTT V5.

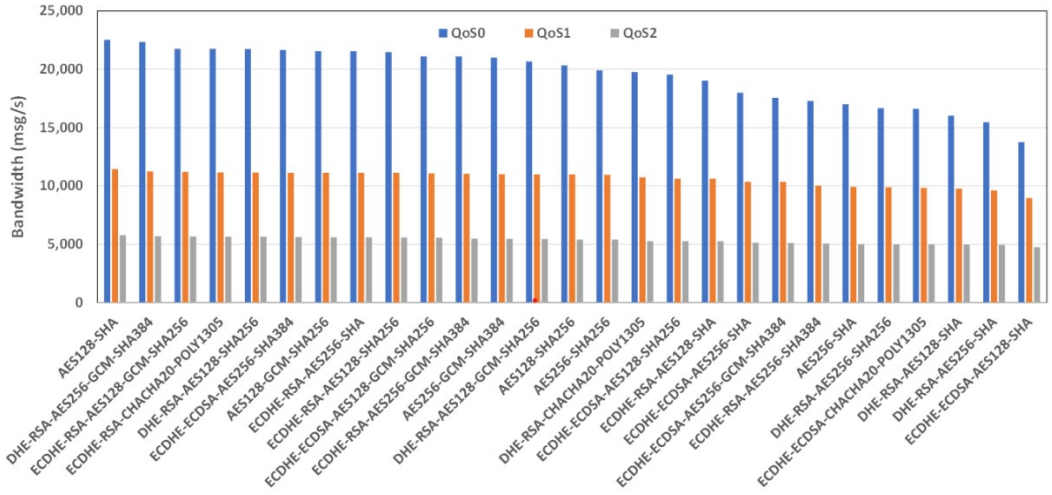


Рисунок 3.11 – Порівняльний тест пропускної здатності для наборів шифрів TLSv1.2 і всіх рівнів QoS на каналі 4G і MQTTV5.0

Сформулюємо коротко одержані результати.

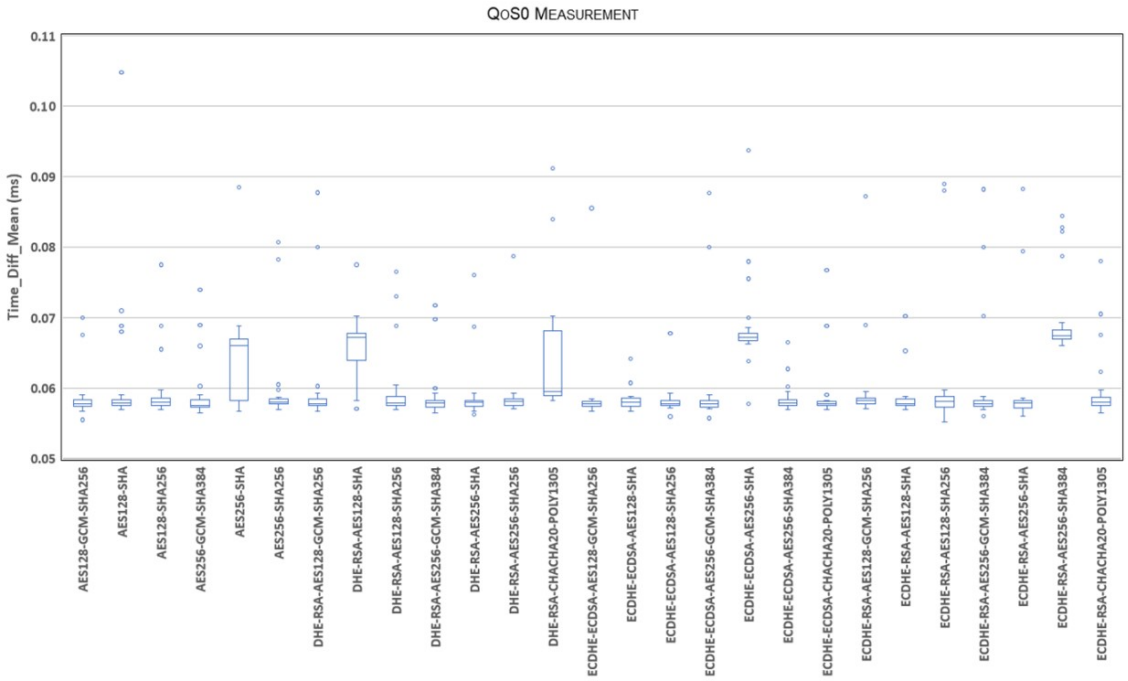


Рисунок 3.12 – Статистичний розподіл затримок для наборів шифрів TLSv1.2 і рівня QoS0 на каналі 4G і MQTTV3

