

A Novel Lightweight MQTT Security Scheme for the Internet of Medical Things

SIJIA TIAN

Doctor of Philosophy

University of York

Department of Computer Science

July 2023

Abstract

Healthcare 4.0 has revolutionised next-generation healthcare services by leveraging the transformation potential of the Internet of Things (IoT). A lightweight and reliable Message Queuing Telemetry Transport (MQTT) is often used to share patient-related information between medical devices and remote locations in heterogeneous and resource-constrained IoT devices. However, lightweight MQTT experiences notable security challenges due to resource constraints, exposure to cyber threats, and the sensitivity of healthcare data. Therefore, this research analyses the performance of lightweight MQTT security strategies and explores the design of novel security solutions tailored for healthcare 4.0 that address critical security attacks while maintaining optimal performance and minimal resource consumption vital for medical applications.

This research begins with evaluating the performance of several lightweight symmetric key cryptography algorithms under two attack scenarios, MitM and DoS. This approach comprehensively analyses various cryptography algorithms under diverse network conditions, resulting in broad validation techniques with diverse metrics and scenarios. We conclude that the FBC and PRESENT algorithms achieve superior results in varying network scenarios.

Subsequently, an improved ciphertext-policy attribute-based encryption (ICP-ABE) is proposed by incorporating a lightweight symmetric encryption scheme using the PRESENT algorithm. This work separates attributes based on blind keys generated using the PRESENT algorithm and significantly reduces the computation complexity associated with the existing CP-ABE scheme.

Finally, an optimised CP-ABE-based lightweight scheme is proposed using attribute-specific blind-key generation with a Fast-PRESENT algorithm to enhance the security and performance in the Internet of Medical Things environment. This scheme reduces the key tracing possibilities of OCP-ABE and the risks related to traceability and attacks that impact MQTT performance. The proposed solutions are analysed and validated for performance using formal, tool-based, and simulation-based analysis to demonstrate superiority. Ultimately, the proposed contributions shape the evolving landscape of Healthcare 4.0 to enable critical life medical services, particularly enhancing patient care and monitoring.

Acknowledgements

I would first like to thank my supervisor, Dr Vasileios Vasilakis. I was able to complete my Ph.D. thesis under his supervision and guidance. When I started this journey, he served as a point person, patiently guiding me in researching IoT security and providing many helpful and innovative suggestions. When I encountered problems, he used his experience and expertise to broaden my horizon so that I could cross one step after another and reach new heights. During my doctoral studies, I had to face the grief of several elders in my family passing away one after another, and I was unable to mourn them in the first time. Thank you to my supervisor for his understanding and help, which eased my inner torment. I also thank my internal assessor Dr. Solomon Amos for his insightful questions during all the assessment meetings. Thanks to my second supervisor Dr. Siamak Shahandashti and my tap member Solomon Amos for the help they gave me during the preparation of my thesis.

I want to thank my family for their love and forgiveness! Thank you to my parents who are far away from home. You are my strongest support, and you support my study and life with actions and love. When I was lost, you gave me selfless support and encouragement, and gave me the courage and strength to move forward. I miss and mourn my deceased grandma and grandpa. I miss you very much. In addition, I would like to thank my friends including Lin Min and Ren Xiaonan for their companionship and help over the years.

Finally, I thank all the professionals and scholars who reviewed my dissertation and participated in the Viva during my busy schedule. I wish my supervisors, friends and family happiness and health forever!

Sijia Tian,
York,
December 2022.

Declaration

I declare that this thesis is a presentation of original work and that I am the sole author. This work has not previously been presented for an award at this, or any other, University. I want to emphasise that all sources are acknowledged as References.

In addition, I ensure that any publications arising from the thesis are acknowledged in this section.

Chapter 5 and 6 is based on the following paper.

Sijia Tian and Vassilios G. Vassilakis. Performance Evaluation of Lightweight Symmetric Cryptographic Algorithms for MQTT Protocol Under Denial-of-Service and Man-in-Middle Attacks. *Under review*.

Chapter 7 is based on the following conference paper.

Sijia Tian and Vassilios G. Vassilakis. A lightweight authentication and privacy preservation scheme for MQTT. Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing (SAC2023), pages 1289–1292, Tallinn, Estonia, 2023. ACM. <https://doi.org/10.1145/3555776.3577817>

Chapter 7 is also based on the following journal article.

Sijia Tian and Vassilios G. Vassilakis. On the Efficiency of a Lightweight Authentication and Privacy Preservation Scheme for MQTT. MDPI Electronics, 2023; 12(14):3085. <https://doi.org/10.3390/electronics12143085>.

Chapter 8 is based on the following journal article.

Sijia Tian, Vassilios G. Vassilakis and Wei Jie. MQTT Lightweight Authentication Scheme for IoMT using Optimised CP-ABE and Indirect Revocation. *Under review*.

Contents

List of Tables	ix
List of Figures	xi
List of Acronyms	xiv
Glossary	xvii
1 Introduction	1
1.1 Internet of Things (IoT) and Healthcare 4.0	1
1.1.1 Healthcare 4.0 and Internet of Medical Things (IoMT)	2
1.1.2 Essentials of MQTT	3
1.2 Technical Challenges for MQTT	5
1.2.1 Security Vulnerability in MQTT Protocol	6
1.3 Security Solutions for MQTT	7
1.4 Significance and Motivation	9
1.4.1 Shortcomings in the Current Security Solutions	11
1.5 Problem Statement	12
1.5.1 Aim of the Thesis	13
1.5.2 Research Questions	13
1.5.3 Objectives of the Thesis	14
1.6 Major Contributions	14
1.7 Thesis Organisation	16
2 Background	19
2.1 IoT Background	19
2.1.1 Layered Structure of IoT	20
2.1.2 Emerging IoT Applications	23
2.1.3 Background of Healthcare IoT	28
2.2 Preliminaries of MQTT Protocol	30
2.2.1 Technical Aspects of MQTT Protocol over IoT	31
2.2.2 Technical Challenges of MQTT Pub/Sub Implementation	51
2.3 Security of MQTT	53
2.3.1 MQTT Security Requirements	54
2.3.2 MQTT Security Threats	57
2.3.3 Impact of Security Attacks on MQTT Performance	67

2.4	Chapter Summary	74
3	Literature Survey	75
3.1	Cryptography based MQTT Security Countermeasures	75
3.1.1	Symmetric Key Cryptography based Solutions	77
3.1.2	Asymmetric Key Cryptography based Solutions	83
3.2	Review of Lightweight Cryptography Algorithms	87
3.2.1	SSL/TLS Encryption based Solutions	88
3.2.2	Authentication-based Solutions	90
3.2.3	Confidentiality-based Solutions	93
3.2.4	Hybrid Solutions	94
3.3	Comparative Analysis of MQTT Security Solutions	97
3.4	Current Lightweight Security Solutions	98
3.4.1	Advantages of Lightweight Cryptography	103
3.4.2	Shortcomings of the Lightweight cryptography	104
3.5	General Research Gaps in MQTT Security and Discussion	105
3.5.1	Summary	107
3.6	Related Work on Lightweight and Attribute-Based Encryption Schemes for MQTT and IoMT	108
3.6.1	Applications of lightweight and ABE in Healthcare and IoMT Environments	108
3.6.2	Performance Evaluations and Limitations of ABE Schemes in Constrained Environments	111
3.6.3	Comparisons Between ABE-Based Lightweight Schemes and Traditional Access Control Mechanisms	112
3.7	Research Gaps in Existing ABE-Based MQTT Security Schemes in IoMT .	114
3.8	Chapter Summary	116
4	Methodology	117
4.1	Need for Novel Lightweight Security Solutions for MQTT	117
4.1.1	Significance of Lightweight Cryptography Analysis	118
4.1.2	Bridging Gaps with Novel Cryptography Solutions	119
4.1.3	Trade-off Between Security and Computation Cost	120
4.2	Use Case for Healthcare and Medical IoT	121
4.2.1	Healthcare 4.0 Overview	121
4.2.2	Key Components of MQTT Enabled Healthcare 4.0	122
4.2.3	Communication Flow of Healthcare 4.0	123
4.2.4	Security for Healthcare Data	123
4.2.5	Security Challenges in Healthcare 4.0	124
4.3	Methodology of Lightweight Security Solutions for MQTT	125
4.3.1	Analysis of Lightweight Symmetric Cryptographic Algorithms for MQTT against Confidentiality-related Attacks	125
4.3.2	Lightweight Security Scheme for Topic Encryption and Attribute-based Authentication	127
4.3.3	Optimisation of Attribute-Based Lightweight Authentication Scheme	127
4.4	Validation and Evaluation Methods and Tools	128
4.4.1	Validation and Evaluation Methods for MQTT Security	128
4.4.2	Tools-based Validation and Evaluation	129

4.4.3	Need for Simulation-based Validation and Evaluation for MQTT . .	130
4.5	Chapter Summary	136
5	Performance Validation for MQTT in Healthcare 4.0	137
5.1	Introduction for MQTT Lightweight Symmetric Key Cryptography	137
5.2	Research Gaps and Motivation	140
5.3	Preliminaries of MQTT and Security Vulnerabilities	141
5.3.1	System Architecture of MQTT Enabled Smart Healthcare 4.0 . . .	142
5.3.2	DoS Attack and Impact on MQTT	144
5.3.3	MitM Attack and Impact on MQTT	145
5.3.4	Lightweight Cryptography based Security for MQTT	145
5.3.5	Symmetric Cryptographic Algorithms	146
5.4	Analysis	149
5.4.1	Attack Analysis	149
5.4.2	Computational Cost Estimation	151
5.5	MQTT Protocol Verification	152
5.5.1	Formal Security Analysis	153
5.5.2	Tool Based Security Analysis	158
5.6	Chapter Summary	165
6	Performance Evaluation for MQTT in Healthcare 4.0	166
6.1	Performance Evaluation using Cooja Simulator	166
6.1.1	Network Scenario Creation	167
6.1.2	Attack Scenario Creation	168
6.2	Simulation Results	170
6.2.1	MQTT without Attack Scenario	170
6.2.2	Performance Evaluation of Different Nodes under MitM or DoS At- tacks Scenarios	176
6.2.3	Performance Evaluation of Various Data Rates under DoS or MitM Attacks Scenarios	182
6.2.4	Performance Evaluation of Different Nodes under DoS and MitM Simultaneous Attack Scenarios	189
6.3	Results Discussion	195
6.3.1	Balance of Performance and Security	200
6.3.2	Results Summary	203
6.4	Chapter Summary	205
7	Lightweight Security Scheme for Topic Encryption and Attribute-based Authentication	206
7.1	Introduction for Improved CP-ABE	207
7.1.1	Motivation to Propose ICP-ABE	209
7.2	The Proposed ICP-ABE	210
7.2.1	Overview of ICP-ABE	210
7.2.2	System Model	215
7.3	Attributes-based Access Control and Topic-based Encryption Using PRESENT	216
7.3.1	Attributes-based Access Control	216
7.3.2	Security Model	222
7.3.3	Topic-based Encryption and Decryption using PRESENT Keys . .	223

7.3.4	Secret Key Sharing Using the PRESENT Algorithm	224
7.3.5	Attribute-Based Signature Scheme and Self Key Revocation Scheme using Attributes	226
7.4	Security and Computational Cost Analysis	226
7.4.1	Security Analysis	227
7.4.2	Computational Complexity Analysis	228
7.4.3	Time Complexity Analysis	231
7.5	Evaluation and Validation	232
7.5.1	Provable Security	232
7.5.2	Simulation based Evaluation	234
7.6	Simulation Results and Discussion	235
7.6.1	Simulation Results	236
7.6.2	Discussion	236
7.6.3	Summary	246
7.7	Chapter Summary	247
8	OCP-ABE: An MQTT-Based Lightweight Authentication Scheme for the Internet of Medical Things	248
8.1	Introduction for OCP-ABE	250
8.2	Problem Formulation and Proposed OCP-ABE Scheme	251
8.2.1	System Entities	254
8.2.2	Access Structure and Attributes	256
8.2.3	System and Attacker Model	257
8.2.4	Overview of the OCP-ABE	260
8.3	Optimised CP-ABE Scheme using Attribute Specific Blind Key Generation and MQTT Flag	265
8.3.1	Fast-PRESENT Algorithm-Based Encryption	266
8.3.2	Lightweight Indirect Revocation Scheme	271
8.3.3	Secure Data Publishing/Subscribing	272
8.4	Attack Analysis and Security Strength Evaluation	272
8.4.1	Attack Analysis	273
8.4.2	Avalanche Effect	275
8.4.3	Correlation Coefficient	276
8.4.4	Semi Equivalent Key Test	277
8.5	Performance Evaluation of OCP-ABE	278
8.5.1	Performance Evaluation of OCP-ABE MQTT without Attacks . . .	279
8.5.2	Comparison Results of ICP-ABE and OCP-ABE	284
8.5.3	Results Summary	289
8.6	Chapter Summary	291
9	Conclusions and Future Work	292
9.1	Contributions	292
9.1.1	Contributions	292
9.2	Future Directions	294
	References	297

List of Tables

2.1	Comparative Analysis of Emerging IoT Applications.	27
2.2	MQTT Characteristics.	33
2.3	Comparison of MQTT Protocol Versions.	35
2.4	Comparison of Various MQTT brokers.	40
2.5	Comparison of QoS Levels Offered by MQTT.	45
2.6	Packet Types in MQTT.	47
2.7	Table of ReturnCode Values.	50
2.8	Comparative Analysis of ACL, RBAC, TBAC, ABAC, and PBAC.	57
2.9	Comparative Analysis of Different MQTT Security Attacks.	66
2.10	Impact of Security Attacks on MQTT Performance.	74
3.1	Comparison Table of Related Works.	97
3.2	Existing MQTT Security Algorithms with Advantages and Limitations. . .	100
4.1	Analysis of Different Lightweight Cryptography Solutions.	118
4.2	Notations Used for Computational Complexity Estimation.	129
5.1	Comparison of AES, DES, PRESENT, LED, and FBC Algorithms.	148
5.2	Attack Prevention Capability of Symmetric Encryption Algorithms.	150
5.3	Computational Cost Analysis of Cryptography Algorithms.	152
5.4	Lists the Symbols Used in Security Analysis.	153
6.1	Simulation Model.	168
6.2	Results and Discussion of MQTT.	196
6.3	FBC-based MQTT for Remote Patient Monitoring System.	199
6.4	Remote Patient Monitoring System Simulation Parameters.	201
7.1	Lists the Symbols.	217
7.2	Access Policies for ICP-ABE and Optimised CP-ABE.. . . .	222
7.3	List of Notations for Complexity.	229
7.4	List of Notations for PRESENT in ICP-ABE.	229
7.5	Attack Prevention Ability of Various Algorithms Against Different Attacks. .	234
7.6	Simulation Model.	235
7.7	Results of Existing Works under Normal and Different Attack Scenarios - 1.	236
7.8	Results of Existing Works under Normal and Different Attack Scenarios - 2.	237
7.9	Results of Existing Works under Normal and Different Attack Scenarios - 3.	238

8.1	List of used Notations.	255
8.2	MF Flag in TCP Message.	266
8.3	Avalanche Effect.	276
8.4	Correlation Coefficient Test.	277
8.5	Semi Equivalent Key Test.	278
8.6	Simulation Model.	279

List of Figures

1.1	Classification of MQTT Security Attacks	6
1.2	Reported MQTT Vulnerabilities	10
2.1	Core Components of IoT	20
2.2	Layered IoT Architecture	21
2.3	Emerging IoT Applications	24
2.4	MQTT Pub/Sub model for IoT Healthcare Scenario	32
2.5	The Pub/Sub Architecture	37
2.6	MQTT Communication Scenario for Healthcare	43
2.7	MQTT CONNECT & CONNACK	46
2.8	CONNECT	47
2.9	CONNACK	48
2.10	Impact of Security Attacks on Latency and Lives	68
2.11	Impact of Security Attacks on Throughput	69
2.12	Impact of Security Attacks on Reliability	71
2.13	Impact of Security Attacks on Energy Consumption	72
2.14	Impact of Security Attacks on Overhead	73
3.1	Cryptography based Countermeasures	77
3.2	Classification of Lightweight Cryptography Solutions	88
4.1	Lightweight Security and MQTT-Enabled Healthcare Monitoring System .	122
4.2	Healthcare IoT Security with Attack Scenario	124
4.3	Proposed Methodology	126
5.1	MQTT Communication for Smart Healthcare Scenario	144
5.2	MQTT-based AES Algorithm for Smart Healthcare Scenario	146
5.3	Lightweight Symmetric Key Cryptography	155
5.4	Attack Pattern Result of DoS Attack	161
5.5	Attack Pattern Result of MitM Attack	162
5.6	JCrypTool Output Results of PRESENT-based MQTT	164
6.1	Simulation Scenario of Remote Patient Monitoring System	169
6.2	Attack Scenario	169
6.3	Throughput and PDR Under Number of Nodes	170
6.4	Delay and Execution Time for Different Number of Nodes Scenario	171

6.5	Energy Consumption and CPU Energy Consumption for Various Numbers of Nodes	172
6.6	Throughput and Packet Delivery Ratio under Various Data Rates	173
6.7	Delay and Execution Time under Various Data Rates	174
6.8	Energy Consumption and CPU Energy Consumption Results for Various Data Rates	175
6.9	Throughput Results for Different Nodes and Attacker Scenarios	177
6.10	PDR Results for Different Nodes and Attacker Scenarios	178
6.11	Delay Results for Different Nodes and Attacker Scenario	178
6.12	Execution Time Results for Different Nodes and Attacker Scenario	179
6.13	Average Energy Consumption Results for Different Nodes and Attacker Scenario	180
6.14	CPU Energy Consumption Results for Different Nodes and Attacker Scenario	181
6.15	Throughput and Packet Delivery Ratio with DoS Attack under Various Data Rate Scenarios	182
6.16	Delay and Execution Time with DoS Attack under Various Data Rate Scenarios	184
6.17	Average Energy Consumption and CPU Energy Consumption with DoS Attack under Various Data Rate Scenarios	185
6.18	Throughput and Packet Delivery Ratio with MitM Attack under Various Data Rate Scenarios	186
6.19	Delay and Execution Time with MitM Attack under Various Data Rate Scenarios	187
6.20	Energy Consumption and CPU Energy Consumption with MitM Attack under Various Data Rate Scenarios	188
6.21	Throughput and PDR for Different Number of Nodes Scenario	189
6.22	Delay and Execution Time for Different Number of Nodes Scenario	190
6.23	Energy Consumption and CPU Energy Consumption for Different Number of Nodes Scenario	191
6.24	Throughput and Packet Delivery Ratio under Various Data Rates	192
6.25	Delay and Execution Time under Various Data Rates	193
6.26	Energy Consumption and CPU Energy Consumption Results for Various Data Rates	194
6.27	MQTT-based Communication for Remote Patient Monitoring System . . .	200
6.28	Remote Patient Monitoring System Using Lightweight MQTT	202
6.29	Encryption and Decryption Time	202
7.1	Architecture for MQTT Secure Data Transmission	212
7.2	Communication Process of ICP-ABE	213
7.3	Block Diagram for Proposed Lightweight Authentication Scheme in MQTT	214
7.4	System Architecture of ICP-ABE	215
7.5	Example of Access Tree Structure	220
7.6	Encryption and Decryption of ICP-ABE	221
7.7	Functions in PRESENT Algorithm	225
7.8	Algorithms Vs Throughput	237
7.9	Algorithms Vs Packet Delivery Ratio	239
7.10	Algorithms Vs Delay	240

7.11 Algorithms Vs Execution Time	241
7.12 Algorithms Vs CPU Energy Consumption	242
7.13 Algorithms Vs Average Energy Consumption	243
7.14 Algorithms Vs Computation Overhead	244
7.15 Algorithms Vs Strength Evaluation Criteria	245
7.16 Algorithms Vs Communication Overhead	245
8.1 Smart IoMT Application Scenario	253
8.2 MQTT Attacks	259
8.3 Block Diagram of Proposed Methodology	261
8.4 Key Sharing and Revocation Operations of OCP-ABE Scheme	263
8.5 Process of OCP-ABE	264
8.6 Operations in Fast-PRESENT Algorithm	267
8.7 Process in Fast-PRESENT Algorithm	268
8.8 Parallel SubByte Operation in Fast n PRESENT	269
8.9 Hash Function to Show Avalanche Effect	275
8.10 Number of Nodes Vs. Throughput	280
8.11 Number of Nodes Vs. PDR	280
8.12 Number of Nodes Vs. Energy Consumption	281
8.13 Number of Nodes Vs. CPU Energy Consumption	282
8.14 Number of Nodes Vs. Delay	283
8.15 Number of Nodes Vs. Execution Time	283
8.16 Performance Results of ICP-ABE and OCP-ABE for Without Attack Scenario	285
8.17 Performance Results of ICP-ABE and OCP-ABE for With Attack Scenario	287
8.18 Performance Results of ICP-ABE and OCP-ABE for DoS Attack Scenario	288
8.19 Performance Results of ICP-ABE and OCP-ABE for MitM Attack Scenario	290

List of Acronyms

3DES-RC4 Triple Data encryption standard-Rivest Cipher 4

ABAC Attribute-Based Access Control

ABE Attribute-Based Encryption

ACL Access Control List

AE Avalanche Effect

AEAD Authenticated Encryption with Associated Data

AES Advanced Encryption Standard

AES-256 Advanced Encryption Standard with 256-bit Key Size

AES-CBC AES Cipher Block Chaining

AES-CTR AES Counter

AI Artificial intelligence

ALK Attribute Length Key

AMQP Advanced Message Queuing Protocol

ANN Artificial Neural Networks

AR Augmented Reality

BEAST Browser Exploit Against SSL/TLS

CC Correlation Coefficient

CLCA Coordinated Load-Changing Attacks

CoAP Constrained Application Protocol

CP-ABE Ciphertext-Policy Attribute-Based Encryption

CRIME Compression Ratio Info-Leak Made Easy

CSTN Configurable Switching and Toggling Network

D2D Device-to-Device

DDoS Distributed Denial-of-Service

DHA Diffie Hellman Algorithm

DoS Denial-of-Service

DP-ABE Dual Policy Attribute-Based Encryption model

ECDHE Elliptic Curve Diffie-Hellman Ephemeral

ECDSA Elliptic Curve Digital Signature Algorithm

E-ECDH enhanced EC-DH

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie Hellman

ECDLP Elliptical Curve Differential Logarithm

-
- ECG** Electrocardiogram
EHRs Electronic Health Records
ERF Efficiency of a Random Function
FBC Fast Bitslice Cipher
GDPR General Data Protection Regulation
GWNs Gateway Nodes
HIBE Hierarchical ID-Based Encryption
HIPAA Health Insurance Portability and Accountability Act
HMAC Hash-based Message Authentication code
HOTP Based One-Time Password
HPC Hardware Performance Counter
HSTS HTTP Strict Transport Security
HW Hamming Weight
IBE Identity-Based Encryption
ICP-ABE Improved Ciphertext-Policy Attribute-Based Encryption
IDS Intrusion Detection System
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IoHTs Internet of Healthcare Things
IoMT Internet of Medical Things
IoT Internet of Things
IoTDePT Detecting Security Threats and Pinpointing Anomalies in an IoT environment
ISO International Organization for Standardisation
iTLS Lightweight Transport-Layer Security Protocol
JSON JavaScript Object Notation
KP-ABE Key Policy Attribute Based Encryption
KSA-PRESENT Key Schedule Algorithm-PRESENT
LED Light Encryption Device
LEDEM Learning-Driven Detection Mitigation
LKA Lightweight Key Agreement
LORENA Low memORy symmEtric-key geNerAtion
LPU Local Processing Unit
LWT Last Will and Testament
M2M Machine-to-Machine
MARAS Mutual Authentication and Authorisation Scheme
MART Mobile Augmented Reality for Tele-Assistance
MitM Man-in-the-Middle
ML Machine Learning
MQTT Message Queuing Telemetry Transport
MWSNs Marine Wireless Sensor Networks
OAuth 2.0 Open Authorisation 2.0

OCP-ABE Optimisation of Ciphertext-Policy Attribute-Based Encryption

OMA Open Mobile Alliance

OTP One-Time Password

PBAC Policy-Based Access Control

PCG Phonocardiogram

PDR Packet Delivery Ratio

PKI Public Key Infrastructure

PPG Photoplethysmogram

Pub/Sub Publish/Subscribe

QoS Quality of Service

QUIC Quick UDP Internet Connections

RBAC Role-Based Access Control

RFID Radio-Frequency Identification

RQ Research Question

RSA Rivest–Shamir–Adleman

RSMB Mosquitto Really Small Message Broker

SCG Seismocardiogram

SDP Symmetric Key Distribution Protocol

SEC-RMC Secure and Reliable Messaging Communication

SEEMQTT Secure End-to-End MQTT

SHA-256 Secure Hash Algorithm 256-bit

S-MQTT Secure MQTT

SCRAM Salted Challenge Response Authentication Mechanism

SDN Software Defined Networking

SMART Secure Mobile Augmented Reality for Tele-Assistance

SPDL Security Policy Definition Language

SQL Structured Query Language

SSI Self-Sovereign Identity

SSL/TLS Secure Sockets Layer/Transport Layer Security

TA Trusted Authority

TBAC Topic-Based Access Control

TCP-SYN Floods Transmission Control Protocol Synchronisation Floods

TCP/IP Transmission Control Protocol/Internet Protocol

TLAMD Testing Framework for Learning-Based Android Malware Detection

TLS Transport Layer Security

TTP Trusted Third-Party

UDP User Datagram Protocol

URI Universal Resource Identifier

WBAN Wireless Body Area Network

XML Extensible Markup Language

XMPP Extensible Messaging and Presence Protocol

XSS cross-site scripting

Glossary

5G 5G is the fifth generation of cellular technology. It is designed to increase speed, reduce latency, and improve flexibility of wireless services.

Access Control List ACL determines who has access to a resource and the actions they can perform. Each resource is associated with a set of permissions

Access Control Access Control is a data security process that enables organizations to manage who is authorized to access corporate data and resources.

Advanced Message Queuing Protocol AMQP is an open-source published standard for asynchronous messaging by wire.

Advanced Encryption Standard with 256-bit Key Size AES-256 specifically refers to AES with a 256-bit key, which offers a very high level of security.

Advanced Encryption Standard-Cipher Block Chaining AES-CBC is a secure mode of operation for AES block cyphers.

Advanced Encryption Standard-Counter AES-CTR is a block cypher mode of operation that uses a counter to encrypt data.

Artificial intelligence AI is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy.

Attribute-Based Access Control ABAC defines user permissions based on user attributes. Each user is assigned a set of attributes and each attribute is assigned a set of permissions.

Attribute-based Attribute-based in authorisation is a broader term, which refers to any authorisation method which uses attributes to determine access.

Augmented Password-Authenticated Key Exchange AugPAKE is a cryptographic protocol designed to securely establish a shared key between two parties using a password, even if the password itself is relatively weak. AugPAKE provides both mutual authentication and resilience against offline dictionary attacks, making it well-suited for secure communications where a shared password is used.

