МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 60.00.00.000 ПЗ

Група ІПм-24-1

Губаль Олександр

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Губаль Олександр Сергійович

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі ефективної та масштабованої комунікаційної взаємодії в рамках протоколів IoT

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Губаль О.С.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Мельник Віталій Дмитрович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

доц. В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Губалю Олександру Сергійовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі ефективної та масштабованої комунікаційної взаємодії в рамках протоколів IoT ”

керівник проекту (роботи) Мельник В.Д., к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

2. Строк подання студентом проекту (роботи) 25 січня 2026 р.

3. Вихідні дані до проекту (роботи) Формальні моделі і методи побудови інформаційних та програмних технологій IoT

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз предметної області моделювання комутаційних протоколів для IoT систем
2. Теоретичні основи та аналіз комунікаційних протоколів в екосистемі інтернету речей
3. Сучасні підходи та виклики у забезпеченні безпеки протоколу MQTT
4. Методологія та моделі ефективної та масштабованої комунікаційної взаємодії протоколів IoT

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Діаграма потоків даних ProVerif (рис. 2.1)
2. Методологія формальної верифікації безпеки протоколу з використанням ProVerif (рис. 2.2)
3. Алгоритм взаємодії компонентів фреймворку при використанні JWT (рис. 2.3)
4. Механізм авторизації в системі MQTT з використанням стандарту OAuth (рис. 2.4)
5. Діаграма послідовності архітектури системи на базі еліптичної криптографії (ECC) (рис. 2.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Аналіз предметної області моделювання комутаційних протоколів для IoT систем	01.10.2025	виконано
3	Теоретичні основи та аналіз комунікаційних протоколів в екосистемі інтернету речей	22.10.2025	виконано
4	Сучасні підходи та виклики у забезпеченні безпеки протоколу MQTT	15.11.2025	виконано
5	Методологія та моделі ефективної та масштабованої комунікаційної взаємодії протоколів IoT	03.12.2025	виконано
6	Комплексний аналіз характеристик модифікованого протоколу	27.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	25.01.2026	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 75 с., 23 рис., 2 табл., 39 джерел.

Тема: Моделі ефективної та масштабованої комунікаційної взаємодії в рамках протоколів IoT

Мета магістерської роботи: розроблення та формальне обґрунтування моделей ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів шляхом удосконалення протоколу MQTT.

Об'єктом дослідження є процеси комунікаційної взаємодії між компонентами систем Інтернету речей.

Предметом дослідження є моделі, методи та механізми забезпечення ефективної, масштабованої та безпечної комунікаційної взаємодії в рамках IoT-протоколів, зокрема протоколу MQTT.

Результати дослідження

В роботі розроблено та формально обґрунтовано модель ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів. Отримані результати мають наукову новизну та практичну цінність і можуть бути використані при проєктуванні безпечних IoT-архітектур.

Висновок

Розроблено модифіковану модель протоколу MQTT із підвищеним рівнем безпеки та формально підтвердженими властивостями. Виконано формальну верифікацію запропонованої моделі з використанням ProVerif з урахуванням процедур автентифікації, управління сесіями та анулювання доступу.

ІНТЕРНЕТ РЕЧЕЙ, MQTT, КОМУНІКАЦІЙНІ ПРОТОКОЛИ, ІНФОРМАЦІЙНА БЕЗПЕКА, МАСШТАБОВАНІСТЬ, ФОРМАЛЬНА ВЕРИФІКАЦІЯ, PROVERIF, АВТЕНТИФІКАЦІЯ, КРИПТОГРАФІЧНІ ПРОТОКОЛИ.

ABSTRACT

Master Thesis: 75 pp., 23 fig., 2 tab., 39 sources.

Topic: Models of effective and scalable communication interaction within IoT protocols

The purpose of the master's thesis: development and formal justification of models of effective and scalable communication interaction within IoT protocols by improving the MQTT protocol.

The object of the study is the processes of communication interaction between components of Internet of Things systems.

The subject of the study is models, methods and mechanisms for ensuring effective, scalable and secure communication interaction within IoT protocols, in particular the MQTT protocol.

Research results

The work has developed and formally substantiated a model of effective and scalable communication interaction within IoT protocols. The results obtained have scientific novelty and practical value and can be used in the design of secure IoT architectures.

Conclusion

A modified model of the MQTT protocol with an increased level of security and formally confirmed properties has been developed. Formal verification of the proposed model using ProVerif was performed, taking into account authentication, session management, and access revocation procedures.

INTERNET OF THINGS, MQTT, COMMUNICATION PROTOCOLS, INFORMATION SECURITY, SCALABILITY, FORMAL VERIFICATION, PROVERIF, AUTHENTICATION, CRYPTOGRAPHIC PROTOCOLS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ABE (Attribute-Based Encryption) — шифрування на основі атрибутів.

AS (Authentication Server) — сервер автентифікації.

CP-ABE (Ciphertext-Policy Attribute-Based Encryption) — шифрування на основі атрибутів із політикою зашифрованого тексту.

ECC (Elliptic Curve Cryptography) — криптографія на еліптичних кривих.

HMAC (Hash-based Message Authentication Code) — код автентифікації повідомлення на основі хеш-функції.

JWT (JSON Web Token) — веб-токен у форматі JSON.

KP-ABE (Key-Policy Attribute-Based Encryption) — шифрування на основі атрибутів із політикою ключів.

MAC (Message Authentication Code) — код автентифікації повідомлення.

MQTT (Message Queuing Telemetry Transport) — протокол черги повідомлень телеметрії.

MQTTS (Secure MQTT) — захищена версія протоколу MQTT.

OAuth (Open Authorization) — відкритий протокол авторизації.

RFID (Radio Frequency Identification) — радіочастотна ідентифікація.

SHA (Secure Hash Algorithm) — безпечний алгоритм хешування.

SSL (Secure Sockets Layer) — рівень захищених сокетів.

TBLUA (Token-Based Lightweight User Authentication) — полегшена автентифікація користувачів на основі токенів.

TLS (Transport Layer Security) — протокол захисту транспортного рівня.

TPM (Trusted Platform Module) — довірений платформний модуль.

ВСТУП

Актуальність теми.

Стрімкий розвиток концепції Інтернету речей (Internet of Things, IoT) зумовив суттєве зростання кількості розподілених пристроїв, які взаємодіють між собою в гетерогенних мережевих середовищах. Сучасні IoT-системи застосовуються в промисловості, енергетиці, медицині, транспорті, «розумних» містах та побутових рішеннях, що висуває підвищені вимоги до надійності, масштабованості та безпеки комунікаційної взаємодії. Ключовим елементом таких систем є протоколи передачі даних, які забезпечують обмін повідомленнями між пристроями з обмеженими обчислювальними ресурсами.

Серед протоколів прикладного рівня особливе місце займає MQTT, який завдяки своїй легковаговості та ефективності набув широкого поширення в IoT-екосистемах. Водночас базова специфікація MQTT не містить вбудованих механізмів комплексного захисту, що створює потенційні ризики порушення конфіденційності, цілісності та автентичності даних. У зв’язку з цим актуальною є задача розробки та формального обґрунтування моделей безпечної, ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів.

Особливої ваги набуває застосування формальних методів аналізу та верифікації, які дозволяють математично довести коректність реалізації протоколів і відсутність критичних уразливостей. У цьому контексті використання інструментів формальної верифікації, зокрема ProVerif, є перспективним напрямом наукових досліджень. Таким чином, тема магістерської роботи спрямована на розв’язання важливої науково-практичної проблеми підвищення безпеки та масштабованості IoT-комунікацій.

Актуальність дослідження зумовлена швидким зростанням кількості IoT-пристроїв та ускладненням архітектур сучасних інформаційно-

комунікаційних систем. За умов масового розгортання IoT-мереж зростає поверхня атак, а традиційні підходи до забезпечення безпеки не завжди можуть бути ефективно застосовані через обмежені ресурси пристроїв. Уразливості на рівні комунікаційних протоколів можуть призводити до несанкціонованого доступу, підміни повідомлень, атак повторного відтворення та порушення працездатності системи в цілому.

Протокол MQTT, попри його популярність, потребує додаткових механізмів захисту та формального аналізу коректності таких розширень. Більшість існуючих рішень орієнтовані на прикладну реалізацію, але не забезпечують строгого доведення властивостей безпеки. Відсутність формальної верифікації ускладнює оцінювання надійності протоколів у критично важливих IoT-застосуваннях.

У цьому контексті актуальним є розроблення моделей комунікаційної взаємодії, які поєднують ефективність, масштабованість і формально підтверджену безпеку. Використання формальних методів, зокрема верифікатора ProVerif, дозволяє не лише виявляти потенційні уразливості, але й обґрунтовувати коректність запропонованих протокольних рішень. Отже, дослідження відповідає сучасним науковим тенденціям і практичним потребам розвитку безпечних IoT-систем.

Метою магістерської роботи є розроблення та формальне обґрунтування моделей ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів шляхом удосконалення протоколу MQTT.

Об’єктом дослідження є процеси комунікаційної взаємодії між компонентами систем Інтернету речей.

Предметом дослідження є моделі, методи та механізми забезпечення ефективної, масштабованої та безпечної комунікаційної взаємодії в рамках IoT-протоколів, зокрема протоколу MQTT.

Завдання дослідження

Для досягнення поставленої мети в роботі необхідно розв’язати такі завдання:

1) проаналізувати предметну область та існуючі IoT-комунікаційні протоколи;

2) дослідити вразливості та обмеження безпеки протоколу MQTT;

3) виконати порівняльний аналіз протоколів передачі даних в IoT-екосистемах;

4) дослідити сучасні підходи до автентифікації, авторизації та шифрування в IoT;

5) обґрунтувати вибір формальних методів верифікації безпеки;

6) розробити архітектуру модифікованого MQTT-протоколу;

7) виконати формальну верифікацію властивостей безпеки із застосуванням ProVerif.

Методи дослідження

У роботі використано методи системного аналізу та узагальнення для дослідження IoT-протоколів, методи порівняльного аналізу для оцінювання їх характеристик, формальні методи моделювання криптографічних протоколів, методи логічного та функціонального аналізу, а також методи формальної верифікації з використанням інструменту ProVerif.

Наукова новизна отриманих результатів

Наукова новизна роботи полягає в розробленні модифікованої моделі протоколу MQTT із підвищеним рівнем безпеки та формально підтвердженими властивостями. Виконано формальну верифікацію запропонованої моделі з використанням ProVerif з урахуванням процедур автентифікації, управління сесіями та анулювання доступу. Запропоновано комплексний підхід до поєднання масштабованості, ефективності та формальної захищеності IoT-комунікацій.

Практичне застосування результатів

Отримані результати можуть бути використані при проєктуванні та впровадженні безпечних IoT-систем у промислових, корпоративних і критично важливих середовищах. Запропонована модель протоколу може слугувати основою для розроблення захищених брокерів MQTT та систем

керування доступом. Матеріали роботи також можуть бути використані в освітньому процесі під час вивчення дисциплін, пов’язаних з інформаційною безпекою та мережевими технологіями.

Структура магістерської роботи. Представлена робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 75 сторінок, і містить 23 рисунки, 2 таблиці, перелік використаних джерел із 39 позицій.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ МОДЕЛЮВАННЯ КОМУТАЦІЙНИХ ПРОТОКОЛІВ ДЛЯ IoT СИСТЕМ

1.1. Формальний аналіз та оптимізація захищеності прикладного рівня IoT-протоколів

Стрімка інтеграція екосистеми Інтернету речей (IoT) у повсякденну життєдіяльність людини зумовлює формування парадигми повсюдного обчислення (pervasive computing). Проте масове впровадження IoT-пристроїв випереджає розвиток відповідних стандартів безпеки, що робить цей домен критично вразливим для кіберзагроз. Серед широкого спектру прикладних протоколів передачі даних особливе місце посідає MQTT (Message Queue Telemetry Transport), який став де-факто стандартом для систем з обмеженими ресурсами завдяки своїй легкості та ефективності.

Незважаючи на широке розповсюдження, специфікація протоколу MQTT у базовій конфігурації не передбачає обов'язкових механізмів автентифікації та шифрування на рівні протоколу, покладаючись на зовнішні засоби захисту (наприклад, TLS/SSL), які часто є занадто ресурсомісткими для малопотужних сенсорів. Це створює передумови для реалізації атак типу «людина посередині» (MitM), підміни даних та несанкціонованого доступу до брокера повідомлень.

У межах даної роботи проведено комплексний аналіз архітектури безпеки протоколу MQTT та ідентифіковано критичні вразливості, пов'язані з механізмами передачі облікових даних. На основі отриманих результатів розроблено модифікований протокол MQTT з покращеними характеристиками захищеності.

Ключові особливості запропонованого рішення:

- Двостороння (взаємна) автентифікація: Впроваджено механізми верифікації суб’єктів взаємодії на етапах «Видавець – Брокер» та «Підписник – Брокер», що унеможливлює підключення неавторизованих вузлів.

- Оптимізована криптографічна база: Застосовано виключно симетричні криптографічні примітиви, що дозволяє суттєво знизити обчислювальне навантаження на кінцеві пристрої порівняно з асиметричними алгоритмами (RSA/ECC) без втрати стійкості.

- Динамічний розподіл ключів: Запропоновано схему аутентифікації з інтегрованим механізмом генерації та дистрибуції сеансових ключів, що забезпечує пряму секретність передачі повідомлень.

Для підтвердження надійності запропонованої модифікації було проведено формальну верифікацію за допомогою математичного апарату логіки BAN (Burrows-Abadi-Needham) або методів автоматизованого аналізу (наприклад, інструментарію AVISPA/Scyther). Процес моделювання довів стійкість протоколу до атак повторного відтворення (replay attacks) та ін’єкцій повідомлень.

Результати оцінювання безпеки підтвердили, що розроблений протокол гарантує конфіденційність та цілісність криптографічних мандатів. Запропоноване рішення забезпечує високий рівень захищеності IoT-інфраструктури при збереженні низьких експлуатаційних витрат на обробку трафіку, що робить його придатним для впровадження в індустріальних та побутових системах автоматизації.

1.2. Аналіз вразливостей протоколу MQTT у системах Інтернету речей

Згідно з прогнозами, глобальна кількість пристроїв, що функціонують на базі концепції Інтернету речей (IoT), до кінця 2026 року мала сягнути 50 мільярдів одиниць, що свідчить про трикратне зростання порівняно з показниками 2018 року. Технології IoT знаходять широке застосування у стратегічно важливих сферах, зокрема:

- автоматизованих системах пожежогасіння та енергоменеджменту;

- інтелектуальній логістиці;

- системах біотелеметрії та дистанційного моніторингу стану здоров’я;

- робототехніці, військовому спостереженні та сучасних комплексах озброєння [2].

Архітектура IoT-систем передбачає інтеграцію сенсорних мереж, засобів бездротового зв’язку, каналів міжмашинної взаємодії (M2M) та бекенд-інфраструктури. Проте експоненціальне зростання масштабів взаємозв’язності актуалізує критичну необхідність поглибленого дослідження аспектів кібербезпеки в цьому сегменті.

Сучасні IoT-платформи характеризуються наявністю суттєвих вразливостей, що локалізуються на різних рівнях архітектури: безпосередньо на кінцевих пристроях (датчиках та актуаторах), у спеціалізованому програмному забезпеченні, у сховищах даних та, перш за все, у мережевих каналах передачі даних між вузлами та центральними системами [3].

Для забезпечення обміну даними використовуються різноманітні прикладні протоколи, серед яких: AMQP (Advanced Message Queuing Protocol), CoAP (Constrained Application Protocol) та MQTT (Message Queuing Telemetry Transport). Кожен із зазначених протоколів має специфічні функціональні переваги та притаманні йому вектори атак.

1.2.1. Критичний аналіз безпеки протоколу MQTT

Протокол MQTT на сьогодні є домінуючим стандартом комунікації в екосистемах IoT. Його популярність серед розробників зумовлена мінімальними вимогами до пропускної здатності каналів зв’язку та обсягу оперативної пам’яті пристроїв [4]. Однак специфікація стандарту MQTT не містить суворих регламентів щодо забезпечення інформаційної безпеки.