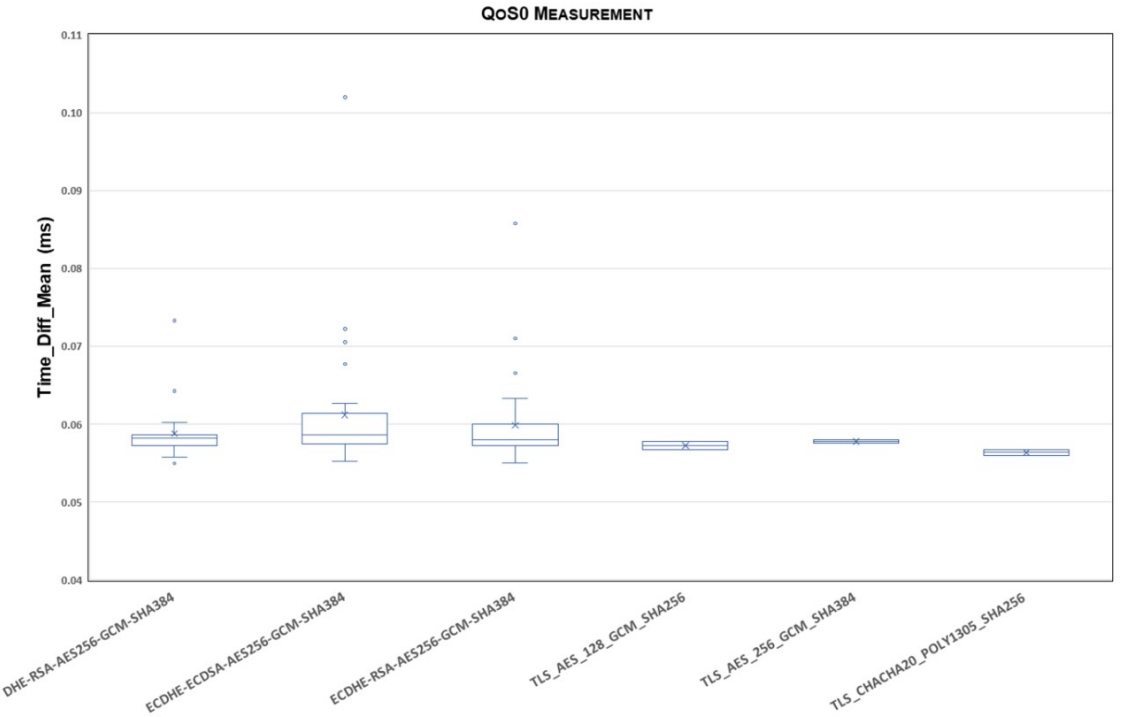


Рисунок 3.13. – Статистичний розподіл затримок для наборів шифрів TLSv1.3 і рівня QoS0 на каналі 4G і MQTT V5

Переваги MQTT над TLS 1.2.

– Широка підтримка: TLS 1.2 широко підтримується на різних платформах і пристроях, забезпечуючи сумісність із широким спектром реалізацій MQTT.

– Встановлена безпека: TLS 1.2 використовується протягом багатьох років і добре зрозумілий розробникам і експертам з безпеки, забезпечуючи надійне шифрування та безпеку для зв'язку MQTT.

– Зріла екосистема: завдяки своїй давній присутності в галузі TLS 1.2 має переваги зрілої екосистеми інструментів, бібліотек і найкращих практик впровадження та керування.

Недоліки MQTT порівняно з TLS 1.2.

– Обмежені функції безпеки: у TLS 1.2 можуть бути відсутні деякі розширені функції безпеки та вдосконалення, наявні в новіших версіях, що потенційно наражає зв'язок MQTT на певні вразливості.

– Накладні витрати на продуктивність: TLS 1.2 може спричинити вищі накладні витрати на продуктивність порівняно з новими версіями TLS, впливаючи на швидкість і ефективність зв'язку MQTT, особливо в середовищах з обмеженими ресурсами.

– Потенційні вразливості: незважаючи на те, що TLS 1.2 забезпечує високий рівень безпеки, він усе ще може бути вразливим до певних відомих уразливостей або атак, що потребує ретельної конфігурації та керування для зменшення ризиків.

Переваги MQTT над TLS 1.3.

– Покращена безпека: у TLS 1.3 представлено кілька покращень безпеки та покращення продуктивності порівняно з TLS 1.2, включаючи надійніші алгоритми шифрування, спрощений процес рукостискання та кращу стійкість до певних типів атак.

– Зменшена затримка: TLS 1.3 зменшує затримку, пов'язану зі встановленням захищеного з'єднання, що забезпечує швидший і чутливіший зв'язок MQTT, що особливо корисно для програм реального часу.

– Пересилання секретності: TLS 1.3 за замовчуванням передбачає пересилання секретності, гарантуючи, що минулі повідомлення неможливо розшифрувати, навіть якщо закритий ключ сервера зламано, підвищуючи загальну безпеку та конфіденційність.

Недоліки MQTT порівняно з TLS 1.3.

– Обмежене застосування: незважаючи на свої переваги, TLS 1.3 може мати не таку широку підтримку, як TLS 1.2, що потенційно може спричинити проблеми сумісності з певними реалізаціями MQTT або вимагати оновлення існуючої інфраструктури.

– Складність: впровадження та керування TLS 1.3 може вимагати додаткових знань і ресурсів порівняно з TLS 1.2, оскільки він представляє нові функції та зміни протоколу, які можуть бути незнайомі розробникам і адміністраторам.

– Проблеми сумісності: у неоднорідних середовищах із сумішшю реалізацій TLS 1.2 і TLS 1.3 можуть виникнути проблеми сумісності, що вимагає ретельного планування та координації для забезпечення безперебійного зв’язку між клієнтами MQTT і брокерами.

Мікропрограмне забезпечення є безпечним варіантом для створення мереж IoT навіть з недорогими пристроями. Першою метою було визначити оптимальні трійки (QoS, набір шифрів, версія TLS), які забезпечують максимально можливу ефективність передачі даних на обмежених пристроях. Було оцінено продуктивність кожної трійки для кожного типу QoS, набору шифрів і версії TLS, реалізованих на двох конкретних обмежених пристроях: маршрутизаторі TP-LINK OpenWrt і двох Raspberry Pi 4.