Authentication Access control for systems: check to see whether the credentials of a user match the credentials in a database of authorised users or in a data authentication server. The process of verifying the identity of a user or device attempting to access a network or system.

Authorisation This is a security mechanism which is the process of granting or denying access to a network resource which allows the user access to various resources based on the user's identity. The process of defining and enforcing permissions regarding what actions authenticated users or devices are permitted to perform.

Bluetooth Bluetooth is a technology standard used to enable short-range wireless communication between electronic devices.

CoAP CoAP is a specialised web transfer protocol used in the IoT with constrained nodes and constrained networks.

Diffie–Hellman Algorithm DH Algorithm is a method for securely exchanging cryptographic keys over a public communications channel.

Distributed Denial-of-Service DDoS attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

Efficiency of a Random Function ERF typically refers to a metric or analysis used to evaluate the performance, computational cost, and resource utilization of a cryptographic function or algorithm that behaves like a random function.

Elliptic Curve Digital Signature Algorithm This is a digital signature algorithm that uses elliptic curve cryptography to authenticate the identities of the communicating parties and ensure the integrity of the transmitted data.

Elliptic Curve Diffie–Hellman Ephemeral This is a key exchange mechanism that allows two parties to securely establish a shared secret over an insecure channel.

Elliptic-Curve Diffie–Hellman ECDH is a key-agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. This shared secret is used to derive another symmetric key.

Elliptic Curve Cryptography ECC is a cryptography approach based on the algebraic structure of elliptic curves over finite fields.

Extensible Messaging and Presence Protocol XMPP is a protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real-time.

ExTru ExTru is a communication protocol designed to enhance security and performance in resource-constrained IoT devices. ExTru incorporates a dynamic encryption mechanism in order to counteract vulnerabilities to physical attacks.

Fixed Header The fixed header is a two-byte header present in all MQTT messages.

General Data Protection Regulation The GDPR is a regulation enacted by the European Union (EU) to protect the privacy and personal data of individuals within the EU.

Health Insurance Portability and Accountability Act HIPAA is a U.S. law that focuses on safeguarding sensitive patient information in the healthcare sector. It establishes national standards for the protection of electronic health information, covering aspects like patient privacy, data security, and breach notifications.

Healthcare 4.0 It mainly applies the concepts of the Fourth Industrial Revolution to the field of health care.

Implants Implants are devices embedded within the human body to monitor or enhance physiological functions.

Last Will and Testament Last Will and Testament (LWT) is a powerful feature in MQTT that allows clients to specify a message that will be automatically published by the broker on their behalf, if or when an unexpected disconnection occurs.

Local Processing Unit LPU is a small, battery-operated device with limited computational capabilities, responsible for processing data collected from various biosensors attached to a patient's body.

LoRaWAN LoRaWAN is a low-power, wide area networking protocol built on top of the LoRa radio modulation technique.

Payload The payload is the data being transmitted by the message.

Policy-Based Access Control PBAC is a security framework that defines user permissions according to policies. Each user receives a set of policies and each policy defines the actions the user can perform.

Radio-Frequency Identification Tags RFID refers to the transmission of digital ID and other data between RFID tags and readers in a wireless or non-contact manner through electromagnetic waves.

Ransomware Ransomware is a type of malware which demands payment from the victim in order to recover control of their computer and data.

Rivest–Shamir–Adleman RSA is a public-key cryptosystem, one of the oldest widely used for secure data transmission.

Role-Based Access Control RBAC is a security framework that defines user permissions according to their role. Each user has a role, and each role has a set of permissions.

Salted Challenge Response Authentication Mechanism SCRAM is a family of modern, password-based challenge–response authentication mechanisms that provide authentication of a user to a server.

Secure Hash Algorithm 256-bit SHA-256 is a cryptographic hash function that generates a fixed 256-bit hash (digest) from input data of any size. Unlike AES, SHA-256 is not used for encryption but rather for data integrity verification.

Sensor Sensor is a device to detect and respond to different environmental changes, such as temperature, light, pressure, or motion.

SPONGENT SPONGENT refers to a lightweight cryptographic hash function designed for resource-constrained environments, such as IoT devices.

Token Authentication Token authentication is a type of authentication method that uses a Token to verify the identity of the user.

Topic-Based Access Control TBAC defines user permissions based on user trust levels. Each user has a trust level, and each trust level has a set of permissions.

Transmission Control Protocol Synchronisation Flood A SYN flood (half-open connection attack) is a denial of service (DDoS) attack designed to exhaust available server resources, rendering the server unable to transmit legitimate traffic.

Trojans Trojan horse is a type of malware disguised as legitimate code or software used to spy on and steal data.

Variable Header Variable Headers are optional headers present in some MQTT messages. It contains additional information about the message.

Volumetric Attack The volumetric attack is a method of bombarding a server with traffic so much that its bandwidth gets exhausted.

Wearable Wearable is a kind of device worn on the body in order to monitor various health and activity metrics.

Wi-Fi Wi-Fi is a wireless networking technology that uses radio waves to provide wireless high-speed Internet access.

Worms A worm is a type of self-replicating malware that usually spreads across a network by exploiting security vulnerabilities.

Zigbee Zigbee is a wireless protocol that is for low-data rate, low-power applications and is an open standard.

This chapter provides a foundational introduction to MQTT-enabled Internet of Things (IoT) in the context of Healthcare 4.0 and outlines the motivation for developing lightweight and secure communication protocols. Section 1.1 introduces the IoT and elaborates on its integration with Healthcare 4.0, emphasising the critical role of real-time communication, data sharing, and device interoperability. Section 1.2 presents the technical challenges MQTT faces in constrained environments, such as heterogeneity, scalability, resource limitations, and the lack of native security. In Section 1.3, various existing security solutions for MQTT are surveyed, including TLS-based security, authentication, authorisation, and lightweight cryptographic strategies. Section 1.4 highlights the significance and motivation of the research by addressing the real-world impact of MQTT vulnerabilities and justifying the need for novel lightweight security approaches. Section 1.5 defines the problem statement, key research questions, and objectives, clarifying the distinct roles of authentication and authorisation in the proposed schemes. Section 1.6 outlines the main contributions of the thesis, while Section 1.7 summarises the structure of the thesis and explains how each chapter contributes to the overall research goals.

1.1 Internet of Things (IoT) and Healthcare 4.0

The Internet of Things (IoT) is a global network of smart devices enabled by the Internet, deployed in various locations to monitor and share real-time data [1, 2]. Technological breakthroughs in information and communication technologies have had a profound impact on the IoT, driving new opportunities in real-time application scenarios such as automated production, intelligent logistics, smart manufacturing, smart transport, smart

living, innovative healthcare, and Industry 4.0 [3]. Among these, the integration of IoT with the emerging concept of Healthcare 4.0 [4] has received significant attention and acts as a key enabler to improve vital medical services [5,6]. Healthcare 4.0 has revolutionised the next generation of healthcare by leveraging advancements in IoT technologies and medical devices to improve patient care, enable continuous monitoring, ensure effective treatment, and facilitate timely intervention—ultimately improving overall patient outcomes.

These critical healthcare applications rely on real-time data transmission and sharing to generate prompt responses and ensure timely medical actions. To effectively manage the heterogeneous and resource-constrained characteristics of IoT devices, a lightweight and reliable Message Queuing Telemetry Transport (MQTT) protocol is often used to share patient-related information between medical devices and remote locations in a Healthcare 4.0 environment [7].

1.1.1 Healthcare 4.0 and Internet of Medical Things (IoMT)

At the heart of Healthcare 4.0 lies the Internet of Medical Things (IoMT), a specialised subset of IoT composed of interconnected medical devices, wearable sensors, healthcare applications, and cloud platforms designed to support real-time data acquisition, remote monitoring, and intelligent clinical decision-making [8,9]. The IoMT serves as the technical foundation of Healthcare 4.0, enabling seamless integration between patients, healthcare professionals, and data-driven services through secure and efficient communication channels.

Security and privacy have become major concerns with the growing impact of IoT in Healthcare 4.0. Medical IoT devices handle vast amounts of sensitive and non-sensitive data, which can be exploited, increasing attack vectors and threatening overall system security [10]. The widespread adoption of IoT technologies—particularly in healthcare—has introduced significant cybersecurity risks. Stealing valuable information, such as real-time monitoring data, can lead to various fraudulent activities, and attackers are incentivised to launch cyberattacks that target patient data. Securing MQTT-enabled healthcare IoT

systems is essential in preventing such attacks.

A robust architecture is vital for IoT-based healthcare systems to protect sensitive data through key security principles: confidentiality, access control, integrity, and availability. Strong authentication and Authorisation frameworks are necessary to prevent unauthorised access, while lightweight encryption mechanisms are essential for protecting data during transmission and storage. Implementing effective security mechanisms in MQTT-enabled Healthcare 4.0 environments presents even greater challenges than traditional networks due to resource limitations and architectural complexity.

Implementing security in MQTT-enabled Healthcare 4.0 poses greater challenges than traditional MQTT-based networks.

Due to the heterogeneity of IoT devices and communication protocols, along with diverse interfaces and services, it is not feasible to rely on standard heavy-weight and complex security solutions from conventional information technology networks [11, 12]. Traditional security measures are often inadequate given the large scale and variety of IoT devices and protocols used in Healthcare 4.0. Although lightweight cryptography offers a promising solution by providing feasible security for such systems, notable limitations persist due to the unique characteristics and diverse use cases of the IoT. Therefore, it is imperative to design and develop new lightweight security solutions, including end-to-end encryption, strong Authentication, fine-grained access control, data integrity, secure communication, efficient key management, and continuous vulnerability assessments—essential for securing the MQTT-enabled healthcare environment.

1.1.2 Essentials of MQTT

IoT has revolutionised how devices interact and communicate, creating an interconnected network capable of sharing information and adapting to dynamically changing environments. To meet the requirements of IoT in healthcare, MQTT [13] is an ideal application-layer communication protocol specifically designed for low-bandwidth communications in resource-constrained IoT devices. This versatile protocol supports routing strategies, including one-to-one, one-to-many, and many-to-many configurations. MQTT

is lightweight, efficient, and well-suited for the IoT environment, which often faces constraints such as device heterogeneity, limited resources, and unreliable network conditions [14]. Notable features of MQTT include the Publish/Subscribe (Pub/Sub) model, Quality of Service (QoS) levels, and retained messaging, making it highly adaptable and an excellent choice for IoT applications. Its lightweight nature is particularly advantageous in environments with intermittent connectivity, where numerous resource-limited devices must operate efficiently.

The MQTT Publish/Subscribe model effectively decouples the message producer (publisher) and consumer (subscriber), enabling asynchronous data exchange. This decoupling allows for scalable and flexible architectures in which IoT devices do not require persistent direct connections. It is especially beneficial for applications such as Industry 4.0, smart cities, connected vehicles, and home automation domains, where numerous heterogeneous devices and sensors continuously transmit data to cloud servers.

For example, the low latency and real-time message delivery offered by MQTT is crucial for instant messaging applications such as Facebook Messenger [15], which uses MQTT for online chat. The publish/subscribe communication model is also well suited for Amazon Web Services, where low-latency data exchanges are often required [15]. The lightweight nature of MQTT makes it an ideal solution for the Everything IoT platform [15], which connects millions of physical products to the digital world. The IoT hub [16] leverages MQTT as its primary protocol for telemetry message exchange. In the OpenStack context [17], MQTT with the Mosquitto broker provides upstream infrastructure support for real-time data transfer.

The main advantages of utilising MQTT include enabling real-time interactions, offering easy integration support, being highly scalable, requiring low resources, and facilitating seamless global data exchange between various products. In addition, its lightweight and efficient Pub/Sub messaging model is ideal for applications in environments with intermittent connectivity, where many devices with limited resources must interact. In particular, within Healthcare 4.0, MQTT plays a critical role in sharing real-time monitoring information from wearable and sensor-enabled medical devices to cloud-based healthcare servers [18]. This real-time communication capability allows healthcare providers

to monitor patient vital signs, ensuring continuous and timely healthcare interventions. Many cloud-based healthcare applications have been designed around medical devices that support MQTT, and its lightweight design minimises the use of device resources when transmitting data to cloud servers, making it ideal for remote healthcare solutions.

However, several technical challenges arise when implementing MQTT in real-world applications, which are discussed in the next section.

1.2 Technical Challenges for MQTT

Although the lightweight design and efficient messaging strategy make MQTT well-suited for IoT applications, several technical challenges must be addressed to achieve successful real-time implementation [19]. The key challenges affecting the deployment of MQTT in real-world IoT scenarios are as follows:

Heterogeneity [20]: IoT comprises numerous small-scale devices from various domains, including actuators, sensors, mobile systems, smart appliances and gateways. These devices use a variety of communication protocols, operate on disparate platforms, and employ various algorithms to process data.

Security and Privacy [21]: By design, MQTT lacks native security features. Consequently, it is vulnerable to several security issues, especially when deployed on resource-constrained devices. Common security challenges in MQTT-based IoT environments include ensuring data privacy, confidentiality, fine-grained authorisation, access control, data integrity, and end-to-end security.

Scalability [22]: As IoT ecosystems grow, scalability becomes a critical challenge. MQTT must support communication among millions of dynamic devices or nodes while maintaining performance and reliability.

Network Reliability [21]: IoT devices frequently operate in unstable network environments, which results in intermittent connectivity. Therefore, MQTT protocols must be designed to tolerate such conditions. Addressing issues such as limited bandwidth, high latency, network congestion, message loss, and fault tolerance is essential for reliable network performance.

Resource Constraints [21]: IoT devices typically have limited resources, including

energy, bandwidth, and memory. To be viable, lightweight MQTT implementations must minimise resource consumption and computational complexity.

Interoperability [23]: The data-sharing format of MQTT must be standardised and adaptable to allow seamless interaction and cooperation among heterogeneous IoT devices. High interoperability ensures that MQTT can support various applications across different platforms.

Quality of Service (QoS) [21]: MQTT supports three levels of QoS, each providing different guarantees regarding message delivery. Selecting the appropriate QoS level is a significant challenge in optimising reliability and performance in various IoT application scenarios.

1.2.1 Security Vulnerability in MQTT Protocol

The widespread adoption and popularity of MQTT have made it a frequent target for various types of cyberattacks [24]. Figure 1.1 classifies the common security attacks encountered in MQTT environments.

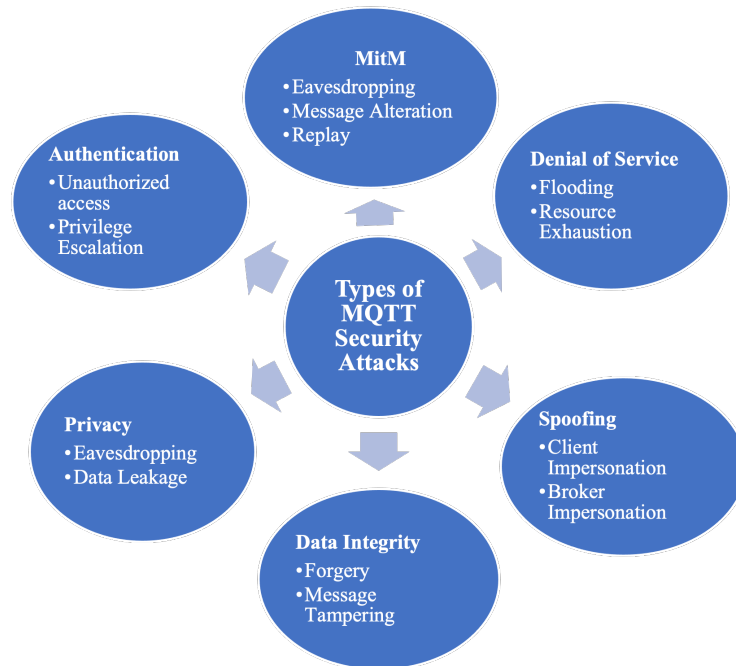


Figure 1.1: Classification of MQTT Security Attacks

1. Man-in-the-Middle (MitM) Attack [25]: Attackers intercept MQTT traffic between clients and brokers to read, modify, or delete messages. These attacks may involve eavesdropping, alteration of messages, or replay.
2. Denial-of-Service (DoS) Attack [26]: In this type of attack, adversaries flood MQTT clients or brokers with excessive traffic, rendering them unusable. Flooding and resource exhaustion are the primary tactics employed.
3. Spoofing Attack [24]: Attackers send falsified messages to clients or brokers to disrupt communication or steal data. This is often done by impersonating a legitimate broker or client.
4. Data Integrity Attack [27]: These attacks involve the forgery or manipulation of messages exchanged between clients and brokers, compromising data accuracy and consistency.
5. Privacy Attack [28]: If an attacker gains access to MQTT brokers or clients through attribute-based vulnerabilities, it can lead to data leakage via unauthorised eavesdropping of sensitive information.
6. Authentication and Authorisation Attack [28]: Unauthorised access or privilege escalation attacks undermine the integrity and confidentiality of the MQTT network by exploiting weak or absent authentication and access control mechanisms.

1.3 Security Solutions for MQTT

The lightweight, flexible, efficient, and reliable nature of the MQTT protocol makes it a compelling choice for various IoT applications, particularly Healthcare 4.0 and medical systems. The publish/subscribe (Pub/Sub) model and multiple QoS levels enable distinct communication patterns and accommodate various reliability requirements. However, to ensure secure and resilient communication in Healthcare 4.0 systems, addressing the security vulnerabilities inherent in MQTT is critical.

By default, MQTT does not operate securely or encrypt data before transmission [14]. MQTT security is a pressing concern for many IoT deployments, especially within Industry 4.0 organisations that depend on interconnected devices. IoT devices are susceptible to attacks that can compromise sensitive information and disrupt device operations and

management, particularly in environments connected to the Internet [24]. Robust MQTT security ensures data authorisation, confidentiality, integrity, and availability within the Healthcare 4.0 domain [29]. Several MQTT security solutions have been proposed, as outlined below.

- **Transport Layer Security (TLS)-based Security:** TLS is a cryptographic protocol designed to protect communications over computer networks [30]. It uses digital certificates to provide encryption, authentication, and integrity between MQTT brokers and clients. Despite these protections, attackers can still capture MQTT topics and messages exchanged between publishers and subscribers. Some traditional authentication mechanisms, such as username and password pairs, validate clients within the Secure Sockets Layer/Transport Layer Security (SSL/TLS) channel. However, this approach only prevents initial impersonation and does not secure the communication content. In addition, message replay attacks are still possible. Due to its complexity and susceptibility to MitM attacks, TLS-based security is often unsuitable for resource-constrained IoT environments.
- **Authentication-based Security:** This method ensures that only authorised clients can connect to the MQTT broker [24,31]. It uses credentials such as usernames, passwords, certificates, and OAuth 2.0 tokens to mitigate authentication-based threats. However, the high implementation and computational costs make it less practical for resource-limited IoT scenarios.
- **Authorisation and Access Control-based Security:** This approach ensures that only authorised clients can perform specific MQTT operations. Mechanisms such as Access Control List (ACL), Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC) are commonly used [24,31]. Among them, ABAC is particularly sophisticated, since it grants or denies access based on user roles, resource types, contextual attributes, and IoT environment attributes.
- **Privacy-Enabled Solutions:** Another key challenge in MQTT communication is to preserve data privacy during evaluation and transmission [32]. Privacy is especially critical for Healthcare 4.0 applications. Although most existing MQTT

security solutions provide encryption during data transmission and decryption at the storage or receiving end, they typically require complete trust in the data storage infrastructure. These conventional solutions cannot ensure end-to-end privacy if a compromised or malicious device leaks sensitive data.

- **Lightweight Cryptography-Based Security:** Lightweight cryptographic algorithms offer adequate security while addressing the performance and resource constraints of IoT devices [33, 34]. However, achieving real-time security performance in diverse IoT applications remains a significant challenge. Although traditional methods may use secure channels with certificate-based key management, such schemes are often resource-intensive and unsuitable for constrained environments. Multiple cryptographic approaches have been proposed to improve the security of MQTT [35, 36]. Integrating robust and lightweight authentication, encryption, and Access Control mechanisms is crucial to ensure efficient and secure MQTT communication in diverse and rapidly evolving IoT environments.

1.4 Significance and Motivation

According to McKinsey’s predictions, more than 50 billion IoT devices are expected to be connected globally by 2025 [37]. This rapid growth has revolutionised society by enabling a wide range of applications, from remote medical monitoring to smart home automation, and is expected to continue evolving in the future [38]. Another study estimates that between 40 and 75 billion IoT devices will be deployed between 2025 and 2033 [39, 40]. These devices include webcams, wearables, sensors, and many others across diverse IoT use cases.

A typical IoMT architecture includes resource-constrained devices that transmit sensitive medical data using lightweight communication protocols such as Message Queuing Telemetry Transport (MQTT). In this context, ensuring security, privacy, and interoperability becomes essential to support time-sensitive healthcare operations while adhering to the computational limitations of medical devices.

Despite its efficiency and popularity, MQTT is vulnerable to numerous cybersecurity threats. Several real-time security breaches have been observed in IoT deployments,

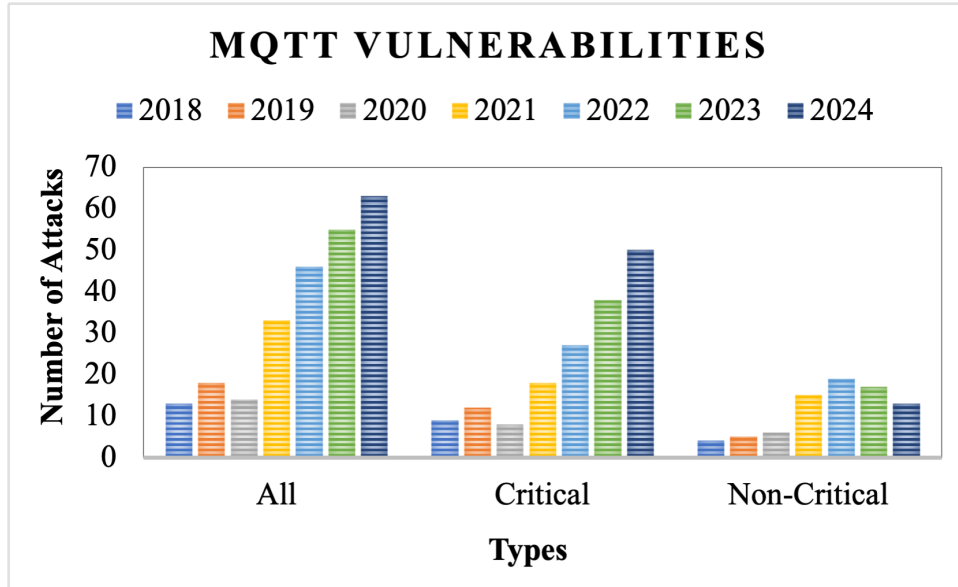


Figure 1.2: Reported MQTT Vulnerabilities

significantly compromising the security and functionality of time-sensitive applications, especially in healthcare. These vulnerabilities have serious consequences, including loss of life and considerable financial damage. For example, a significant MQTT-based cyber-attack in 2020 on a medical system caused sensitive data leaks and service interruptions. The economic impact included recovery costs of approximately \$500,000, regulatory fines of \$200,000, and an estimated \$1 million in revenue losses due to service unavailability. In 2023, IBM reported that an MQTT data leak resulted in a \$4.45 million financial loss. According to Kaspersky, nearly 18 out of 33 known MQTT vulnerabilities are classified as high risk, and 91% healthcare providers have been affected by these types of attacks [41]. Figure 1.2 illustrates the reported vulnerabilities in MQTT from 2018 to 2024. These findings highlight the critical need for effective and secure MQTT communication, which motivates this research to develop novel security strategies tailored for healthcare IoT systems.

As discussed in Chapter 1, the evolving role of IoMT in Healthcare 4.0 introduces new challenges for MQTT communication, particularly regarding authentication, access control, and lightweight encryption. Lightweight cryptographic algorithms offer high security with minimal computational overhead, which is crucial to maintaining performance in real-time MQTT-based IoT applications, especially in Healthcare 4.0 [24, 42]. These

schemes strike a balance between security and resource efficiency. Evaluating various lightweight cryptography approaches helps identify the most suitable solution for specific IoT environments. In parallel, ABAC mechanisms provide fine-grained access control by evaluating user roles, resource types, and environmental factors [24]. Such mechanisms are well-suited to manage complex access policies across heterogeneous IoT devices and data streams. Together, lightweight cryptography and ABAC offer a promising solution for securing evolving MQTT-based IoT systems, thus motivating the design and evaluation of new cryptographic strategies for secure Healthcare 4.0 communications.

Therefore, the secure design of MQTT protocols in Healthcare 4.0 is intrinsically tied to the characteristics and constraints of IoMT, motivating the research contributions presented in this thesis.

1.4.1 Shortcomings in the Current Security Solutions

Despite many proposed security solutions, several critical research gaps remain in addressing MQTT-specific security challenges. A key limitation is that many IoT devices that use MQTT are resource restricted, making conventional security approaches, such as heavyweight cryptography and TLS, impractical [30]. Therefore, it is essential to investigate and innovate lightweight security schemes tailored explicitly for IoMT environments.

Firstly, conventional authentication based on usernames and passwords is inadequate for meeting the stringent security requirements of IoT applications. Furthermore, certificate-based mutual authentication is often too resource-intensive for constrained devices. These limitations underline the need for lightweight multifactor authentication strategies.

Secondly, various lightweight cryptographic solutions exhibit inconsistent performance in different IoT contexts [24, 42], yet many studies lack a detailed comparative analysis under diverse conditions.

Thirdly, many MQTT security implementations lack formal verification, a critical step in identifying design flaws and ensuring protocol integrity. Formal methods and model verification techniques should be used to verify the accuracy and security properties of lightweight cryptographic schemes rigorously [43].

Addressing these gaps requires the adoption of scalable, formally verified, and context-sensitive security frameworks. This research focuses on introducing and optimising lightweight security models with strong authentication capabilities, ultimately enhancing the resilience and performance of MQTT against various threats in Healthcare 4.0.

1.5 Problem Statement

Applying conventional cryptographic methods directly to resource-constrained IoT devices is often impractical due to their high computational complexity. Lightweight and low-complexity cryptographic algorithms have been proposed to address this issue using optimised design methodologies. However, these solutions exhibit varying levels of performance depending on the characteristics of the IoT environment, including dynamic network topologies, device heterogeneity, data sensitivity, and environmental constraints.

Although numerous lightweight cryptographic schemes exist, many suffer from limited real-world validation and lack comprehensive analysis across diverse network scenarios and metrics. As a result, there is a critical need to systematically evaluate these lightweight methods to design novel security strategies that impose minimal computational burden and are specifically suited for intelligent IoT devices. The optimisation of lightweight security protocols tailored to specific application requirements is essential to strengthen overall security while maintaining the performance of MQTT-based communication in resource-limited environments.