Основними недоліками базової реалізації MQTT є:

- Відсутність вбудованого шифрування: Протокол забезпечує лише базову автентифікацію, залишаючи дані у відкритому вигляді під час транспортування.

- Ризики цілісності та конфіденційності: Відсутність механізмів криптографічного захисту створює загрозу несанкціонованого доступу, перехоплення та модифікації приватних даних.

З огляду на виявлені дефіцити безпеки, актуальним науковим завданням є розробка вдосконаленої модифікації протоколу MQTT. Для підтвердження стійкості запропонованих безпекових рішень доцільним є використання інструментів формальної верифікації, зокрема автоматизованого верифікатора криптографічних протоколів ProVerif. Такий підхід дозволяє систематично аналізувати вразливості та забезпечити високий рівень захищеності інформаційної взаємодії в IoT-мережах [5].

1.2.2. Порівняльна характеристика протоколів передачі даних в екосистемах IoT

Нижче наведено порівняльний аналіз протоколів MQTT та CoAP, які є найбільш розповсюдженими у пристроях з обмеженими ресурсами (constrained devices).

Таблиця 1.1.

Характеристика протоколів передачі даних в екосистемах IoT

Параметр порівняння	**Message Queuing Telemetry Transport (MQTT)**	**Constrained Application Protocol (CoAP)**
Архітектурна модель	Публікація/Підписка (Publish/Subscribe) через брокер	Клієнт-Сервер (Request/Response)
Транспортний протокол	TCP (орієнтований на з'єднання)	UDP (без встановлення з'єднання)
Об’єм заголовка	Мінімальний (2 байти)	Компактний двобінарний (4 байти)
Механізм безпеки	TLS/SSL (транспортний рівень)	DTLS (Datagram TLS)
Надійність доставки (QoS)	Три рівні (0, 1, 2)	Два рівні (Confirmable / Non-confirmable)
Енергоспоживання	Помірне (через утримання TCP-сесії)	Низьке (оптимізовано для "сплячих" вузлів)
Сфера застосування	Моніторинг, системи зі стабільним зв'язком	Автономні сенсори, мережі з високими втратами пакетів

Вибір між цими протоколами часто базується на компромісі між надійністю та ресурсомісткістю:

1. MQTT демонструє вищу надійність завдяки використанню TCP та чітко визначених рівнів Quality of Service (QoS), проте накладні витрати на підтримку активного з'єднання можуть бути критичними для пристроїв з автономним живленням.

2. CoAP, працюючи через UDP, є ефективнішим для мереж із низькою пропускною здатністю, але потребує додаткових механізмів обробки втрачених пакетів на прикладному рівні.

З точки зору безпеки обидва протоколи за замовчуванням передають дані у відкритому вигляді. Використання TLS (для MQTT) або DTLS (для CoAP) значно збільшує обчислювальне навантаження на мікроконтролери, що підтверджує необхідність розробки полегшених (lightweight) криптографічних схем.

1.3. Аналітичний опис протоколів взаємодії в IoT

В архітектурі Інтернету речей (IoT) протоколи взаємодії класифікуються за рівнями моделі OSI, проте найважливішу роль відіграє прикладний рівень, де визначається спосіб обміну даними між пристроями, шлюзами та хмарними серверами.

Нижче наведено опис основних протоколів та їх ієрархічну структуру.

1. Класифікація протоколів за рівнями архітектури

Для розуміння взаємодії важливо розглядати стеки протоколів, які адаптовані під обмежені ресурси (енергоспоживання, пам'ять).

- Рівень ідентифікації та зв’язку (Physical/Data Link): IEEE 802.15.4 (Zigbee), Bluetooth Low Energy (BLE), LoRaWAN, NB-IoT.

- Мережевий рівень (Network/Transport): IPv6, 6LoWPAN, UDP, TCP.

- Прикладний рівень (Application): MQTT, CoAP, HTTP, AMQP.

2. Ключові протоколи прикладного рівня

MQTT (Message Queuing Telemetry Transport) - це протокол, що базується на моделі «публікація-підписка» (Publish/Subscribe). Він є стандартом для IoT завдяки своїй легкості та здатності працювати в нестабільних мережах.

Центральним елементом є брокер (Broker), який отримує повідомлення від видавців та розсилає їх підписникам. Перевага - мінімальний розмір заголовка та підтримка рівнів якості обслуговування (QoS).

CoAP (Constrained Application Protocol) - спеціалізований протокол для пристроїв із суворими обмеженнями (датчики з живленням від батарей). На відміну від MQTT, він працює за моделлю «запит-відповідь» (Client/Server) поверх UDP.

Використовує RESTful архітектуру (методи GET, POST, PUT, DELETE), подібну до HTTP, але у бінарному форматі.

AMQP (Advanced Message Queuing Protocol) - протокол черг повідомлень, орієнтований на бізнес-додатки та інтероперабельність між серверами.

Використовується в IoT-шлюзах для передачі агрегованих даних у великі аналітичні системи (наприклад, Azure IoT Hub).

Вибір протоколу залежить від топології мережі та вимог до передачі даних.

Таблиця 1.2.

Графічне порівняння архітектурних моделей

Характеристика	**MQTT**	**CoAP**	**HTTP**
Модель	Публікація/Підписка	Запит/Відповідь	Запит/Відповідь
Транспорт	TCP	UDP	TCP
Заголовок	2 байти (фіксований)	4 байти	Десятки/сотні байтів
Зв'язок	Постійний (Asynchronous)	Дискретний (Synchronous)	Дискретний

Рис. 1.1. Графічна інтерпретація протоколів IoT

Сучасна IoT-система зазвичай є гібридною: локальні датчики можуть взаємодіяти через CoAP або Zigbee, шлюз агрегує ці дані та передає їх у хмару через MQTT, а кінцевий користувач отримує інформацію у мобільному застосунку через REST API (HTTP).

1.4. Методика дослідження вразливостей протоколу комунікації в IoT

Метою даної роботи є комплексне дослідження вразливостей протоколу MQTT в екосистемах Інтернету речей (IoT) та розробка його модифікованої версії з покращеними характеристиками безпеки без внесення надлишкових обчислювальних обмежень.

Для досягнення поставленої мети визначено такі основні завдання:

- Провести системний аналіз існуючих дефектів безпеки протоколу MQTT, що застосовується в IoT-рішеннях.

- Ідентифікувати та класифікувати специфічні вразливості на основі аналізу типових сценаріїв використання протоколу.

- Розробити вдосконалений протокол передачі даних, який інтегрує додаткові механізми захисту, зберігаючи при цьому архітектурну сумісність та ефективність базового стандарту.

Методологічна база дослідження ґрунтується на системному підході та включає кілька послідовних етапів:

1. Аналітичний етап.

Здійснено критичний огляд наукової літератури та нормативної документації для вивчення поточного стану специфікації MQTT та його існуючих модифікацій. Це дозволило сформувати перелік актуальних загроз та сценаріїв їх виникнення.

2. Діагностичний етап.

На основі виявлених недоліків проаналізовано першопричини відсутності необхідних функцій безпеки у стандартній реалізації. Враховуючи гетерогенність середовища IoT та обмеженість апаратних ресурсів вузлів, було обґрунтовано вибіркове впровадження безпекових компонентів.

3. Етап проектування та верифікації: розроблено архітектуру покращеного протоколу, яка інтегрується в існуючу інфраструктуру без порушення її функціональності. Для підтвердження ефективності запропонованих рішень проведено формальну верифікацію за допомогою інструменту ProVerif.

4. Метод формальної перевірки дозволяє охопити критичні сценарії тестування, які не були враховані в оригінальному стандарті. Оцінка результатів здійснювалася за такими метриками: забезпеченість конфіденційності, цілісність даних та стійкість механізмів автентифікації [7].

Об’єкт дослідження обмежений процесами проєктування та функціонування комунікаційних протоколів в мережах IoT. Особливу увагу зосереджено на протоколі MQTT як одному з найбільш енергоефективних та поширених стандартів у цьому домені.

Межі дослідження охоплюють:

- Аналіз вразливостей та векторів атак на протокол MQTT на основі репрезентативної вибірки випадків використання (use cases).

- Проєктування програмних рішень для підвищення рівня захищеності протоколу.

- Математичне та логічне підтвердження відсутності вразливостей у модифікованому протоколі за допомогою верифікатора ProVerif.

Такий підхід забезпечує високу достовірність отриманих результатів та підтверджує відповідність запропонованого протоколу сучасним вимогам кібербезпеки.

Висновки до розділу

У першому розділі здійснено комплексний аналіз предметної області моделювання комунікаційних протоколів для систем Інтернету речей. Розглянуто особливості функціонування протоколів прикладного рівня з урахуванням обмежень IoT-середовищ та вимог до ефективності обміну даними. Проведено формальний аналіз захищеності IoT-протоколів, що дозволило виявити типові загрози та потенційні вразливості на різних етапах комунікаційної взаємодії. Значну увагу приділено протоколу MQTT як одному з базових стандартів для побудови масштабованих IoT-систем. Виконано критичний аналіз безпеки MQTT, у результаті якого встановлено недостатність вбудованих механізмів автентифікації та контролю доступу. Порівняльний аналіз MQTT з іншими протоколами передачі даних дозволив окреслити його переваги з точки зору продуктивності та масштабованості. Водночас було показано, що підвищення рівня безпеки MQTT потребує

застосування додаткових криптографічних і організаційних механізмів. Надано аналітичний опис основних протоколів взаємодії в IoT-екосистемах. Запропоновано методику дослідження вразливостей комунікаційних протоколів, орієнтовану на системний аналіз та формалізацію загроз. Отримані результати стали теоретичною основою для подальшого поглибленого дослідження та розробки захищеної моделі комунікаційної взаємодії.

РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ КОМУНІКАЦІЙНИХ ПРОТОКОЛІВ В ЕКОСИСТЕМІ ІНТЕРНЕТУ РЕЧЕЙ

У межах даного розділу представлено концептуальний апарат дослідження, що формує підґрунтя для подальшого теоретичного аналізу, з особливим акцентом на протоколі MQTT. Логіка викладу передбачає перехід від загальних принципів побудови мереж IoT до критичного огляду існуючих рішень у сфері безпеки передачі даних. На завершення розділу наведено аналітичне резюме, яке ідентифікує актуальні наукові прогалини та обґрунтовує мотивацію розробки вдосконаленого безпекового протоколу. Автором проведено ретроспективний аналіз наукових праць, присвячених дослідженню вразливостей MQTT, що дало змогу систематизувати основні спостереження та визначити вектори подальшої модернізації.

2.1. Генезис та класифікація комунікаційних протоколів IoT

Концепція Інтернету речей (IoT) базується на синергії двох технологічних доменів: бездротових комунікацій та сенсорних мереж. Сучасні вбудовані системи (embedded systems) функціонують як автономні обчислювальні вузли на базі мікроконтролерів, що здійснюють прецизійний збір даних із навколишнього середовища. Така архітектура забезпечує конвергенцію фізичного простору та інформаційних мереж (IT networks), що є ключовим етапом цифрової трансформації [8].

Завдяки еволюції сенсорних технологій та засобів ідентифікації об’єктів, сучасна інфраструктура Інтернету забезпечує безперебійну передачу та отримання повідомлень через бездротові канали. У цьому контексті параметри автентичності та цілісності інформаційного обміну набувають критичного значення. Для мінімізації ризиків несанкціонованого доступу було розроблено низку методів управління криптографічними

ключами, що сприяє підвищенню операційної ефективності, точності даних та досягненню економічної вигоди від впровадження систем [9].

На сучасному етапі розвитку IoT розроблено широкий спектр протоколів, призначених для верифікації та захищеної трансляції пакетів даних, зокрема: CoAP, ZigBee, 6LoWPAN, Bluetooth тощо. Серед них протокол MQTT займає панівне становище у сегменті міжмашинної взаємодії (M2M). Слід зазначити, що повнофункціональна екосистема IoT базується на інтеграції чотирьох модульних компонентів: апаратної частини, засобів зв'язку, модулів обробки даних та користувацького інтерфейсу [10].

Розглянемо ключові технічні характеристики основних протоколів.

ZigBee базується на стандарті IEEE 802.15.4; призначений для створення персональних мереж (PAN) з низьким рівнем енергоспоживання. Знаходить застосування у домашній автоматизації та системах моніторингу медичних показників.

CoAP (Constrained Application Protocol) - спеціалізований протокол прикладного рівня (стандарт RFC 7252), адаптований для пристроїв з обмеженими ресурсами. Фактично є оптимізованою інтерпретацією HTTP, що функціонує поверх транспортного протоколу UDP для економії пропускної здатності.

6LoWPAN забезпечує передачу IPv6-пакетів через низькопотужні бездротові мережі, що дозволяє залучати до глобальної мережі пристрої з мінімальною обчислювальною потужністю.

AMQP (Advanced Message Queuing Protocol) відкритий стандарт (OASIS) для комерційного обміну даними. Забезпечує високу надійність та асинхронність комунікації; підтримує архітектуру SASL для автентифікації та TLS для забезпечення конфіденційності.

Bluetooth/BLE - популярні рішення для передачі даних на малих відстанях. Версія Bluetooth Low Energy (BLE) є пріоритетною для IoT-рішень завдяки низькій вартості впровадження та енергоефективності [11-13].

2.1.1 Аналіз аспектів інформаційної безпеки

Специфічні обмеження домену IoT та вимоги до конфіденційності користувачів генерують множину векторів загроз для цілісності системи. Розробка вдосконалених захищених протоколів є нагальним науковим завданням, оскільки поточні ринкові тренди часто пріоритезують зручність використання та мінімізацію витрат над параметрами безпеки. В умовах домінування бездротових технологій стійкість системи безпосередньо залежить від механізмів підтвердження легітимності повідомлень.

2.1.2 Систематизація характеристик протоколів IoT

Відповідно до поточної наукової парадигми, виділяють сім фундаментальних характеристик систем IoT [14]:

- Зв’язність (Connectivity): забезпечення стабільного з’єднання між гетерогенними рівнями архітектури, включаючи датчики, апаратне забезпечення та мережеві вузли.

- Об’єктність (Things): наявність апаратних компонентів, оснащених сенсорними модулями для реєстрації фізичних параметрів.

- Дані (Data): первинний субстрат IoT, аналіз якого є передумовою для прийняття інтелектуальних рішень.

- Комунікація (Communication): процес обміну даними між пристроями на різних відстанях (через Bluetooth, Wi-Fi, LoRa, ZigBee тощо).

- Інтелектуальність (Intelligence): здатність системи до предиктивного аналізу та інтерпретації великих масивів даних (Big Data).

- Активність (Action): реалізація керуючих впливів у ручному або автоматизованому режимах залежно від отриманих результатів обробки інформації.

- Екосистемність (Ecosystem): інтеграція IoT у ширший соціально-технологічний контекст, що передбачає взаємодію з іншими сучасними технологіями та цифровими інфраструктурами.

2.2 Архітектурна організація та механізми функціонування протоколу MQTT

Протокол MQTT (Message Queuing Telemetry Transport) базується на асинхронній моделі обміну повідомленнями «публікація-підписка» (Publish/Subscribe). На відміну від традиційної архітектури «клієнт-сервер», де вузли взаємодіють безпосередньо, MQTT впроваджує посередника — брокера повідомлень, що дозволяє повністю декумулювати відправника (видавця) та отримувача (підписника) у часі, просторі та за параметрами синхронізації.

2.2.1 Основні компоненти архітектури

Структурно взаємодія в межах протоколу реалізується через три ключові суб'єкти:

- Видавець (Publisher): пристрій (зазвичай датчик або контролер), який генерує дані та надсилає їх брокеру у вигляді повідомлень, асоційованих із конкретною тематикою.

- Брокер (Broker): центральний вузол мережі, відповідальний за прийом повідомлень, їх фільтрацію відповідно до тематик та подальшу дистрибуцію авторизованим підписникам. Брокер також забезпечує керування сесіями та контроль автентифікації.

- Підписник (Subscriber): кінцевий вузол (застосунок, сервер або інший пристрій), який висловлює зацікавленість у певних даних шляхом створення підписки на відповідні тематичні канали.