На основі проведених експериментів алгоритми, які забезпечують найкращу ефективність передачі даних, наведені в табл. 3.1. Ця таблиця містить огляд отриманих результатів. Зокрема, при використанні з’єднання Ethernet 1 Гбіт найкращим і найгіршим наборами шифрів є ECDHE-RSA-CHACHA20-POLY1305 і ECDHE-ECDSA-AES256-SHA відповідно, якщо Wi-Fi 2,4 ГГц, то найкращі та найгірші набори шифрів AES256-GCM-SHA384 і DHE-RSA-

CHACHA20-POLY1305. Обидва ці значення були отримані з урахуванням версії MQTT 3.11, і версії TLS 1.2. Нарешті, для області 4G/LTE найкращим і найгіршим набором шифрів є AES128-SHA з TLSv1.2 і MQTT 5.0, а також TLS-AES-128-GCM-SHA256 з MQTT 3.11 і TLSv1.3. У цьому випадку аналіз версій MQTT на з'єднаннях типу 4G/LTE проводився як для версії 3.1x, так і для версії 5.x протоколу, саме для перевірки продуктивності, досягнутої з менш керованим вектором зв'язку.

Що стосується другої мети роботи, набори шифрів, які гарантують найкращу сумісність із поточними пристроями з проведених експериментів, здається, належать до сімейства TLSv1.2, прозоро прийнятого всіма перевіреними клієнтами. У той же час TLSv1.3 керується нативно лише за допомогою mqttx і hive- mqtt -cli.

Що стосується третьої мети, всі протестовані клієнти дозволяють використовувати версії 3.1, 3.11 і 5.0 MQTT. Тим не менш, клієнти mosquitto можуть узгоджувати лише TLSv1.2 і TLSv1.3, але використовують лише набори шифрів TLSv1.2 і не можуть спілкуватися безпосередньо через WebSocket. Mqttx і mqtt -cli, з іншого боку, легко узгоджують будь-який набір шифрів, сімейство TLS і версію MQTT.

Таблиця 3.1. Найгірші та найкращі результати з точки зору виміряної пропускної здатності (msg/s), якщо значення QoS0 використані як еталон.

Засоб комунікації	Комплект шифров	MQTT	TLS	Найгірший	Найкращий
ГІГАБІТНИЙ ETHERNET	ECDHE-RSA-CHACHA20-POLY1305	V3.11	V1.2		21,295
ГІГАБІТНИЙ ETHERNET	ECDHE-ECDSA-AES256-SHA	V3.11	V1.2	17,353	
WiFi-2.4	AES256-GCM-SHA384	V3.11	V1.2		22,623
WiFi-2.4	DHE-RSA-CHACHA20-POLY1305	V3.11	V1.2	10 527	
WiFi-2.4	ECDHE-ECDSA-AES256-GCM-SHA384	V3.11	V1.3		22 525
WiFi-2.4	DHE-RSA-AES256-GCM-SHA384	V3.11	V1.3	21,553	
WiFi-2.4	AES128-GCM-SHA256	V5.0	V1.2		20 661
WiFi-2.4	ECDHE-RSA-CHACHA20-POLY1305	V5.0	V1.2	10 597	
WiFi-2.4	ECDHE-RSA-AES256-GCM-SHA384	V5.0	V1.3		21,553
WiFi-2.4	DHE-RSA-AES256-GCM-SHA384	V5.0	V1.3	20,242	
4G-LTE	DHE-RSA-CHACHA20-POLY1305	V3.11	V1.2		22,239
4G-LTE	ECDHE-RSA-AES128-GCM-SHA256	V3.11	V1.2	12 819	
4G-LTE	AES128-SHA	V5.0	V1.2		22 526
4G-LTE	ECDHE-ECDSA-AES128-SHA	V5.0	V1.2	13,743	
4G-LTE	ECDHE-ECDSA-AES256-GCM-SHA384	V3.11	V1.3		22,333
4G-LTE	TLS-AES-128-GCM-SHA256	V3.11	V1.3	5376	
4G-LTE	TLS-CHACHA20-POLY1305-SHA256	V5.0	V1.3		21,461
4G-LTE	ECDHE-RSA-AES256-GCM-SHA384	V5.0	V1.3	21,113	

ВИСНОВКИ

Підхід до безпеки на основі VLAN для мереж IoT забезпечує надійний захист даних і високу продуктивність. Комплексна оцінка та порівняльний аналіз наборів шифрів і рівнів QoS дають змогу зрозуміти оптимізацію розгортання IoT, тоді як заходи конфіденційності даних відповідають унікальним потребам безпеки в чутливих областях.

Безсумнівно, бездротові технології є корисними в багатьох сценаріях, особливо в зовнішньому середовищі, де зв'язок обмежений, наприклад у сільській або гірській місцевості. Бездротові з'єднання також корисні під час керування зашифрованими з'єднаннями, які охоплюють великі мережі датчиків, навіть якщо ці мережі складаються з різних типів датчиків. Можна напряму з'єднувати маршрутизатори як активних учасників мережі за допомогою протоколів, таких як локальні мережі Zigbee, з'єднані через шлюзи Raspberry Pi або мережі LoRa/ LoRaWAN , які також можна підключати до мереж LTE/4G/5G. Для цього потрібне спеціальне обладнання та встановлення спеціального програмного забезпечення.

Майбутня робота може включати подальший аналіз з використанням різних типів посередників MQTT (EMQX, RabbitMQ, NanoMQ, VernMQ), налаштованих у дедалі складніших топологіях.

У деяких випадках TLS 1.2 можна вважати більш придатним для апаратного забезпечення з обмеженнями, насамперед через його простіші криптографічні операції та більш широку підтримку серед існуючих пристроїв і платформ Інтернету речей.