Among existing solutions, Attribute-Based Encryption (ABE) schemes have gained attention to secure IoT communications. Although ABE offers robust security through fine-grained access control, it suffers from computational overhead and memory demands, making it challenging to deploy in publish/subscribe-based MQTT environments, particularly in healthcare settings where real-time performance is critical. Therefore, a lightweight, privacy-preserving, and authentication-capable scheme is necessary to support secure, low-latency data sharing in MQTT-enabled Healthcare 4.0 systems.

Optimising ABE with enhancements such as self-key revocation, dynamic attribute selection, and lightweight cryptographic primitives can significantly reduce computational overhead, making ABE schemes more feasible for real-time and resource-constrained ap-

plications. Such optimisation and evaluation are essential for enabling effective and secure communications in next-generation healthcare IoT systems.

1.5.1 Aim of the Thesis

Given these challenges, this thesis aims to design, implement and evaluate lightweight, secure, and authentication-enabled MQTT communication strategies tailored for resource-constrained Healthcare 4.0 environments. Special emphasis is placed on optimising Attribute-Based Encryption (ABE) schemes to enable fine-grained access control, reduce computational overhead, and support real-time, secure communication across dynamic IoMT networks.

The proposed work seeks to strengthen security while maintaining performance in publish/subscribe MQTT models, thereby enabling efficient and scalable data sharing in Healthcare 4.0 systems without overburdening constrained medical devices.

1.5.2 Research Questions

Based on the issues and challenges discussed in Section 1.4, this thesis seeks to answer the following key Research Question (RQ):

- **RQ1:** Why is it essential to analyse the performance of lightweight symmetric key cryptographic algorithms in MQTT-enabled IoT environments?
- **RQ2:** What are the potential methods and tools for analysing and validating the performance of five lightweight cryptographic algorithms?
- **RQ3:** Why is a novel Attribute-based authentication-enabled MQTT strategy needed for IoT environments?
- **RQ4:** What type of lightweight MQTT authentication strategy can be designed to secure publisher-subscriber communication?
- **RQ5:** What are the benefits and limitations of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) in MQTT-enabled IoT environments?
- **RQ6:** How can an optimised lightweight MQTT security and authentication scheme be applied to healthcare-specific IoT environments?

1.5.3 Objectives of the Thesis

Given the above problems and challenges, this thesis sets out the following research objectives to address the identified research questions. These objectives will also be elaborated upon in Chapters 5 to 8:

- To evaluate the performance of different lightweight symmetric key security algorithms under two distinct security attack models within an MQTT-enabled Healthcare 4.0 environment. This will be done using the Cooja simulator [44], a widely used tool in the Contiki operating system for IoT testing and validation [45]. Cooja allows network simulation before hardware deployment and provides real-time monitoring of simulation events.
- To propose an enhanced attribute-based encryption scheme, based on ciphertext-policy, that incorporates a lightweight authentication design and is highly adaptable to MQTT-enabled IoT environments.
- To introduce a novel authentication scheme aimed at securing lightweight MQTT communication in the Internet of Medical Things (IoMT) [7], incorporating proper authorisation and access control mechanisms, and clearly distinguishing between authentication (verifying identities) and authorisation (controlling access based on attributes).

Clarification: Throughout this thesis, while the terms authentication and authorisation can be used jointly, it is explicitly emphasised that the Improved Ciphertext-Policy Attribute-Based Encryption (ICP-ABE) and Optimisation of Ciphertext-Policy Attribute-Based Encryption (OCP-ABE) schemes are designed primarily to provide authorisation through fine-grained attribute-based access control, with supplementary authentication achieved through verification of entity attributes.

1.6 Major Contributions

This thesis presents three significant contributions to enhance the robustness, efficiency, and adaptability of security mechanisms for MQTT communication in IoT applications.

1. Performance Evaluation of MQTT in Healthcare 4.0

- (a) The initial phase of the research analyses the security and performance of symmetric encryption schemes within MQTT-enabled smart Healthcare 4.0 systems. The evaluation considers data confidentiality, communication efficiency, computational complexity, and energy consumption.
- (b) Three evaluation methodologies are adopted: formal security analysis, tool-based verification, and simulation-based testing. Formal and tool-based methods involve theoretical formulations and software tools to assess algorithmic efficiency.
- (c) Cooja-based simulation models are utilised under attack and non-attack conditions across various network scenarios to validate lightweight cryptographic algorithms' real-time performance and security strength.
- (d) This phase offers a multi-perspective analysis to identify the most suitable lightweight encryption strategies for secure MQTT communication, considering available countermeasures.

2. Lightweight Authentication Scheme Based on ICP-ABE for MQTT

- (a) This research contributes to a lightweight authentication mechanism that integrates efficient cryptographic primitives with an improved CP-ABE scheme. The goal is to secure MQTT communication among IoT devices without compromising performance.
- (b) The scheme separates attribute auditing and key extraction, allowing key generation without revealing attribute information, enhancing user privacy. By avoiding MQTT's keep-alive settings, the model also prevents SlowDoS [46] attacks from degrading network performance.
- (c) The length of the secret key is reduced using blind tokens embedded with selective attributes. Only MQTT clients with the correct blind token can perform publish/subscribe operations, ensuring strong mutual authentication.
- (d) The cryptographic properties of the proposed model are formally analysed

using mathematical proofs. Furthermore, its effectiveness is validated through provable security analysis and simulations in the Cooja environment.

3. OCP-ABE Lightweight Authentication Scheme

- (a) This contribution presents an OCP-ABE-based lightweight cryptographic scheme designed to enhance MQTT security and performance in the IoMT environment. The approach combines attribute-specific blind key generation with the Fast-PRESENT encryption algorithm.
- (b) Using Fast-PRESENT for blind key sharing and self-key revocation significantly reduces traceability risks, defending against threats such as masquerading, DoS, and guessing attacks.
- (c) Parallel execution of S-box operations within Fast-PRESENT enhances encryption speed, optimising runtime performance.
- (d) A dynamic attribute-based signature scheme enables both direct and indirect user revocation, preventing repeated access by users with the same attribute set. This feature also reduces ciphertext size, lowering system complexity while maintaining security.
- (e) The scheme is evaluated for resilience against various attack vectors and validated through Cooja-based simulation analysis.

This thesis proposes lightweight, robust authentication and privacy-preserving solutions for MQTT-enabled, resource-constrained healthcare IoT environments. It also offers seamless support for authorisation and access control during data flow between a broker and clients, reinforcing its primary role in authorisation alongside its authentication features. The proposed solutions are validated through multiple approaches, including formal proofs, software-based verification tools, and Cooja-based simulations.

1.7 Thesis Organisation

This thesis comprises nine chapters, presenting the research design, methodology, and key findings.

Chapter 2 introduces the background of MQTT-enabled IoT and IoMT systems, emerging IoT applications, key technological challenges, and existing security issues in MQTT. It also reviews essential studies that contribute to a foundational understanding of MQTT security and the broader research landscape. Chapter 3 discusses existing security solutions for MQTT, including ABE-based approaches, and identifies limitations and gaps in current research. The chapter highlights the motivation for adopting lightweight security mechanisms to address MQTT vulnerabilities, particularly in Healthcare 4.0 environments. It also identifies the research gaps that inform the design of the proposed ICP-ABE and OCP-ABE frameworks. Chapters 7 and 8 then focus on the implementation, system model, and performance evaluation of these frameworks.

Chapter 4 outlines the research methodology, including the three evaluation strategies adopted for lightweight MQTT algorithm validation. It discusses the use of simulation tools, focusing on the Cooja simulator, and explains its relevance for evaluating the security and performance of MQTT-based systems.

Chapter 5 reviews various symmetric key cryptographic algorithms and highlights relevant studies. This chapter marks the study's first phase by applying and validating lightweight symmetric encryption algorithms in MQTT communication for Healthcare 4.0 applications. It integrates formal security analysis and tool-based validation approaches to examine the robustness and effectiveness of each encryption scheme.

Chapter 6 extends the previous analysis by evaluating the real-time performance of the selected symmetric encryption algorithms using the Cooja simulator. It performs simulations under various network and attack scenarios to determine the most suitable cryptographic algorithms for lightweight MQTT frameworks.

Chapter 7 introduces the second research contribution. It proposes a lightweight authentication and privacy-preserving scheme tailored to MQTT communication. The design leverages selected symmetric encryption algorithms and an attribute-based authentication framework. The simulation results are presented to demonstrate the performance of the model and the privacy guarantees.

Chapter 8 presents the third contribution, offering a refined lightweight security scheme for IoMT environments. Improves ICP-ABE by enhancing encryption speed through

parallel S-box execution (Fast-PRESENT) and integrating indirect user revocation to tackle traceability and unauthorised access issues. The chapter provides simulation-based performance evaluation to support the effectiveness of the model.

Chapter 9 summarises the three research contributions, restates how each research question was addressed, identifies study limitations, and proposes directions for future research.

This chapter provides background information on MQTT-enabled IoT systems and offers a comprehensive review of the literature related to MQTT security. Section 2.1 introduces the fundamental components of the IoT, the layered IoT architecture, and various potential applications that leverage MQTT-enabled IoT. In addition, we introduce the components, unique requirements, characteristics, and security challenges of IoMT in Healthcare 4.0.

Section 2.2 discusses the technical aspects of MQTT within IoT environments, including its lightweight design, the Publish/Subscribe (Pub/Sub) model, the MQTT broker, topic hierarchy and filtering mechanisms, topics and subscriptions, QoS levels, retained messages, and its scalable, interoperable architecture. This section also highlights its robust and adaptive features by highlighting MQTT's suitability in complex healthcare IoT (IoMT) scenarios.

In addition, Section 2.3 explores the importance of securing MQTT communication, identifies different categories of security attacks, and assesses their impact on MQTT performance. The section compares how various security attacks influence the core security requirements, such as confidentiality, integrity, availability, and system performance.

2.1 IoT Background

The rapidly expanding IoT forms a network of interconnected intelligent devices capable of exchanging information and communicating over the Internet [47,48]. These devices range from simple sensors and actuators to complex household appliances. The core objective of IoT is to enable everyday objects to collect, process, and share data, thereby enhancing automation, operational efficiency, and decision-making across various sectors [49]. The

core components of an IoT system are illustrated in Figure 2.1.

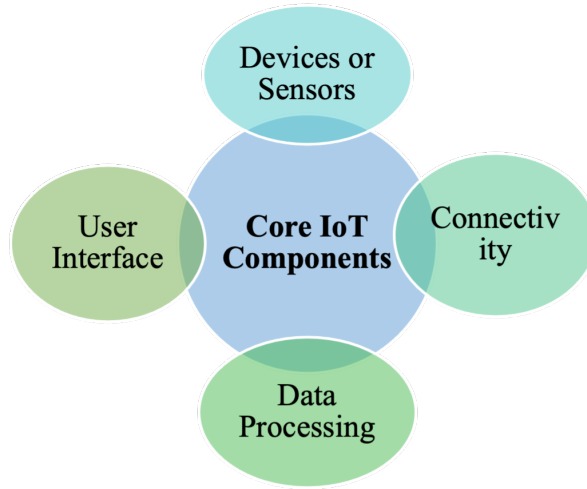


Figure 2.1: Core Components of IoT

Devices or Sensors: These are physical objects equipped with sensors, embedded software, and communication technologies. They are typically resource-constrained and are responsible for monitoring their environment. Examples include wearable health monitors, intelligent lighting systems, temperature sensors, smart thermostats, autonomous vehicles, and connected cars.

Connectivity: IoT devices require reliable communication mechanisms to interact with each other and central systems. They commonly use wireless technologies such as Wi-Fi, Zigbee, Bluetooth, LoRaWAN, and 5G [50].

Data Processing: After data collection, processing is performed locally on edge devices or cloud servers. Data analytics may involve basic filtering for immediate response or advanced computations for deriving insights.

User Interface: This allows end users to interact with the IoT system. Through interfaces, users can receive notifications (e.g., emergency alerts), monitor device status, or control operations through applications connected to cloud services.

2.1.1 Layered Structure of IoT

IoT architecture is often described using layered models to address its complexity [51]. A standard model is the four-layer structure that comprises perception, network, middle-

ware, and application layers [52]. However, more detailed models, such as the seven-layer structure illustrated in Figure 2.2, provide a finer breakdown of IoT components and their respective functionalities [53].

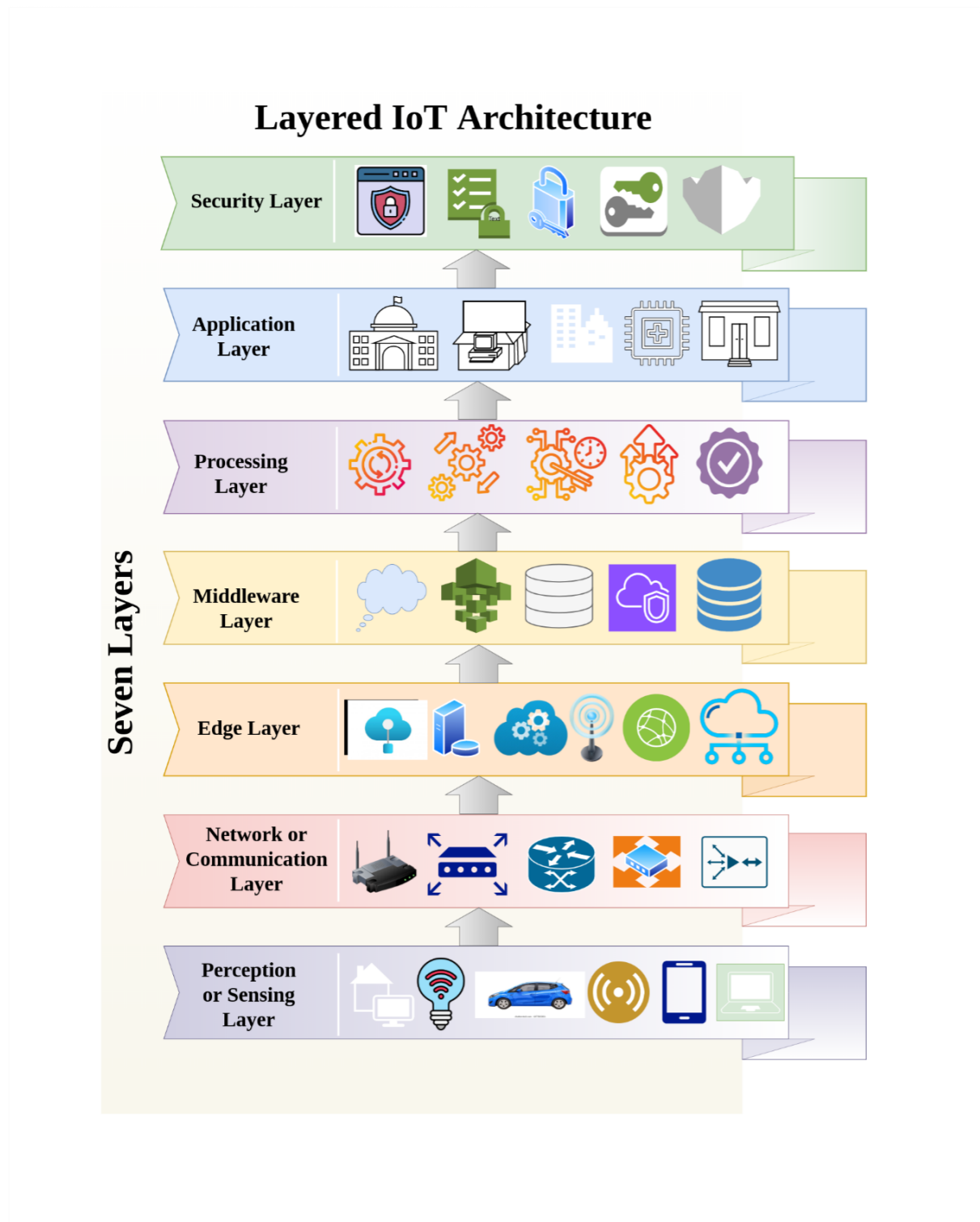


Figure 2.2: Layered IoT Architecture

Perception or Sensing Layer [53]: This layer collects data from the physical environment through intelligent IoT devices. These sensing devices monitor a range of physical parameters, such as temperature, blood pressure, heart rate, humidity, motion, and light, depending on the application context [54]. Actuators, in turn, interact with the physical world in response to control signals, such as operating motors, lights, or remote healthcare tools. The primary function of the perception layer is to observe and report environmental conditions by converting physical stimuli into digital signals for further processing.

Network or Communication Layer [53]: Situated above the sensing layer, the network layer is responsible for transmitting the collected data to cloud platforms or centralised servers. Data are relayed through network components, such as routers, gateways, and edge nodes. This layer ensures effective communication among IoT devices using wireless technologies, including Wi-Fi, Zigbee, Bluetooth, LoRaWAN, and 4G/5G cellular networks [55]. Gateways consolidate data from diverse sources and facilitate communication across the lower and upper layers. Its primary role is to support reliable, efficient and secure data exchange in heterogeneous IoT environments.

Edge Layer [53]: Located close to the point of data generation, such as sensors, gateways, or local edge nodes, this layer enables real-time information processing. It allows rapid responses to time-critical events by performing preliminary data analysis locally [56]. By filtering, aggregating, and compressing raw data at the edge, only relevant insights are transmitted to the cloud, reducing latency and bandwidth usage. This mechanism is crucial for latency-sensitive applications such as autonomous vehicles, remote health monitoring, and industrial automation.

Middleware Layer [53]: The middleware layer is an intermediary between hardware and higher-level software systems. Collects, aggregates, and preprocesses data from multiple IoT devices to ensure it is formatted consistently for analytics or decision making [57]. This layer supports interoperability across heterogeneous devices, enabling integration without custom configuration. It also processes events such as alarms, threshold breaches, and status changes in real-time. It provides standardised APIs that abstract the complexity of low-level protocols to facilitate application development.

Processing Layer [53]: This layer is responsible for analysing and managing the large volumes of data generated by IoT systems. Positioned between the network and application layers, it transforms raw data into actionable insights through data analytics and context-aware computing [58]. It can autonomously trigger actions, adjust device parameters, or issue alerts based on predefined rules or detected patterns. Furthermore, it enables dynamic decision making by identifying the context in which data is generated and adapting processing strategies accordingly.

Application Layer [53]: The application layer interfaces with end-users by providing services derived from processed IoT data. It encompasses user-facing components such as mobile applications, dashboards, and web portals, allowing users to interact with and manage IoT systems [59]. The application logic and business rules of this layer allow automation and intelligent control in domains such as home automation, digital healthcare, smart transportation, smart cities, and industrial systems [49].

Security Layer [53]: The security layer ensures that fundamental security requirements, such as confidentiality, integrity, availability, and access control, are met across all layers of the IoT ecosystem. It is essential to protect data, devices, and communication protocols from cyber threats [60]. This is achieved by implementing cryptographic mechanisms, including encryption, decryption, hashing, and digital signatures. It also handles secure key management, authentication, and access validation. Lightweight security protocols, such as OAuth, MQTT-based authentication, Datagram Transport Layer Security, Constrained Application Protocol (CoAP), and Kerberos, are supported to suit resource-constrained IoT devices [51]. In addition, firewalls, intrusion detection, and prevention systems are integrated to detect and mitigate unauthorised access to or malicious traffic. Collectively, these measures strengthen the resilience and trustworthiness of IoT infrastructures.

2.1.2 Emerging IoT Applications

Using modern technologies such as edge computing and 5G, IoT applications are rapidly evolving and transforming various industries [61]. These emerging applications extend

beyond conventional use cases, enabling novel opportunities in multiple sectors, including healthcare, transportation, agriculture, manufacturing, and smart city infrastructure. Selected examples of these applications are illustrated in Figure 2.3.

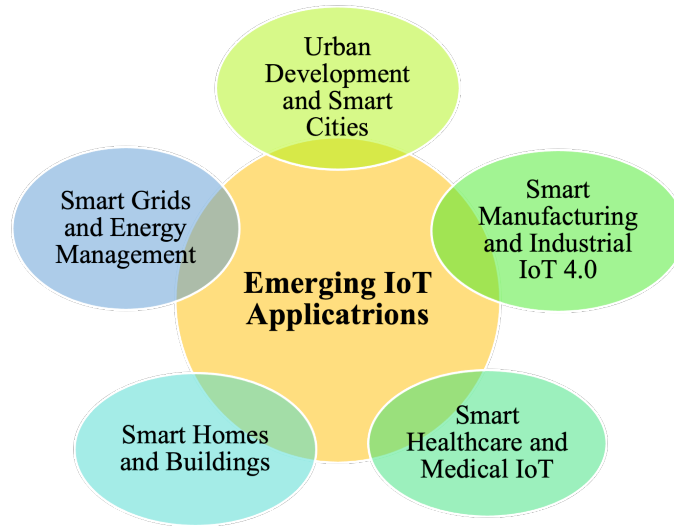


Figure 2.3: Emerging IoT Applications

Urban Development and Smart Cities

IoT-enabled smart cities aim to enhance the quality of urban life by optimising a range of services, including energy management, transportation, waste collection and public safety. Effective smart city planning is essential for creating sustainable, efficient, and liveable urban environments [62, 63]. By integrating advanced technologies such as data analytics and IoT, smart cities can address critical urban challenges such as traffic congestion, energy inefficiency, pollution control, intelligent waste disposal, and strained public services, while simultaneously improving the overall quality of life of citizens. Furthermore, the 5G and edge computing combination facilitates the delivery of modernised and responsive city management services. For instance, intelligent traffic systems equipped with smart sensors use real-time data from connected vehicles to adjust traffic lights and reduce congestion dynamically.

Smart Manufacturing and Industrial IoT 4.0

Industry 4.0 [64] is a defining element of the fourth industrial revolution, characterised by integrating physical manufacturing processes with advanced digital technologies, including IoT, edge computing, machine learning, robotics, cloud computing, and big data analytics. The adoption of IoT in industrial contexts drives unprecedented levels of automation, efficiency, and seamless connectivity across manufacturing systems. Industry 4.0 is marked by deploying interconnected sensors, devices, and machinery that facilitate real-time data collection, communication, and analysis [65,66]. Industrial IoT (IIoT) devices are embedded with sophisticated software, standardised communication protocols, and robust network connectivity, enabling automated processes and actionable insights in routine and critical decision-making environments. Typical IIoT applications include energy management, predictive maintenance, real-time monitoring, environmental control, and process optimisation in manufacturing, transportation, oil and gas, and logistics.

Smart Healthcare and Medical IoT

IoT is increasingly critical in the healthcare industry by integrating advanced digital technologies and connected smart devices, such as wearables and implants, to improve patient care, optimise clinical operations and improve healthcare outcomes. Innovations in medical IoT enable continuous real-time monitoring, personalised treatment plans, remote patient care, accurate diagnostics, and enriched medical data analytics, thus transforming traditional healthcare systems. Integrating Internet-enabled medical devices with healthcare institutions facilitates real-time collection of patient health data [67,68]. For example, wearable sensors and smart implants continuously monitor chronic conditions such as heart disease and diabetes, transmitting data to healthcare providers to enable proactive care and early intervention. Bright pill dispensers remind patients to take medications on schedule and automatically notify healthcare professionals if a dose is missed.

Smart Homes and Buildings

Smart homes and buildings utilise IoT technologies to improve energy efficiency, security, functionality, and convenience. These systems enable users to monitor environmental conditions and autonomously manage everyday tasks and building operations. IoT-enabled smart buildings can monitor and control lighting, ventilation, heating, air conditioning, energy usage, and security systems [69]. For example, home automation systems dynamically adjust temperature and airflow according to occupancy patterns and external weather conditions, thereby maximising energy efficiency [70]. These systems provide secure, real-time, data-driven solutions that enhance user comfort and building intelligence by integrating edge computing and IoT with cryptographic techniques.

Smart Grids and Energy Management

IoT revolutionises energy management within innovative grid environments by improving efficiency, sustainability, and reliability. Smart grids incorporate advanced digital technologies into traditional energy infrastructures to facilitate real-time monitoring, control, and optimisation of electricity generation, distribution, and consumption [71]. These systems also empower consumers to manage energy usage more effectively, support the integration of renewable energy sources, and improve the resilience of the grid. Smart grids enabled with IoT can dynamically adjust energy distribution based on real-time demand and availability, allowing consumers to trade surplus energy generated by solar panels or battery systems [72]. IoT technologies contribute significantly to the development of intelligent, sustainable energy networks through decentralised energy trading and adaptive load balancing.

Table 2.1 captures the essentials of emerging IoT applications, their transformative potential in various sectors, and the technical barriers they face that must be solved for the continued boom.

Table 2.1: Comparative Analysis of Emerging IoT Applications.

Application Type	Key principle	Advantages	Technical barriers	Future Scope
Urban Development and Smart Cities	IoT-enabled infrastructure, traffic, and waste management	Public service enhancements, energy efficiency, and congestion reduction	Management of massive data, high cost, scalability, and privacy problems	5G integration, artificial intelligence, and innovative mobility solutions
Smart Manufacturing and Industrial IoT 4.0	Predictive maintenance, industrial automation, data analytics	Maximized efficiency, downtime reduction, and cost savings	Integration with legacy systems, cybersecurity, and privacy	Artificial intelligence and 5G-enabled automation
Smart Healthcare and Medical IoT	Wearable devices, remote monitoring, data analytics	Reduced hospital visits, Efficient and timely personalized patient care	Information privacy, device security, and regulatory compliance	Artificial intelligence-enabled secure and timely diagnostics
Smart Homes and Buildings	Automated homes, remote control, voice control, security	Energy saving, convenience, and enhanced home security	High cost, novel attacks, device interoperability, privacy issues	Energy management solutions and green technology
Smart Grids and Energy Management	Real-time monitoring and efficient management of distributed energy resources	Enhanced grid reliability, energy efficiency, and renewable integration	High implementation costs, cybersecurity, and privacy risks	Blockchain-integrated solutions, microgrids, and renewable energy

2.1.3 Background of Healthcare IoT

Healthcare IoT is a broad domain that revolutionises digital healthcare systems through the integration of healthcare information received from smart medical devices and medical sensors into different operations, such as improving patient monitoring, disease diagnostics, and operational efficiency [73, 74]. By exploiting real-time data gathering, remote patient monitoring, and artificial intelligence-driven automation, Healthcare IoT improves patient care with reduced costs [75, 76].

However, widespread adoption of Healthcare IoT introduces several challenges, including security risks, interoperability problems, data management complexities, and regulatory compliance requirements [77]. These challenges are intensified by the resource-constrained nature of many medical devices and the need for low-latency, secure communication in life-critical systems.

Key Components of Healthcare IoT

Healthcare IoT includes several critical components that facilitate timely, efficient, and secure healthcare service delivery.

Medical Devices and Sensors: They play a key role, including wearables for ECG monitors and smartwatches, implantable devices for insulin pumps and pacemakers, and hospital equipment for smart beds and ventilators. These smart devices continuously collect vital information about the patient, such as heart rate, glucose levels, and oxygen saturation, further transmitting the collected data to processing units for examination [78].