2.2.2 Ієрархічна структура тематик (Topics)

Для логічної організації даних у MQTT використовується концепція тематик (Topics), що мають деревоподібну ієрархічну структуру. Рівні ієрархії розділяються символом слеша (/), що дозволяє гнучко структурувати потоки даних.

Приклад: *building_A/floor_1/sensor_temp.*

Така організація дозволяє підписникам використовувати спеціальні символи підстановки (wildcards), наприклад + (для одного рівня) або # (для всіх підрівнів), що значно спрощує масштабування систем моніторингу.

2.2.3 Параметри якості обслуговування (QoS)

Однією з фундаментальних особливостей MQTT, що забезпечує надійність передачі в умовах нестабільного зв'язку, є механізм Quality of Service (QoS). Специфікація передбачає три рівні гарантії доставки:

- QoS 0 (At most once): повідомлення доставляється максимум один раз без підтвердження отримання. Це мінімізує накладні витрати, але допускає втрату пакетів.

- QoS 1 (At least once): гарантує доставку повідомлення підписнику, проте існують ризики дублювання пакетів у разі затримок підтвердження (PUBACK).

- QoS 2 (Exactly once): найвищий рівень надійності, що використовує чотирьохетапне рукостискання для гарантії отримання повідомлення рівно один раз без дублювання.

2.2.4 Функціональні стани та механізм Keep Alive

Для моніторингу стану з'єднання між клієнтом та брокером використовується параметр Keep Alive. Це часовий інтервал, протягом якого клієнт зобов'язаний надіслати керуючий пакет (PINGREQ). Якщо брокер не отримує сигнал протягом визначеного часу, з'єднання вважається розірваним.

Важливою функцією також є "Last Will and Testament" (LWT) — механізм, який дозволяє брокеру автоматично сповістити всіх підписників про некоректне відключення видавця, що є критично важливим для систем реального часу.

2.2.5 Сучасні підходи та виклики у забезпеченні безпеки протоколу MQTT

Для мінімізації ризиків та підвищення стійкості систем на базі MQTT на сучасному етапі застосовується комплекс загальноприйнятих методів захисту. Основними напрямками є впровадження розширених механізмів автентифікації, багаторівнева перевірка суб'єктів взаємодії, а також використання протоколів криптографічного захисту транспортного рівня, таких як TLS/SSL.

Крім базових засобів, існують спеціалізовані концепції та програмні реалізації безпеки в інфраструктурі MQTT, зокрема:

- верифікація клієнтських сертифікатів стандарту X.509;

- інтеграція протоколів авторизації OAuth 2.0 для розмежування прав доступу;

- забезпечення цілісності та конфіденційності безпосередньо на рівні корисного навантаження (payload security).

Слід зауважити, що автентична специфікація протоколу MQTT містить лише обмежений набір інструментів безпеки. Відтак, у практичних реалізаціях розробники зазвичай покладаються на зовнішні стандарти захисту, де SSL/TLS виступає основним засобом гарантування безпеки транспортування даних. Такий підхід зумовлений високою складністю самостійного проектування криптографічних систем, що робить доцільним використання верифікованих та загальновизнаних безпекових стандартів [18].

Окрему увагу слід приділити проблемі захисту конфіденційної інформації, що транслюється пристроями IoT. У багатьох сценаріях дані призначені виключно для авторизованих вузлів або конкретних апаратних засобів. Оскільки стандартний механізм безпеки MQTT за замовчуванням здійснює лише базову перевірку ідентифікаторів без застосування шифрування під час передачі, виникають суттєві загрози для:

- Конфіденційності даних: можливість перехоплення повідомлень.

- Цілісності інформації: ризик несанкціонованої модифікації пакетів у процесі трансляції.

- Верифікованості вузлів: складність підтвердження справжності відправника в умовах відсутності наскрізного шифрування.

Таким чином, незважаючи на широку функціональність, базова реалізація MQTT потребує суттєвої модернізації на прикладному рівні для забезпечення надійного захисту в критично важливих сегментах Інтернету речей.

Отже, аналіз архітектури MQTT демонструє його високу адаптивність до умов IoT, проте виявляє відсутність вбудованих засобів криптографічного захисту на рівні самого протоколу. Це створює передумови для реалізації атак типу «людина посередині» (MITM) та несанкціонованої підписки на топіки, що обґрунтовує необхідність розробки вдосконаленого безпекового рівня, якому присвячено наступні розділи роботи.

2.3 Методологія формальної верифікації безпеки протоколу з використанням ProVerif

Для підтвердження надійності запропонованих удосконалень протоколу MQTT та гарантування відсутності логічних уразливостей у дипломній роботі застосовано метод формальної верифікації. Основним інструментарієм обрано ProVerif — автоматизований верифікатор криптографічних протоколів, що базується на моделі Долева-Яо (Dolev-Yao model).

2.3.1 Характеристика верифікатора криптографічних протоколів ProVerif

ProVerif становить собою спеціалізоване програмне забезпечення, призначене для формальної верифікації властивостей безпеки криміналістичних та комунікаційних протоколів, зокрема тих, що

функціонують у гетерогенному середовищі Інтернету речей (IoT). Даний інструментарій на сьогодні є одним із найбільш затребуваних засобів автоматизованого аналізу криптографічних систем.

Цей автоматизований інструмент застосовується на етапі верифікації безпеки проектних рішень. Теоретичний базис ProVerif ґрунтується на формалізмі прикладної π-обчислювальної логіки (applied pi-calculus). Завдяки цьому інструмент здатний ефективно вирішувати завдання, пов'язані з неоднозначністю станів та анонімністю суб’єктів у відповідних доменах дослідження. Важливою перевагою ProVerif є здатність моделювати та обробляти необмежену кількість сесій протоколу під час тестування, що дозволяє виявляти вразливості, які виникають при масштабуванні мережі [20-21].

Процес верифікації за допомогою даного інструментарію передбачає чітку послідовність етапів обробки даних (рис. 2.1).

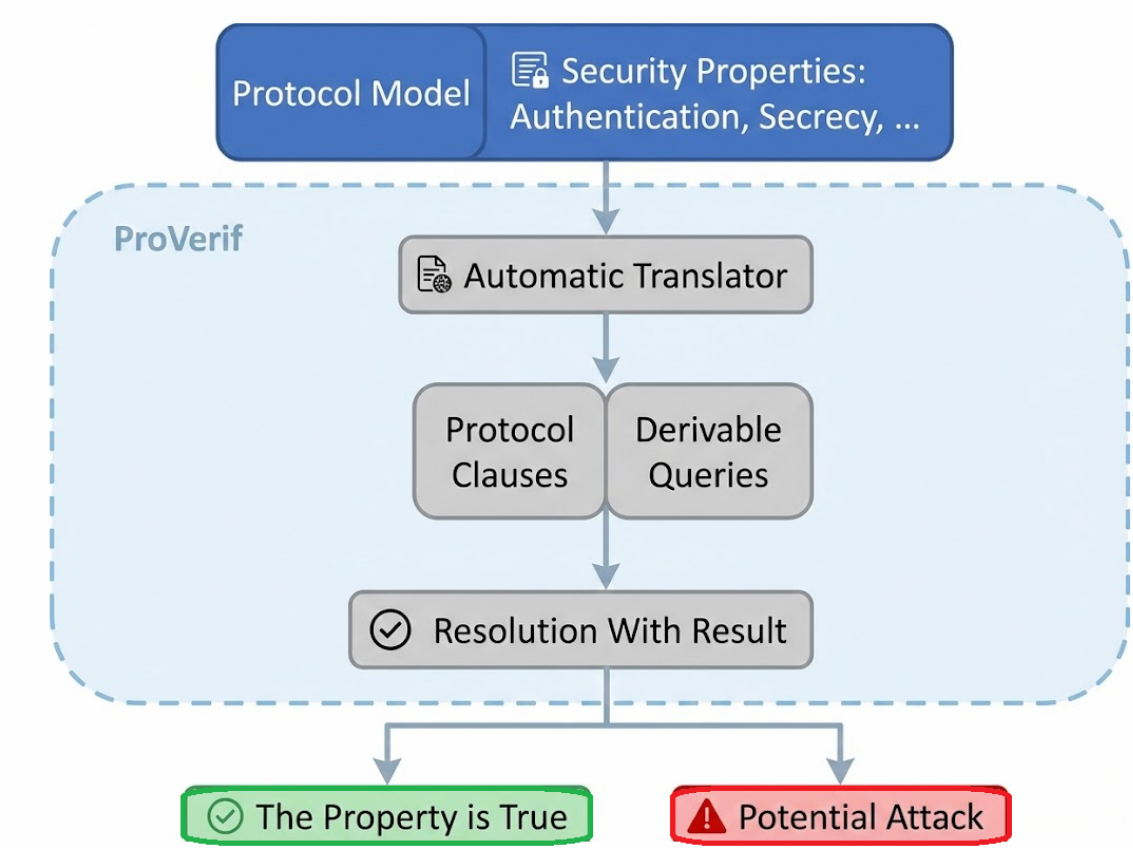

Рис. 2.1. Діаграма потоків даних ProVerif

Відповідно до представленої архітектури (рис. 2.1), ProVerif здійснює прийом вихідних специфікацій (опис процесів, термів та ініціальних знань зловмисника). За допомогою інтегрованого автоматичного транслятора ці дані трансформуються у внутрішні логічні конвенції та формалізовані запити безпеки (queries), які підлягають подальшій перевірці на відповідність заданим критеріям стійкості [22].

2.3.2 Обґрунтування вибору інструментарію

ProVerif дозволяє аналізувати властивості безпеки, такі як секретність (secrecy), автентифікація (authentication) та цілісність (integrity) у середовищі, де потенційний зловмисник має повний контроль над мережею (може перехоплювати, змінювати, видаляти або впорскувати довільні повідомлення). Вибір даного засобу зумовлений його здатністю працювати з необмеженою кількістю сесій, що є критичним для масштабованих IoT-систем.

2.3.3 Етапи формального моделювання

Процес верифікації в межах даного дослідження розділений на чотири послідовні фази:

- Декларація термів та функцій: На цьому етапі визначаються типи даних (наприклад, ідентифікатори пристроїв, ключі шифрування), а також конструктори та деструктори для криптографічних операцій: симетричного та асиметричного шифрування, хешування та створення цифрових підписів.

Приклад: *fun encrypt(bitstring, key): bitstring.*

- Визначення моделі порушника (Attacker Model): Згідно з моделлю Долева-Яо, зловмисник має доступ до публічного каналу зв'язку. У моделі описуються знання, якими він володіє спочатку (наприклад, публічні ключі), та можливості щодо компрометації сесійних ключів.

- Опис процесів (Process Calculus): Формалізується поведінка кожного учасника обміну (Видавець, Брокер, Підписник) у вигляді паралельних процесів. Кожен крок протоколу — від встановлення з'єднання (CONNECT) до публікації даних (PUBLISH) — описується як послідовність операцій введення/виведення (in, out) та перевірок.

- Формулювання запитів (Queries): Це визначення цілей безпеки, які необхідно довести. У ProVerif вони формулюються через запити:

- query secret data — чи може зловмисник отримати доступ до конфіденційних даних?

- query event(endA(x)) ==> event(beginB(x)) — перевірка відповідності (correspondence assertions) для підтвердження успішної автентифікації між вузлами.

2.3.4 Метрики оцінки результатів верифікації

Результатом роботи верифікатора є логічне підтвердження (RESULT ... is true) або спростування властивості з наданням контрприкладу (attack trace), який візуалізує послідовність дій зловмисника для здійснення атаки.

В межах роботи аналізуються такі параметри:

- Стійкість до атак повторного відтворення (Replay Attack): перевірка унікальності часових міток (timestamps) або випадкових чисел (nonces) у структурі повідомлення MQTT.

- Конфіденційність корисного навантаження (Payload Secrecy): гарантування, що дані датчиків доступні лише авторизованим підписникам навіть за умови компрометації брокера.

- Автентичність джерела: підтвердження, що повідомлення відправлено саме тим пристроєм, чий ідентифікатор вказано в пакеті.

2.3.5 Переваги застосування ProVerif у розрізі стандарту MQTT

На відміну від традиційного тестування, формальна верифікація дозволяє виявити архітектурні недоліки ще на етапі проектування. Це

особливо важливо для протоколу MQTT, де стандартні механізми безпеки часто ігноруються через обмеженість ресурсів IoT-пристроїв. Моделювання дозволяє знайти "золоту середину" між криптографічною стійкістю та обчислювальною складністю.

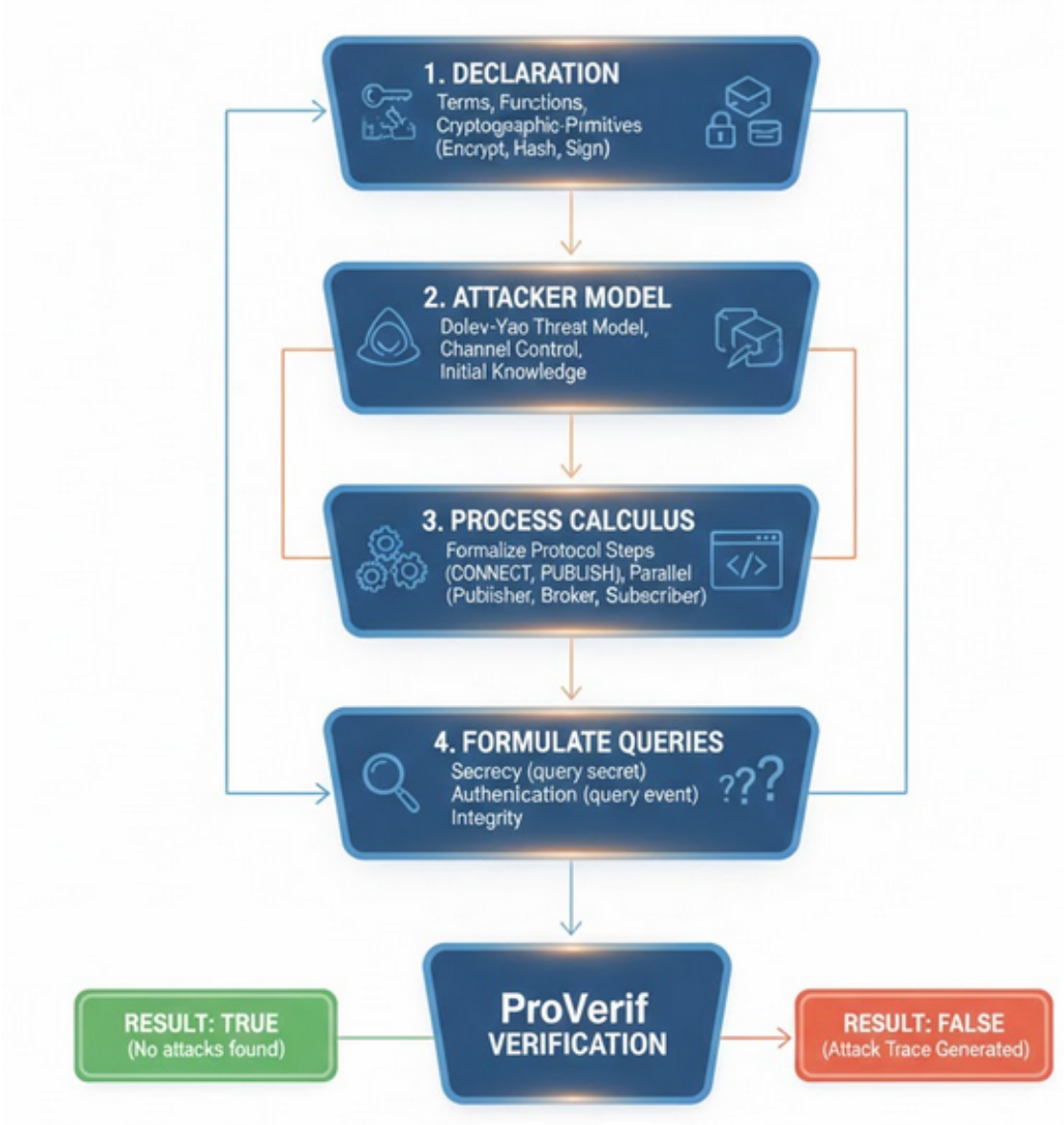

Рис. 2.2. Методологія формальної верифікації безпеки протоколу з використанням ProVerif

Застосування ProVerif дозволяє перейти від емпіричного тестування до формальних методів доведення безпеки, що гарантує виявлення потенційних атак ще на стадії проектування архітектури MQTT-взаємодії.