Загалом TLS 1.3 пропонує значні переваги у захисті зв'язку в середовищах IoT. Він вирішує унікальні проблеми, пов'язані з пристроями з обмеженими ресурсами, динамічними мережевими умовами та зміною загроз безпеці. Покращення затримки, безпеки, простоти та продуктивності роблять

його ідеальним для захисту цілісності, конфіденційності та доступності систем і даних IoT.

У той час як MQTT через TLS 1.2 пропонує широку підтримку та зрілу екосистему, MQTT через TLS 1.3 забезпечує покращену безпеку та зменшену затримку. Вибір між ними залежить від таких факторів, як вимоги до безпеки, продуктивність і сумісність з існуючою інфраструктурою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chaudhary A., Peddoju S. K., Kadarla K. Study of internet-of-things messaging protocols used for exchanging data with external sources. *2017 IEEE 14th international conference on mobile ad-hoc and sensor systems (MASS)*, м. Orlando, FL, 22–25 жовт. 2017 р. 2017. URL: https://doi.org/10.1109/mass.2017.85

2. Serpanos D., Wolf M. Internet-of-Things (iot) systems. Cham : Springer International Publishing, 2018. URL: https://doi.org/10.1007/978-3-319-69715-4

3. Gupta B. B., Quamara M. An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols. *Concurrency and computation: practice and experience*. 2018. Т. 32, № 21. URL: https://doi.org/10.1002/cpe.4946

4. Internet of things architectures: a comparative study / M. G. d. Santos та ін. 2020. 13 с. (Препринт. 2004.12936). URL: https://doi.org/10.48550/arXiv.2004.12936.

5. Amazon, google and microsoft solutions for iot: architectures and a performance comparison / P. Pierleoni та ін. *IEEE access*. 2020. Т. 8. С. 5455–5470. URL: https://doi.org/10.1109/access.2019.2961511

6. Lombardi M., Pascale F., Santaniello D. Internet of things: a general overview between architectures, protocols and applications. *Information*. 2021. Т. 12, № 2. С. 87. URL: https://doi.org/10.3390/info12020087

7. Sparavigna A. Labels discover physics: the development of new labelling methods as a promising research field for applied physics. 2008. 16 с. (Препринт. 0801.2700). URL: https://doi.org/10.48550/arXiv.0801.2700.

8. Python-based FPGA implementation of AES using migen for internet of things security / K. O. Setetemela та ін. *2019 IEEE 10th international conference on mechanical and intelligent manufacturing technologies (ICMIMT)*, м. Cape Town, South Africa, 15–17 лют. 2019 р. 2019. URL: https://doi.org/10.1109/icmimt.2019.8712074

9. 5G waveforms for iot applications / I. B. F. de Almeida та ін. *IEEE communications surveys & tutorials*. 2019. Т. 21, № 3. С. 2554–2567. URL: https://doi.org/10.1109/comst.2019.2910817

10. LoRaWAN – A low power WAN protocol for Internet of Things: a review and opportunities / J. d. C. Silva та ін. *2017 2nd international multidisciplinary conference on computer and energy science (splitech)*. 2017. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8019271.

11. Analysis the structures of some symmetric cipher algorithms suitable for the security of iot devices / K. F. Jasim та ін. *Cihan university-erbil scientific journal*. 2021. Т. 5, № 2. С. 13–19. URL: https://doi.org/10.24086/cuesj.v5n2y2021.pp13-19

12. Ahamed J., Rajan A. V. Internet of Things (IoT): application systems and security vulnerabilities. *2016 5th international conference on electronic devices, systems and applications (ICEDSA)*, м. Ras Al Khaimah, 6–8 груд. 2016 р. 2016. URL: https://doi.org/10.1109/icedsa.2016.7818534

13. Perception layer security in the internet of things / K. Aarika та ін. *Procedia computer science*. 2020. Т. 175. С. 591–596. URL: https://doi.org/10.1016/j.procs.2020.07.085

14. A secure scheme for group communication of wireless iot devices / B. A. Alohali та ін. *2018 11th international symposium on communication systems, networks and digital signal processing (CSNDSP)*, м. Budapest, 18–20 лип. 2018 р. 2018. URL: https://doi.org/10.1109/csndsp.2018.8471871

15. Pan J., Yang Z. Cybersecurity challenges and opportunities in the new "edge computing + iot" world. *CODASPY '18: eighth ACM conference on data and application security and privacy*, м. Tempe AZ USA. New York, NY, USA, 2018. URL: https://doi.org/10.1145/3180465.3180470

16. Wara M. S., Yu Q. New replay attacks on zigbee devices for internet-of-things (iot) applications. *2020 IEEE international conference on embedded software and systems (ICESS)*, м. Shanghai, 10–11 груд. 2020 р. 2020. URL: https://doi.org/10.1109/icess49830.2020.9301593

17. The end of eavesdropping attacks through the use of advanced end to end encryption mechanisms / L. Maglaras та ін. *IEEE INFOCOM 2022 - IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, м. New York, NY, USA, 2–5 трав. 2022 р. 2022. URL: https://doi.org/10.1109/infocomwkshps54753.2022.9798072

18. Guezzaz A., Benkirane S., Azrour M. A novel anomaly network intrusion detection system for internet of things security. *IoT and smart devices for sustainable environment*. Cham, 2022. С. 129–138. URL: https://doi.org/10.1007/978-3-030-90083-0_10