IoT Gateways and Edge Computing: They guarantee efficient data handling by filtering, encrypting, and transmitting sensitive medical information while shrinking network congestion and latency. With the support of cloud computing and big data analytics, Healthcare IoT stores and processes vast healthcare data, facilitating artificial intelligence-driven predictive diagnostics and early disease identification [75]. Moreover, for robust network infrastructure building, it is crucial to incorporate 5G, Wi-Fi, Bluetooth, and RFID that facilitate seamless connectivity for real-time patient monitoring.

Healthcare IoT Architecture: A multi-layered structure supports data capture,

transmission, analysis, and decision-making [79]. The perception layer (device layer) consists of smart medical sensors and wearables that gather real-time patient data. The network layer ensures secure data transmission through IoT gateways and communication protocols. Once collected, the data moves to the processing layer, where edge computing and cloud-based AI analytics assess anomalies and generate predictive insights for disease management. The application layer provides user interfaces for healthcare professionals, integrating electronic health records, AI-powered diagnostics, and mobile health applications for real-time decision making. This structured architecture improves data security, efficiency, and accessibility across diverse healthcare environments.

Healthcare IoT Challenges and Security Considerations

Despite its transformative benefits, Healthcare IoT faces significant challenges that hinder its full potential [80–82].

Security and Privacy: Healthcare IoT devices are highly vulnerable to cyber threats such as MitM and DoS attacks, leading to data breaches and compromised patient safety. Strong encryption, authentication, and secure key management are necessary to protect sensitive health information. Unauthorised access can disrupt medical operations, making robust access control essential. Regular security updates and anomaly detection systems are crucial to mitigate evolving threats.

Resource Limitations: Many healthcare IoT devices are battery-operated, requiring efficient power management to ensure long-term functionality. Frequent battery replacements or recharging may not be practical in critical healthcare environments. Low-power communication protocols and energy-efficient hardware design are essential to extend the useful life of the device. Adaptive power management techniques can help optimise energy consumption while maintaining performance.

Interoperability Issues: Different manufacturers use proprietary protocols, making it difficult for IoT devices to communicate seamlessly. The lack of standardised communication frameworks creates integration challenges between various medical systems. This issue affects data consistency, device compatibility, and efficient healthcare workflows. Establishing global interoperability standards is necessary to ensure smooth data

exchange.

Network and Latency Constraints: IoT networks often suffer from bandwidth limitations and high latency, which can delay real-time patient data transmission. These limitations are critical for remote monitoring and emergency response applications, where delayed decisions can be life-threatening. Congested networks or inefficient routing protocols further affect performance. Optimising data transmission and using edge computing can help reduce latency issues.

Regulatory Compliance: Healthcare IoT systems must comply with strict regulations of the Health Insurance Portability and Accountability Act (HIPAA), General Data Protection Regulation (GDPR), and other regional data protection laws. Managing patient data securely while adhering to legal frameworks is challenging, especially for cross-border data sharing. Non-compliance can lead to hefty penalties and loss of trust in healthcare institutions. Implementing secure data storage, encryption, and audit mechanisms is crucial for meeting regulatory standards.

Relevance of MQTT in Healthcare IoT

MQTT is widely adopted in Healthcare IoT due to its lightweight publish/subscribe model, which enables asynchronous and low-overhead communication between resource-constrained devices and central servers [7]. Its scalability and efficiency make it ideal for IoMT applications such as patient monitoring and cloud-based diagnostics.

However, MQTT lacks native encryption and access control support, exposing sensitive medical data to risks such as eavesdropping, spoofing, and replay attacks. Recent studies have highlighted vulnerabilities in MQTT-based IoMT systems and stressed the need for layered security solutions [83, 84]. Enhancing MQTT with fine-grained authorisation and lightweight encryption mechanisms is essential to meet the performance and security needs of Healthcare 4.0 environments.

2.2 Preliminaries of MQTT Protocol

This section provides a foundational overview of the MQTT protocol by highlighting its features in IoT healthcare scenarios and examining its desirable properties. The pub-

lish/subscribe (Pub/Sub) communication model forms the core messaging pattern of the lightweight MQTT protocol, making it particularly suitable for resource-constrained IoT environments.

In a Pub/Sub system, any IoT client or device can be a publisher, transmitting messages associated with specific topics. The MQTT broker plays a crucial role by decoupling the communication between publishers and subscribers: it prevents direct exposure between the two parties [85]. This decoupled architecture enhances scalability and flexibility, allowing secure and efficient data dissemination. The lightweight nature of MQTT, combined with the Pub/Sub model, makes it highly effective for various IoT applications, particularly in healthcare, as illustrated in Figure 2.4.

In healthcare contexts, implantable, wearable and remote sensors are often highly compact and constrained in processing power, battery life, and storage capacity [86]. The lightweight and scalable design of MQTT makes it a highly appropriate communication protocol for these devices.

2.2.1 Technical Aspects of MQTT Protocol over IoT

MQTT was initially designed to address the challenge of reliable data transmission over low-power and lossy networks [87]. It establishes data connectivity in low-bandwidth environments by adhering to operational principles that emphasise simplicity, efficiency, and flexibility. The protocol employs a publish/subscribe (Pub/Sub) communication model to minimise network overhead, making it significantly lighter than other IoT application-layer protocols [88].

Table 2.2 outlines the key characteristics that make MQTT a popular choice for various IoT applications, including intelligent vehicles, smart homes, telemedicine, and industrial systems. Its lightweight and efficient design renders MQTT particularly suitable for the healthcare IoT environment, where resource-constrained devices require low-bandwidth communication and minimal overhead to support timely data transmission. This reliable communication capability makes MQTT Pub/Sub a suitable candidate for critical healthcare applications where real-time responsiveness is essential to avoid life-threatening

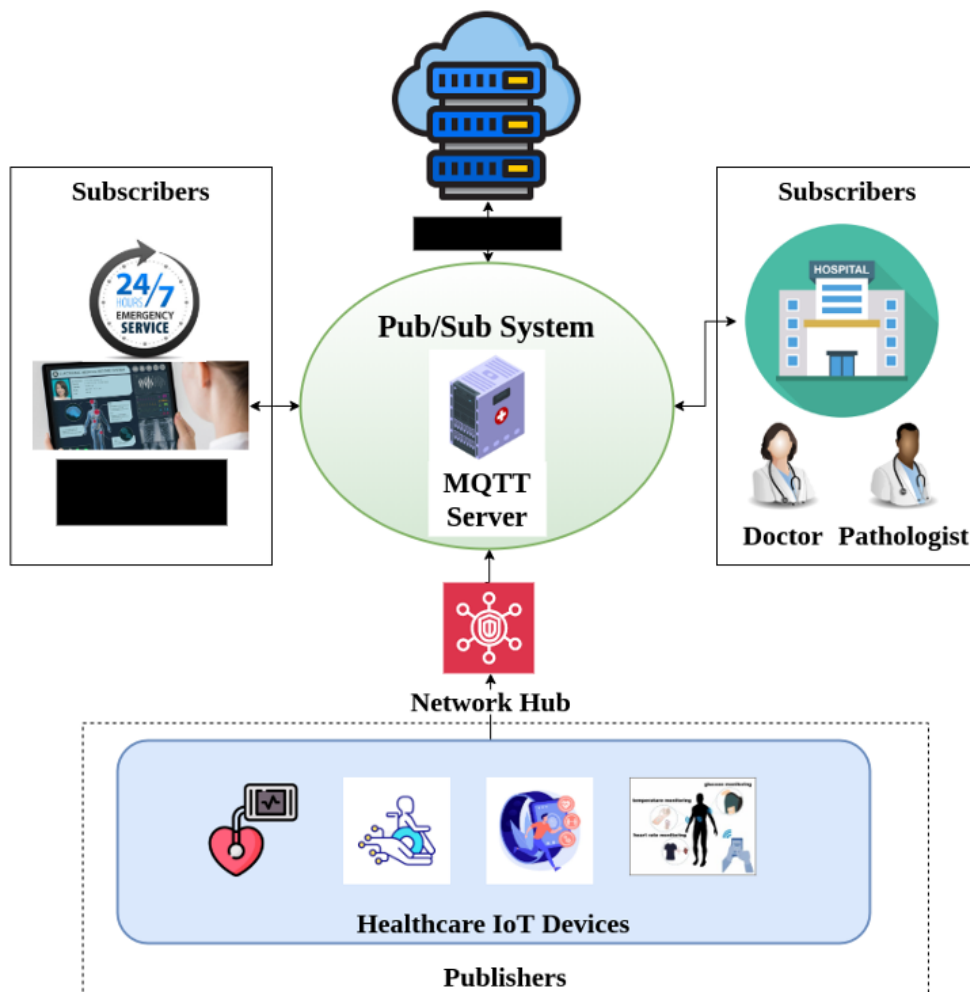


Figure 2.4: MQTT Pub/Sub model for IoT Healthcare Scenario

delays.

Table 2.2: MQTT Characteristics.

Characteristics	Description
Pub/Sub Model	Supports one-to-many message distribution, decoupling message producers from consumers [87].
Transport Mechanism	Provides content masking through a lightweight messaging mechanism [88].
Network Connection	Utilises Transmission Control Protocol/Internet Protocol (TCP/IP) for reliable network connectivity [88].
Reliability	Supports three Quality of Service (QoS) levels for message delivery: QoS 0, QoS 1, and QoS 2 [87].
Lightweight	Offers low-bandwidth transmission with minimal overhead (e.g., a fixed 2-byte header) and reduced protocol exchanges [89].
Notification Mechanism	Uses the Last Will and Testament (LWT) feature to notify both parties of unexpected disconnections [90].
Interoperability	Compatible with a wide range of platforms and devices [91].
Efficient Data Management	Supports data aggregation and filtering to reduce network load [92].
Payload Agnostic Features	Allows flexible transmission of various data formats [93].

MQTT optimises the publish/subscribe communication model and operates over TCP/IP to ensure reliable data transmission [88]. It is compatible with other TCP/IP-based application-layer protocols while offering significantly lower overhead [90]. Unlike CoAP, which follows a request/response model over User Datagram Protocol (UDP) [94], MQTT's Pub/Sub architecture offers greater reliability and flexibility, making it more suitable for secure multicast communication in IoT environments. This model also helps reduce bandwidth consumption and processing overhead, extending battery life for resource-constrained devices.

To ensure secure communications, MQTT brokers commonly support Secure Sockets Layer/Transport Layer Security (SSL/TLS)-based authentication, thereby enhancing data privacy and security on the Internet [87]. MQTT transmits messages as small data packets, which reduces latency, bandwidth usage, and energy consumption. It also supports various levels of Quality of Service (QoS), allowing applications to specify message delivery guarantees. Furthermore, MQTT supports persistent messaging, where messages

are retained on the broker even if the publisher disconnects. This feature is especially valuable in scenarios where subscribers must receive messages that were published during offline periods.

The MQTT protocol follows a mMachine-to-Machine (M2M) communication paradigm [87], exemplified by its scalable Pub/Sub architecture. It can connect thousands of clients to a single server (broker), which supports large-scale IoT deployments. The two primary components in MQTT communication are the client and the broker. The protocol has evolved to accommodate the increasing demands of modern IoT networks, particularly with regard to the timely delivery of messages and high reliability. Several versions of MQTT have been introduced to support these growing requirements:

- **MQTT v3.1 [95]:** The first widely adopted version, MQTT v3.1, was designed for resource-constrained environments. It employs a lightweight messaging structure based on the Pub/Sub model, in which publishers send messages on specific topics, and subscribers receive relevant messages through brokers. However, this version lacks built-in security mechanisms, offering only basic authentication (e.g., username and password). Additional layers, such as TLS, are required to ensure secure communication. Moreover, MQTT v3.1 was extensively used before its formal standardisation.
- **MQTT v3.1.1 [95]:** This version introduced minor improvements over v3.1 and was officially standardised by OASIS. The primary goal of v3.1.1 was to enhance interoperability and consistency across different MQTT implementations. Improvements in broker behaviour and protocol error handling helped standardise MQTT communication across vendors, making it more robust and widely adopted in industrial applications. Improves behaviour consistency across various implementations by resolving ambiguities in the MQTT v3.1 specification. It also introduced standardisation of client identifiers to enhance performance efficiency. However, this version retains basic security measures, relying on Transport Layer Security (TLS) for encryption and simple username/password authentication. Moreover, MQTT v3.1.1 lacks several advanced features, such as message expiration, extended sub-

scription options, and shared subscriptions, which are essential for more complex IoT environments.

- **MQTT v5 [93]:** This major update addresses many limitations of its predecessors and introduces a range of new features to enhance the protocol’s functionality, scalability, and adaptability to advanced IoT messaging scenarios. Key enhancements include the addition of custom metadata in messages, improved error reporting, expanded subscription options, support for request/response patterns, enhanced authentication mechanisms, and the use of topic aliases. Although MQTT v5 offers significant advantages for sophisticated IoT deployments, the added features introduce greater protocol complexity, which may be unnecessary or burdensome for simpler IoT applications. Furthermore, not all legacy MQTT clients and brokers support MQTT v5, potentially leading to interoperability challenges in heterogeneous IoT environments.

Table 2.3 presents a comparative analysis of the three versions of MQTT.

Table 2.3: Comparison of MQTT Protocol Versions.

Versions	MQTT v3.1	MQTT v3.1.1	MQTT v5
Year	2010	2014	2019
Standardization	Proprietary	OASIS Standard	OASIS Standard
Pub/Sub Model	Yes	Yes	Yes
QoS Levels	0, 1, 2	0, 1, 2	0, 1, 2
Retained Messages	Yes	Yes	Yes
Session Persistence	Limited	Limited	Enhanced
Message Expiry	No	No	Yes
Negative Acknowledgments	No	No	Yes
Reason Codes	No	Limited	Yes
Subscription Options	Basic	Basic	Advanced
Shared Subscriptions	No	No	Yes
Request/Response Pattern	No	No	Yes
Authentication	Basic	Basic	Enhanced-OAuth
Topic Aliases	No	No	Yes
Custom Properties	No	No	Yes
Complexity	Low	Low	High

Publisher and Subscriber Model

With the rapid increase in connected devices, managing the large volume of data generated in IoT environments presents significant challenges. Therefore, it is imperative to adopt efficient data communication paradigms. The Pub/Sub model is a widely recognised and practical approach that provides flexibility and scalability for data dissemination [96].

The Pub/Sub model is a robust communication paradigm that decouples information producers (publishers) from consumers (subscribers) [97]. This decoupling enables asynchronous and event-driven interactions between IoT components, facilitating precise and timely data exchange. In a healthcare context, publishers—typically smart medical devices—broadcast data on specific topics, while subscribers, such as healthcare professionals, receive only the information relevant to them by subscribing to those topics. Communication between publishers and subscribers is mediated by a broker, which efficiently transfers messages between the parties.

This architecture supports resilient communication even during network failures or dynamic changes in the IoT environment. Based on Pub/Sub principles, publishers and subscribers operate asynchronously, meaning neither must wait for the other to complete a transaction. This asynchronous design improves the responsiveness and performance of the system. Furthermore, the model supports communication between multiple publishers and subscribers without requiring direct interaction [98]. This high degree of decoupling ensures flexibility and scalability, which are essential for managing large-scale IoT deployments.

The Pub/Sub model applies highly to scenarios that require many interconnected devices, such as healthcare, smart cities, and industrial automation. It comprises three core components: the publisher, the broker, and the subscriber [99].

Publisher: Publishers are entities that send messages to specific topics. Monitoring devices such as wearables and implants act as publishers in healthcare applications. They are not required to know the identity or existence of subscribers. This anonymity and decoupling enhance scalability and system robustness.

Subscriber: Subscribers are individuals or systems that receive messages by sub-

scribing to topics of interest. Healthcare 4.0 includes doctors, diagnostic systems, and patient-monitoring platforms that subscribe to topics such as pulse rate, heart rate, or blood pressure. Subscribers can filter messages based on topic relevance, improving overall system efficiency and reducing unnecessary data processing.

Message Broker: The message broker is the central component of the Pub/Sub architecture. Manages the transmission of messages from publishers to subscribers, ensuring efficient routing and delivery. Brokers also handle additional responsibilities such as message persistence, quality of service management, and connection state tracking to provide reliable and ordered communication.

Topics: Topics act as logical channels that group related messages. This abstraction enables publishers and subscribers to operate independently, without the need to know each other's identities or operational details. Figure 2.5 illustrates the architecture and communication workflow of the Pub/Sub model.

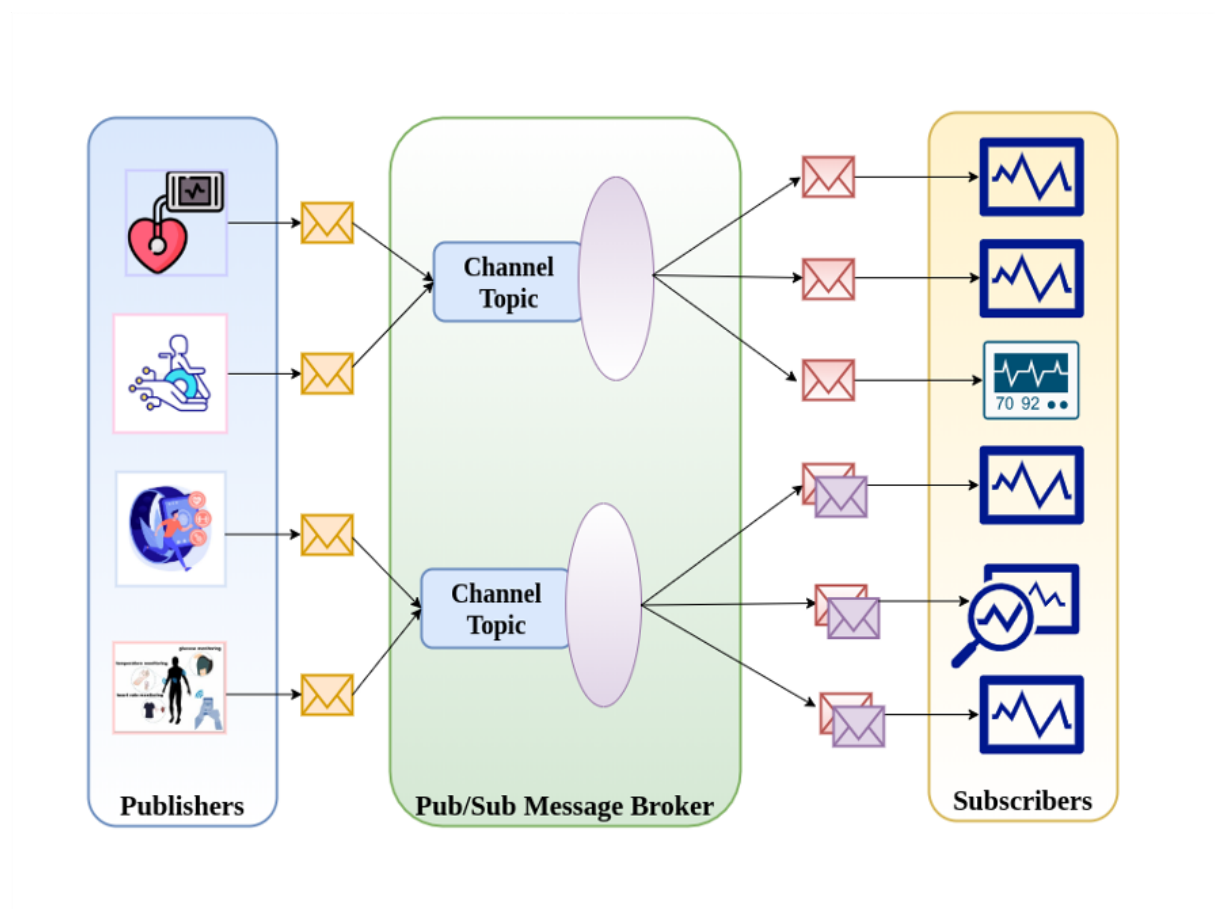


Figure 2.5: The Pub/Sub Architecture

The communication process within the Pub/Sub model typically follows these steps:

- Step 1: The publisher creates a message, which may include various data types, such as text, binary data, or image-based events. In healthcare, messages can contain patient data, such as pulse, heart rate, or blood pressure.
- Step 2: The publisher transmits the message to a specific topic at defined intervals.
- Step 3: The message broker receives and stores the message.
- Step 4: One or more subscribers subscribe to the topic. In healthcare, these might include doctors or automated monitoring systems.
- Step 5: The message broker forwards the message to all relevant subscribers.
- Step 6: Subscribers receive and process the message, enabling timely interventions or responses based on patient data.

MQTT Broker

The MQTT broker is a critical component of the protocol [100], functioning as the central node that manages communication among publishers, subscribers, and IoT devices within the network. It serves as an intermediary that receives messages from publishers and routes them to subscribers based on their topic subscriptions.

The broker plays a vital role in ensuring message delivery reliability, system efficiency, and communication security - all essential for MQTT-based operations in healthcare IoT. MQTT v5, the most recent protocol version, introduces advanced features that enhance these capabilities compared to previous versions, such as MQTT v3 and v3.1.1. Several well-established MQTT v5 brokers are available, including:

Eclipse Mosquitto [101]: Eclipse Mosquitto is a widely used open-source MQTT broker that supports MQTT v5 and earlier versions (v3.1.1 and v3.1). It is lightweight and particularly suitable for resource-constrained devices and full-scale server deployments. Mosquitto offers comprehensive support for advanced MQTT v5 features such as user-defined properties, reason codes, and session expiration. By ensuring reliable message delivery with different QoS levels, such as QoS 0, QoS 1, and QoS 2, the Mosquitto broker is well-suited to Healthcare IoT environments. It also supports integration with

TLS, ensuring secure communication.

HiveMQ [102]: HiveMQ is a commercial MQTT broker designed for high-performance, scalability and security deployments. It supports the full range of MQTT v5 features, making it particularly appropriate for enterprise-grade applications. These features include shared subscriptions, enhanced authentication options, and topic aliases. HiveMQ also offers clustering capabilities to ensure high availability and scalability through horizontal expansion.

EMQX [103]: EMQX is an open-source MQTT broker engineered for high scalability and reliability. It supports MQTT v5 and can manage millions of concurrent IoT connections, making it ideal for large-scale IoT deployments. EMQX offers comprehensive MQTT v5 functionality, including advanced session management, improved authentication, and support for reason codes. Its distributed architecture with clustering enables effective load balancing, and its plugin system allows integration with other protocols and services.

VerneMQ [104]: VerneMQ is another open-source, distributed MQTT broker that supports the MQTT v5 protocol. It is designed for environments requiring high availability and scalability, especially within Industry 4.0 applications. VerneMQ supports all key MQTT v5 features such as persistent message transfer, session expiration, and user-defined properties. Its strengths include horizontal scalability, robust clustering, load balancing, fault tolerance, and uninterrupted service delivery.

AWS IoT Core [105]: AWS IoT Core is a fully managed MQTT broker service offered by Amazon Web Services. Supports multiple IoT protocols, with particular emphasis on MQTT v5. This service allows seamless integration with the broader AWS ecosystem, including the AWS Lambda, DynamoDB, and S3 services. AWS IoT Core provides enhanced error reporting, shared subscriptions, and automatic scaling to support millions of IoT devices. It is ideal for constructing complex and scalable IoT workflows.

Table 2.4 presents a comparative overview of various MQTT brokers and their characteristics.

Table 2.4: Comparison of Various MQTT brokers.

Various Brokers	Type of Deployment	Scalability	Clustering Support	Security Algorithm	Use Cases	Cost	Commercial Support
Eclipse Mosquitto	Medium	On-premises	Constrained	Basic TLS	Small and medium-scale IoT deployments	open-source and Free	No
HiveMQ	High	Cloud, On-premises	High	Advanced TLS, OAuth2, and SSO	Large-scale IoT deployments	Payable	Yes
EMQX	Very High	Cloud, On-premises	High	Advanced TLS, OAuth2, and JWT	Massive IoT deployments	Open-source and Free	Yes
VerneMQ	High	On-premises	High	Advanced TLS, OAuth2, and JWT	Large-scale IoT deployments	Open-source and Free	Yes
AWS IoT Core	Very High	Cloud	Depending on AWS	Integrated with AWS IAM	Large-scale and cloud-based IoT	Payable	Yes

Topic Hierarchy and Filters

In MQTT v5, topic hierarchies and filters play a critical role in structuring and managing the flow of messages between publishers and subscribers.

Topic Hierarchy [93]: The structured use of topics is fundamental to effectively routing messages. MQTT topics serve as addressing mechanisms, ensuring messages are delivered only to interested subscribers. Each published message is associated with a specific topic, and subscribers receive messages by subscribing to relevant topics. The key features of topic hierarchies include the following:

- **Slash separator [93]:** MQTT uses forward slashes (“/”) to define hierarchical topics, allowing efficient organisation and categorisation of data.
- **Structure Hierarchy [93]:** Topics can be arranged in a tree-like structure. For example, in healthcare, a broad topic like `healthcare/patient` may branch into subtopics such as `healthcare/patient/heart-rate` or `healthcare/patient/blood-pressure`. This hierarchy helps to navigate complex information by breaking it down into more manageable ways and logically relating it to different groups.
- **Topic Naming Flexibility [106]:** MQTT allows clients to define both primary and nested subtopics according to their specific use cases. Since MQTT does not impose any predefined topic structures, this flexibility is essential for segmenting large datasets into meaningful levels for efficient organisation and retrieval.
- **No Fixed Levels [106]:** The MQTT topic hierarchy does not enforce fixed topic levels. Publishers and subscribers can define as many levels as necessary to suit application-specific needs, enhancing adaptability and scalability.
- **Efficient Message Transfer [107]:** The hierarchical structure enables MQTT brokers to efficiently route messages to subscribers based on topic matching, reducing unnecessary traffic and improving system responsiveness.

The dynamic management of topic hierarchies typically involves three primary operations: creation, modification, and maintenance. These are influenced by system evolution, changes in data structure, integration of new devices, and service adaptation to changing user or system requirements.

Topic Filters [108]: Topic filters are pattern-matching mechanisms subscribers use to specify topics of interest. These filters support wild-card characters, allowing a single subscription to cover multiple related topics. MQTT supports the following wild-card types:

- **Single-Level Wildcard (+):** This wild-card matches exactly one level in a topic hierarchy. For example, the filter `healthcare/+/heart-rate` would match topics such as `healthcare/patient1/heart-rate` and `healthcare/patient2/heart-rate`, regardless of the intermediate level.
- **Multi-Level Wildcard (#):** This wild-card matches one or more levels in a topic structure. For instance, the filter `healthcare/#` would match all topics starting with `healthcare`, such as `healthcare/patient1/heart-rate`, `healthcare/patient2/blood-pressure`, etc. It is beneficial to subscribe to broad topic groups.
- **Hybrid Use of Wildcards:** Filters can combine single- and multi-level wildcards to create powerful and flexible subscription rules. This feature is especially advantageous in complex IoT scenarios with diverse data sources, allowing efficient monitoring with minimal subscription overhead.

Broad monitoring scenarios benefit significantly from hybrid wild-card filtering, where the entire topic structure is unknown or dynamically changing. Although single-level and multi-level filters serve different purposes, both lack the specificity sometimes required in complex IoT applications [93]. Hybrid filtering combines the strengths of both mechanisms, allowing for granular yet flexible subscription rules across diverse topic structures.