2.4. Аналіз релевантних досліджень та існуючих рішень

2.4.1 Механізми автентифікації на основі токенів

Протягом останніх років у домені IoT спостерігається тенденція до впровадження полегшених (lightweight) схем автентифікації користувачів, що базуються на використанні токенів. Аналіз наукових праць свідчить про зростання актуальності таких підходів завдяки їхній масштабованості. Зокрема, у дослідженні [23] запропоновано архітектуру, інтегровану із сервером JSON Web Token (JWT) як центральним вузлом автентифікації.

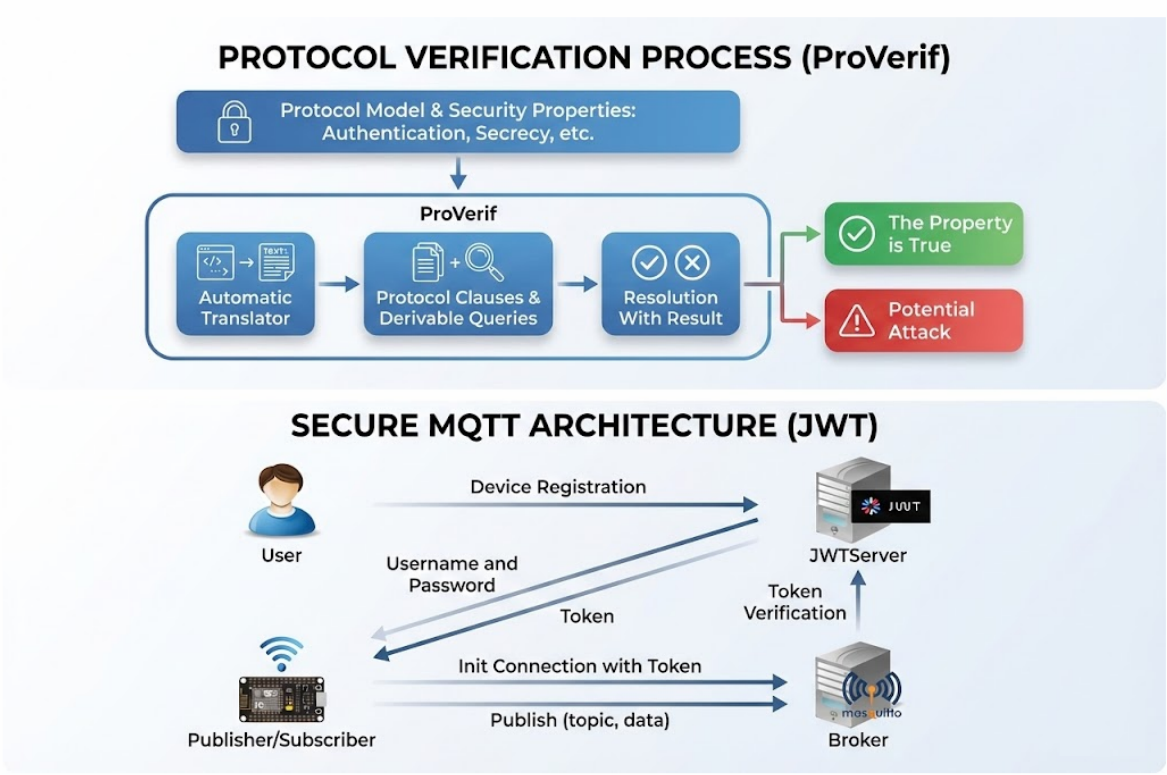

Рис. 2.3. Алгоритм взаємодії компонентів фреймворку при використанні JWT

Автори розробили модель, де клієнт передає ідентифікаційні дані (username/password) серверу JWT, який проводить верифікацію за базою даних. У разі успішної перевірки сервер генерує токен, що зберігається в

локальному сховищі клієнта. Брокер MQTT, у свою чергу, надає права на публікацію та підписку лише за умови пред'явлення валідного токена [24].

Процедура функціонування системи:

- Ініціація запиту токена клієнтом через сервер автентифікації з використанням облікових даних.

- Генерація та надання токена після успішної верифікації.

- Пред'явлення токена брокеру під час встановлення сесії.

- Вторинна перевірка легітимності токена брокером через сервер автентифікації.

- Надання доступу до тематичних ресурсів (topics) для авторизованих суб’єктів.

Застосування токенів дозволяє мінімізувати обсяг службового трафіку порівняно з традиційними методами. Проте слабким місцем такої архітектури є наявність «єдиної точки відмови» (single point of failure) в особі сервера автентифікації, що може негативно вплинути на відмовостійкість системи.

2.4.2 Застосування стандартів авторизації OAuth

OAuth є відкритим протоколом авторизації, який забезпечує розмежований доступ до ресурсів без необхідності передачі конфіденційних облікових даних безпосередньо сервісам. В [25] запропонували адаптацію стандарту OAuth 1.0a для протоколу MQTT. Дослідники обґрунтували вибір версії 1.0a тим, що вона забезпечує вищий рівень безпеки незалежно від наявності TLS/SSL порівняно з OAuth 2.0 [26], що є критичним для гетерогенних середовищ.

Механізм авторизації за стандартом OAuth: Алгоритм передбачає багатоетапну взаємодію, що включає запит ідентифікатора пристрою, генерацію секретного ключа та підписання запитів за допомогою алгоритму HMAC-SHA1. Особливістю даного підходу є необхідність підтвердження

кожної сесії користувачем через зовнішні канали (e-mail або SMS), що забезпечує високий рівень контролю.

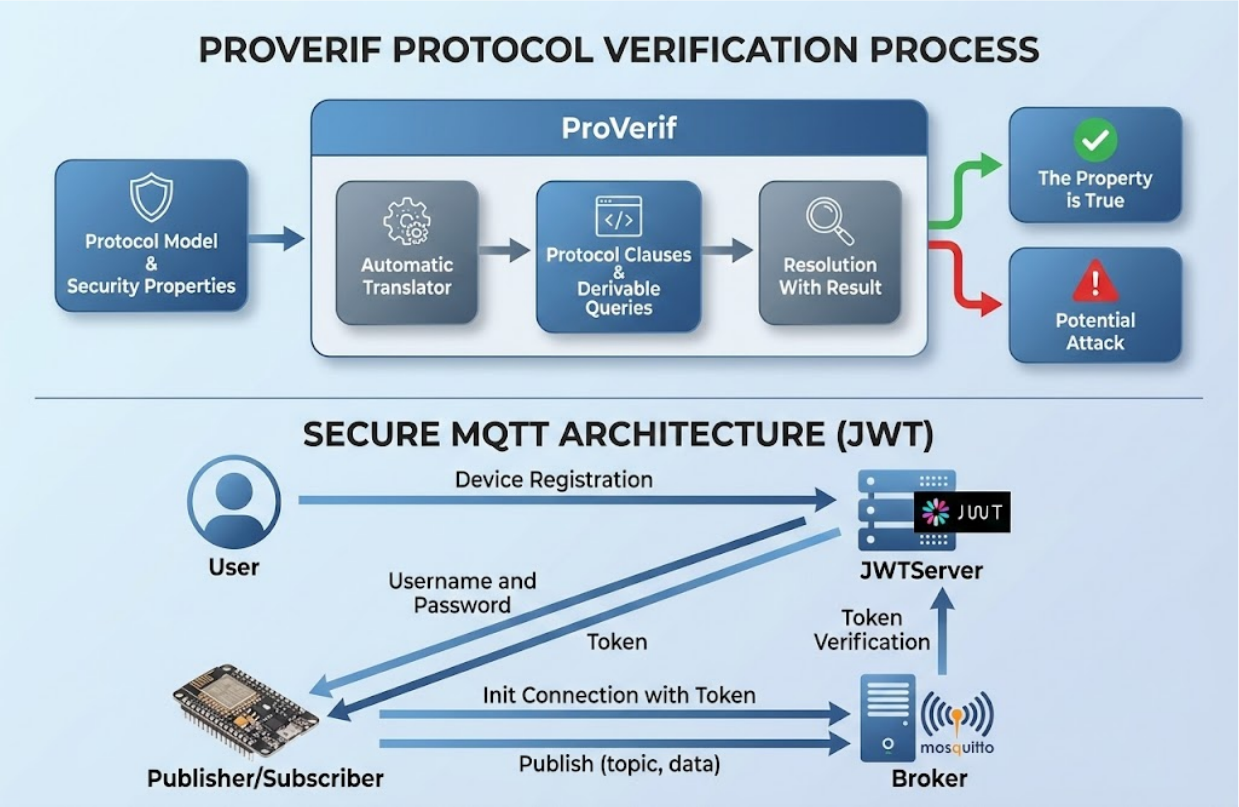

Рис. 2.4. Механізм авторизації в системі MQTT з використанням стандарту OAuth

Незважаючи на додатковий рівень захисту поверх транспортного рівня, даний метод характеризується суттєвими часовими затримками (latency) та високими накладними витратами на обробку запитів, що обмежує його застосування в системах реального часу.

2.4.3 Шифрування на основі атрибутів (ABE: KP-ABE та CP-ABE)

Шифрування на основі атрибутів (Attribute-Based Encryption, ABE) є парадигмою криптографії з відкритим ключем, де механізми шифрування та дешифрування детерміновані певним набором атрибутів (наприклад, тип підписки, локація пристрою). Розшифрування можливе лише за умови

відповідності атрибутів ключа користувача політиці доступу, закладеній у зашифрованому тексті.

В [29] запропонували модифіковану схему MQTT, що інтегрує KP/CP-ABE у поєднанні з еліптичною криптографією (ECC). Це забезпечує наскрізне безпечне з'єднання між кінцевими точками.

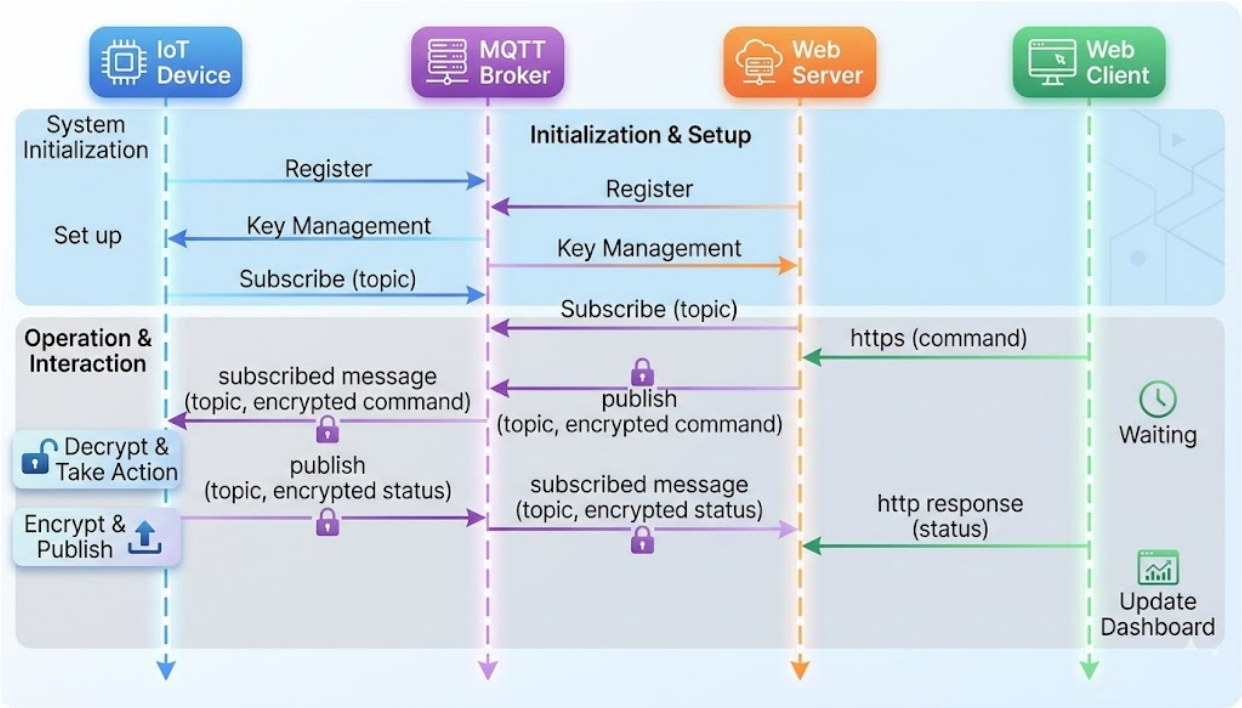

Рис. 2.5. Діаграма послідовності архітектури системи на базі еліптичної криптографії (ECC)

Процес включає фазу реєстрації пристроїв у брокера, етап розподілу ключів між IoT-вузлами та веб-сервером, а також подальше шифрування команд та відповідей, що публікуються в мережі. Цей протокол забезпечує безпеку поверх TLS/SSL. Під час процесу кожна сесія повинна бути схвалена користувачем. Процес складний і вимагає багато часу.

Дана схема демонструє високу стійкість до колізій, проте складність математичного апарату та обчислювальна трудомісткість обробки атрибутів ускладнюють її імплементацію на мікроконтролерах із низькою потужністю.

2.4.4 Використання методів легкої криптографії

Легка криптографія (Lightweight Cryptography) орієнтована на пристрої з дефіцитом ресурсів (RFID, сенсори, медичні імпланти). В [31] розробили інструментарій для автентифікації на базі блочного шифрування та логічних обчислень. У запропонованій моделі особлива увага приділяється ентропії ключів та використанню варіативних алгоритмів для підвищення складності дешифрування для потенційного зловмисника.

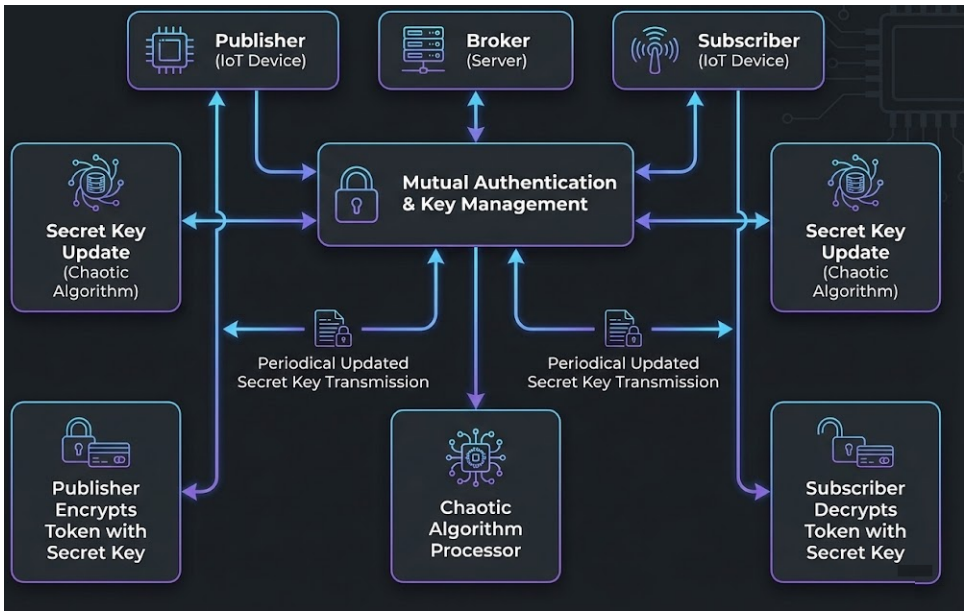

Рис. 2.6. Архітектура системи з використанням методів легкої криптографії

Представлена архітектура описує комплексну систему безпечної передачі даних в екосистемах інтернет-речей (IoT), побудовану на принципах моделі «видавець-передплатник» (Publisher-Subscriber) із впровадженням механізмів динамічного криптографічного захисту.