19. Rathee G., Kerrache C. A., Calafate C. T. An Ambient Intelligence approach to provide secure and trusted Pub/Sub messaging systems in IoT environments. *Computer networks*. 2022. С. 109401. URL: https://doi.org/10.1016/j.comnet.2022.109401

20. Rathee G., Kerrache C. A., Lahby M. TrustBlkSys: a trusted and blockchained cybersecure system for iiot. *IEEE transactions on industrial informatics*. 2022. С. 1–8. URL: https://doi.org/10.1109/tii.2022.3182984

21. Rathee G., Kerrache C. A., Ferrag M. A. A blockchain-based intrusion detection system using viterbi algorithm and indirect trust for iiot systems. *Journal of sensor and actuator networks*. 2022. Т. 11, № 4. С. 71. URL: https://doi.org/10.3390/jsan11040071

22. A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology / S. Singh та ін. *Future generation computer systems*. 2022. Т. 129. С. 380–388. URL: https://doi.org/10.1016/j.future.2021.11.028

23. Cache-Timing attacks still threaten iot devices / S. Takarabt та ін. *Codes, cryptology and information security*. Cham, 2019. С. 13–30. URL: https://doi.org/10.1007/978-3-030-16458-4_2

24. Internet-of-Things security and vulnerabilities: case study / G. Alqarawi та ін. *Journal of applied security research*. 2022. С. 1–17. URL: https://doi.org/10.1080/19361610.2022.2031841

25. Salim M. M., Rathore S., Park J. H. Distributed denial of service attacks and its defenses in IoT: a survey. *The journal of supercomputing*. 2019. Т. 76, № 7. С. 5320–5363. URL: https://doi.org/10.1007/s11227-019-02945-z

26. Salah S., Amro B. M. Big picture: analysis of DDoS attacks map - systems and network, cloud computing, SCADA systems, and IoT. *International journal of internet technology and secured transactions*. 2022. Т. 1, № 1. С. 1. URL: https://doi.org/10.1504/ijitst.2022.10047199

27. Thankappan M., Rifà-Pous H., Garrigues C. Multi-Channel man-in-the-middle attacks against protected wi-fi networks: a state of the art review. *Expert systems with applications*. 2022. С. 118401. URL: https://doi.org/10.1016/j.eswa.2022.118401

28. Dorri A., Mishra S., Jurdak R. Vericom: A Verification and Communication architecture for IoT-based blockchain. *Ad hoc networks*. 2022. С. 102882. URL: https://doi.org/10.1016/j.adhoc.2022.102882

29. English K. V., Obaidat I., Sridhar M. Exploiting memory corruption vulnerabilities in connman for iot devices. *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, м. Portland, OR, USA, 24–27 черв. 2019 р. 2019. URL: https://doi.org/10.1109/dsn.2019.00036

30. Exploiting bluetooth vulnerabilities in e-health iot devices / M. Zubair та ін. *ICFNDS '19: 3rd international conference on future networks and distributed systems*, м. Paris France. New York, NY, USA, 2019. URL: https://doi.org/10.1145/3341325.3342000

31. Papaspirou V., Maglaras L., Ferrag M. A. A tutorial on cross site scripting attack - defense. (Препринт. 202012.0063/v1). URL: https://doi.org/10.20944/preprints202012.0063.v1.

32. FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things / O. Friha та ін. *Journal of parallel and distributed computing*. 2022. Т. 165. С. 17–31. URL: https://doi.org/10.1016/j.jpdc.2022.03.003

33. ML-based IDPS enhancement with complementary features for home iot networks / P. Illy та ін. *IEEE transactions on network and service management*. 2022. С. 1. URL: https://doi.org/10.1109/tnsm.2022.3141942

34. Secure lightweight cryptosystem for IoT and pervasive computing / M. Abutaha та ін. *Scientific reports*. 2022. Т. 12, № 1. URL: https://doi.org/10.1038/s41598-022-20373-7

35. Hashemi S., Zarei M. Internet of Things backdoors: resource management issues, security challenges, and detection methods. *Transactions on emerging telecommunications technologies*. 2020. URL: https://doi.org/10.1002/ett.4142

36. Vignau B., Khoury R., Halle S. 10 years of iot malware: a feature-based taxonomy. *2019 IEEE 19th international conference on software quality, reliability and security companion (QRS-C)*, м. Sofia, Bulgaria, 22–26 лип. 2019 р. 2019. URL: https://doi.org/10.1109/qrs-c.2019.00088

37. Nabiyev B. R. Investigation of computer incidents for cyber-physical infrastructures in industrial control systems. *NATO science for peace and security series – D: information and communication security*. 2022. URL: https://doi.org/10.3233/nicsp220041

38. Security and privacy for green iot-based agriculture: review, blockchain solutions, and challenges / M. A. Ferrag та ін. *IEEE access*. 2020. Т. 8. С. 32031–32053. URL: https://doi.org/10.1109/access.2020.2973178

39. Sadhu P. K., Yanambaka V. P., Abdelgawad A. Internet of things: security and solutions survey. *Sensors*. 2022. Т. 22, № 19. С. 7433. URL: https://doi.org/10.3390/s22197433

40. Chaudhary P., Gupta B. B., Singh A. K. Adaptive cross-site scripting attack detection framework for smart devices security using intelligent filters and attack ontology. *Soft computing*. 2022. URL: https://doi.org/10.1007/s00500-022-07697-2

41. Malicious code detection under 5G hetnets based on a multi-objective RBM model / Z. Cui та ін. *IEEE network*. 2021. Т. 35, № 2. С. 82–87. URL: https://doi.org/10.1109/mnet.011.2000331

42. RMDD: cross layer attack in internet of things / V. K. Asati та ін. *2018 international conference on advances in computing, communications and informatics (ICACCI)*, м. Bangalore, 19–22 верес. 2018 р. 2018. URL: https://doi.org/10.1109/icacci.2018.8554471

43. Radosavac S., Benammar N., Baras J. S. Cross-layer attacks in wireless ad hoc networks. *Conference on information sciences and systems, , march 17–19, 2004* : Conference, 17 берез. 2004 р. URL: https://user.eng.umd.edu/~baras/publications/papers/2004/RadostovacBB_2004.htm.