Topics and Subscriptions

In a healthcare setting, users may wish to subscribe to different types of patient monitoring data through a web-based interface. However, in MQTT communication, clients do not subscribe to a user-facing data stream, but rather to a specific “topic” used internally by the MQTT server to route messages. All MQTT messages are published on topics that act as logical message identifiers.

Each MQTT message comprises three parts: the Fixed Header, the Variable Header,

and the Payload [109]. Figure 2.6 illustrates how the MQTT server (broker) coordinates client communication using topic-based message routing.

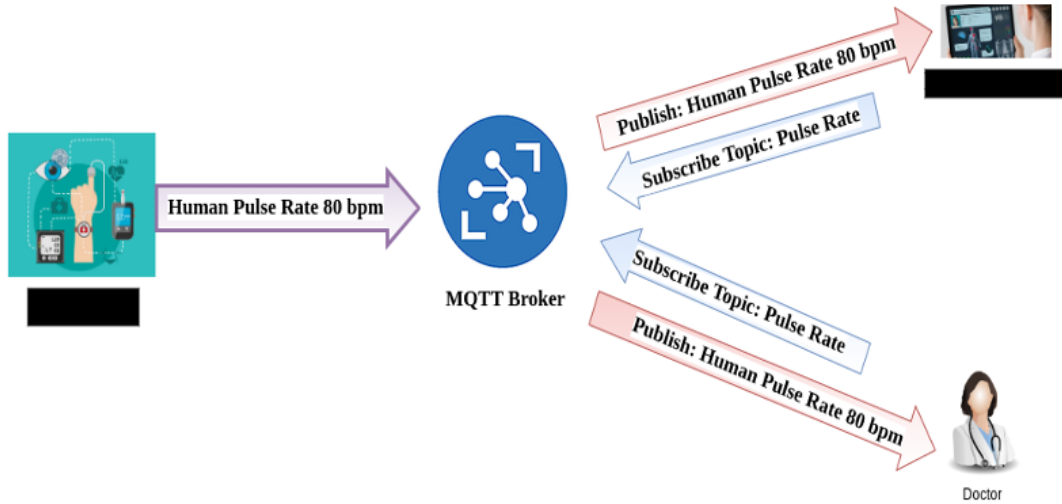


Figure 2.6: MQTT Communication Scenario for Healthcare

This example involves three clients: one publisher, two subscribers, and one broker. The broker uses topics to manage communication between them [110]. To obtain a patient's pulse rate using a mobile phone and a monitoring system, subscribers subscribe to the topic "Pulse Rate" on the broker. When the subscriber posts a message to the broker's "Pulse Rate" topic, the broker checks which clients have subscribed to this topic. If it finds that the subscribed clients are mobile phones or monitoring systems, the broker forwards them the "Pulse Rate" message received from the publishers. In this example, healthcare sensors are the publisher of the topic "Pulse Rate", while mobile and monitoring systems act as subscribers. The core hub of MQTT communication is the MQTT broker. With the server receiving, storing, processing, and sending MQTT information, clients can be independent of each other when publishing and subscribing to information. They can be spatially separated and temporally asynchronous.

Quality of Service

Quality of Service (QoS) is a fundamental feature of MQTT, as it determines the reliability and assurance level for message delivery between publishers and subscribers [87]. MQTT

supports three QoS levels: QoS 0, QoS 1, and QoS 2 [87]. Each level represents a different trade-off between reliability and resource use. The QoS level is specified when a message is published, and subscribers can also indicate their preferred QoS level during subscription.

QoS 0 - At Most Once Delivery: QoS 0 is the lowest QoS level in MQTT and provides “at most once” messaging [88]. At this level, messages are sent from the publisher to the broker only once and are not tracked by the broker. The broker immediately delivers the message to the subscriber without acknowledgement or further assurance. QoS 0 is suitable for scenarios where message loss can be tolerated, such as general information updates or sensor data transfers. Since there is no acknowledgement or tracking, messages can be quickly published and forwarded to subscribers. Resource utilisation is minimised, as intermediaries do not store or track messages. With many publishers and subscribers, minimising resource utilisation is advantageous, and status updates for connected devices, where occasional missed messages are acceptable.

QoS 1 - At Least Once Delivery: QoS 1 guarantees that a message will be delivered at least once [88]. The publisher sends a message to the broker and then forwards it to the subscriber. Upon successful delivery, the broker sends a PUBACK message back to the publisher [111, 112]. The broker will retransmit the message until confirmation is received if no acknowledgement is received. Although this ensures message delivery, it may result in duplicate messages if acknowledgements are lost. QoS 1 is suitable for applications where message loss is unacceptable, but duplicate messages can be handled, such as emergency alerts, alarm systems, or remote device control.

QoS 2 - Exactly Once Delivery: QoS 2 provides the highest level of reliability in MQTT, ensuring that each message is delivered exactly once [88]. This is achieved through a four-step handshake process between the broker and the subscriber. First, the publisher sends a message to the broker. The broker then acknowledges receipt and begins a controlled exchange with the subscriber to ensure that the message is neither lost nor duplicated. This level is essential for applications requiring strict message integrity, such as financial transactions or mission-critical medical telemetry. Although QoS 2 incurs the highest communication overhead, it provides maximum delivery assurance. QoS 2 ensures that messages are delivered exactly once, avoiding duplication or loss [113].

The handshake mechanism between the broker and the subscriber guarantees successful message delivery - an essential requirement for applications demanding consistent system state or synchronisation across distributed environments.

Table 2.5 compares the different QoS levels in MQTT, highlighting their respective guarantees, advantages, and use cases.

Table 2.5: Comparison of QoS Levels Offered by MQTT.

QoS Level	Delivery Guarantee	Advantages	Use Cases
QoS 0	At most once	Fast delivery with minimal overhead	Non-critical updates, low-priority telemetry, highly scalable systems
QoS 1	At least once	Reliable delivery using acknowledgement mechanisms	Critical alerts, healthcare monitoring, industrial automation
QoS 2	Exactly once	End-to-end delivery guarantee with full reliability	Financial transactions, distributed logging, mission-critical telemetry

MQTT Control Packet Format

The MQTT control packet format supports device-to-device communication through the MQTT broker. This section explores its structure, packet types, and associated functionality. The design is intentionally lightweight to accommodate the bandwidth and resource limitations of constrained IoT devices [114].

Each MQTT packet consists of three components: a fixed header, an optional variable header, and a payload.

Fixed Header: The fixed header is mandatory and always present. It is at least 2 bytes in size and contains the following fields:

1. **Packet Type:** Indicates the type of MQTT control packet (e.g., CONNECT, PUBLISH, SUBSCRIBE).
2. **Flags:** Represents QoS level and duplication status (DUP flag).
3. **Remaining Length:** A variable-length field that specifies the total length of the variable header and payload.

Variable Header: This header is optional and appears only for specific packet types. It contains metadata such as topic name, packet identifier (message ID), and optional fields such as username and password. [93].

Payload: The payload section carries the actual application message or data that is being transferred. Its structure and presence depend on the type of packet. For example, in a CONNECT packet, the Payload includes client identification, while in a PUBLISH packet, it contains the message content. The Payload can consist of any form of data, such as text, binary, or JSON.

The MQTT protocol defines various control packets that enable flexible and reliable communication between clients and brokers. These include: CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP, SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK, PINGREQ, PINGRESP, and DISCONNECT [111]. Each packet type serves a specific function to support lightweight and reliable messaging. For example, the CONNECT packet initiates a connection between the client and the broker, while the CONNACK packet provides the broker's response to this request.

Table 2.6 summarises the primary types of MQTT packets and their roles in client-broker interactions.

Figures 2.7, 2.8, and 2.9 illustrate the structure of CONNECT and CONNACK control packets.

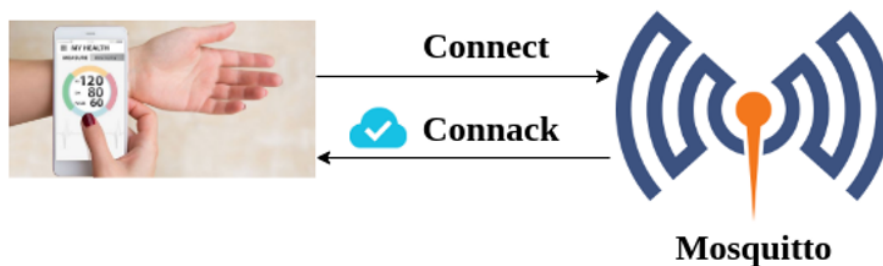


Figure 2.7: MQTT CONNECT & CONNACK

Figures 2.8 and 2.9 illustrate the information contained in the CONNECT and CONNACK packets. Each packet contains multiple fields essential for establishing a client-

Table 2.6: Packet Types in MQTT.

Packet	Description
CONNECT	Used to establish a connection with a broker.
CONNACK	Acknowledgement of a CONNECT packet.
PUBLISH	Used to publish a message on a topic.
PUBACK	Acknowledgement of a PUBLISH packet.
PUBREC	Acknowledge the broker's receipt of the PUBLISH packet to the publisher.
PUBREL	Used to release the PUBLISH packet by the publisher to the broker.
PUBCOMP	Acknowledge the broker's receipt of the PUBREL packet to the publisher.
SUBSCRIBE	Used to subscribe to a topic.
SUBACK	Acknowledgement of a SUBSCRIBE packet.
UNSUBSCRIBE	Used to unsubscribe from a topic.
UNSUBACK	Acknowledgement of a UNSUBSCRIBE packet.
PINGREQ	Used to send a heartbeat to the broker.
PINGRESP	Acknowledgement of a PINGREQ packet.
DISCONNECT	Used to disconnect from the broker.


MQTT-Packet:	
CONNECT	
contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

Figure 2.8: CONNECT

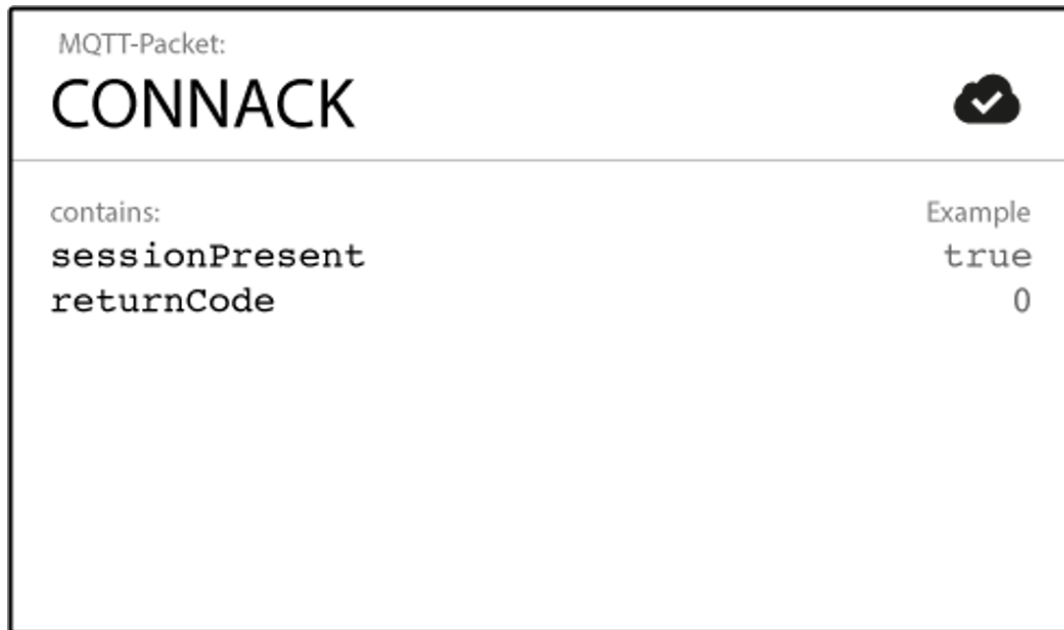


Figure 2.9: CONNACK

broker session [115]. In Figure 2.8, the left column presents the names of the CONNECT packet fields, while the right column displays the corresponding values. For example, the CONNECT packet shown contains a field labelled `ClientId`, with the value “client-1”. The `ClientId` is a unique identifier that MQTT brokers use to distinguish clients. If two clients attempt to connect using the same `ClientId`, the broker assumes that they represent the same client.

Session Clean and Connection Reliability

Session cleaning [116] is a feature that enables the broker to determine how to handle undelivered messages. When a client acknowledges a message, this confirmation allows the broker to discard the message safely. However, if no acknowledgement is received, the broker assumes that the message was not successfully delivered and takes remedial actions:

1. Stores the undelivered message for future transmission.
2. Retransmits the message and waits for an acknowledgement from the client.

As illustrated in Figure 2.8, the CONNECT packet includes a `cleanSession` flag,

which can be set to **True** or **False**. If set to **True**, the broker will not retain session data or undelivered messages when the client disconnects. This setting is appropriate for stateless communication scenarios. If set to **False**, the broker stores the state of the session, including the subscription details and any messages that were not delivered successfully. This is critical when message reliability and security are essential. Users who select **cleanSession=False** should also consider using a QoS level above 0 to ensure delivery guarantees.

KeepAlive [117]: In MQTT, the KeepAlive mechanism ensures that both the broker and the client are aware of the status of the connection with each other. Due to potential network instability, the broker cannot always confirm whether a client is still active. The KeepAlive timer enables periodic verification. If no communication occurs within the KeepAlive interval, the client sends a **PINGREQ** message. If the connection is intact, the broker replies with a **PINGRESP** message [118].

In Figure 2.8, the **KeepAlive** interval is set to 60 seconds. If no data is exchanged within this time, the client initiates a ping to the broker. This lightweight heartbeat mechanism helps maintain the session state and detect disconnections in a timely manner, in response to the **PINGREQ** sent by the client. Conversely, if the client does not receive a **PINGRESP** from the broker, it assumes that the connection has been lost. Furthermore, when the broker does not receive any packet from the client within 1.5 times the **KeepAlive** interval, it considers the client disconnected.

SessionPresent [93]: This flag indicates whether the broker has an existing session associated with the client. If **cleanSession=False** and the client reconnects, the broker sets **SessionPresent=True** to indicate that the previous session state has been resumed.

ReturnCode [93]: The settings the broker sends to the user's **sessionPresent** are determined. Table 2.7 summarises the possible values and their meanings.

When **cleanSession** is set to “True”, the broker determines **sessionPresent** to be “False”, which means that the broker will not save any information. In contrast, when **cleanSession** is “False”, the corresponding **sessionPresent** is “True”. In addition to these two examples, each packet type in MQTT follows a specific structure and provides different functionality to support various communication requirements.

Table 2.7: Table of ReturnCode Values.

Value	Meaning
0	Connection accepted successfully.
1	Connection refused: unsupported protocol version. The broker does not support the MQTT version requested by the client.
2	Connection refused: identifier rejected. The client identifier is malformed or not accepted.
3	Connection refused: server unavailable. The broker is currently unable to accept connections.
4	Connection refused: bad username or password. The authentication credentials provided by the client are invalid.
5	Connection refused: not authorised. The client does not have permission to connect to the broker.

Message Retaining

A retained message is defined as a message sent with the “retain” flag set to true [119]. The MQTT broker stores this retained message and delivers it to new subscribers who subscribe to the associated topic [119]. Publishers set the retain flag to true when publishing a message, prompting the broker to store it as the latest retained message for the specified topic. This feature immediately ensures new subscribers receive the most recent retained message, supporting high availability and initialisation with up-to-date state data.

For effective message handling, publishers can also send a retained message with an empty payload, instructing the broker to clear any retained message associated with that topic. This mechanism helps to reduce storage overhead on the MQTT broker. The retained message feature works seamlessly with all QoS levels, with delivery reliability aligned to the QoS level negotiated between publisher and subscriber.

Although this feature improves usability and reliability—especially in applications where the latest status must always be known—it is essential to carefully manage retained messages to prevent unnecessary broker storage usage and ensure only relevant data are retained.

2.2.2 Technical Challenges of MQTT Pub/Sub Implementation

Although the MQTT Pub/Sub model offers many benefits, it faces several technical challenges, including network orchestration and scalability, security and confidentiality, cross-platform integration, standardisation, computational cost, latency, and real-time processing constraints [14].

Network Orchestration and Expansion

Scalability is a key requirement for modern IoT systems [120]. However, as the number of connected devices and data sources increases, the MQTT architecture may struggle to efficiently deliver messages to all relevant subscribers [121]. This can lead to high routing, filtering, and delivery costs, degradation of performance, and increased latency.

To mitigate these limitations, distributed communication architectures such as mesh networks and hierarchical IoT infrastructures have been proposed [122]. These decentralised models distribute the communication load across multiple nodes, improving efficiency and scalability.

Information Security and Confidentiality

Security is a critical concern in IoT systems due to the sensitivity of the data and the risk of unauthorised access or attacks. In the MQTT decoupled Pub / Sub architecture, the broker becomes a central point of vulnerability, potentially subject to disruption or data interception [123]. Moreover, MQTT does not include built-in encryption for message payloads, which can compromise data confidentiality and integrity [124].

Although end-to-end authentication is supported, the lack of default encryption requires additional protective mechanisms. Secure communication protocols (e.g., TLS), access control, and encryption schemes must be implemented carefully to ensure the confidentiality and security of IoT data transmissions.

Cross-Platform Integration

IoT ecosystems involve various devices and protocols, including Zigbee, Bluetooth, Wi-Fi, LoRa, and 4G/5G cellular networks [125]. Achieving seamless communication across such heterogeneous systems is challenging due to the absence of universal standards.

Furthermore, IoT data are often produced in varying formats (e.g., JavaScript Object Notation (JSON), Extensible Markup Language (XML), binary) [126]. Effective integration requires robust middleware solutions capable of translating and standardising these formats. In addition, many application domains impose proprietary interfaces or unique requirements, further complicating interoperability, particularly in real-time environments such as healthcare or industrial automation.

Standardisation

Numerous organisations have developed IoT standards—such as Institute of Electrical and Electronics Engineers (IEEE) [127], Internet Engineering Task Force (IETF) [128], Open Mobile Alliance (OMA) [129], and International Organization for Standardisation (ISO) [129]. However, coordination among these bodies is essential for interoperability across IoT platforms.

Competing standards and sector-specific requirements hinder the adoption of unified frameworks [130]. Furthermore, regulatory discrepancies between regions complicate the deployment of global IoT networks, especially concerning data protection, frequency allocation, and device certification.

Computation Cost

Although the Pub/Sub model simplifies communication by decoupling publishers and subscribers, it introduces complexity in configuration and maintenance [131]. Defining topics, registering devices, and configuring access control policies can be time-consuming [132].

The computational burden of handling message routing, filtering, and topic management increases as the network scales. Consistent performance requires efficient broker

implementations and possibly offloading tasks to distributed edge nodes.

Latency and Real-time Processing

The asynchronous nature of the MQTT Pub/Sub model introduces inherent latency [133]. Although tolerable for general applications, this delay becomes problematic for time-critical systems requiring immediate responsiveness.

For such scenarios, hybrid architectures combining Pub/Sub with request/response or stream processing paradigms have been proposed [134]. These facilitate real-time decision making through direct interaction and low-latency processing.

However, MQTT message delivery is not guaranteed to preserve publication order. In systems where message sequencing is critical, additional mechanisms must be implemented to maintain the integrity of temporal data flows.

2.3 Security of MQTT

From a security perspective, MQTT requires significant improvements beyond its default protocol design to address vulnerabilities in various communication attacks [135]. Attackers may intercept MQTT topics and messages exchanged between the publisher and the broker or between the broker and subscribers. Several techniques have been proposed to protect against such threats, including using SSL/TLS channels as complementary security layers [136].

TLS-based authentication prevents attackers from initially impersonating publishers or subscribers. However, this mechanism does not guarantee protection for subsequent communications once a connection is established. Privacy preservation is another primary concern in MQTT communication, especially in time-sensitive and life-critical healthcare applications [137].

Traditional MQTT security solutions often assume a high level of trust in data storage, which can lead to privacy breaches if malicious devices leak sensitive IoT data. Although secure channels, digital certificates, and key management can be employed to secure MQTT communications, these mechanisms pose challenges in IoT environments due to resource constraints. The storage and handling of certificates, as well as key-exchange

procedures, are often computationally intensive. Moreover, the secure communication channel itself can be susceptible to encryption attacks such as Browser Exploit Against SSL/TLS (BEAST) and Compression Ratio Info-Leak Made Easy (CRIME) [138].

2.3.1 MQTT Security Requirements

Despite the inherent simplicity of MQTT, implementing an additional security layer is essential to ensure safe communication among entities. Core security requirements, such as confidentiality, authentication, authorisation, access control, availability, and integrity, are critical for protecting sensitive data in healthcare IoT applications [139].

Advanced technologies enable continuous patient monitoring and help healthcare professionals make diagnoses and treatment decisions. However, without adequate security, the communication process between these devices can expose critical data to potential threats.

Malicious actors can exploit IoT devices. For example, adversaries or cybercriminals can use, expose, or modify users' personal or sensitive information through compromised devices. Therefore, securing IoT devices against unauthorised cyberattacks has become a critical concern in developing IoT systems [140].

Based on hardware architecture and communication patterns, Zhang et al. [141] categorised the root causes of IoT security vulnerabilities into two main groups. This classification comes from IoT devices designed according to various environmental constraints, including form factor, size, functionality, sensor type, and network connectivity. Consequently, managing and securing such a heterogeneous ecosystem is complex and challenging.

For example, there remains a lack of standardised regulation and clear business models in the IoT domain [142]. Mandal et al. [143] introduced the "IoT Security Dataset", which includes 7147 samples focused exclusively on IoT security discourse. Their analysis revealed that 68% of these samples did not meet the definition of actionable practices, and 73% of the actionable recommendations were related to the software development lifecycle phase. These findings underscore the critical role of manufacturers and developers in

ensuring robust IoT security.

Encryption and Data Confidentiality

In Pub/Sub communications, private user data must be encrypted to protect against unauthorised access or tampering [144]. Encryption is fundamental to ensure data confidentiality and integrity in secure IoT architectures. Despite its importance, many IoT devices and protocols lack proper encryption due to resource limitations and network design constraints.

Recent studies focus on integrating lightweight yet robust encryption algorithms and key management schemes to protect sensitive payloads in MQTT messages, ensuring that user data remains secure throughout the publishing and subscribing processes.

Authentication

MQTT employs authentication mechanisms that typically involve clients providing a username and password during connection requests to the broker [145]. The broker verifies these credentials using an external authentication service or an internal user database.

Advanced frameworks such as OAuth 2.0 have also been adopted. OAuth 2.0 enables users to authorise third-party applications to access their resources without directly sharing log-in credentials [146]. An external authorisation server issues access tokens upon successful authentication in such a setup. MQTT can be integrated with OAuth 2.0 to validate these tokens before allowing access to its resources.

MQTT brokers may validate tokens using custom authentication plugins or extensions [147]. These custom strategies can address the specific security requirements of particular applications. However, they require appropriate support from both the broker and the client. Implementing such strategies involves modifying the broker to support token-based validation and ensuring clients can generate or handle authentication tokens accordingly.

Commonly used authentication mechanisms include X.509 digital certificates for mutual authentication [148], Token Authentication schemes [149], and the Salted Challenge Response Authentication Mechanism (SCRAM) technique [150].

Non-Reputation and Integrity

Non-repudiation ensures that an entity involved in a communication cannot deny having sent or received a message [151]. This property is critical in establishing accountability within IoT systems. Digital signatures are a primary technique used to achieve non-repudiation. In this process, the sender signs the message using their private key, and the recipient verifies it using the corresponding public key. This mechanism proves the origin and integrity of the message.

Non-repudiation mechanisms often include audit logs containing timestamps, user identities, and message traces. However, MQTT does not natively support digital signatures in its protocol [152]. Thus, it is necessary to implement such mechanisms at the application level. For example, publishers may attach digital signatures to messages, which subscribers can verify upon receipt. MQTT brokers can also log message transactions and client connections to create an audit trail that supports non-repudiation.

Integrity ensures that a message has not been altered during transmission. Hashing techniques are commonly used to verify integrity [153]. A hash value is computed for the original message, which is transmitted along with the message. Upon receipt, the receiver recomputes the hash and compares it with the received hash to verify the integrity of the data [154].

Non-repudiation and integrity are critical for ensuring secure and trustworthy communication in MQTT-based IoT environments.

Authorisation and Access Control

Establishing robust authorisation and access control remains a significant challenge in MQTT-enabled systems. In resource-constrained IoT environments, conventional authorisation strategies often fail to satisfy the fundamental requirements for secure communication. This inadequacy exposes users to identity theft and unauthorised access.

Effective authorisation mechanisms help mitigate such issues by ensuring publishers and subscribers possess valid and legitimate identities. The MQTT security framework supports a variety of access control models, as summarised in Table 2.8. These include

Access Control List (ACL) [155], Role-Based Access Control (RBAC) [156], Topic-Based Access Control (TBAC) [157], Attribute-Based Access Control (ABAC) [158], and Policy-Based Access Control (PBAC) [159].

Each access control policy offers unique advantages, and the choice of mechanism depends on specific design requirements. Using different access control models improves the flexibility, scalability, and adaptability of security frameworks for various MQTT-based IoT applications.

Table 2.8: Comparative Analysis of ACL, RBAC, TBAC, ABAC, and PBAC.

Factors	ACL	RBAC	TBAC	ABAC	PBAC
Type	User-based	Role-based	Topic-based	Attribute-based	Policy-based
Granularity	Low	Moderate	Moderate	High	High
Flexibility	Low	Moderate	Low	High	High
Scalability	Low	High	Moderate	High	High
Real-time Adaptability	Low	Moderate	Low	High	High
Context-Awareness	Nil	Low	Low	High	High
Complexity	Low	Moderate	Low	High	High
Management	Direct	Role-based	Topic-based	Attribute-based	Policy-based
Examples	HTTP	LDAP	MQTT	XACML	XACML
Use cases	Small networks	Enterprise Networks	Pub/Sub IoT environment	IoT and smart grids	Federated IoT environment

2.3.2 MQTT Security Threats

Since most medical data collected from patients is highly private, it is crucial to ensure data security during transmission, storage, and processing [160]. This section discusses the impact of data breaches on IoT security and the possibilities and challenges of solving data breaches. Chin et al. [161] provided a comprehensive overview of the security threats to data information in the Intelligent Grid based on the IoT. Various security threats affect the integrity and reliability of data in intelligent IoT-based infrastructures. In the smart grid context, data breaches and network attacks can compromise domestic energy consumption patterns and the integrity and availability of services by modifying

critical data. Unauthorised access and interference threaten both data confidentiality and operational reliability.

Researchers have proposed various defensive mechanisms to address these challenges, including threat simulations and attack surface analysis from an adversarial perspective. Furthermore, implementing layered data protection strategies is essential to secure IoT devices against such intrusions.