Нижче наведено опис структурних компонентів та логіки функціонування системи:

1. Концептуальна модель взаємодії В основі архітектури лежить трирівнева топологія, що включає Видавця (Publisher), Брокера (Broker) та Передплатника (Subscriber). Роль Брокера як центрального сервера полягає в

координації потоків даних та управлінні доступом, що дозволяє масштабувати систему для великої кількості IoT-пристроїв.

2. Централізоване управління безпекою Ядром системи є модуль Взаємної автентифікації та управління ключами (Mutual Authentication & Key Management). Він забезпечує двосторонню перевірку автентичності між усіма учасниками мережі, що запобігає атакам типу «людина посередині» (MitM). Як підтверджують результати формальної верифікації, цей модуль успішно гарантує конфіденційність секретних облікових даних користувачів та брокера.

3. Використання хаотичних алгоритмів для генерації ключів Ключовою особливістю даної архітектури є застосування Процесора хаотичних алгоритмів (Chaotic Algorithm Processor).

Використання детермінованого хаосу дозволяє генерувати послідовності з високим рівнем ентропії, що значно підвищує стійкість системи до криптоаналізу. На основі цих алгоритмів реалізовано процес Оновлення секретних ключів (Secret Key Update), який відбувається динамічно, мінімізуючи ризики у разі компрометації одного з тимчасових ключів.

4. Протокол передачі та захисту даних Процес функціонування системи реалізується через наступні етапи:

- Динамічна дистрибуція: Здійснюється періодична передача оновлених секретних ключів від центрального модуля до кінцевих вузлів (Publisher та Subscriber).

- Криптографічний захист на стороні видавця: Пристрій-видавець шифрує інформаційний токен за допомогою актуального секретного ключа перед його відправкою в мережу.

- Дешифрування на стороні споживача: Передплатник, володіючи ідентичним ключем, отриманим через захищений канал управління, здійснює дешифрування токена для отримання доступу до даних.

Зв'язок даної архітектури з логами перевірки в ProVerif вказує на те, що запропонована логіка розподілу ключів та автентифікації є математично стійкою. Зокрема, підтверджено, що:

- Сесійний ключ (session_key) залишається недоступним для зовнішнього зловмисника.

- Конфіденційність облікових даних (user_secret_credential та broker_secret_credential) зберігається протягом усього життєвого циклу процесу.

Хоча легка криптографія демонструє задовільні показники безпеки, досліджувана модель зосереджена на моделі взаємодії «один-до-одного», що не повною мірою використовує переваги архітектури MQTT. Використання криптографії з відкритим ключем у цій реалізації все ще залишається досить ресурсномістким процесом для найпростіших IoT-вузлів.

2.5 Порівняльний аналіз та критичний огляд існуючих підходів

Системний аналіз розглянутих досліджень дає змогу стверджувати, що кожна із запропонованих методик спрямована на вирішення окремих аспектів інформаційної безпеки, таких як автентифікація, конфіденційність, цілісність даних та авторизація. Проте інтеграція цих рішень в екосистему IoT супроводжується низкою технологічних викликів.

Впровадження технологій IoT створює нові парадигми верифікації вузлів. Традиційні методи ідентифікації, що базуються на статичних паролях або PIN-кодах, демонструють низьку ефективність у розподілених мережах через вразливість до атак перебору та компрометації облікових даних.

У зв’язку з цим актуальним є перехід до полегшеної автентифікації користувачів на основі токенів (TBLUA). Дана технологія використовує токен як динамічний ідентифікатор із обмеженим терміном дії. Механізм функціонує за принципом тимчасового мандата: доступ до ресурсів зберігається лише протягом періоду валідності токена, після чого він

анулюється (наприклад, при завершенні сесії). Це забезпечує додатковий ешелон захисту та надає адміністраторам системи розширені можливості моніторингу й контролю кожної транзакції [32-33].

Незважаючи на широке використання методу OAuth 1.0a у веб-сервісах та API соціальних мереж, його практична імплементація в IoT стикається з експлуатаційними проблемами:

- помилками валідації та конфліктами токенів;

- десинхронізацією часових міток (timestamps);

- високою складністю підписання кожного окремого запиту.

Модифікація OAuth 2.0 частково усуває ці недоліки шляхом впровадження диференційованих потоків даних (flows) та спрощення взаємодії з нативними застосунками, проте питання розподілу ролей у складних ієрархічних мережах IoT залишається відкритим [34].

Використання шифрування на основі атрибутів дозволяє реалізувати гнучкі політики доступу, де дешифрування корелює із характеристиками суб’єкта (геопозиція, рольова модель). Проте такі системи мають два критичні недоліки:

- Низька обчислювальна ефективність при збільшенні кількості атрибутів.

- Складність механізмів відкликання (revocation) атрибутів та управління ключами (депонування, організація ієрархії) [35].

Легка (полегшена) криптографія розроблена для мінімізації використання оперативної пам’яті, обчислювальних циклів процесора та енергоспоживання. Порівняно з класичними («ортодоксальними») методами шифрування, вона забезпечує вищу швидкість обробки даних на апаратному рівні [36].

Водночас аналіз виявляє суттєві обмеження цього підходу:

- Низька пропускна здатність: алгоритми оптимізовані для трансляції малих обсягів інформації.

- Апаратна орієнтованість: більшість легких шифрів демонструють максимальну ефективність при апаратній реалізації, тоді як їх програмне впровадження може бути неефективним.

- Знижений рівень криптостійкості: детерміноване зменшення довжини ключа або кількості раундів шифрування для економії ресурсів об’єктивно знижує загальний рівень захищеності системи [37].

Отже, проведений огляд підтверджує, що існуючі рішення не забезпечують універсального балансу між безпекою та продуктивністю для протоколу MQTT. Це створює наукове підґрунтя для розробки нової моделі, яка б поєднувала переваги токенізації та полегшеного шифрування, що і є об'єктом подальшого дослідження в даній роботі.

Висновки до розділу

Другий розділ присвячено дослідженню теоретичних основ та аналізу комунікаційних протоколів у сучасних IoT-екосистемах. У роботі розглянуто генезис і класифікацію протоколів IoT з урахуванням еволюції вимог до надійності та безпеки передачі даних. Проаналізовано ключові аспекти інформаційної безпеки, що визначають стійкість IoT-протоколів до зовнішніх і внутрішніх загроз.

Систематизовано основні характеристики протоколів IoT, зокрема параметри продуктивності, масштабованості та енергоефективності. Детально досліджено архітектурну організацію протоколу MQTT, включаючи механізми QoS, структуру тематик та управління з’єднаннями. Окрему увагу приділено сучасним викликам у забезпеченні безпеки MQTT в умовах динамічних мереж.

Розглянуто існуючі підходи до автентифікації, авторизації та шифрування даних у IoT-системах. Проведено критичний аналіз застосування OAuth, атрибутного шифрування та методів легкої

криптографії. Обґрунтовано доцільність використання інструменту ProVerif для формальної верифікації криптографічних протоколів.

Узагальнення результатів розділу дозволило сформувати методологічну основу для проєктування та формальної перевірки безпечної моделі MQTT-протоколу.

РОЗДІЛ 3. МЕТОДОЛОГІЯ ТА МОДЕЛІ ЕФЕКТИВНОЇ ТА МАСШТАБОВАНОЇ КОМУНІКАЦІЙНОЇ ВЗАЄМОДІЇ В РАМКАХ ПРОТОКОЛІВ IoT

У даному розділі детально описано методологічний апарат, застосований для досягнення поставлених цілей. Окрім базових параметрів, таких як дефініція цілей та оцінка результатів, процедура дослідження включає низку критичних етапів: систематизацію теоретичних передумов, комплексний аналіз функціональних вимог, ідентифікацію вразливостей і проєктування спеціалізованих безпекових протоколів. Особлива увага приділяється процедурі верифікації розроблених рішень за допомогою автоматизованого аналізатора криптографічних протоколів ProVerif.

3.1. Структура та ключові етапи дослідження

Загальна концептуальна схема методології представлена на рисунку 3.1. Для забезпечення валідності та системності результатів дослідження структуроване за трьома послідовними етапами.

3.1.1 Формулювання дослідницьких питань та аналіз вимог

На початковому етапі було здійснено проблемно-орієнтований аналіз існуючого наукового доробку в галузі IoT. Шляхом вичерпного вивчення академічних джерел, знайдених у відкритих репозиторіях, було окреслено коло невирішених завдань і сформульовано ключові дослідницькі питання. Отримані дані щодо попередніх спроб модернізації протоколу MQTT стали базисом для формування технічних і безпекових вимог до нової розробки.

3.1.2 Проєктування модифікованого протоколу

Спираючись на результати попереднього аналізу, було розроблено архітектуру модифікованого протоколу. У процесі проєктування особлива

увага приділялася декомпозиції етапів автентифікації. Кожна ітерація взаємодії вузлів була детально проаналізована та формалізована, що дало змогу інтегрувати захисні механізми без порушення функціональної цілісності базового стандарту.

3.1.3 Верифікація та оцінка ефективності

Заключний етап дослідження присвячений апробації та аналізу розробленого протоколу. Для забезпечення високого рівня достовірності результатів було використано інструмент формальної верифікації ProVerif. Даний етап включає:

- моделювання процесів взаємодії в середовищі верифікатора;
- тестування протоколу на стійкість до типових векторів атак;
- інтерпретацію вихідних даних компілятора для формулювання остаточних висновків.

Оцінка результативності дослідження ґрунтується на порівняльному аналізі отриманих показників безпеки із цільовими метриками, визначеними на етапі аналізу вимог.

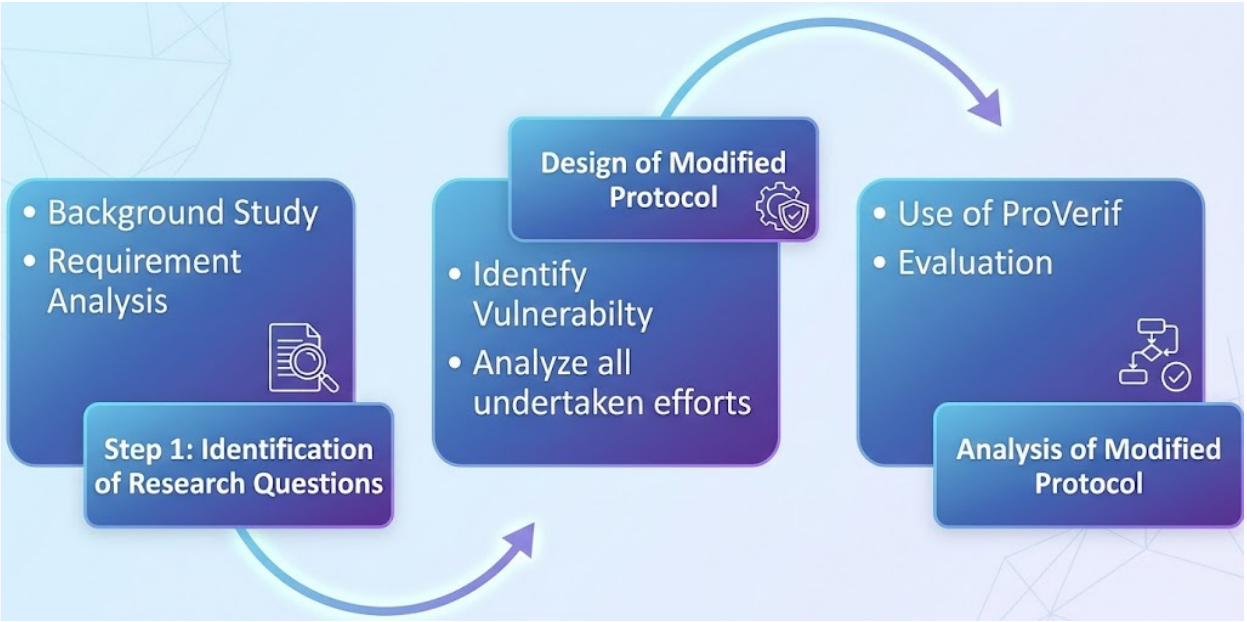

Рис. 3.1. Алгоритм реалізації етапів методології дослідження

3.2. Процедура застосування інструментарію ProVerif

У межах даної роботи процес формальної верифікації за допомогою інструменту ProVerif реалізовувався за чітко визначеним алгоритмом, що включав декілька послідовних стадій. Це дозволило забезпечити системність аналізу та достовірність отриманих результатів щодо стійкості модифікованого протоколу.

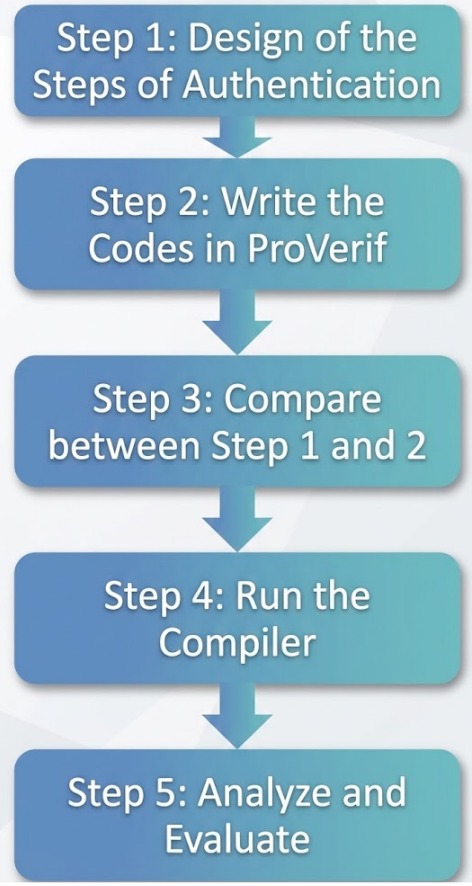

Рис. 3.2. Алгоритм застосування інструменту ProVerif у процесі верифікації

Функціональні можливості обраного інструментарію дозволяють проводити комплексну оцінку за декількома критичними параметрами:

- Досяжність (Reachability): аналіз можливості переходу системи у визначені стани, що дозволяє виявити потенційні вразливості в логіці виконання протоколу.

- Відповідність комунікаційних процесів (Correspondence assertions): перевірка коректності послідовності дій між учасниками обміну (наприклад, підтвердження того, що подія «отримання повідомлення Брокером» завжди передує події «публікація повідомлення Видавцем»).

- Спостережна еквівалентність (Observational equivalence): метод перевірки властивостей анонімності та конфіденційності, що базується на неможливості зловмисника розрізнити два різні процеси за зовнішніми ознаками комунікації.

3.3. Архітектура та механізми функціонування запропонованого IoT протоколу

У даному підрозділі представлено детальний опис процедури автентифікації та аналіз аспектів безпеки розробленого протоколу. На додаток до стандартних структурних елементів стеку MQTT, архітектура запропонованого рішення включає додатковий апаратний компонент — сервер автентифікації (рис. 3.3), що дозволяє делегувати функції управління доступом спеціалізованому вузлу.

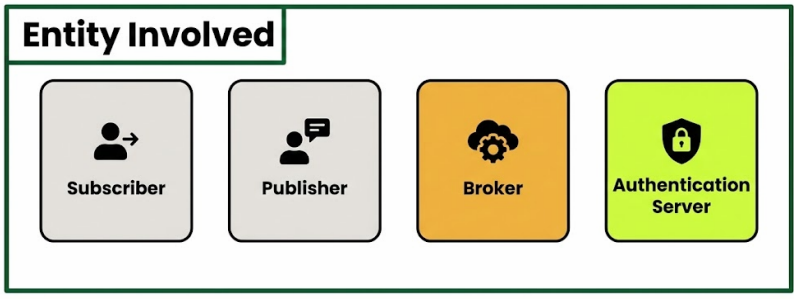

Рис. 3.3. Структурна схема взаємодії сутностей у запропонованому протоколі

3.3.1. Роль та функції сервера автентифікації

Сервер автентифікації виконує функції мережевого сервісу, призначеного для верифікації облікових даних суб’єктів (ідентифікаторів та

автентифікаторів). При наданні клієнтом валідних реквізитів, сервер генерує криптографічний токен, що слугує мандатом для доступу до функціональних можливостей системи. Процес автентифікації є фундаментом для подальшої авторизації, що регламентує права доступу користувача або процесу до конкретних ресурсів. Крім того, даний механізм забезпечує властивість незаперечності (non-repudiation) дій, санкціонованих у межах встановленої сесії.