44. Cross-Layer attack and defense in cognitive radio networks / W. Wang та ін. *GLOBECOM 2010 - 2010 IEEE global communications conference*, м. Miami, FL, USA, 6–10 груд. 2010 р. 2010. URL: https://doi.org/10.1109/glocom.2010.5684069

45. A novel Two-Factor HoneyToken Authentication Mechanism / V. Papaspirou та ін. *2021 international conference on computer communications and networks (ICCCN)*, м. Athens, Greece, 19–22 лип. 2021 р. 2021. URL: https://doi.org/10.1109/icccn52240.2021.9522319

46. Rana M., Mamun Q., Islam R. Lightweight cryptography in IoT networks: a survey. *Future generation computer systems*. 2022. Т. 129. С. 77–89. URL: https://doi.org/10.1016/j.future.2021.11.011

47. Hedayati R., Mostafavi S. A lightweight image encryption algorithm for secure communications in multimedia internet of things. *Wireless personal communications*. 2021. Т. 123, № 2. С. 1121–1143. URL: https://doi.org/10.1007/s11277-021-09173-w

48. draft-garcia-core-security-03. Security considerations in the ip-based internet of things. Вид. офіц. 2013. 44 с. URL: https://datatracker.ietf.org/doc/draft-garcia-core-security/03/.

49. draft-sarikaya-core-sbootstrapping-05. Security bootstrapping solution for resource-constrained devices. Вид. офіц. 2013. URL: https://datatracker.ietf.org/doc/draft-sarikaya-core-sbootstrapping/.

50. Securing communication in 6LoWPAN with compressed IPsec / S. Raza та ін. *2011 international conference on distributed computing in sensor systems (DCOSS)*, м. Barcelona, Spain, 27–29 черв. 2011 р. 2011. URL: https://doi.org/10.1109/dcoss.2011.5982177

51. Rescorla E. Diffie-Hellman key agreement method. RFC Editor, 1999. URL: https://doi.org/10.17487/rfc2631

52. Saied Y. B. Collaborative security for the internet of things : Thèse. 2013. 122 с. URL: https://theses.hal.science/tel-00879790v1.

53. Camtepe S. A., Yener B. Key distribution mechanisms for wireless sensor networks: a survey. *ResearchGate*. URL: https://www.researchgate.net/publication/267241899_Key_Distribution_Mechanisms_for_Wireless_Sensor_Networks_a_Survey.

54. Key management systems for sensor networks in the context of the Internet of Things / R. Roman та ін. *Computers & electrical engineering*. 2011. Т. 37, № 2. С. 147–159. URL: https://doi.org/10.1016/j.compeleceng.2011.01.009

55. Wang Y., Attebury G., Ramamurthy B. A survey of security issues in wireless sensor networks. *IEEE communications surveys & tutorials*. 2006. Т. 8, № 2. С. 2–23. URL: https://doi.org/10.1109/comst.2006.315852

56. Rabin M. O. Digitalized signatures and public-key functions as intractable as factorization, MIT/LCS/TR-212. *MIT Libraries*. URL: https://hdl.handle.net/1721.1/149499.

57. State of the art in ultra-low power public key cryptography for wireless sensor networks / G. Gaubatz та ін. *Proceedings. third IEEE international conference on pervasive computing and communications workshops. percom 2005 workshops*, м. Kauai Island, HI, 8–12 берез. 2005 р. 2005. URL: https://doi.org/10.1109/percomw.2005.76

58. Turner S., Polk T. Prohibiting secure sockets layer (SSL) version 2.0. RFC Editor, 2011. URL: https://doi.org/10.17487/rfc6176

59. A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication / T. Kothmayr та ін. *2012 IEEE 37th conference on local computer networks workshops (LCN workshops)*, м. Clearwater, FL, USA, 22–25 жовт. 2012 р. 2012. URL: https://doi.org/10.1109/lcnw.2012.6424088

60. Lithe: lightweight secure coap for the internet of things / S. Raza та ін. *IEEE sensors journal*. 2013. Т. 13, № 10. С. 3711–3720. URL: https://doi.org/10.1109/jsen.2013.2277656

61. Thubert P. Compression format for ipv6 datagrams over IEEE 802.15.4-based networks / ред. J. Hui. RFC Editor, 2011. URL: https://doi.org/10.17487/rfc6282

62. Towards viable certificate-based authentication for the internet of things / R. Hummen та ін. *The 2nd ACM workshop*, м. Budapest, Hungary, 19 квіт. 2013 р. New York, New York, USA, 2013. URL: https://doi.org/10.1145/2463183.2463193

63. Granjal J., Monteiro E. End-to-end transparent transport-layer security for Internet-integrated mobile sensing devices. *2016 IFIP networking conference (IFIP networking) and workshops*, м. Vienna, Austria, 17–19 трав. 2016 р. 2016. URL: https://doi.org/10.1109/ifipnetworking.2016.7497235

64. Kohl J., Neuman C. The kerberos network authentication service (V5). RFC Editor, 1993. URL: https://doi.org/10.17487/rfc1510

65. Ray S., Biswas G. P. Establishment of ecc-based initial secrecy usable for IKE implementation. *ResearchGate*. URL: https://www.researchgate.net/publication/233483472_Establishment_of_ECC-based_Initial_Secrecy_Usable_for_IKE_Implementation.