Meng et al. [162] analysed the architecture of intelligent home platforms and highlighted security issues inherent to consumer-focused Internet architectures, such as unauthorised access and spoofing. These vulnerabilities can negatively impact user experience and cause financial losses. To mitigate these risks, the authors recommended network encryption, intrusion detection systems, and continuous monitoring.

Doss et al. [163] emphasised the importance of data integrity in IoT development. Their findings show that improved authentication methods play a vital role in preserving the confidentiality and authenticity of user data. As security risks in IoT are multidimensional, a comprehensive approach involving multiple protective layers is necessary. Siby et al. [164] noted that privacy threats pervade many technological aspects of the IoT, including network communications. Management of IoT security remains difficult due to the heterogeneity and scale of connected devices.

Several researchers have proposed robust policy frameworks integrating encryption algorithms, secure network protocols, authentication schemes, and access control mechanisms to counteract these vulnerabilities. For example, Moin et al. [165] presented a blockchain-based solution to improve IoT security. Salman et al. [166] introduced an identity-based authentication protocol tailored for heterogeneous IoT environments. Tu et al. [167] used a Q-learning algorithm to determine optimal thresholds for impersonation attack detection. Nakhodchi et al. [168] studied threats such as malware and data breaches in smart agriculture and stressed the need for secure data collection and transmission. Torky and Hassanein [169] proposed a blockchain-based solution to address security issues in precision agriculture. The approach tracks and protects data throughout its lifecycle, from collection to application, ensuring integrity and trust. In addition, authentication and data encryption technologies are employed to safeguard the confidentiality

and reliability of user and system data.

Hussain et al. [170] presented a security framework tailored for real-time health monitoring systems that use both the CoAP [171] and MQTT protocols. Their solution integrates encryption, authentication, access authorisation, and audit mechanisms to ensure data integrity, confidentiality, and availability from edge devices to cloud infrastructures. The widespread adoption of SSL/TLS protocols provides additional protection during data transmission against threats such as interference, tampering, and eavesdropping [172].

This thesis focuses on designing a security framework grounded in the Pub/Sub communication model. Consequently, robust encryption algorithms and fine-grained access control policies are critical to securing information exchange. Furthermore, incorporating modules to identify adversaries and detect attack patterns will strengthen the resilience of MQTT. Finally, due to the diversity and heterogeneity of IoT devices, data breaches can originate from various sources [173]. Therefore, designing a secure communication scheme requires a holistic approach that balances performance with multi-layered security.

Denial-of-Service

The act of a Denial-of-Service (DoS) attack [174] involves inundating a network or server with traffic from various sources, ultimately resulting in failure. In the IoT environment, DoS attacks can scale from a single target to a massive attack against an entire system. Due to their interconnection, IoT devices are often the prime targets of DoS attacks. IoT devices are widely used in industries such as healthcare and transportation, where handling sensitive and confidential information is a common practice. Attacks targeting these areas can cause significant disruptions to system services and are considered highly sensitive. The extent of damage and the impact could cause considerable damage to the development of society. For example, if an attacker makes a power network or transport system unavailable, this can bring an entire city to a standstill or blackout, causing people to panic. Therefore, researchers must consider DoS attacks when researching techniques and security schemes for IoT security.

Volumetric Attack [175], Transmission Control Protocol Synchronisation Flood (TCP-

SYN Floods), and application layer attacks are the most common DoS attacks [113]. Volumetric attacks attack a server with a large amount of malicious traffic to completely exhaust its bandwidth. TCP SYN Flood attacks are a clever way to exploit the communication protocol by overwhelming the server with excessive connections, making it impossible for the server to handle genuine traffic and connections. This result finally leads to device failure due to overload.

Distributed Denial-of-Service (DDoS) attacks [176] are an advanced form of DoS attacks that use multiple compromised devices to generate high traffic volumes aimed at disrupting target services. In contrast, application-layer attacks exploit vulnerabilities in software applications, making services inaccessible to regular users. In MQTT's Pub/Sub model, attackers can flood subscribers with spam messages, overloading the broker and resulting in service crashes. Furthermore, if the speed limits in MQTT service configurations are too high, the system becomes more susceptible to volumetric attacks.

Ravi et al. [177] proposed Learning-Driven Detection Mitigation (LEDEM), a semi-supervised machine learning-based mechanism designed to detect and mitigate DDoS attacks. Vishwakarma et al. [178] explored malware and botnets as underlying drivers of distributed DoS attacks in IoT environments. Chaabouni et al. [179] examined intrusion detection techniques based on Machine Learning (ML) algorithms, highlighting their effectiveness in improving IoT security. Parra et al. [180] proposed a distributed cloud-based deep learning framework to detect and mitigate phishing and botnet threats. However, the early stage of Software Defined Networking (SDN) deployment in IoT still leads to issues such as low detection accuracy, high memory consumption, network overhead, and inefficient processing.

To mitigate these concerns, Ujjan et al. [181] proposed the use of sFlow, adaptive polling-based sampling, Snort-based Intrusion Detection System (IDS), and deep learning models to reduce the impact of DDoS attacks in IoT networks. Wireless communication in the IoT is especially vulnerable to a range of threats, including DoS, Man-in-the-Middle (MitM), and message modification attacks. Designing a robust IoT security framework requires strong authentication and authorisation mechanisms and thorough auditing to detect suspicious activity [10]. In addition, encrypted message exchanges between pub-

lishers and subscribers protect against interception and unauthorised access.

Man-in-the-Middle

In general, attackers performing a MITM attack intercept communications between two parties without their knowledge. These adversaries secretly relay or modify the exchanged messages, often intending to eavesdrop, alter, or impersonate either participant. In MQTT-based communications, MitM attacks may occur during the connection establishment phase if encryption or authentication is improperly configured or absent. Attackers can exploit MITM attacks to manipulate MQTT communications, one of the most prevalent threats in IoT systems. An adversary intercepts and potentially alters messages exchanged between legitimate users in such attacks. This intrusion is critical in time-sensitive applications, where most transmitted messages contain sensitive information.

A notable example is dynamic eavesdropping, in which an attacker independently establishes connections with both victims and relays messages between them, creating the illusion of a direct, private conversation. In reality, the attacker has complete control over the communication, enabling interception, modification of messages, or injection of malicious data [182, 183]. MitM attackers can also mimic legitimate users, insert themselves into trusted communication flows, and gradually disrupt the system functionality.

Standard cryptographic strategies like TLS offer end-to-end authentication to mitigate MitM attacks. TLS is widely regarded as a foundational security strategy, often employed with the endorsement of trusted certificate authorities to authenticate one or both parties in IoT communications [25, 184, 185]. However, these cryptographic methods are not always suitable for resource-constrained IoT environments with limited computational capabilities [186–189]. For example, [190] proposed a detection and localisation strategy for MitM attacks in wireless sensor networks, but its performance was suboptimal in general IoT contexts.

Eslava et al. [191] developed a firewall-based system that uses a Raspberry Pi gateway to monitor network traffic and analyse communication patterns using a cloud-enabled database. Similarly, Saif et al. [192] introduced a symmetric cryptographic approach to

secure real-time healthcare monitoring applications. Their system encrypts the data in transit and allows it to be processed without decryption within Local Processing Unit (LPU), thereby reducing overhead while maintaining confidentiality of the data.

Malware Attacks

Intelligent living systems allow users to monitor real-time data and manage daily routines efficiently using IoT devices. However, ensuring the security of these devices is essential to protect the broader IoT ecosystem against unauthorised access, surveillance, and data tampering. IoT devices are vulnerable to various forms of malware, including Worms [193], Trojans [194], Ransomware [195]. Such malware can interfere with or compromise the operation of IoT networks. Malware attacks are known for their complexity and diversity [196]. In MQTT-based systems, an attacker may exploit the Pub/Sub model by subscribing to a topic and injecting malicious payloads into published messages. Once received, these infected messages can compromise subscriber systems, enabling data theft or remote control of IoT clients.

The heterogeneous nature of IoT networks, where devices, services, and protocols are often developed independently, further complicates the implementation of unified security measures. Furthermore, the dynamic topology of IoT systems, where devices frequently join or leave the network, creates challenges to detect and isolate infected nodes [197].

Farooq et al. [198] proposed an analytical model to study the propagation of malware in Device-to-Device (D2D) (Device-to-Device) communication within wireless IoT networks. While their model captures critical factors related to botnet formation, its assumptions limit its applicability across more complex environments. Neshenko et al. [199] presented a comprehensive vulnerability classification framework for IoT ecosystems. Although effective for theoretical exploration, the applicability of the framework can be restricted by the diversity and rapid evolution of IoT devices. Arnaboldi et al. [200] introduced the concept of a Coordinated Load-Changing Attacks (CLCA), a coordinated attack in which malware-infected IoT devices synchronise the operation of high-power appliances to destabilise smart grid infrastructures. Acarali et al. [201] applied epidemic modelling principles to simulate malware spread in wireless sensor networks, providing information

on infection dynamics and control strategies.

Identifying the exact source of an attack, such as the specific compromised IoT node, is also a challenge. Rattanalerdnusorn et al. [202] proposed Detecting Security Threats and Pinpointing Anomalies in an IoT environment (IoTDePT), a framework for fine-grained detection and identification of malware threats in IoT networks. Their work also explores the potential of federated learning in IoT malware detection and highlights the security implications of this decentralised learning paradigm.

To address these limitations, Liu et al. [203] proposed a Testing Framework for Learning-Based Android Malware Detection (TLAMD) system based on machine learning to detect adversarial malware targeting IoT devices. Their method leverages Hardware Performance Counter (HPC) to distinguish between malware and benign applications using an Artificial Neural Networks (ANN). However, most adversarial sample generation methods require access to model parameters and focus predominantly on image data, limiting their applicability in IoT contexts.

On a broader scale, overcoming malware threats in IoT environments requires comprehensive collaboration between IoT device manufacturers, service providers, and end users. This includes establishing robust security standards, improving the cybersecurity literacy of stakeholders, and investing in advanced detection and prevention strategies. However, this process is hindered by the lack of consensus on shared technical standards, system definitions, and usage policies throughout the ecosystem. In response, Rodriguez-Mota et al. [204] initiated a research effort to develop tools to describe and analyse the attack surface of IoT systems. Solutions to this problem can fully exploit the potential of IoT technology and extend its possibilities to enterprises, governments, and individuals.

Physical Attacks

Attacks on IoT devices can damage their physical components. Attackers can attack IoT communication devices by stealing or destroying servers and routers, or blocking electromagnetic signals [205]. In MQTT communications, when an attacker uses physical attacks to gain physical access to the MQTT broker, the topic information stored on the MQTT broker can be damaged, modified, or deleted. Furthermore, attackers can use

malware attacks on the MQTT communication system to gain unauthorised access to the system or private user data.

Adequate protection against physical attacks in IoT communication is a significant challenge [206], as IoT devices vary in type, size, functionality, and communication capabilities. This thesis restricted IoT devices to resource-constrained states because resource-constrained devices are weaker in processing power, memory, and storage space and more susceptible to security attacks. Second, devices with limited resources usually only have essential services and communication capabilities, and communication protocols such as MQTT that support resource-constrained devices do not have strong encryption and security mechanisms [207].

Another challenge is the structure of the communication of the Pub/Sub model, which is usually distributed and decentralised [208]. An attacker can affect the entire communication system by accessing a single node. Second, MQTT is a clear text protocol [209], and the Pub/Sub model usually uses unencrypted channels [210] to communicate. This means that any user with network access can intercept and listen to encrypted messages. In addition, MQTT devices are often deployed in remote locations [211], so communications using wireless connections such as mobile networks, Wi-Fi or Bluetooth are vulnerable to interference and MitM attacks. These attacks can cause loss of user privacy data and unauthorised access. When sensitive information is involved, it can also lead to personal injury. Consequently, it is concluded that using lightweight security protocols and strong authentication with authorisation mechanisms improves communication efficiency and reduces resource consumption while ensuring communication and information security.

Given the vulnerability of IoT devices, ensuring their stability is essential. One of the most effective measures is to ensure that all IoT devices are tested for authentication and access control through secure channels and implement threat detection mechanisms [212]. The security detection mechanism can also help detect and respond to physical attacks. Sharaf et al. [213] studied the problem of device authentication in the IoT. An object authentication framework was proposed to authenticate objects in the IoT using device-specific information called fingerprints. Yaqoob et al. [214] presented ransomware attacks and security issues in IoT. These flaws increase manufacturing costs and allow silent prob-

lems to occur only under specific loads, thus threatening security and lives. To understand the potential dangers and secure manufacturing systems, Pan et al. [215] proposed two taxonomies: one to categorise cyberphysical attacks targeting manufacturing processes and the other to describe quality control measures to counter these attacks.

Security solutions for IoT devices should be computationally efficient [216]. To meet all these requirements, Gope et al. [217] proposed a lightweight and privacy-preserving two-factor authentication scheme for IoT devices, where physical unclonable features are considered. Due to wireless communication between intelligent IoT devices and Gateway Nodes (GWNs), several security threats can arise in IoT environments, including replay, MitM, impersonation, malicious device deployment, and physical device capture attacks. To mitigate such security threats, Malani et al. [218] designed a new certificate-based device access control scheme in IoT environments, which is secure against the attacks mentioned above and retains anonymity.

Another approach is to apply the principles of secure design. It employs cryptographic algorithms and secure communication protocols such as TLS to ensure safe data transfer between devices. Security features may include password protection, encryption, and access control. It is also possible to ensure that IoT devices are placed in a secure environment by restricting their physical access to monitor and protect them from physical damage. Lin et al. [219] demonstrated a new offline guessing attack on Steiner, Tsudik, and Waidner's protocol. Padmavathi [220]. The HB+ protocol and many variants are vulnerable to a simple MitM attack. Kamali et al. [221] introduced ExTru, an encrypted communication protocol based on stream cyphers with the addition of a Configurable Switching and Toggling Network (CSTN), which not only improves the communication performance of lightweight IoT devices but also improves the communication performance with traditional channels resistant to side channels. It also consumes much less energy than conventional antise channel cyphers. However, physical attacks can still occur on IoT devices despite these measures. In this case, a comprehensive security protection mechanism and an incident response plan must be developed to minimise the impact of the attack [222].

Table 2.9 compares the impact of different security attacks on the healthcare environ-

ment enabled by MQTT.

Table 2.9: Comparative Analysis of Different MQTT Security Attacks.

Factors	DoS	MitM	Malware	Physical
Main In- tention	Service dis- ruption	Inject Malicious readings	Infect targeting functions	Tampering or steal- ing
Attack Method	Broker over- loading	Message Alteration	Malicious software	Broke physical in- frastructure
Targeting Devices	MQTT broker and de- vices	MQTT broker and devices	MQTT broker, de- vices, and server	IoT devices and components
Frequently suffered use cases	Remote Patient Monitoring, sensors like Electrocardiogram (ECG)	Electronic health- care records, patient data	Healthcare databases and servers, monitoring devices	Hospital infrastruc- ture, Gateways, monitors
Confident- iality	Low	High	High	Dynamic
Integrity	Medium	High	High	High
Availability	High	Medium	Medium	High
Detection Complex- ity	Low to Medium	High	Medium	Low
Preventive Measures	Anti-DoS and QoS settings	Strong encryption authentication at the topic level	Anti-virus, network segmentation	Effective physical structure and con- trol
Mitigation strategies	Secure bro- kers	Mutual authentica- tion	Real-time access control, sandboxing	Continuous moni- toring
Effects	Critical mis- sion failure, service out- age, delay	Incorrect diagnosis, delayed treatment	Patient safety com- promisation, data loss, ransom de- mand	Medical device thefts or damages, patient care interrup- tion
System Ef- ficiency	Low to medium	Low to medium	Low	Medium
Recovery Effort	High	Medium	High	High
The risk level of Human Live	High	High	High	High

2.3.3 Impact of Security Attacks on MQTT Performance

In particular, diversified security attacks on the IoT can significantly impact the performance and reliability of these systems. The interconnected nature of heterogeneous IoT devices and their use of real-time data sharing pose novel security breaches [223]. These security attacks can severely degrade the functionality of the entire IoT system by introducing different vulnerabilities and compromising user safety. This attack leads to inaccessibility of critical IoT services to users, resulting in downtime, which is especially harmful in critical applications such as healthcare, where continuous service is crucial to saving human lives. DoS and MitM attacks introduce attackers to deny or manipulate communication among devices, leading to delayed, corrupted, or false data transmission, respectively. This attack significantly affects the accuracy and reliability of the system. Analysing the impact of such attacks on MQTT-enabled systems is very crucial.

Latency

MQTT security attacks, such as DoS and MitM, can significantly impact communication latency [224]. This impact creates remarkable economic or life losses, especially in healthcare-enabled IoT environments where concurrent real-time data exchange and time-sensitive responses are critical. In a healthcare IoT system, a DDoS attack on the broker can delay the delivery of essential patient data, such as heart rate and oxygen levels, from continuous monitoring systems, thus extending response times in emergency care. The latency effects are highly dependent on the type of attack and the way it targets the MQTT protocol. The DoS attacker floods unnecessary traffic into the MQTT broker or IoT devices, aiming to overwhelm their resources [225]. However, the MitM attacker intercepts the message transmission by potentially altering messages among MQTT clients, such as publishers, subscribers, and brokers, resulting in excessive processing time. All types of attacks impact latency because messages must pass through a system affected by attackers before reaching the desired destination.

For example, a significant increase in latency can create delayed responses, leading to life-critical risks in critical healthcare situations, such as heart attack alerts. Patients

under constant monitoring or in intensive care suffer a difference between life and death due to seconds of delay in alert messages. A DoS attack could cause multiple fatalities by crippling communication for a prolonged time in a time-critical healthcare environment. Moreover, the impact of the DoS attacker on latency leads to a high risk of live losses, and the effects of MitM on latency lead to medium to high-level live losses [182, 226]. MitM attackers are estimated to create \$2 billion annual losses worldwide, and nearly 50% of MitM attackers are involved in the alteration of sensitive information, resulting in life-critical increases [227]. Figure 2.10 demonstrates the impact of various security attacks on latency and human lives.

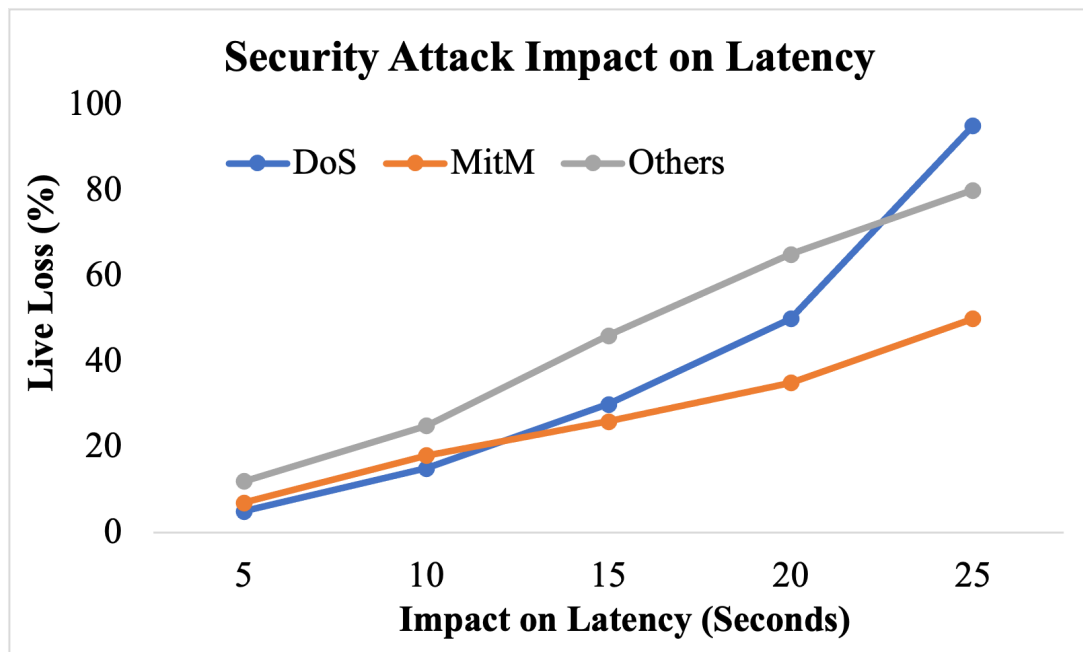


Figure 2.10: Impact of Security Attacks on Latency and Lives

Throughput

Throughput is the successful data transmission through a system over a specific time interval, which poses a distinct security threat that MQTT can reduce [228]. Among them, DoS and DDoS attacks are the most severe threats that overwhelm the MQTT broker by sending excessive traffic into the network, making it critical or impossible to process legitimate messages [229]. A DDoS attacker on a healthcare MQTT broker can

prevent monitoring information from being transferred efficiently, minimising the system's throughput. In a healthcare IoT environment, MitM attacks can insert or alter information obtained from patient monitoring devices, shrinking the vital amount of data reaching the monitoring systems, thereby reducing throughput. Successful legitimate message transmission decreases as the healthcare environment becomes highly congested with malicious network traffic, significantly reducing throughput [230].

Therefore, MQTT security attacks can significantly minimise throughput in the IoT by introducing high overhead, causing unnecessary resource exhaustion, and leading to higher data corruption. These adverse effects range from legitimate traffic denial to processing delays. Thus, they force tampered or replayed information reprocessing, all of which diminish the system's ability to effectively manage the intended massive volume of information. In addition, a drop in throughput creates the most critical consequences in time-sensitive and constant monitoring of IoT environments, such as healthcare.

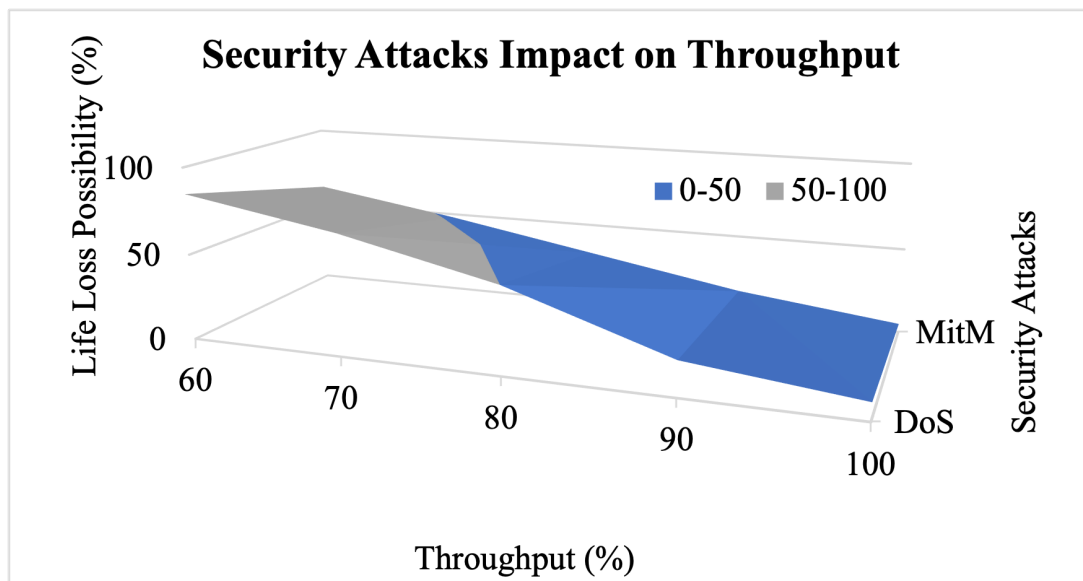


Figure 2.11: Impact of Security Attacks on Throughput

Figure 2.11 plots the impact of various MQTT security threats on throughput in healthcare data transmission. In particular, a reduction in throughput increases the risk of life losses. This scenario illustrates that reduced throughput by 60 leads to 85% and 62% live loss risks in the presence of DoS and MitM attackers, respectively [231]. The high-throughput scenario minimises the risks by 10 and 5 in the same scenario, respectively.

Reliability

Various security attacks on MQTT profoundly affect the reliability of the IoT system by compromising availability, integrity, and consistency of delivery [232]. In the MQTT context, reliability represents the ability of the system to deliver messages consistently and in the correct order to the intended destination without duplication or loss of messages. MQTT security attacks target broken orders or create message loss or duplicates, which degrades the reliability of the IoT system. In a healthcare IoT system, an attacker sends incorrect readings to the medical monitoring system by altering the information, undermining the reliability of the healthcare application. For example, MitM attackers intercept and possibly change the information presented in messages before reaching the desired destination, leading to corrupted or incorrect data delivery. This attack undermines the system's reliability, since receivers cannot trust that the received information is accurate.

In contrast, in the presence of a DoS attack, the attacker overwhelms the broker by dropping or delaying legitimate messages, minimising the reliability of message delivery. In other types, attackers use tokens or certificates for weak authentication and authorisation strategies, which allow unauthorised users to publish or subscribe to MQTT topics, resulting in incorrect or malicious data [233]. Encryption mechanisms are often introduced to mitigate man-in-the-middle and eavesdropping attacks. However, a high encryption overhead may introduce latency in unaddressed cases, affecting the reliability of time-sensitive applications such as healthcare.

In a DoS attack scenario, almost 20% to 30% of the messages are dropped or denied to the MQTT broker, resulting in low reliability. This attack severely affects timely care and increases life losses [234]. Furthermore, almost 60% to 80% of the data are dropped due to message alteration caused by MitM attacks. The attacker can change the content of the messages, causing the MQTT subscriber to receive incorrect or altered information from the brokers, which diminishes the reliability.

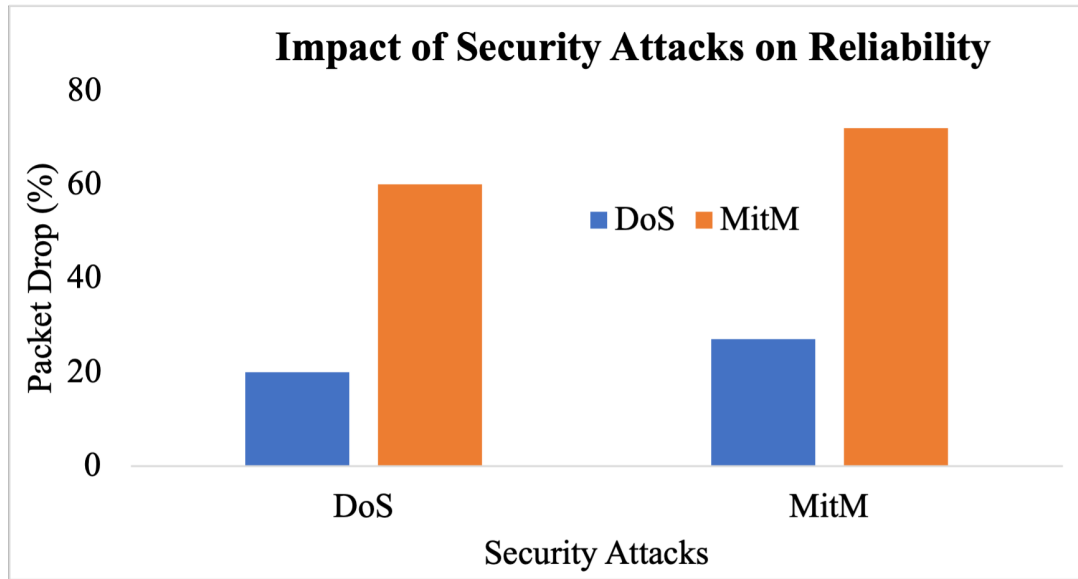


Figure 2.12: Impact of Security Attacks on Reliability

Resource Consumption

Some MQTT security attacks impact device resources, with the aim of deteriorating the helpful life of the network [235]. For instance, DoS or DDoS attacks flood excessive message traffic or connection requests to the MQTT broker for unnecessary resource exhaustion. The main intention of the DoS attack is to reduce brokers' processing capabilities, thereby significantly escalating resource consumption and scaling down the IoT performance efficiency. Unnecessary consumption of CPU and memory resources by brokers leads to less resource availability for legitimate operations. However, MitM attackers can introduce excess network traffic to intercept, delay, or retransmit messages. Frequent data retransmissions due to attack behaviour escalate CPU or memory usage [236].