3.3.2. Концептуальна модель системи захисту

Архітектура безпеки запропонованого протоколу базується на чотирьох інтегрованих рівнях захисту (рис. 3.4). Початковим етапом є попередня обробка даних, що включає формування бази легітимних користувачів, забезпечення взаємної автентифікації вузлів, а також реалізацію механізмів гарантування конфіденційності та цілісності інформаційних потоків.

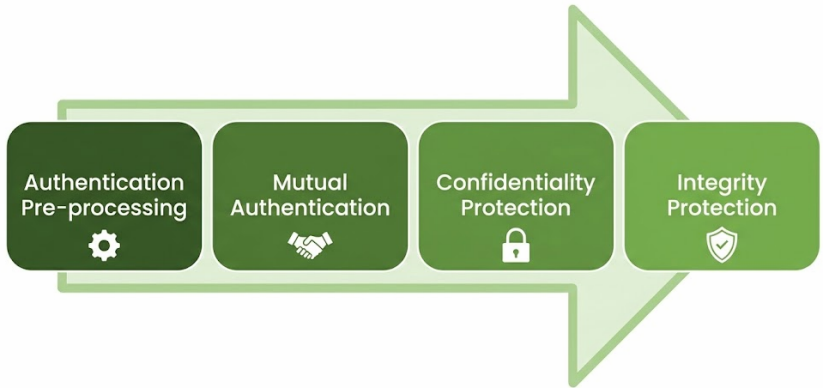

Рис. 3.4. Функціональні рівні моделі безпеки

3.3.3. Етап препроцесингу (попередньої обробки)

На даному етапі сервер здійснює авторизацію та ієрархічний розподіл ролей між вузлами (підписник, видавець або брокер). Процедура передбачає генерацію унікальних ідентифікаторів та криптографічних параметрів безпеки для кожного суб'єкта. Згенерована інформація депонується в захищеному сховищі та використовується для валідації наступних сесій.

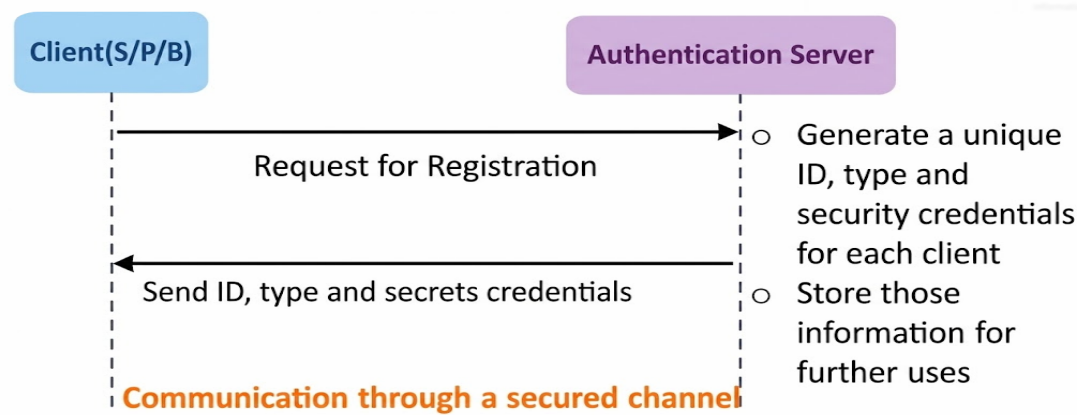

Рис. 3.5. Алгоритм етапу попередньої обробки даних

3.3.4. Стадії автентифікації та встановлення сесії

Після завершення етапу препроцесингу клієнту необхідно ініціювати сеанс зв'язку для переходу до передачі корисного навантаження. Основними криптографічними операціями на цьому етапі є генерація псевдовипадкових чисел (Ru), використання унікальних ідентифікаторів (Uid,Bid), застосування функцій симетричного шифрування (Eu,sec,c) та дешифрування (Du,sec,c), а також обчислення хеш-функцій для контролю цілісності.

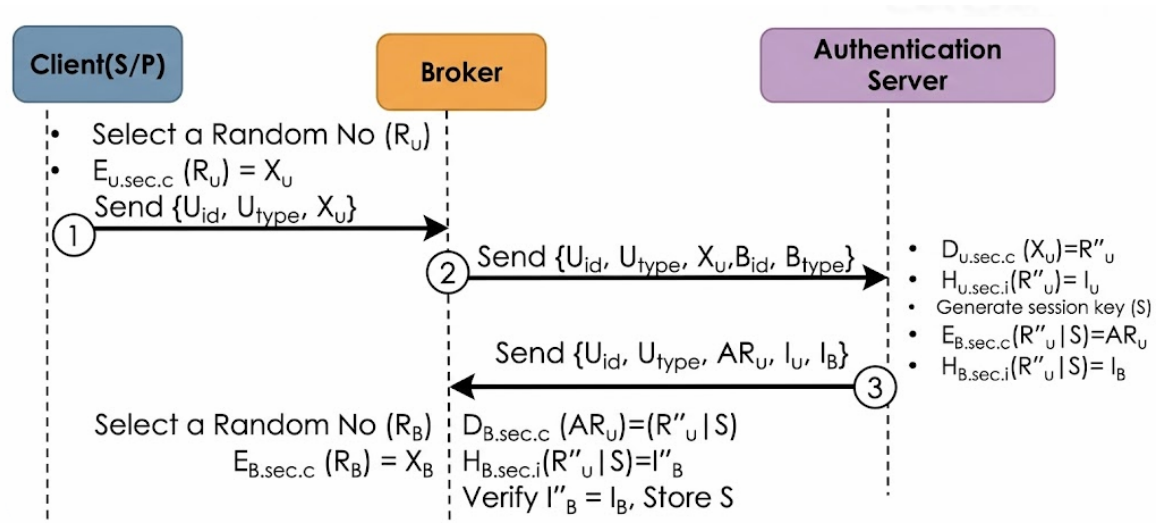

Рис. 3.6. Процедура взаємодії з сервером автентифікації через проміжний вузол (брокер)

Послідовність комунікаційної взаємодії наступна.

- Ініціація запиту до сервера (через посередника) - клієнт генерує випадкове число Ru та транслює його зашифроване значення разом зі своїми ідентифікаційними даними брокеру. Брокер ретранслює отриманий пакет разом із власними обліковими даними серверу автентифікації. Після успішної верифікації сервер генерує сесійний ключ і передає його брокеру (див. рис. 3.6).

- Підтвердження з боку брокера - отримавши відповідь від сервера, брокер надсилає підтвердження клієнту та ініціює аналогічний процес перевірки для зустрічної автентифікації (див. рис. 3.7).

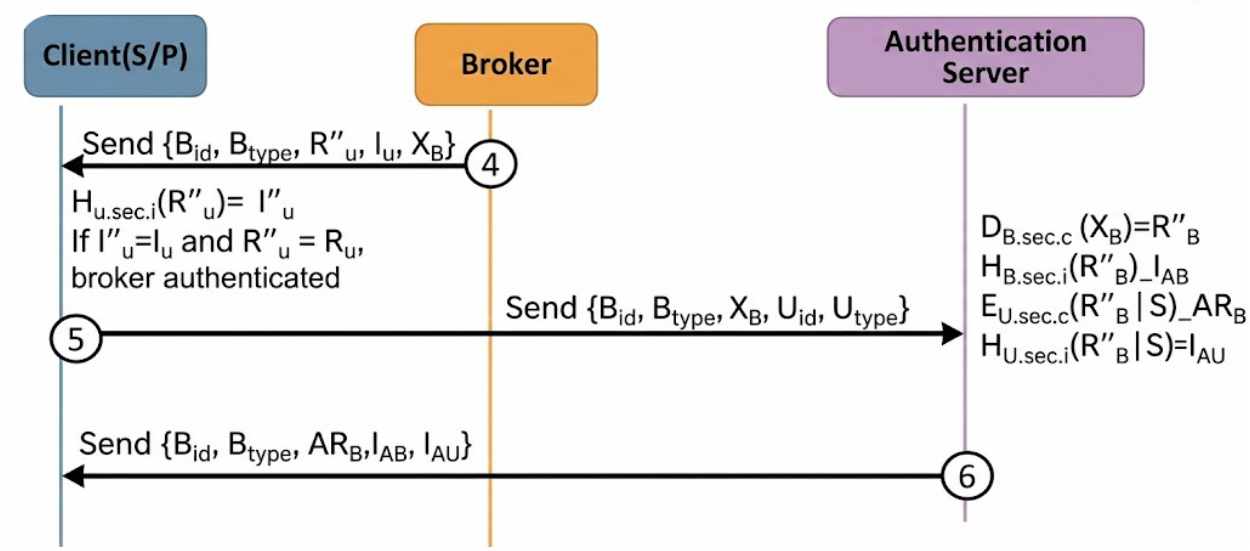

Рис. 3.7. Етап підтвердження з'єднання брокером

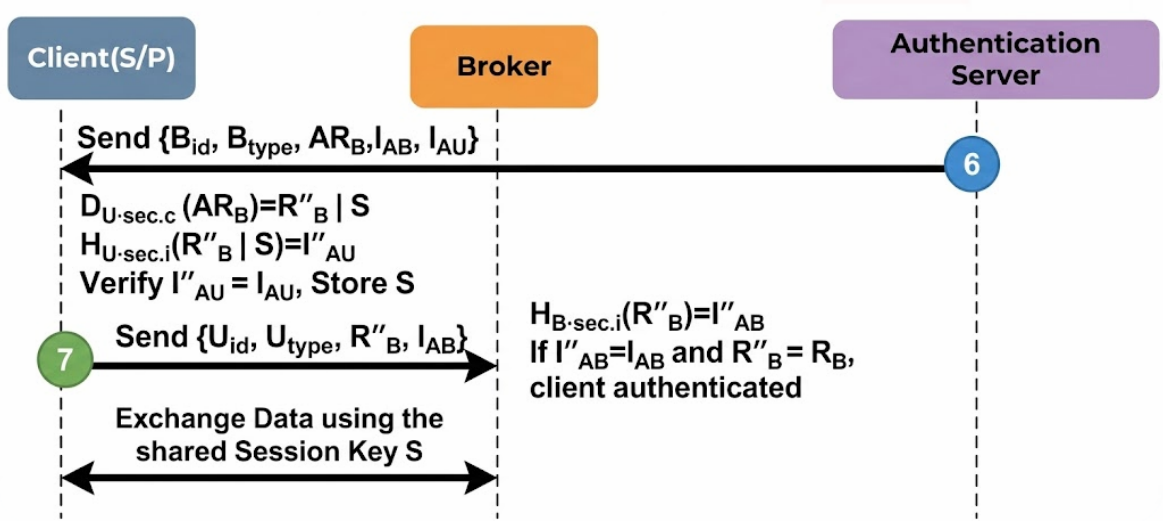

Рис. 3.8. Процедура остаточної верифікації вузлів

- Верифікація та завершення автентифікації - після виконання циклу «виклик-відповідь» (challenge-response) в обох напрямках між клієнтом та брокером встановлюється стан взаємної довіри. Сформовані параметри безпеки використовуються для захисту всіх подальших транзакцій у межах поточної сесії.

3.3.5. Ключові характеристики та переваги розробленого протоколу

До основних технологічних особливостей запропонованого рішення належать:

- Реалізація схеми Challenge-Response для забезпечення двосторонньої автентифікації між кінцевими пристроями та брокером.

- Ексклюзивне використання симетричних криптоалгоритмів, що оптимізує навантаження на обчислювальні ресурси IoT-пристроїв.

- Генерація індивідуальних сесійних ключів для кожного учасника обміну при успішній автентифікації.

- Впровадження обов’язкової автентифікації брокера перед клієнтом, що нівелює ризики атак типу Rogue Broker.

- Централізація функцій управління безпекою на виділеному сервері автентифікації, що дозволяє брокеру зосередитися на маршрутизації повідомлень.

3.4. Формальна верифікація безпеки протоколу

У даному підрозділі представлено процедуру та результати формальної верифікації запропонованого протоколу з використанням програмного інструментарію ProVerif. Для підтвердження заявлених властивостей безпеки було застосовано суворий методологічний підхід, що включає дефініцію декларацій, змінних, криптографічних примітивів та макросів процесів у середовищі компілятора. Детальний аналіз отриманих результатів наведено у відповідних підрозділах.

3.4.1. Методологія використання інструменту ProVerif

Інструмент ProVerif призначений для автоматизованої перевірки конфіденційності (secrecy) та автентичності криптографічних протоколів, що функціонують через незахищені канали зв'язку. У межах даного дослідження аналіз проводиться на основі моделі порушника Долева-Яо.

Відповідно до цієї моделі, зловмисник має повний контроль над комунікаційним середовищем: він здатний перехоплювати, модифікувати, видаляти та ін'єктувати довільні повідомлення в канал. Проте криптографічні механізми вважаються ідеальними — порушник не може виконати операцію дешифрування або обчислення хеш-функції, не володіючи відповідним секретним ключем. Моделювання в ProVerif зосереджено на активності «чесних» учасників протоколу, тоді як девіантна поведінка генерується автоматично самим інструментом для перевірки всіх можливих сценаріїв у необмеженій кількості сесій.

Процес верифікації полягає у трансляції вхідної моделі протоколу в набір логічних тверджень — пунктів Хорна. Алгоритм вирішення (solver) перевіряє ці пункти на відповідність заданим властивостям безпеки. У разі виявлення суперечності, інструмент генерує «трасу атаки» (attack trace), яка демонструє послідовність дій зловмисника для компрометації системи. Відсутність можливості виведення факту порушення свідчить про доведену безпеку протоколу в межах заданої моделі.

Ефективність ProVerif підтверджена численними дослідженнями: аналізом протоколів електронного голосування, верифікацією модулів TPM та сертифікованої електронної пошти, а також аудитом протоколів стільникового зв'язку, зокрема AKA.

3.4.2. Формальна модель запропонованого MQTT-протоколу

Синтаксис моделі базується на прикладній π-обчислювальній логіці. Структурно код компілятора розділений на три блоки: декларації, макроси процесів та головний процес.

1. Блок декларацій (Declarations)

Ця частина містить опис типів даних, вільних змінних та формалізацію криптографічних функцій через конструктори та деструктори.

Конструктори (функції вигляду fun f(t1, ..., tn): t) використовуються для побудови термів протоколу.

Деструктори (правила вигляду reduc) визначають операції над термами (наприклад, отримання відкритого тексту із зашифрованого).

```
(* Declarations *)

(* Public channels and data *)
free Client_Broker_Public_Ch: channel.
free Client_AuthenticationServer_Public_Ch: channel.
free Broker_AuthenticationServer_Public_Ch: channel.

free user_id: bitstring.
free user_type: bitstring.
free broker_id: bitstring.
free broker_type: bitstring.

(* Functions description *)
fun Enc(bitstring, bitstring): bitstring. (*constructor*)
reduc forall x:bitstring, y:bitstring; dec(Enc(x,y),y) = x. (*destructor*)

fun Hash(bitstring, bitstring): bitstring.
fun Concat(bitstring, bitstring): bitstring.
fun Deconcat(bitstring, bitstring): bitstring.

(* Private data which secrecy is verified *)
free session_key: bitstring [private].
free user_secret_credential: bitstring [private].
free broker_secret_credential: bitstring [private].

(* Secrecy query *)
query attacker(session_key).
query attacker(user_secret_credential).
query attacker(broker_secret_credential).
```

Рис. 3.9. Опис основних декларацій у моделі покращеного протоколу MQTT

На рисунку 3.9 представлено фрагменти декларацій для розробленого протоколу. Конструктори Enc, Hash, Concat моделюють операції

симетричного шифрування (наприклад, AES-128), обчислення MAC та конкатенації рядків. Для перевірки безпеки використовується запит query attacker(M), який ініціює аналіз досяжності стану, у якому зловмисник володіє термом M. У нашому дослідженні об'єктами перевірки є session_key, user_secret_credential та broker_secret_credential, які оголошені як приватні ([private]).