66. The case for elliptic curve cryptography. *https://www.nsa.gov/Cybersecurity/*.

67. Shamir A. Identity-Based cryptosystems and signature schemes. *Advances in cryptology*. Berlin, Heidelberg. С. 47–53. URL: https://doi.org/10.1007/3-540-39568-7_5

68. Yang L., Chao Ding, Meng Wu. Establishing authenticated pairwise key using Pairing-based Cryptography for sensor networks. *2013 8th international conference on communications and networking in china (CHINACOM)*, м. Guilin, China, 14–16 серп. 2013 р. 2013. URL: https://doi.org/10.1109/chinacom.2013.6694650

69. Gentry C. Practical identity-based encryption without random oracles. *Advances in cryptology - EUROCRYPT 2006*. Berlin, Heidelberg, 2006. С. 445–464. URL: https://doi.org/10.1007/11761679_27

70. Boneh D., Franklin M. Identity-Based encryption from the weil pairing. *SIAM journal on computing*. 2003. Т. 32, № 3. С. 586–615. URL: https://doi.org/10.1137/s0097539701398521

71. On the energy cost of communication and cryptography in wireless sensor networks / G. de Meulenaer та ін. *2008 IEEE international conference on wireless and mobile computing, networking and communications (WIMOB)*, м. Avignon, France, 12–14 жовт. 2008 р. 2008. URL: https://doi.org/10.1109/wimob.2008.16

72. Szczechowiak P., Collier M. TinyIBE: identity-based encryption for heterogeneous sensor networks. *2009 international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, м. Melbourne, Australia, 7–10 груд. 2009 р. 2009. URL: https://doi.org/10.1109/issnip.2009.5416743

73. Bormann C., Ersue M., Keranen A. Terminology for Constrained-Node Networks. RFC Editor, 2014. URL: https://doi.org/10.17487/rfc7228

74. draft-moskowitz-hip-rg-dex-06. HIP Diet EXchange (DEX). Вид. офіц. 2014. URL: https://datatracker.ietf.org/doc/draft-moskowitz-hip-rg-dex/.

75. Heer T., Jokela P., Henderson T. Host identity protocol version 2 (hipv2) / ред. R. Moskowitz. RFC Editor, 2015. URL: https://doi.org/10.17487/rfc7401

76. HIP security architecture for the ip-based internet of things / F. Vidal Meca та ін. *2013 workshops of 27th international conference on advanced information networking and applications (WAINA)*, м. Barcelona, 25–28 берез. 2013 р. 2013. URL: https://doi.org/10.1109/waina.2013.158

77. Pre-Shared key ciphersuites for transport layer security (TLS) / ред.: P. Eronen, H. Tschofenig. RFC Editor, 2005. URL: https://doi.org/10.17487/rfc4279

78. Eschenauer L., Gligor V. D. A key-management scheme for distributed sensor networks. *The 9th ACM conference*, м. Washington, DC, USA, 18–22 листоп. 2002 р. New York, New York, USA, 2002. URL: https://doi.org/10.1145/586110.586117

79. A key management scheme for wireless sensor networks using deployment knowledge / Wenliang Du та ін. *Ieee infocom 2004*, м. Hong Kong, PR China. URL: https://doi.org/10.1109/infcom.2004.1354530

80. Haowen Chan, Perrig A., Song D. Random key predistribution schemes for sensor networks. *2003 symposium on security and privacy*, м. Berkeley, CA, USA. URL: https://doi.org/10.1109/secpri.2003.1199337

81. A key pre-distribution scheme for secure sensor networks using probability density function of node deployment / T. Ito та ін. *The 3rd ACM workshop*, м. Alexandria, VA, USA, 7 листоп. 2005 р. New York, New York, USA, 2005. URL: https://doi.org/10.1145/1102219.1102233

82. Hwang D. D., Lai B.-C. C., Verbauwhede I. Energy-Memory-Security tradeoffs in distributed sensor networks. *Ad-Hoc, mobile, and wireless networks*. Berlin, Heidelberg, 2004. С. 70–81. URL: https://doi.org/10.1007/978-3-540-28634-9_6

83. A scalable and efficient key establishment protocol for wireless sensor networks / A. Fanian та ін. *2010 ieee globecom workshops*, м. Miami, FL, USA, 6–10 груд. 2010 р. 2010. URL: https://doi.org/10.1109/glocomw.2010.5700195

84. Liu D., Ning P., Li R. Establishing pairwise keys in distributed sensor networks. *ACM transactions on information and system security*. 2005. Т. 8, № 1. С. 41–77. URL: https://doi.org/10.1145/1053283.1053287

85. Blom R. An optimal class of symmetric key generation systems. *Advances in cryptology*. Berlin, Heidelberg. С. 335–338. URL: https://doi.org/10.1007/3-540-39757-4_22

86. Seys S., Preneel B. Key establishment and authentication suite to counter dos attacks in distributed sensor networks. COSIC, 2012.

87. A. Perrig та ін. *Wireless networks*. 2002. Т. 8, № 5. С. 521–534. URL: https://doi.org/10.1023/a:1016598314198

88. Lai B., Kim S., Verbauwhede I. Scalable session key construction protocol for wireless sensor networks. *IEEE workshop on large scale realtime and embedded systems (LARTES)*, 7 груд. 2010 р. URL: https://www.scirp.org/reference/referencespapers?referenceid=1909042.