In an IoT healthcare system, the additional process required for encrypting and verifying medical data is incorporated into monitoring devices at the level of resource consumption attached to patients and the MQTT broker. The attackers may repeat illegitimate authentication attempts, thus overloading the MQTT broker with repeated authentication checks and increasing CPU and memory usage. Furthermore, resource depletion attacks the target CPU, memory, and storage by overwhelming devices with excessive or unnecessary tasks or messages, leading to high energy consumption and potential net-

work failures. In addition, implementing security solutions like TLS for MQTT for strong authentication and integrity introduces additional computational overhead, resulting in high resource consumption [30].

In addition, MQTT security attacks cause a substantial escalation in the consumption of resources across IoT systems. Figure 2.13 shows the impact of security attacks on overhead. Figure 2.13 shows the effects of security attacks on overhead, demonstrating the unnecessary energy consumption caused by DoS and MitM attacks.

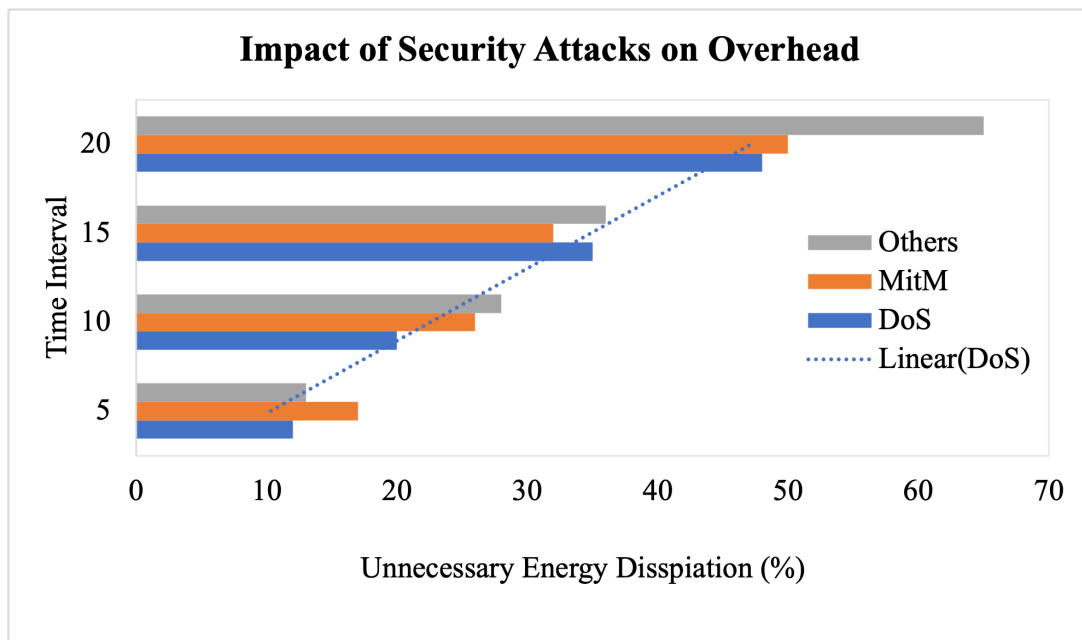


Figure 2.13: Impact of Security Attacks on Energy Consumption

Overhead

By familiarising themselves with the extra computational and energy demands, MQTT security attacks raise overhead, hindering the performance efficiency, scalability, and reliability of healthcare systems, especially in resource-constrained environments [237]. DoS attackers force MQTT brokers and devices to process large volumes of malicious traffic, directly increasing the CPU usage and memory consumption, leading to additional overhead [238]. Patient monitoring devices in the healthcare environment require real-time data transmission. Hence, the DDoS attacker restricts the MQTT broker from immediately updating critical patient information by forcing it to prioritise malicious traffic

handling, resulting in high processing overhead and reduced system efficiency. Healthcare IoT should perform resource reallocation to mitigate attack behaviours, diverting processing power to handle malicious traffic instead of focusing on critical healthcare functions.

MitM attacks enforce IoT in healthcare by implementing strong encryptions such as TLS to provide security to communication channels [239]. This attack adds computational overhead to IoT monitoring devices and brokers to encrypt and decrypt messages. Thus, MQTT security attacks introduce significant overhead to IoT systems, especially in healthcare, through frequent encryption, authentication, and resource allocation. This extra overhead can reduce the level of performance of healthcare settings, create delays in time-sensitive real-time patient reading transmission, and deplete resources such as bandwidth and energy. In addition, it is crucial to mitigate the impact of MQTT security attacks to effectively balance security and resource efficiency and ensure that IoT is highly secure, timely, responsive and unique in healthcare. Figure 2.14 shows the impact of security attacks on overhead. MitM introduces a higher overhead in the healthcare IoT environment than in the DoS attack scenario.

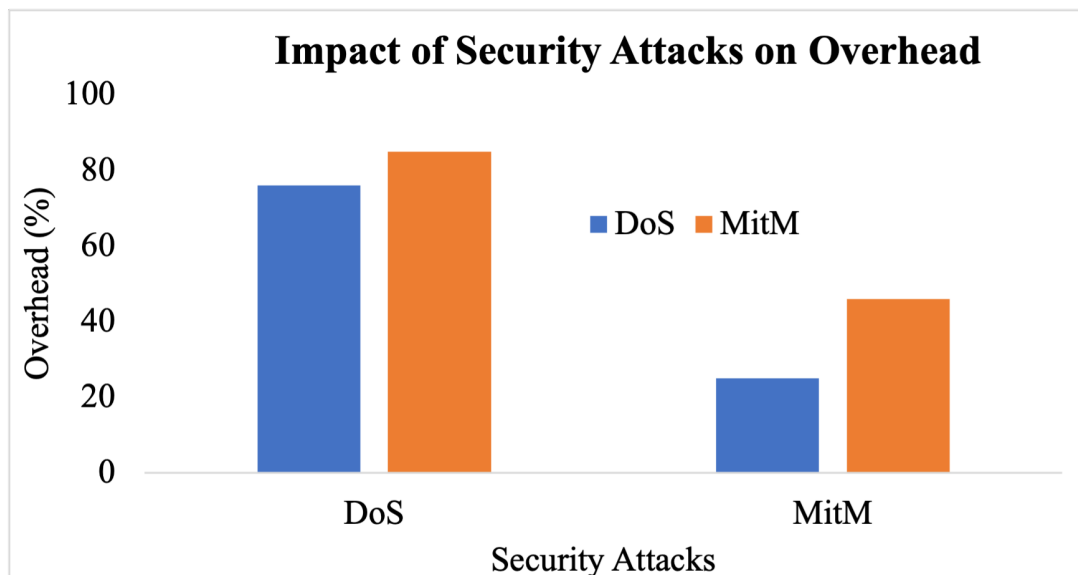


Figure 2.14: Impact of Security Attacks on Overhead

Table 2.10: Impact of Security Attacks on MQTT Performance.

Factors	DoS	MitM	Malware	Physical
Impact on Latency	High	Medium to high	High	Low to High
Impact on Throughput	Low to High	Medium	Medium	Medium
Impact on Reliability	High	High	High	High
Impact on Resource Consumption	Very High	Medium	High	Medium to High
Impact on Overhead	High	High	Medium to High	Medium

2.4 Chapter Summary

This chapter established a solid technical foundation for understanding MQTT-enabled IoT environments. It introduced the essential components of IoT, including device types, connectivity protocols, layered architecture, and application domains. In particular, the chapter introduced a focused background on Healthcare IoT, explaining how interconnected medical devices and sensors facilitate real-time monitoring and diagnosis in Healthcare 4.0. The discussion highlighted the unique challenges faced in this domain, such as limited computational resources, strict privacy requirements, and the need for secure and lightweight communication protocols like MQTT.

The core technical principles of the MQTT protocol were explored in detail, covering its lightweight design, operational model, and quality of service levels. The suitability of MQTT for healthcare applications was also discussed. The chapter critically assessed the security requirements of MQTT in resource-constrained IoT settings. It explored the protocol's vulnerabilities and the types of attacks it faces while also analysing how different security threats affect its performance. These insights are the basis for evaluating and proposing improved lightweight security strategies in the following chapters.

This chapter offers a comprehensive review of work related to MQTT security and MQTT in Healthcare IoT. Section 3.1 reviews the literature on MQTT security and classifies the security schemes based on their working operations and different algorithms. Section 3.2 discusses lightweight security solutions based on various security requirements of MQTT. Section 3.3 presents a comparative analysis of MQTT security solutions and the essentials of proposing lightweight solutions to IoT-based environments. Section 3.4 combines the previous reviews analysis to compare the advantages and limitations of lightweight encryption schemes. Section 3.5 summarises the general research gaps of these security schemes, which highlights the lightweight security schemes for the IoT healthcare environment. Section 3.6 reviews the existing MQTT schemes based on ABE in IoMT, and the literature is analysed from three aspects: application, performance and comparison of access control.

Finally, Section 3.7 summarises the research gaps in ABE-based schemes in IoMT and demonstrates the necessity of proposing a lightweight multi-authorisation security scheme suitable for handling secure and privacy-preserving data exchange in the dynamic and resource-limited Healthcare environment.

3.1 Cryptography based MQTT Security Counter-measures

Several studies have focused on symmetric and asymmetric key-based encryption techniques to address various security challenges while maintaining computational efficiency in healthcare IoT scenarios [240]. The study in [240] offers a comprehensive review of cryptographic key management strategies, particularly those tailored for resource-constrained

healthcare environments. It categorises conventional cryptographic methods into symmetric and asymmetric schemes and evaluates their suitability for IoT-driven medical devices. Symmetric key approaches are considered ideal for low-power sensors due to their reduced computational requirements; however, they face challenges related to secure key distribution and limited scalability.

In contrast, asymmetric schemes provide enhanced scalability and facilitate secure key exchanges, but their computational overhead renders them unsuitable for real-time, resource-constrained healthcare applications. The study also investigates the trade-offs between security and performance through hybrid cryptographic strategies, which combine the strengths of both symmetric and asymmetric approaches. Key management protocols are evaluated based on energy consumption, latency, computational overhead, and robustness to security threats. In addition, the study examines real-world use cases, including patient monitoring, secure medical data exchange, and remote diagnostics, while identifying several research gaps. In particular, it emphasises the need for lightweight cryptographic solutions and highlights the importance of flexible, context-sensitive key management systems in future IoT-enabled healthcare environments.

Despite ongoing advancements in IoT security, various security and privacy issues remain unresolved. For example, when security threats emerge at the IoT application layer, devices and services can be shut down or maliciously compromised [241]. A prominent example is the propagation of malicious code, such as a “worm” that rapidly spreads across the Internet and targets embedded systems operating on specific platforms [242]. Another concern is malicious manipulation of node-based applications [243], whereby adversaries exploit software vulnerabilities to install harmful packages, which can result in data tampering or operational disruption.

Within the three-layer IoT architecture [244], resource-limited devices must be able to mitigate cascading system compromises when security breaches occur in individual architectural layers. As IoT adoption grows worldwide, the need for secure communication protocols becomes increasingly urgent [245]. Among these, MQTT—a lightweight publish/subscribe application-layer protocol—is widely employed across constrained IoT environments [21].

Authentication, access control, encryption, and message integrity are typically integrated into MQTT security solutions. When effectively combined, these measures protect against unauthorised access, data interception, and tampering. Cryptographic countermeasures are broadly classified into two main categories, as illustrated in Figure 3.1.

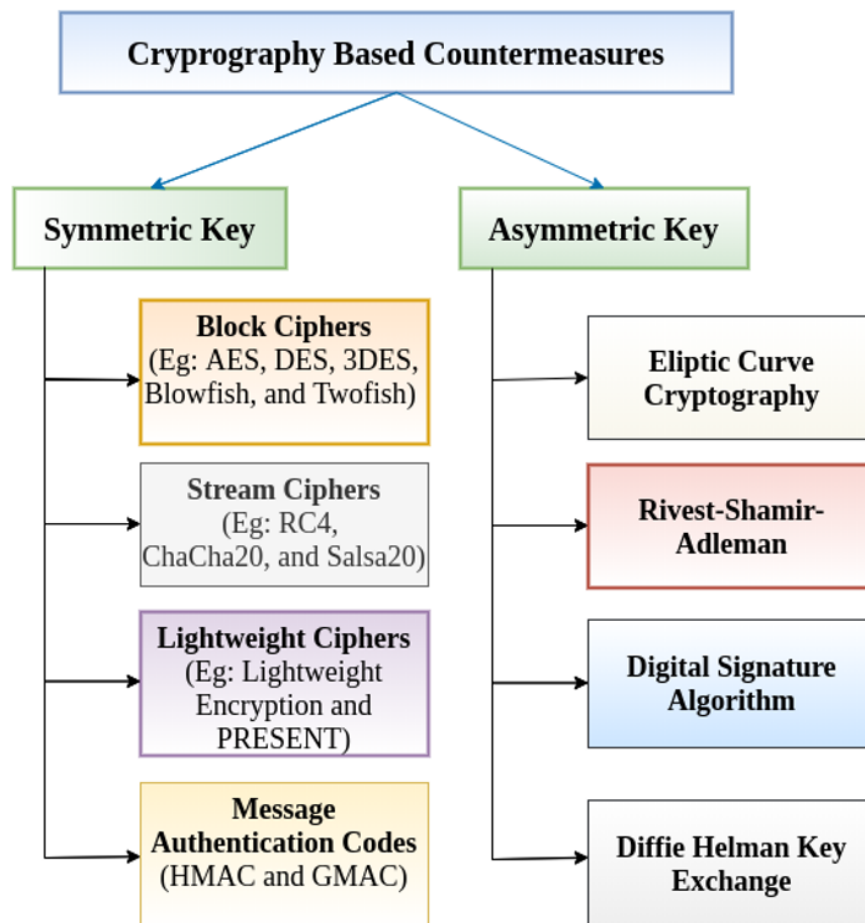


Figure 3.1: Cryptography based Countermeasures

3.1.1 Symmetric Key Cryptography based Solutions

Symmetric algorithms are encryption methods that use the same key for encryption and decryption. This approach offers efficiency and speed, as the involved entities do not need

to exchange keys securely. Due to their performance advantages, symmetric algorithms are frequently used in MQTT-based security schemes to balance robust protection and computational feasibility. Several studies have proposed symmetric key cryptographic solutions tailored for IoT environments [246–249].

Pereira et al. [250] evaluated cryptographic algorithms to assess their suitability for various IoT platforms. Based on the simulation results, the authors recommended the selection of symmetric encryption algorithms based on the network traffic level in IoT applications. Furthermore, they suggest that the Advanced Encryption Standard-Counter (AES-CTR) mode provides faster and less complex encryption/decryption operations with the Constrained Application Protocol (CoAP) compared to the Advanced Encryption Standard-Cipher Block Chaining (AES-CBC) mode. However, this work does not analyse the cryptographic algorithms used in the MQTT protocol. Kumar et al. [251] analysed MQTT over Quick UDP Internet Connections (QUIC). Using QUIC in MQTT reduces packet processing costs and delays and improves network throughput compared to TCP in MQTT. QUIC is designed to work in low-bandwidth scenarios, such as when sensors are part of unreliable networks. Moreover, TCP does not fully utilise the available bandwidth for the first few round trips, hindering the complete use of the advantages of MQTT. However, most existing schemes cannot analyse the security of basic MQTT and lightweight encryption schemes in low-bandwidth scenarios.

Varghese et al. [252] proposed the Secure Mobile Augmented Reality for Tele-Assistance (SMART) system, which addresses the security and privacy issues associated with Mobile Augmented Reality for Tele-Assistance (MART) applications. In SMART frameworks, Vargase et al. used the Advanced Encryption Standard (AES) algorithm [253] to encrypt images before they are transmitted over the network, ensuring that only authorised users can see images in private areas. However, using AES algorithms in SMART systems requires a shared key between the encrypted data sent and received. Therefore, it is challenging to manage the key if the same key is exchanged between the Augmented Reality (AR) device and the remote server. However, encryption can break down if the key is compromised or incorrect. Bisne et al. [254] proposed a composite security scheme of ABE and dynamic S-box AES to provide access control and confidentiality for mes-

sages exchanged in MQTT-based IoT systems. This composite security framework uses dynamic S-box AES to make it more difficult for attackers to crack encryption keys and provides fine-grained access control for MQTT messages. Second, the composite security framework is scalable and can protect MQTT messages in large-scale IoT deployments. However, the composite framework is relatively complex and does not provide a strong defence against MitM attacks. Additionally, the dynamic S-box AES affects communication performance.

Yerlikaya et al. [255] proposed a security mechanism for authentication and authorisation for the MQTT protocol. The mechanism combines user-name/password authentication, Hash-based Message Authentication code (HMAC)-Based One-Time Password (HOTP), TLS, and AES. Although TLS is a secure transport layer protocol, it cannot provide confidentiality for authentication and authorisation. Yerlikaya et al. used AES to encrypt MQTT messages and authenticate and encrypt user identities and authority credentials. Furthermore, the topic-based authorisation scheme defines the permission set for each topic. However, when subject permissions are stored in the broker, the broker is the single failure point of the topic-based authorisation scheme. If the broker is attacked, all the permissions for the topic are in danger.

Rhbech et al. [256] proposed an implementation of IoMT based on citizen geolocation and incorporating Artificial intelligence (AI)-based facial recognition. Rhbech et al. proposed an enhanced MQTT protocol called Secure MQTT (S-MQTT). S-MQTT combines AES and Rivest–Shamir–Adleman (RSA) [257] algorithms for mutual authentication and key exchange. AES encrypts the data exchanged between the devices, while RSA is used to authenticate the devices and exchange keys. This mechanism ensures that only authorised devices can communicate and exchange data. However, since S-MQTT uses symmetric and asymmetric encryption, this can increase the overall communication overhead for applications with implementation requirements.

Varma et al. [258] discussed MQTT, a lightweight messaging protocol in which payloads are used as drivers for data transmission, in Varma et al. AES is used to encrypt the MQTT payload. This approach can be applied at the transport or application layers. Encrypting the payload at the transport layer provides end-to-end security but may

incur some performance overhead. Encrypting the payload at the application layer is less secure, but more efficient. However, due to the use of the TLS-based encryption protocol, this is not lightweight for resource-constrained devices and requires higher computational cost [28]. Furthermore, this scheme lacks access control settings. Shilpa et al. [259] proposed a Secure and Reliable Messaging Communication (SEC-RMC) protocol that performs MQTT data transfer with enhanced encryption through Mosquitto [101]. In this study, the session key is encrypted using AES between Mosquitto and the client. Key exchange is performed using RSA encryption to ensure data confidentiality. The broker only verifies the topic information with the database and then shares it with the user.

Although the scheme ensures safe communications, using asymmetric keys for key exchanges is a significant performance burden for resource-constrained devices [260]. Second, symmetric encryption also adds overhead because it requires the exchange of session keys between MQTT brokers and clients.

Most MQTT applications for secure communication explore Advanced Encryption Standard (AES) [261]. Although AES requires less communication time, it has the major drawback of high memory consumption, posing significant implementation challenges for resource-limited IoMT devices. Iyer et al. [262] evaluated MQTT using SIMON [263] and SPECK [264] under the MitM attack. They used the lightweight hash function SPONGENT to support resource-constrained IoT devices. SPECK provides a better security level while consuming minimal battery and memory resources. The primary limitation of the SIMON and SPECK algorithms is the exclusion of the S-box concept in the encryption process. Performing operations on both plaintext and ciphertext can be costly, and it is essential to enhance the confusion property during ciphertext generation, especially for the IoMT environment.

Sadio et al. [265] proposed using the Authenticated Encryption with Associated Data (AEAD)-CHACHA 20-POLY1305 algorithm to secure restricted node communication through MQTT/MQTT-SN. The security scheme assumes the presence of a gateway between the publisher and the broker. The topic data are encrypted and authenticated successfully using a pre-shared secret key and ChaCha20-Poly1305 AEAD. Although this

security framework uses a lightweight and efficient algorithm for message exchange and data encryption between client and server, this security framework is less secure than other frameworks, such as TLS, and is not widely available. Furthermore, the ChaCha20-Poly1305 AEAD algorithm must provide a relatively new and adequate proof of security verification. Moreover, they did not evaluate the suitability of other symmetric encryption algorithms for MQTT communication. However, the ChaCha20-Poly1305 AEAD fails to ensure strong security against various attacks and introduces high complexity due to the critical key management structure. Although Sabri et al. [266] demonstrated that ChaCha20 is well suited for real-time applications in medical IoT devices, it remains challenging for battery-powered healthcare devices in low-power environments due to energy efficiency and key management issues.

Bogdanov et al. [267] introduced a PRESENT algorithm, a lightweight structured cryptography protocol adapted to resource-limited network environments. It is a lightweight symmetric block cypher that offers better IoT performance and security trade-offs. However, the linear functions of the PRESENT key scheduling algorithm are employed to identify the relationship among round keys, leading to a slow and predictable bit transition. However, the limited block size and short key lengths of the PRESENT algorithm hinder its performance in providing security against novel and multiple attacks. Therefore, Imdad et al. [268] introduced a PRESENT algorithm based on the Key Schedule Algorithm-PRESENT (KSA-PRESENT) scheme to solve these issues. The KSA-PRESENT scheme enhances the randomness and unpredictability of the round key for cryptographic security by combining the PRESENT-128 packet cypher with a more complex non-linear function to generate the round key.

Sahmi et al. [269] used the Augmented Password-Authenticated Key Exchange (AugPAKE) and PRESENT algorithms to secure MQTT communications. PRESENT explores small S-boxes and executes them individually. It needs to balance the confusion property of secret keys and computational complexity. Moreover, AugPAKE intends to create a secure key exchange session. However, the long-term session keys exploited by AugPAKE increase the vulnerability against different attacks. Furthermore, it lacks an evaluation of the suitability of other symmetric encryption algorithms for MQTT com-

munication in hostile environments, leaving unaddressed security concerns.

The work in [270] presents a Low memORy symmEtric-key geNerAtion (LORENA) method incorporating a secret key agreement group protocol for environments in the Internet of Healthcare Things (IoHTs). This protocol effectively addresses the resource constraints of healthcare devices and achieves better performance. However, achieving better security and trade-offs is a critical issue in LORENA.

A secure patient authentication scheme in [271] leveraging symmetric encryption to protect patient records. While their approach enhanced security, it lacks a detailed performance evaluation against real-time attack scenarios and does not address the scalability issue of the proposed authentication mechanism. Symmetrical searchable encryption in [272] efficiently shares personal health record data with high security. Although this method improves data confidentiality, it fails to fully address the computational overhead of resource-constrained IoT devices, raising concerns about its practicality for low-power medical devices. The work in [273] presents an anonymous patient monitoring system exploiting wireless medical sensor networks, emphasising security and anonymity. However, it does not consider real-time threat detection mechanisms and proactive intrusion prevention strategies, leaving gaps in attack resilience. A lightweight collaborative authentication scheme [274] with key protection for electronic health records. Although this method reduces computational overhead, it thoroughly fails to explore scalability issues in large-scale IoT networks, potentially limiting its effectiveness in broader deployments.

In summary, due to their efficiency, symmetric encryption algorithms such as AES, PRESENT, and ChaCha20 are well-suited for resource-constrained IoT environments. However, they pose challenges in secure key management, especially in distributed systems like IoMT. To address these concerns, hybrid frameworks and enhanced AES variants (e.g. dynamic S-box AES) have been explored to provide better security and performance balance.

Based on previous analysis of different security schemes, we can determine that using symmetrical encryption algorithms to encrypt transmitted information improves security in the security framework and completes user authentication. However, traditional symmetric algorithms applied to security systems have limitations, such as key manage-

ment [275]. Second, conventional symmetric encryption algorithms may not be suitable for resource-constrained IoT devices due to higher computational costs [276]. In lightweight security schemes, symmetric keys cannot be safely stored in resource-constrained devices because of their limited memory and processing power. Therefore, combining symmetric encryption algorithms and other security schemes in a security system can improve its efficiency and security.

3.1.2 Asymmetric Key Cryptography based Solutions

In addition to the symmetric encryption algorithms mentioned above, asymmetric encryption algorithms are also a good choice for security schemes. In security frameworks, asymmetric encryption algorithms can encrypt messages and establish a secure channel between MQTT clients and brokers. In addition, asymmetric algorithms can be used for client authentication. The asymmetric encryption algorithms used in MQTT security systems include RSA, Elliptic Curve Cryptography (ECC) [277], Diffie–Hellman Algorithm (DHA) [278], and Elliptic-Curve Diffie–Hellman (ECDH) [279]. Mektoubi et al. [280] proposed a new communication method to protect communication through the MQTT protocol. This method uses ECC for key negotiation and authentication.

In addition, they also compared the RSA cryptographic algorithm and the ECC algorithm in MQTT and analysed the performance of both algorithms in encrypting and decrypting the communication process. ECC key sizes are much smaller in finite fields regarding cryptological algorithms, such as RSA. Therefore, ECC is better suited for resource-constrained devices than RSA. Second, the security of ECC is closely related to the difficulty of Elliptical Curve Differential Logarithm (ECDLP). If ECDLP is challenging to solve, then ECC is secure. Furthermore, ECC is much lighter in devices with resource constraints [281–283]. The work in [284] presents an elliptic curve-based signature encryption scheme that effectively combines the functionality of digital signatures and encryption, reducing computational costs and communication overhead compared to traditional signature encryption schemes.

To address the issues associated with SSL/TLS in MQTT in traditional approaches,

such as high complexity, Diro et al. [285] proposed a fog computing-based security scheme that relies on ECC algorithms to provide secure communication for MQTT without compromising security. The proposed scheme exchanges secret keys securely using a reduced number of handshake messages. The ECC algorithm uses short private keys, which reduces message sizes compared to other asymmetric cryptographic algorithms. However, the high computation intensity and complexity in key management in a resource-limited environment make it perform poorly in terms of delay and energy consumption. However, it is unclear whether this scheme suits resource-constrained devices with limited resources. In their paper, Diro et al. analysed the performance of MQTT with the AES scheme following a DoS attack, considering communication and storage overhead and delays. AES provided a fast response, but it has high memory consumption.