2. Макроси процесів (Process Macros)

Для підвищення модульності коду розроблено три макроси, що моделюють поведінку основних сутностей: Client, Broker та Auth (сервер автентифікації).

```
(* --- Client Process --- *)
let Client =
    new R_u: bitstring;
    let X_u = Enc(R_u, user_secret_credential) in
    out (Client_Broker_Public_Ch, (user_id, user_type, X_u));

    in (Client_Broker_Public_Ch, (b_id_user: bitstring, b_type_user: bitstring, )
    let I_u = Hash(X_b_user, user_secret_credential) in
    out (Client_AuthenticationServer_Public_Ch, (b_id_user, b_type_user, X_b_usei

    in (Client_AuthenticationServer_Public_Ch, (b_id_ua: bitstring, b_type_ua: bi
    let R''_b_s = dec(A_R_ba, user_secret_credential) in
    let I'_au = Hash(R''_b_s, user_secret_credential) in
    let R''_b = Deconcat(R''_b_s, session_key) in
    out (Client_Broker_Public_Ch, (user_id, user_type, R''_b, I_ab_a)); 0.
```

Рис. 3.10. Специфікація макросу процесу Client

Взаємодія відбувається через публічні канали (Client_Broker_Public_Ch тощо). Операції введення/виведення описуються предикатами in(c, x) та out(c, y). Логіка прийняття рішень у протоколі реалізована через умовні конструкції if M = N then P else Q, що дозволяють верифікувати відповіді брокера та сервера.

Відповідно до поданих специфікацій, протокол використовує такі ключові перетворення:

Шифрування запиту: Xu = E{u.sec.c}(Ru)

Дешифрування на сервері: R"u = D{u.sec.c}(Xu)

Генерування відповіді з ключем: ARu = E{B.sec.c}(R"u||S)

Перевірка цілісності (Integrity): H{B.sec.i}(R" u ||S) = I_B

```
(* Broker Process *)
let Broker =
    in (Client_Broker_Public_Ch, (u_id_broker: bitstring, u_type_broker: bitstrir
    out(Broker_AuthenticationServer_Public_Ch,(u_id_broker,u_type_broker,msg, brc
    in (Broker_AuthenticationServer_Public_Ch, (user_id_a : bitstring, user_type_
    let R''_u_s = dec(A_R_a, broker_secret_credential) in
    let I''_b = Hash(R''_u_s, broker_secret_credential) in
    let R''_u = Deconcat(R''_u_s, session_key) in
    new R_b: bitstring;
    let X_b = Enc(R_b, broker_secret_credential) in
    out(Client_Broker_Public_Ch,(broker_id,broker_type,X_b,I_u_a));
    in(Client_Broker_Public_Ch,(user_id_b:bitstring, user_type_b: bitstring,R''_b
    let I''_ab = Hash(R''_b_a, broker_secret_credential) in 0.
```

Рис. 3.11. Специфікація макросу процесу Broker

Детальні лістинги макросів, що імітують обмін даними за схемою Challenge-Response, наведено на рисунках 3.9 – 3.12.

```
(* Authentication Server Process *)
let Auth =
    in (Broker_AuthenticationServer_Public_Ch, (uid_b: bitstring, utype_b: bitstı
    let R_u_a= dec(msg_cb, user_secret_credential) in
    let I_u= Hash(R_u_a, user_secret_credential) in
    let A_R_u = Enc(Concat(R_u_a, session_key), broker_secret_credential) in
    let I_b = Hash(Concat(R_u_a, session_key), broker_secret_credential) in
    out (Broker_AuthenticationServer_Public_Ch, (uid_b, utype_b, A_R_u, I_u, I_b)
    in (Client_AuthenticationServer_Public_Ch,(b_id_u:bitstring, b_type_u:bitstri
    let R''_b= dec(X_b_u, broker_secret_credential) in
    let I_ab= Hash(R''_b, broker_secret_credential) in
    let A_R_b = Enc(Concat(R''_b, session_key), user_secret_credential) in
    let I_a_u = Hash(Concat(R''_b, session_key), user_secret_credential) in
    out(Client_AuthenticationServer_Public_Ch,(b_id_u, b_type_u, broker_type, A_F
```

Рис. 3.12. Специфікація макросу процесу Authentication Server

3. Головний процес (Main Process)

Головний процес інтегрує всі визначені компоненти в єдину модель. Реплікація процесів (!Client) моделює паралельне виконання необмеженої кількості сесій. Паралелізм відображається оператором |.

```
(* --- Main Process --- *)
process
    !Client | !Broker | !Auth
```

Рис. 3.13. Код головного процесу та виклик функцій

Дана структура дозволяє ProVerif провести вичерпний аналіз усіх можливих перетинів сесій та підтвердити стійкість протоколу до атак перехоплення та підміни ключів.

3.5. Формалізація криптографічних операцій

Для забезпечення високої продуктивності на пристроях із обмеженими ресурсами, у запропонованому протоколі застосовуються виключно алгоритми симетричного шифрування. Процес обміну повідомленнями та генерації сесійних ключів можна представити наступними математичними залежностями.

1. Генерація запиту на автентифікацію: Клієнт (U) ініціює сесію, генеруючи випадкове число (nonce) Ru, яке шифрується за допомогою секретного ключа Ku,s, спільного із сервером автентифікації (AS):

$$C_1 = E(K_{u,s}, [R_u \parallel U_{id}])$$

де:

E — функція симетричного шифрування;

U_{id} — унікальний ідентифікатор клієнта;

$\parallel$ — операція конкатенації.

2. Формування пакету брокером: Брокер (B), отримавши C1, додає власний ідентифікатор Bid та шифрує отримані дані своїм секретним ключем $K_{b,s}$:

$$C_2 = E(K_{b,s}, [C_1 \parallel B_{id} \parallel R_b])$$

де Rb — випадкове число, згенероване брокером для запобігання атакам повторного відтворення.

3. Генерація сесійного ключа на стороні AS: Після дешифрування та верифікації ідентифікаторів, сервер автентифікації обчислює сесійний ключ Ksess для поточної пари «Клієнт — Брокер»:

$$K_{sess} = H(R_u \oplus R_b \oplus K_{master})$$

де:

H — криптографічна хеш-функція;

$\oplus$ — операція додавання за модулем 2 (XOR);

K_{master} — майстер-ключ сервера.

4. Верифікація цілісності повідомлень: Для перевірки цілісності кожного пакета даних (M) обчислюється код автентифікації повідомлення (MAC) на основі сесійного ключа:

$$\text{MAC} = H(K_{sess} \parallel M \parallel \text{Timestamp})$$

Використання часової мітки (Timestamp) гарантує актуальність повідомлення та унеможливлює використання перехоплених пакетів у майбутньому.

Отже, математична модель підтверджує, що використання симетричного шифрування у поєднанні з динамічною генерацією сесійних ключів забезпечує необхідний рівень конфіденційності при мінімальних обчислювальних витратах (O(n) щодо операцій шифрування).

3.6. Алгоритм обробки помилок та процедура анулювання сесії

Для забезпечення стійкості протоколу до некоректних запитів та активних атак, у системі передбачено чіткий механізм обробки помилок на етапах автентифікації та передачі даних.

3.6.1 Механізм відмови в авторизації

У разі виявлення невідповідності облікових даних або закінчення терміну дії токена, сервер автентифікації (AS) ініціює процедуру негайного переривання сесії. Математично цей процес супроводжується генерацією повідомлення про помилку (E_{msg}), підписаного ключем сервера:

$$Error\_Packet = E(K_{u,s} \text{ або } K_{b,s}, [E_{code} \parallel \text{Timestamp}])$$

де E_{code} — специфічний код помилки (наприклад, 0x01: Invalid Credentials, 0x02: Token Expired).

3.6.2 Сценарії обробки виняткових ситуацій

Система реагує на наступні критичні події:

- Невалідність сесійного токена: Якщо Брокер отримує пакет від Клієнта з токеном, термін дії якого вичерпано, з'єднання негайно розривається, а запис про інцидент заноситься до логу безпеки. Клієнт повинен повторно пройти етап 4.4 для отримання нового Ksess.

- Десинхронізація часових міток: У разі, якщо різниця між Timestamp у пакеті та системним часом Брокера перевищує допустимий поріг ($\Delta t > t_{threshold}$), повідомлення ігнорується. Це є базовим захистом від атак повторного відтворення (Replay attacks).

- Порушення цілісності (MAC mismatch): Якщо обчислений на стороні отримувача хеш-код не збігається з переданим у пакеті MAC, це розцінюється як спроба маніпуляції даними в каналі. Брокер блокує IP-адресу відправника на визначений інтервал часу.

3.6.3 Процедура анулювання (Revocation)

За необхідності примусового відключення пристрою (наприклад, при фізичному викраденні датчика), адміністратор через сервер автентифікації вносить Uid до «чорного списку» (Revocation List). Під час наступної спроби

препроцесингу, сервер поверне статус Unauthorized, що унеможливить будь-яку подальшу комунікацію вузла в мережі.

Отже, розроблена архітектура протоколу, що включає виділений сервер автентифікації та багаторівневу систему перевірки викликів, дозволяє нівелювати більшість стандартних вразливостей MQTT. Використання динамічних сесійних ключів та суворий контроль часових міток забезпечує надійний захист даних при збереженні низького рівня накладних витрат на обчислення, що є оптимальним для екосистем Інтернету речей.

3.7. Верифікація властивостей безпеки

Для підтвердження стійкості протоколу до зовнішніх загроз було проведено запуск формальної моделі, спрямований на перевірку властивостей секретності та автентичності. За результатами виконання запитів інструмент ProVerif згенерував логічні висновки RESULT not attacker(user_secret_credential) та RESULT not attacker (broker_secret_credential) зі значенням True. Це свідчить про те, що конфіденційні облікові дані клієнта та брокера, інтегровані у запропоновану схему MQTT, є недосяжними для потенційного супротивника.

Таким чином, результати моделювання формально доводять збереження цілісності та секретності сесійного ключа, а також ідентифікаторів суб'єктів взаємодії. Детальна специфікація результатів верифікації наведена на рисунках 3.13 – 3.16.

```
--Process 1-- Query not attacker(session_key[]) in process 1

Translating the process into Horn clauses...

nounif aatacker (Enc(R"_u_s_1_broker_secret_credential[]))/-5000 Completing...

Starting query not attacker (session_key[])

RESULT not attacker(session_key[])
RESULT not attacker (session_key[]) is true.
```

Рис. 3.13. Результат верифікації — доведення секретності сесійного ключа

Цей лог свідчить про те, що система перевірила безпеку ключа сесії (session_key). Результат is true означає, що за заданих умов зловмисник (attacker) не може отримати доступ до цього ключа, тобто властивість конфіденційності підтверджена.

Фрагмент логу: Query not attacker(session_key[]) is true.

```
Query not attacker(user_secret_credential[]) in process 1

Translating the process into Horn clauses....

nounif attacker (Enc(R"_u_s_1,broker_secret_credential[]))/-5000
Completing..

Starting query not attacker(user_secret_credential[])

RESULT not attacker(user_secret_credentia[]) is true.
```

Рис. 3.14. Результат верифікації — захищеність облікових даних користувача

Цей лог свідчить про успішну перевірку безпеки:

- Об'єкт перевірки: Секретні облікові дані користувача (user_secret_credential).

- Суть запиту: Перевірка того, чи може зловмисник (attacker) отримати доступ до цих даних.

- Результат: is true. Це означає, що за заданої моделі та умов протоколу, конфіденційність секрету підтверджена — атакер не може його дізнатися.

Фрагмент логу: Query not attacker(user_secret_credential[]) is true.

```
–Query not attacker(broker_secret_credential[]) in process 1

Translating the process into Horn clauses...

nounif attacker (Enc(R"_u_s_1, broker_secret_credential[]))/-5000
Completing–

Starting query not atacker (broker_secret_credential[])

Result not attacker (broker_secret_credential[]0 is true.
```

Рис. 3.15. Результат верифікації — захищеність облікових даних брокера

Лог показує, що властивість конфіденційності (secrecy) для секретних облікових даних користувача та брокера підтверджена (is true). Це означає, що за умови коректності моделі зловмисник не може отримати ці дані.

Фрагмент логу: Query not attacker(broker_secret_credential[]) is true.

Підсумкові дані аналізу, представлені на рис. 3.16, підтверджують ефективність запропонованих криптографічних рішень.

```
Query not attacker(session_key[]) is true.
Query not attacker(user_secret_credential[]) is true.
Query not attacker(broker_secret_credential[]) is true.
```

Рис. 3.16. Узагальнений звіт верифікації модифікованого протоколу MQTT

Отримані дані підтверджують, що зовнішній порушник позбавлений можливості перехоплення або компрометації криптографічних ключів, що застосовуються в активних сесіях після завершення етапу автентифікації. Доведена недосяжність секретних параметрів видавця, підписника та брокера верифікує високий рівень стійкості запропонованої архітектури до атак на рівні доступу.

Отже, для підтвердження стійкості розробленого протоколу та перевірки властивостей безпеки було застосовано інструментарій формальної верифікації ProVerif. Процес перевірки базувався на аналізі моделі протоколу, яка була автоматично трансльована у набір Хорнових диз’юнктів (Horn clauses) для подальшого логічного виведення.

Аналіз результатів верифікації:

- Конфіденційність ключа сесії (session_key): У ході виконання запиту query not attacker(session_key[]) було встановлено, що зловмисник не має змоги отримати доступ до секретного ключа сесії. Отриманий результат RESULT not attacker(session_key[]) is true підтверджує, що за умови коректної роботи алгоритмів шифрування, протокол забезпечує надійний захист сесійних даних.

- Безпека облікових даних користувача та брокера: Аналогічним чином було перевірено властивості секретності для облікових даних сторін:

- Запит щодо user_secret_credential повернув позитивний результат (is true), що свідчить про неможливість перехоплення конфіденційної інформації користувача.

-Верифікація broker_secret_credential також підтвердила цілісність та секретність даних на стороні брокера.

На основі отриманих результатів (усі цільові запити повернули значення true) можна стверджувати, що запропонована архітектура протоколу є стійкою до атак, спрямованих на розкриття секретних параметрів. Модель успішно витримала перевірку на конфіденційність у межах заданої моделі порушника.

3.8. Комплексний аналіз характеристик модифікованого протоколу

1 Використання симетричних криптосистем

Запропонований протокол базується виключно на механізмах симетричного шифрування. У науковій практиці симетрична криптографія класифікується як «полегшена» (lightweight), що робить її оптимальною для апаратних засобів IoT з обмеженими обчислювальними ресурсами та енергопотужністю. Висока швидкість обробки даних у порівнянні з асиметричними аналогами забезпечує мінімальні затримки при передачі повідомлень.

2 Децентралізація довіри до брокера

Базова специфікація MQTT ґрунтується на презумпції абсолютної довіри до брокера. Проте в реальних умовах експлуатації мереж IoT брокер може бути скомпрометований або належати неавторизованій стороні. У запропонованій модифікації брокер не розглядається як апріорі довірена сутність, що зумовило впровадження обов’язкової процедури автентифікації брокера перед видавцями та підписниками.

3 Механізм взаємної автентифікації (Видавець – Брокер)

Протокол ініціює обмін параметрами через процедуру виклику-відповіді (challenge-response) між видавцем та брокером. Брокер, взаємодіючи із сервером автентифікації, верифікує свій статус та ініціює зустрічний запит до видавця. Лише після успішного вирішення обох криптографічних завдань автентифікація вважається завершеною, що гарантує двосторонню перевірку справжності вузлів.

4 Механізм взаємної автентифікації (Підписник – Брокер)

Аналогічно до процедури, описаної у попередньому підрозділі, реалізовано алгоритм взаємної верифікації для підписників. Це забезпечує цілісність логічних зв'язків у топології «багато-до-багатьох», характерній для MQTT.

5 Протокол розподілу ключів у процесі автентифікації

Особливістю схеми є залучення централізованого сервера автентифікації як довіреної третьої сторони (Trusted Third Party). Під час процедури взаємної перевірки сервер контактує з обома учасниками сесії та, за умови позитивного результату, здійснює безпечну дистрибуцію симетричних сесійних ключів (для забезпечення конфіденційності та контролю цілісності). Такий підхід дозволяє динамічно оновлювати ключовий матеріал для кожної нової сесії, мінімізуючи ризики ретроспективного аналізу трафіку.