89. Mattsson J., Tian T. MIKEY-TICKET: ticket-based modes of key distribution in multimedia internet keying (MIKEY). RFC Editor, 2011. URL: https://doi.org/10.17487/rfc6043

90. MIKEY: multimedia internet keying / J. Arkko та ін. RFC Editor, 2004. URL: https://doi.org/10.17487/rfc3830

91. Boudguiga A., Olivereau A., Oualha N. Server assisted key establishment for WSN: a mikey-ticket approach. *2013 12th IEEE international conference on trust, security and privacy in computing and communications (trustcom)*, м. Melbourne, Australia, 16–18 лип. 2013 р. 2013. URL: https://doi.org/10.1109/trustcom.2013.16

92. Protocol for carrying authentication for network access (PANA) / D. Forsberg та ін. ; ред. Y. Ohba. RFC Editor, 2008. URL: https://doi.org/10.17487/rfc5191

93. Extensible authentication protocol (EAP) / B. Aboba та ін. ; ред. H. Levkowetz. RFC Editor, 2004. URL: https://doi.org/10.17487/rfc3748

94. Kanda M., Ohba Y., Das S. PANA applicability in constrained environments. Sources, 2012.

95. SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN) / H. R. Hussen

та ін. *2013 fifth international conference on ubiquitous and future networks (ICUFN)*, м. DA NANG, Vietnam, 2–5 лип. 2013 р. 2013. URL: https://doi.org/10.1109/icufn.2013.6614820

96. Saied Y. B., Olivereau A. D-HIP: A distributed key exchange scheme for HIP-based Internet of Things. *2012 IEEE thirteenth international symposium on "A world of wireless, mobile and multimedia networks" (wowmom)*, м. San Francisco, CA, USA, 25–28 черв. 2012 р. 2012. URL: https://doi.org/10.1109/wowmom.2012.6263785

97. Tailoring end-to-end IP security protocols to the Internet of Things / R. Hummen та ін. *2013 21st IEEE international conference on network protocols (ICNP)*, м. Goettingen, Germany, 7–10 жовт. 2013 р. 2013. URL: https://doi.org/10.1109/icnp.2013.6733571

98. El Moustaine E., Laurent M. A lattice based authentication for low-cost RFID. *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, м. Nice, France, 5–7 листоп. 2012 р. 2012. URL: https://doi.org/10.1109/rfid-ta.2012.6404569

99. Feige U., Fiat A., Shamir A. Zero-knowledge proofs of identity. *Journal of cryptology*. 1988. Т. 1, № 2. С. 77–94. URL: https://doi.org/10.1007/bf02351717

100. draft-mglt-6lo-diet-esp-00. Diet-ESP: a flexible and compressed format for ipsec/esp. Вид. офіц. 2015. 37 с. URL: https://datatracker.ietf.org/doc/draft-mglt-6lo-diet-esp/00/.

101. Marin L., Jara A., Skarmeta A. F. Shifting primes: optimizing elliptic curve cryptography for smart things. *2012 sixth international conference on innovative mobile and internet services in ubiquitous computing (IMIS)*, м. Palermo, Italy, 4–6 лип. 2012 р. 2012. URL: https://doi.org/10.1109/imis.2012.199

102. Low-cost standard signatures in wireless sensor networks: a case for reviving pre-computation techniques? / G. Ateniese та ін. *NDSS symposium 2013*, 23 квіт. 2013 р. URL: https://www.ndss-symposium.org/wp-content/uploads/2017/09/06_5.pdf.

103. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices / I. Chatzigiannakis та ін. *2011 IEEE 8th international conference on mobile ad-hoc and sensor systems (MASS)*, м. Valencia, Spain, 17–22 жовт. 2011 р. 2011. URL: https://doi.org/10.1109/mass.2011.77

104. Kivinen T. Minimal internet key exchange version 2 (ikev2) initiator implementation. RFC Editor, 2016. URL: https://doi.org/10.17487/rfc7815

105. Internet key exchange (ikev2) protocol / ред. C. Kaufman. RFC Editor, 2005. URL: https://doi.org/10.17487/rfc4306

106. Corno F., De Russis L., Mannella L. Helping novice developers harness security issues in cloud-IoT systems. *Journal of reliable intelligent environments*. 2022. URL: https://doi.org/10.1007/s40860-022-00175-4

107. A simulated approach to evaluate side-channel attack countermeasures for the Advanced Encryption Standard / L. Crocetti та ін. *Integration*. 2019. Т. 68. С. 80–86. URL: https://doi.org/10.1016/j.vlsi.2019.06.005

108. Hardware design of an advanced-feature cryptographic tile within the european processor initiative / P. Nannipieri та ін. *IEEE transactions on computers*. 2023. С. 1–14. URL: https://doi.org/10.1109/tc.2023.3278536

109. GitHub - krylovsk/mqtt-benchmark: MQTT broker benchmarking tool. *GitHub*. URL: https://github.com/krylovsk/mqtt-benchmark

110. GitHub - emqx/mqttx: A powerful and all-in-one MQTT 5.0 client toolbox for desktop, CLI and websocket. *GitHub*. URL: https://github.com/emqx/MQTTX

111. GitHub - hivemq/mqtt-cli: MQTT CLI is a useful command line interface for connecting various MQTT clients supporting MQTT 5.0 and 3.1.1. *GitHub*. URL: https://github.com/hivemq/mqtt-cli