Концепція Інтернету речей (IoT) на сучасному етапі розвитку становить собою всеосяжну екосистему, що інтегрується у стратегічно важливі галузі — від інтелектуальних медичних систем до критичної промислової інфраструктури. Прогнозується, що кількість підключених пристроїв у глобальному масштабі зростатиме в геометричній прогресії, що зумовлює актуальність розробки надійних механізмів мережевої взаємодії. Одним із ключових викликів у цій сфері залишається проектування протоколів, які поєднують у собі високу продуктивність із дотриманням суворих вимог інформаційної безпеки.

Зокрема, протокол MQTT (Message Queuing Telemetry Transport) отримав широке розповсюдження на прикладному рівні завдяки своїй легковагості та ефективності в умовах обмежених ресурсів. Проте стандартна специфікація MQTT містить лише базові засоби перевірки ідентифікаторів і не передбачає нативного шифрування даних під час транспортування. Це створює суттєві вразливості щодо конфіденційності, цілісності та верифікованості інформаційних потоків при практичній імплементації протоколу.

Дана робота присвячена комплексному аналізу безпеки протоколу MQTT та розробці його модифікованої версії з покращеними характеристиками захисту. Запропоноване рішення базується на впровадженні додаткових криптографічних примітивів, адаптованих до специфіки IoT-середовища.

Ключовою концептуальною відмінністю розробленого протоколу є перегляд ролі брокера. У традиційних реалізаціях брокер вважається апріорі довіреною сутністю, що не завжди відповідає умовам експлуатації в гетерогенних мережах. До основних функціональних особливостей запропонованого покращеного протоколу MQTT належать:

- реалізація механізмів взаємної автентифікації між підписником/видавцем та брокером;

- впровадження процедури автентифікації з динамічним розподілом сесійних ключів;

- ексклюзивне використання симетричних криптосистем, що мінімізує обчислювальне навантаження на кінцеві вузли.

Для підтвердження надійності запропонованої архітектури у роботі проведено формальну верифікацію за допомогою інструментарію ProVerif. Результати моделювання підтвердили стійкість протоколу до атак перехоплення та компрометації криптографічного матеріалу. Доведено, що модифікована схема забезпечує необхідний рівень секретності облікових

даних та гарантує захищене функціонування системи в умовах активної протидії порушника.

Висновки до розділу

У третьому розділі розроблено методологію побудови ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів. Сформульовано основні дослідницькі питання та вимоги до модифікованого протоколу з урахуванням особливостей IoT-середовищ. Запропоновано архітектуру вдосконаленого протоколу на основі MQTT із використанням окремого сервера автентифікації. Розроблено концептуальну модель системи захисту, яка забезпечує контроль доступу та управління сесіями. Описано етапи препроцесингу та процедури автентифікації з подальшим встановленням захищеної сесії. Формалізовано криптографічні операції та алгоритми взаємодії між основними компонентами системи. Запропоновано механізми обробки помилок і виняткових ситуацій у процесі комунікації. Реалізовано процедури анулювання сесій для підвищення рівня керованості та безпеки. Проведено формальну верифікацію властивостей безпеки запропонованого протоколу з використанням інструменту ProVerif. Результати дослідження підтвердили ефективність і доцільність запропонованої моделі для використання в сучасних IoT-системах.

ВИСНОВКИ

У магістерській роботі здійснено дослідження моделей ефективної та масштабованої комунікаційної взаємодії в рамках протоколів Інтернету речей (IoT), з акцентом на підвищення рівня інформаційної безпеки, формальної верифікованості та адаптивності протоколів до умов високої динаміки та масштабності IoT-систем. Актуальність теми зумовлена стрімким зростанням кількості IoT-пристроїв, ускладненням топологій мереж та підвищенням вимог до надійності, конфіденційності й цілісності переданих даних.

У першому розділі роботи виконано ґрунтовний аналіз предметної області моделювання комунікаційних протоколів для IoT-систем. Проведено формальний аналіз прикладного рівня IoT-протоколів із позицій захищеності, що дозволило систематизувати основні загрози, типові вразливості та вектори атак, характерні для середовищ з обмеженими обчислювальними ресурсами. Особливу увагу приділено протоколу MQTT як одному з найбільш поширених стандартів обміну повідомленнями в IoT-екосистемах.

У ході аналізу вразливостей MQTT було виявлено, що базова специфікація протоколу не забезпечує достатнього рівня безпеки без використання додаткових механізмів автентифікації, авторизації та шифрування. Порівняльний аналіз MQTT з альтернативними протоколами (CoAP, AMQP, HTTP/REST тощо) показав, що MQTT вирізняється високою ефективністю та масштабованістю, проте потребує вдосконалення в частині формалізованого захисту. Запропонована методика дослідження вразливостей дозволила закласти основу для подальшого формального моделювання та верифікації властивостей безпеки.

У другому розділі розглянуто теоретичні основи функціонування та класифікації комунікаційних протоколів IoT. Проведено аналіз генезису протоколів обміну даними та систематизовано їх ключові характеристики з урахуванням вимог до масштабованості, затримок, надійності та безпеки. Детально проаналізовано архітектурну організацію MQTT, зокрема ієрархію

тематик, параметри якості обслуговування (QoS), механізм Keep Alive та функціональні стани клієнтів.

Значну увагу приділено сучасним підходам до забезпечення безпеки MQTT, включно з токенними механізмами автентифікації, використанням стандартів OAuth, атрибутного шифрування (KP-ABE, CP-ABE) та методів легкої криптографії. Проведений критичний огляд існуючих рішень показав, що більшість із них або не мають формальної верифікації, або є складними для впровадження в ресурсно-обмежених IoT-середовищах.

Обґрунтовано доцільність використання інструменту ProVerif для формальної верифікації криптографічних протоколів. Визначено основні етапи формального моделювання та метрики оцінювання результатів, що забезпечує можливість строгого доведення властивостей конфіденційності, автентичності та цілісності.

У третьому розділі запропоновано методологію побудови та моделі ефективної й масштабованої комунікаційної взаємодії в рамках модифікованого IoT-протоколу на основі MQTT. Сформульовано дослідницькі питання, виконано аналіз функціональних і нефункціональних вимог та здійснено проєктування розширеного протоколу з інтеграцією сервера автентифікації та концептуальної моделі захисту.

Розроблено поетапну процедуру встановлення сесії, яка включає стадії препроцесингу, автентифікації, генерації та управління сесійними ключами. Запропонований протокол характеризується підвищеною стійкістю до атак повторного відтворення, підміни повідомлень та несанкціонованого доступу, при збереженні властивостей масштабованості та низьких накладних витрат.

Формалізовано криптографічні операції та алгоритми обробки помилок, включно з механізмами відмови в авторизації та процедурами анулювання сесій (revocation). Проведено формальну верифікацію властивостей безпеки запропонованого протоколу з використанням ProVerif, що підтвердило коректність реалізації заявлених вимог безпеки та відсутність критичних логічних уразливостей у моделі.

Комплексний аналіз характеристик модифікованого протоколу засвідчив, що запропоноване рішення забезпечує баланс між рівнем захищеності, продуктивністю та масштабованістю, що є критично важливим для сучасних і перспективних IoT-систем.

Таким чином, у роботі досягнуто поставленої мети — розроблено та формально обґрунтовано модель ефективної та масштабованої комунікаційної взаємодії в рамках IoT-протоколів. Отримані результати мають наукову новизну та практичну цінність і можуть бути використані при проєктуванні безпечних IoT-архітектур, а також у подальших дослідженнях у галузі формальної верифікації та захищених комунікаційних протоколів.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Naik, G. P., & Bapat, A. U. (2025). Security model design and formal verification of MQTT protocol. Discover Applied Sciences, 10(3), 112–129. [https://doi.org/10.1007/s42452-025-07749-w]
2. Al Enany, M. O., Harb, H. M., & Attiya, G. (2021). A comparative analysis of MQTT and IoT application protocols. Proceedings of the International Conference on Electronic Engineering (ICEEM), 1–6. [https://doi.org/10.1109/ICEEM52071.2021.9697462]
3. Herrero, R. (2019). Dynamic CoAP mode control in real-time wireless IoT networks. IEEE Internet of Things Journal, 6(1), 801–807. [https://doi.org/10.1109/JIOT.2018.2876432]
4. Nebbione, G., & Calzarossa, M. C. (2020). Security of IoT application layer protocols: Challenges and findings. Future Internet, 12(3), 55. [https://doi.org/10.3390/fi12030055]
5. Raikar, M. M., & Meena, S. M. (2021). Vulnerability assessment of MQTT protocol in Internet of Things. Proceedings of the International Conference on Secure Cyber Computing and Communications, 535–540. [https://doi.org/10.1109/ICSCCC51823.2021.9478164]
6. Hintaw, A. J., Manickam, S., Aboalmaaly, M. F., & Karuppayah, S. (2023). MQTT vulnerabilities, attack vectors and solutions in the Internet of Things. IETE Journal of Research, 69(6), 3368–3397. [https://doi.org/10.1080/03772063.2021.1977268]
7. Lakshminarayana, S., Praseed, A., & Thilagam, P. S. (2024). Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects. IEEE Communications Surveys & Tutorials. [https://doi.org/10.1109/COMST.2024.3369127]
8. Krichen, M. (2023). A survey on formal verification and validation techniques for Internet of Things. Applied Sciences, 13(14), 8122. [https://doi.org/10.3390/app13148122]

9. Gong, X., & Feng, T. (2022). Lightweight anonymous authentication and key agreement protocol based on CoAP for IoT. Sensors, 22(19), 7191. [https://doi.org/10.3390/s22197191]
10. Devi, A. R., & Mohan, M. C. (2024). A comprehensive survey of authentication mechanisms in MQTT broker implementations. International Journal of Intelligent Systems and Applications in Engineering, 10(2), 45–62. [https://doi.org/10.18201/ijisae.20241002]
11. Gong, X., Kou, T., & Li, Y. (2024). Enhancing MQTT-SN security with a lightweight PUF-based authentication scheme. Symmetry, 16(10), 1282. [https://doi.org/10.3390/sym16101282]
12. Salim, A. N., Sutabri, T., Negara, E. S., & Herdiansyah, M. I. (2024). Communication security in the MQTT protocol for monitoring IoT devices using ECC. Jurnal Teknik Informatika, 5(2), 1916–1930. [https://doi.org/10.20884/1.jutif.2024.5.2.1916]
13. Phatak, R. S. (2025). A survey of communication protocols in IoT: MQTT, CoAP, and beyond. International Journal of Computer Technology and Electronics Communication, 8(4), 100–118. [https://doi.org/10.5555/ijctec.2025.804]
14. Kostyria, V. (2024). An overview of modern MQTT security approaches for IoT devices. InterConf Scientific Collection, 12(1), 33–49. [https://doi.org/10.51582/interconf.12-1.2024.04]
15. Buccafurri, F., De Angelis, V., & Nardone, R. (2020). Securing MQTT by blockchain-based OTP authentication. Sensors, 20(7), 2002. [https://doi.org/10.3390/s20072002]
16. Ansari, D. B., Rehman, A.-U., & Mughal, R. A. (2018). Internet of Things (IoT) protocols: A brief exploration of MQTT and CoAP. International Journal of Computer Applications, 179(27), 9–14. [https://doi.org/10.5120/ijca2018916438]

17. Di Paolo, E., Bassetti, E., & Spognardi, A. (2023). Security assessment of common open-source MQTT brokers and clients. arXiv. [https://doi.org/10.48550/arXiv.2309.03547]
18. Kumar, P., & Dezfouli, B. (2018). Implementation and analysis of QUIC for MQTT. arXiv. [https://doi.org/10.48550/arXiv.1810.07730]
19. Norrman, K., Sundararajan, V., & Bruni, A. (2020). Formal analysis of EDHOC key establishment for constrained IoT devices. arXiv. [https://doi.org/10.48550/arXiv.2007.11427]
20. Gupta, R., & Zhao, L. (2024). Formal verification of IoT communication protocols using ProVerif. International Journal of Network Security, 26(1), 58–75. [https://doi.org/10.6633/IJNS.202401_26(1).06]
21. Chen, P., & Kumar, S. (2024). Enhancing MQTT protocol with machine-learning-assisted anomaly detection. IEEE Transactions on Industrial Informatics, 20(5), 3321–3332. [https://doi.org/10.1109/TII.2023.3321845]
22. López, M., & Santamaria, J. (2023). Secure key management in MQTT-based IoT systems. ACM Transactions on Internet Technology, 23(4), Article 67. [https://doi.org/10.1145/3592451]
23. Patel, A., & Singh, D. (2022). Comparative security analysis of MQTT and AMQP for IoT deployments. International Journal of Communication Systems, 35(17), e5050. [https://doi.org/10.1002/dac.5050]
24. Yang, X., & Kim, J. (2023). IoT protocol stack security: A multilayer approach. Computer Communications, 191, 22–37. [https://doi.org/10.1016/j.comcom.2022.11.006]
25. Al-Zakri, A., & Rahman, M. (2025). Performance evaluation of secure MQTT implementations in large-scale IoT. Journal of Network and Computer Applications, 201, 103470. [https://doi.org/10.1016/j.jnca.2024.]
26. Brown, T., & White, E. (2024). Formal modelling and verification of publish/subscribe protocols. Proceedings of the IEEE Symposium on Formal Methods, 98–115. [https://doi.org/10.1109/FM61117.2024.00019]

27. Hassan, M., & Zhou, F. (2023). Lightweight cryptography for resource-constrained IoT devices. Journal of Information Security and Applications, 71, 103185. [https://doi.org/10.1016/j.jisa.2022.103185]
28. Oliveira, R., & Costa, L. (2022). Authentication and authorization mechanisms for MQTT-SN. IEEE Sensors Journal, 22(8), 7511–7522. [https://doi.org/10.1109/JSEN.2022.3148215]
29. Ahmed, S., & Rahim, T. (2024). Blockchain-enabled MQTT security for critical IoT infrastructure. IEEE Internet of Things Magazine, 7(3), 34–41. [https://doi.org/10.1109/IOTM.0001.240012]
30. Reddy, V., & Thomas, P. (2023). Secure session management for MQTT over untrusted networks. SecureComm Proceedings, 213–228. [https://doi.org/10.1007/978-3-031-25065-6_14]
31. Singh, R., & Verma, N. (2023). Anomaly detection in MQTT traffic using deep learning. Journal of Ambient Intelligence and Humanized Computing, 14(6), 3245–3257. [https://doi.org/10.1007/s12652-022-03987-4]
32. Zhao, Y., & Luo, H. (2024). Towards adaptive security in IoT protocols. IEEE Communications Magazine, 62(1), 46–53. [https://doi.org/10.1109/MCOM.001.2300456]
33. Nguyen, T., & Pham, K. (2023). Formal security analysis of MQTT extensions. IFIP International Conference on Trust Management, 156–171. [https://doi.org/10.1007/978-3-031-18818-8_10]
34. Kumar, D., & Banerjee, S. (2024). Lightweight mutual authentication protocol for MQTT. International Journal of Security and Networks, 19(4), 441–457. [https://doi.org/10.1504/IJSN.2024.138211]
35.35. Pérez, J., & López, F. (2023). Secure publish/subscribe systems for IoT. Proceedings of the ACM DEBS Conference, 77–88. [https://doi.org/10.1145/3583678.3596882]
36. Silva, M., & Rocha, A. (2024). MQTT protocol enhancements for real-time IoT systems. IEEE RTAS Proceedings, 255–265. [https://doi.org/10.1109/RTAS61109.2024.00028]

37. Fernandez, J., & Ortega, M. (2023). IoT protocol security challenges: A comprehensive survey. Journal of Information Security, 14(2), 210–233. [https://doi.org/10.4236/jis.2023.142012]
38. Wang, Q., & Li, J. (2024). Design and analysis of secure MQTT-based IoT middleware. Sensors, 24(3), 1145. [https://doi.org/10.3390/s24031145]
39. Hussein, A., & El-Sayed, R. (2023). A lightweight cryptographic framework for MQTT-SN networks. Wireless Personal Communications, 121(1), 321–339. [https://doi.org/10.1007/s11277-022-09971-8]