Lohachab et al. [286] also advocated using ECC to provide authentication and communication security for MQTT messages. The proposed authentication mechanism implements security and key management in a secure, efficient, and scalable manner. Using ECC provides a high level of security, and the key-negotiation protocol protects the shared keys. The approach is also efficient, as it does not incur more significant performance overheads. The key negotiation protocol is relatively complex, adding to the overall complexity of the method. The complexity of key negotiation protocols can increase security, but security levels can be limited during implementation and deployment. Second, this method uses Public Key Infrastructure (PKI), meaning the device's public key is stored in a public directory. This approach allows attackers to obtain public keys from the device and use them to intercept messages or to imitate legitimate devices. Elgenaidi and Newe [287] used a hybrid secure key management scheme to achieve security and reliability in Marine Wireless Sensor Networks (MWSNs).

Elgenaidi et al. used RSA to encrypt and decrypt the newly generated keys. However, RSA can take longer to run in applications, which can be a limitation in resource-constrained MWSNs. Furthermore, the bandwidth of the MWSN network may be limited, which can seriously affect its performance if the RSA operation time is shorter. The authentication scheme in this scenario is based on public-key encryption, which can be more computationally intensive and vulnerable to replay attacks. Simplified security

mechanisms combined with more efficient encryption algorithms can reduce the required communication and achieve high performance without sacrificing security, making it more suitable for resource-constrained MWSNs.

Sanaa et al. [288] proposed a security scheme for IoT networks based on MQTT and COAP. This scheme uses ECC to encrypt exchanged messages. ECC is preferred to optimise complexity. The ECC implementation used in this scheme is based on Curve25519. Curve25519 is a slight elliptical curve that is ideal for lightweight IoT devices. The Curve25519 implementation in the paper is only 1,280 bytes, which makes it very efficient to implement on lightweight IoT devices. Second, the sender needs to generate a random number and encrypt the random number using the receiver's public key in the encryption process. Using a random number to authenticate the sender prevents replay attacks. Encrypting the message using the random number and the sender's private key provides confidentiality. ECC algorithms are lighter than traditional asymmetric algorithms such as RSA, but more processing power is needed than Improved Ciphertext-Policy Attribute-Based Encryption (ICP-ABE) [29] algorithms. However, this scheme proposed by Sanaa et al. is still suitable for resource-constrained IoT devices. In terms of security, it cannot deal with all security threats, such as MitM and DoS attacks.

Yusoff et al. [289] used ECC to protect MQTT messages in smart home applications. ECC is more efficient than traditional public-key encryption algorithms such as RSA and is suitable for resource-constrained IoT devices such as smart home applications. Yusoff et al. used the RBAC system to authorise access. This method gives each device a role in determining which operations the device can perform. Then, the scheme encrypts the message payload using a shared key. The encrypted message load is sent to other devices. Payload encryption and access authorisation play a key role in the security of MQTT communications.

Ahmed and Kannan [290] used DHA and ECDH to propose an improved ECDH system, enhanced EC-DH (E-ECDH). This system provides lightweight and secure communication for real-time IoT applications. They also used case studies from the medical industry as research analyses to demonstrate that E-ECDH offers better performance and security. Since ECDH uses the X.509 certificate to authenticate devices, it is vulnerable

to MitM attacks. As a result, Ahmad et al. used the Hash function to authenticate exchanged public keys. This approach makes E-ECDH more resistant to MitM attacks. However, E-ECDH is computationally expensive for devices with resource constraints because E-ECDH uses elliptical curves as key-exchange protocols. Second, E-ECDH must use secure channels to exchange public keys, ensuring that public keys are not intercepted and that safe channels are sufficiently secure.

Sowjanya et al. [291] offered a new lightweight security scheme for IoMT using an ECC-based security scheme proposed by Li et al. [292], simplifying secret key management processes. An introduced disinfectant concept helps sterilise health data blocks and convert the data blocks into valid data for the disinfection file. Helps to maintain private information secrets while remotely performing data integrity audits. However, the attribute set for access control policies in MQTT frequently varies, leading to recurrent revocation of keys and increasing control overhead. Furthermore, private channel-based key transmission is often impractical in MQTT-enabled IoMT environments, resulting in significant security communication challenges.

Zhao et al. [293] developed authentication systems with lightweight security schemes such as ECC and PRESENT between medical devices and MQTT brokers to address these security concerns. Due to resource restrictions, greater security with an optimal key length is required. Minor keys tend to be insecure, while being too long could cause high-delay communication. Doctors in the healthcare industry use IoMT technology to investigate patients with COVID-19. This innovative approach enables remote monitoring and control of medical equipment and data, and allows physicians to monitor patients and isolate them when necessary. However, ensuring security while satisfying the privacy of patient data becomes increasingly challenging, especially during pandemics such as COVID-19 and influenza A, due to the increased data exchange and frequent access demands in emerging IoMT systems [294].

Asymmetric encryption also has authentication functions compared to symmetric encryption algorithms. In addition, symmetric encryption algorithms provide better key management security [295, 296]. The work in [297] optimises security for resource-constrained IoT devices using asymmetric cryptography and blockchain. Despite en-

hanced security, it fails to evaluate diverse IoT platforms, particularly in heterogeneous network environments with varying resource and security constraints. An IoMT image cryptosystem based on spatial watermarking and asymmetric encryption in [298] improves data integrity. However, it did not consider the trade-off between security and processing latency, which is crucial for time-sensitive healthcare applications.

However, asymmetric encryption algorithms that require high bandwidth and memory are not always suitable for resource-constrained IoT devices. Thus, choosing the appropriate encryption algorithm also requires balancing the relationship between performance and security, as well as the basic requirements and features of the security scheme.

In contrast, asymmetric encryption methods like ECC and RSA provide robust mechanisms for key exchange and authentication. These methods are particularly beneficial for securing MQTT communication in IoT systems. However, their computational overhead and memory requirements may hinder their applicability in highly constrained environments. Lightweight adaptations such as Curve25519 and E-EC DH offer promising alternatives.

While symmetric encryption offers computational efficiency ideal for real-time healthcare applications, its key distribution challenges remain a significant limitation. In contrast, asymmetric cryptography improves security and flexibility at the cost of greater resource consumption. Consequently, hybrid models that combine symmetric or asymmetric and lightweight techniques are increasingly proposed to balance these trade-offs in IoMT systems.

3.2 Review of Lightweight Cryptography Algorithms

Lightweight cryptographic schemes have recently been adopted to enhance security in medical IoT environments, where device processing power, memory, and energy are inherently limited. However, several studies have reported that these lightweight algorithms remain vulnerable to various security attacks [299–304]. Numerous reviews have systematically compared these lightweight cryptographic algorithms across diverse platforms and conditions [305–307]. Based on their encryption strategies and the specific security requirements they fulfil, these cryptographic solutions can be classified as shown in

Figure 3.2.

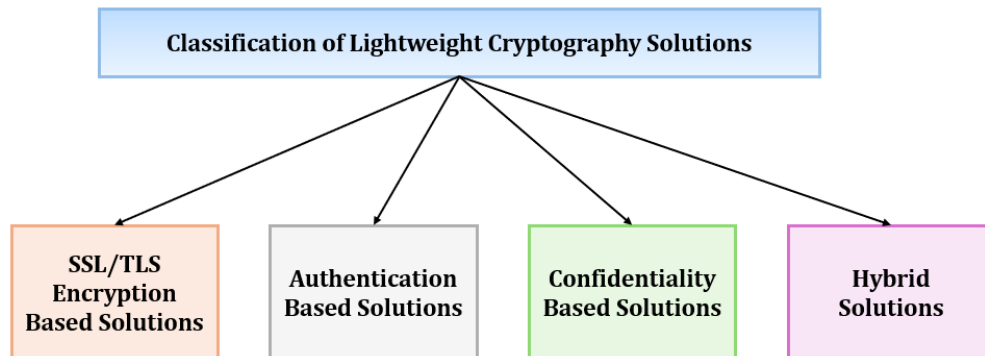


Figure 3.2: Classification of Lightweight Cryptography Solutions

3.2.1 SSL/TLS Encryption based Solutions

Several studies have implemented TLS to secure MQTT communications [308]. The strength of this approach lies in its ability to ensure robust security, ease of implementation, and compatibility with a wide range of systems. Secure Sockets Layer/Transport Layer Security (SSL/TLS) offers a comprehensive security framework that supports encryption, authentication, confidentiality, and integrity, thus mitigating various MQTT-related attacks.

Kannoja et al. [309] proposed a benchmark comparing different TLS cypher suites, including Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)-Elliptic Curve Digital Signature Algorithm (ECDSA) and RSA. Their results indicated that the ECDHE-ECDSA cypher suite significantly outperformed RSA in Ubuntu environments.

Li et al. [310] introduced a lightweight secure transport protocol known as Lightweight Transport-Layer Security Protocol (iTLS), extending the work in [311]. iTLS enables secure data transmission from the client during the first communication round by dynamically generating early keys associated with different identities. This mechanism maintains a high level of secrecy and facilitates implicit mutual authentication without relying on certificates. By eliminating additional round trips, iTLS significantly reduces communi-

cation latency and overhead. It also utilises ephemeral secret tickets obtained from prior connections to generate ephemeral server keys, thereby supporting complete forward secrecy. iTLS is fully compatible with TLS 1.3 and can be easily converted to a DTLS version.

The study in [30] implements SSL/TLS security in MQTT-enabled IoT applications, evaluating its performance on a typical IoT testbed comprising Raspberry Pi 4 and ESP32 nodes. The research primarily analyses the impact of SSL/TLS on energy consumption, computational overhead, complexity, and storage requirements. The results reveal that TLS-enabled nodes consume approximately 25% more battery than their non-TLS counterparts.

Kumar et al. [312] investigated using SSL/TLS certificates to protect systems from common web vulnerabilities. SSL/TLS certificates were shown to protect sensitive information, prevent unauthorised access, and mitigate accidental and malicious attacks. The study outlines procedures for obtaining, installing, and configuring SSL/TLS certificates on web servers, emphasising HTTPS deployment to protect sensitive data. Additional security strategies were also incorporated, such as strong password policies and defence mechanisms against attacks like Structured Query Language (SQL) injection and cross-site scripting (XSS). Furthermore, they found that integrating SSL/TLS with HTTP Strict Transport Security (HSTS) improves overall system security. Perugini et al. [313] proposed extensions to the TLS 1.3 handshake to support two new Self-Sovereign Identity (SSI) authentication modes tailored for IoT systems.

Kumar [314] highlighted the need for lightweight cryptographic implementations that balance security with performance in resource-constrained devices. The study identifies challenges in efficient key management and scalability, particularly for large-scale IoT deployments. A comprehensive survey by Al-Turjman [315] underscored the absence of standardised frameworks for integrating post-quantum cryptography into current TLS/SSL protocols, an important consideration for future-proofing the healthcare IoT against quantum computing threats. Sharma et al. [316] compared TLS 1.3 and TLS 1.2, highlighting the benefits of the former in terms of reduced handshake time and improved energy efficiency. However, the study also notes limited research into the compatibility of TLS 1.3

with legacy healthcare IoT devices. Importantly, these studies do not adequately explore the trade-offs between security overhead and real-time data transmission requirements in healthcare applications.

Although traditional security approaches such as TLS [228, 317, 318] offer adequate protection for MQTT communication, ease of implementation and system interconnectivity remain essential. Nevertheless, SSL/TLS-based security solutions often impose substantial resource demands. The overhead associated with handshake operations and the complexity of certificate management make TLS suboptimal for resource-limited IoT devices using MQTT [28]. Although iTLS benefits from previous key establishment strategies, conventional TLS implementations involve multiple handshake stages and frequent cryptographic computations, leading to increased latency and communication overhead, particularly concerning for time-sensitive healthcare applications. Furthermore, message size increases due to encryption, potentially impacting network bandwidth usage and efficiency.

3.2.2 Authentication-based Solutions

Implementing lightweight encryption authentication schemes within MQTT has emerged as a viable approach to mitigate the performance constraints of traditional security frameworks in IoT environments.

Seker et al. [319] developed a role-based Mutual Authentication and Authorisation Scheme (MARAS) for lightweight IoT applications. MARAS provides access to topics and client publish or subscribe permissions through the Open Authorisation 2.0 (OAuth 2.0) protocol. However, MARAS uses multiple security schemes to protect data confidentiality, integrity, and availability in lightweight IoT applications. However, since MARAS uses a central key management system, it can lead to the possibility that all keys are compromised when the central key management system is attacked. Second, using a trusted third party to manage keys also carries the risk of compromising keys. Heartbeat recognition with authentication is emerging due to the reliable modality [320].

A promising biometric trait is the recognition of heartbeats that precisely capture

cardiac gestures using non-invasive measurement strategies, such as Electrocardiogram (ECG), Seismocardiogram (SCG), Photoplethysmogram (PPG), and Phonocardiogram (PCG). These signals are measured from the body surface of the individuals. These signals reflect factors of sympathetic and parasympathetic interaction within the body of humans [321]. It is very complex and challenging to control or modify fingerprints and faces, the heart waveform in conventional biometric traits, which is highly possible in heartbeat recognition. Conventional certificate-based authentication strategies are heavy in resource-limited IoT. An identity-based authentication scheme allows devices to exploit the identity of users as a public key and generates a private key [322].

A group-based authentication scheme in [323] involves access requests generated from multiple devices with limited resources on the Internet of Things. Group authentication schemes allow multiple users or devices to establish a shared key, resulting in minimal overhead [323–325]. Users or devices within a group can encrypt and decrypt the information using a shared key while ensuring security [326]. The group shared key has two steps: generating and distributing the pre-shared and final key [327]. Dynamic joining and exit processes in this group authentication can minimise the cost by enabling users to switch among various [328].

Alshahrani and Traore [329] developed a lightweight and reliable mutual authentication scheme for smart home IoT environments through cumulative keyed hash chains. It helps authenticate the sender's identity using an incremental key hash chain. This work validates the suggested security scheme using the Burrows-Abadi-Needham logic. However, it fails to secure the MQTT communication as the MQTT broker does not implement the TLS mechanism, leaving IoMT-related critical vulnerabilities. Haripriya et al. [330] demonstrated this problem in their research: they proposed a lightweight Intrusion Detection System (IDS) to analyse the scheme proposed in [329], which is vulnerable to a spoofing attack. Haripriya et al. also demonstrated the security shortcomings of MQTT brokers that do not implement TLS, particularly in IoMT environments. MQTT with a cumulative key hash fails to differentiate standard message packets from spoof messages. Therefore, an attacker can access the MQTT client information. In addition, the vulnerability of the TCP layer, such as SYN flooding, motivates the attackers to inter-

cept communications between publishers and subscribers, constructing several half-open TCP sessions. This open session exposes serious IoMT security risks by compromising data integrity and network stability. It has unnecessarily depleted broker resources [331]. Thus, developing a robust security framework with less complexity is essential and highly suitable for tiny IoMT devices. This approach is crucial to accomplishing the essentials of balanced security with the operational constraints of small-scale devices.

In addition, group-based authentication is highly suitable for distributed and scalable healthcare IoT. Technological breakthroughs bring an emergency in biometric determination [332]. This technique uses biometric sensors, such as fingerprint scanners, voice recognition, facial recognition cameras, and microphones, to collect biometric information [333]. Biometric-based authentication strategies are highly effective when conventional password-based authentication is impractical or less secure [334,335]. An improved lightweight user authentication scheme [336] has been proposed to ensure security in the Internet of Medical Things. However, biometric authentication poses privacy concerns and requires high initial costs for deployment [337]. It is more vulnerable to spoofing attacks that damage the entire system.

Haleh et al. [338] proposed a small scheme for IoMT, a lightweight authentication and key management model to balance secure communication and resource utilisation. However, challenges to handling security without overburdening the capabilities of constrained IoMT devices continue to be overcome. Slight has improved the LACO model [339] by incorporating measures that notably reduce computational overhead and strengthen countermeasures against IoMT security attacks that are malicious sensor and server emulation. However, a greater degree of optimisation is crucial to adopting these measures completely for resource-limited IoMT scenarios.

A multi-factor authentication scheme [340] for IoT-based healthcare services, improving security against impersonation and replay attacks. However, their approach lacked an in-depth evaluation of computational overhead and its impact on resource-limited medical devices. The work in [341] proposes an improved ECC-based authentication and encryption framework for medical sensor data from IoT. It demonstrates enhanced security and efficiency. However, it lacks scalability for large-scale healthcare IoT environments com-

prising emerging threats. A secure three-factor authentication scheme [342] leveraging IoT-enabled devices for healthcare applications. While their solution strengthened security by integrating biometric authentication, the energy consumption of IoT devices under this scheme remained unexplored.

A lightweight and anonymity-preserving user authentication scheme was developed in [343] for H-IoT. Their method effectively reduced authentication delays, but the system's robustness against emerging cyber threats was not extensively analysed. A secure and efficient authentication and authorisation architecture using smart gateways in the H-IoT has been proposed in [344]. Their model enhanced secure data access and reduced latency, yet it lacked a comparative analysis with contemporary SSL/TLS encryption-based security frameworks.

3.2.3 Confidentiality-based Solutions

Data confidentiality-based solutions ensure that authorised users can only access the data, protecting against unauthorised access and information disclosure. Attribute-Based Encryption (ABE) are widely used to ensure data confidentiality [345].

Mendoza-Cardenas et al. [346] proposed a security framework that uses CP-ABE encryption to protect MQTT-based IoT applications. The framework uses CP-ABE to encrypt data sent over the MQTT network. The encryption key is associated with attributes; only a user with the correct attributes can decrypt the data. Although this scheme ensures data authorisation and confidentiality, the CP-ABE encryption scheme requires multiple cryptographic operations. This approach leads to high computational costs, especially on resource-constrained devices like the Raspberry Pi. Using smaller key hierarchies or more efficient key-exchange protocols can reduce the load and performance overhead of the system application.

Xiao et al. [347] proposed an IoT security architecture based on the Triple Data encryption standard-Rivest Cipher 4 (3DES-RC4) hybrid secure encryption algorithm while communicating instantly based on the MQTT protocol. The main objective of this security framework is to protect sensitive information about students from unauthorised

access. This solution uses encryption techniques such as 3DES-RC4 to ensure data is in transit and at rest. The authorisation framework improves the flexibility and security of the security framework by using the combination of RBAC and ABAC. In the university mental health education system, system administrators define user and authorisation roles based on different attributes. However, since the system uses RC4 encryption, the RC4 algorithm is vulnerable to security attacks [348]. Second, the system should be equipped with an audit mechanism to detect any vulnerabilities in the system.

A Privacy Protector [349] is a privacy-preserving data collection framework for IoT-based healthcare systems. It exploits encryption and differential privacy techniques to safeguard patient data. However, the study did not assess computational overhead and scalability in real-time healthcare settings. An optimised security strategy [350] for IoT devices focusing on confidential healthcare data exchange. Incorporate advanced cryptographic techniques to ensure secure communication. Despite its effectiveness, it lacks an in-depth analysis of energy efficiency and latency, critical factors in real-time patient monitoring systems. The work in [351] is an IoT-based anonymous function to enhance security and privacy in healthcare sensor networks. Specifically, it focuses on anonymising patient identities while ensuring secure data transmission. However, it lacks strong security against emerging threats.

3.2.4 Hybrid Solutions

MQTT uses many other approaches to secure communication, except the three encryption algorithms mentioned above; for example, blockchain technology is often used to implement a secure MQTT framework. Refaey et al. [352] proposed a security framework based on the MQTT protocol, based on the Symmetric Key Distribution Protocol (SDP) [353], which provided many security features such as privacy, data encryption, and message authentication. Password login methods prevent attackers from attempting to access the end device by changing the login credentials. However, the framework has some limitations. For example, the authentication credentials depend on the SDP controller, which acts as a trusted third party and affects the whole framework when the

controller is compromised. Compared to our proposed ICP-ABE framework, which is proposed in Chapter 7, the SDP scheme lacks a user privacy protection mechanism. The SDP framework requires a shared key to be shared between publishers and subscribers so that subscribers can receive the publisher's authentication information. It means that publishers cannot anonymise themselves to protect their privacy.

Hamad et al. [354] proposed a security framework for IoT applications based on the Pub/Sub communication model. Secure End-to-End MQTT (SEEMQTT) is efficient, lightweight and suitable for resource-constrained IoT devices. SEEMQTT is designed to split keys into multiple parts and store them in different key stores. The key management of SEEMQTT is based on secret sharing and delegation of trust. Identity-Based Encryption (IBE) can be implemented in SEEMQTT by generating password keys using Trusted Third-Party (TTP) encryption technology [355], creating public keys for each subscriber and keeping private keys corresponding to those keys. The publisher encrypts the message with the subscriber's public key, and the subscriber decrypts the message using the private key. Although IBE provides a higher level of security, TTP must be trusted. Furthermore, the security of SEEMQTT is not as flexible as other security frameworks. For example, shares cannot be withdrawn once issued. If the KeyStore is updated, it will be compromised if the shares are compromised.

Fan et al. [356] proposed an MQTT encryption scheme based on hierarchical identity encryption called MQHIBE. Hierarchical ID-Based Encryption (HIBE) is an Identity-Based Encryption (IBE) that allows the assignment of encryption keys. In MQHIBE, the hierarchy of encryption keys is represented by a unique public key and several private keys for each user that can be used to decrypt encrypted messages using the public keys of the ancestors of the hierarchy. The main drawback of this scheme is the significant issues of escrow and forward secrecy. The trusted third party that generates the HIBE encryption keys holds the master key. Trusted third parties can decrypt all encrypted messages using HIBE. If a trusted third party is compromised, HIBE can be vulnerable to attacks. Second, if a master key is compromised, all messages encrypted in the past can be decrypted. Buccafurri and Romolo [357] proposed a blockchain-based security framework for resource-constrained devices. This framework uses blockchain and One-

Time Password (OTP)s to authenticate clients who want to connect to the broker.

Specifically, when a client wants to connect to the broker, it should first send a connection request to the broker. Then, the broker uses blockchain technology to verify the client's OTP, and if the OTP is valid, the broker confirms that the client can connect. There are many advantages to using blockchain technology for authentication. For example, the OTP is a one-time code from a trusted third party that cannot be tampered with. Second, the distributed structure of the blockchain makes it less prone to single points of failure. However, we must consider factors such as the size of the blockchain, the number of communications, and the bandwidth of the device before designing a security framework because all of these factors will affect the performance and communication conditions of the device.

Noguchi et al. [358] proposed a lightweight and secure key sharing system based on a secret sharing scheme for resource-constrained IoT devices. The system uses a threshold secret sharing scheme (k, n) to share secret keys among IoT devices. The (k, n) threshold secret sharing scheme allows one to share a secret among n participants so that any k of the n participants can reconstruct the secret, but fewer than k participants cannot. Since secret sharing is small, the scheme is well-suited for resource-constrained devices. However, the key-sharing system is vulnerable to internal attacks because it requires a central server to store and manage secret sharing, which can lead to a single point of failure.

The work in [359] proposes a novel hybrid encryption method for securing healthcare data in IoT-enabled infrastructures. It integrates lightweight cryptographic techniques to balance security and efficiency. However, it fails to analyse its impact on resource-constrained IoT devices and real-time performance evaluations. An optimised hybrid encryption framework in [360] tailored for smart home healthcare applications. Improve data confidentiality and security while ensuring optimised computational efficiency. However, it extensively addresses key management complexities and scalability in large-scale IoT healthcare networks.

3.3 Comparative Analysis of MQTT Security Solutions

The review, particularly the comparative analysis of security solutions discussed in Table 3.1, has advantages and limitations in both cryptography and security requirements.

Table 3.1: Comparison Table of Related Works.

Type	Cryptography Solutions	Key Storage	Speed	Security Level	Key Length	Key Management	Complexity	Authentication	Confidentiality	Integrity	Attack Vulnerability	Computation Cost	Energy Consumption	Overhead	Performance Level
Symmetric	[252]]	Y	L	M	S	Y	M	N	Y	N	H	M	M	M	L
	[254]]	Y	H	H	S	Y	H	N	Y	N	L	H	H	M	L
	[256]]	Y	H	M	S	Y	H	Y	Y	Y	M	VH	H	VH	M
	[258]]	Y	L	M	S	Y	M	N	N	N	H	H	M	VH	M
	[259]]	Y	L	H	S	N	M	N	Y	N	H	H	M	VH	H
Asymmetric	[286]]	N	L	H	Lo	N	H	Y	Y	Y	L	H	H	M	H
	[287]]	Y	M	H	S	N	H	N	Y	Y	H	H	H	H	H
	[288]]	Y	L	M	Lo	Y	VH	N	N	Y	L	H	H	VH	H
	[289]]	Y	M	H	Lo	Y	VH	N	N	Y	M	H	H	VH	H
	[290]]	N	L	H	Lo	Y	H	N	N	Y	L	H	H	H	H
SSL/TLS	[310]]	D	L	H	Lo	Y	M	Y	Y	Y	H	M	H	H	M
	[30]]	Y	L	M	S	N	M	N	N	N	M	H	H	H	M
	[312]]	D	L	H	Lo	N	H	Y	Y	Y	H	VH	VH	VH	H
Authentication	[319]]	N	H	M	N	N	M	Y	N	N	H	M	M	L	M
	[320]]	N	M	H	N	N	L	Y	N	N	H	M	L	L	H
	[323]]	N	L	H	N	N	H	Y	N	N	VH	M	H	M	M
Confidentiality-ABE	[346]]	Y	L	H	S	Y	M	N	Y	N	M	M	M	M	H
	[347]]	Y	M	H	S	Y	M	N	Y	Y	H	M	M	H	M

Continued on next page

Table 3.1 – continued from previous page

Type	Cryptography Solutions	Key Storage	Speed	Security Level	Key Length	Key Management	Complexity	Authentication	Confidentiality	Integrity	Attack Vulnerability	Computation Cost	Energy Consumption	Overhead	Performance Level
Hybrid	[354]]	D	M	VH	D	Y	M	N	Y	N	M	H	H	VH	H
	[356]]	D	M	VH	D	Y	M	N	Y	N	M	VH	H	VH	H

• Y-Yes, N-No, L-Low, M-Medium, H-High, VH-Very High, D-Dynamic, Lo-Long, S-Short.

3.4 Current Lightweight Security Solutions

The constantly evolving IoT system increases the demand for lightweight and robust security solutions compatible with devices that are limited in resources. MQTT is the widely exploited IoT protocol, and conventional, standard and robust cryptographic solutions are unsuitable for IoT environments due to resource-intensive operations [361]. Hence, there is a crucial need for lightweight security solutions, as they can rectify MQTT security problems with a minimum level of resource consumption [362].

This lightweight encryption strategy offers the device or user a secure way to establish communication among MQTT clients and the server. This lightweight algorithm protects IoT devices from eavesdropping and interception in an MQTT session. Several lightweight cryptography solutions are available [363], broadly classified as asymmetric and symmetric [364]. Asymmetric key cryptography strategies utilise different keys for the encryption and decryption of messages, whereas symmetric algorithms exploit similar keys. Symmetric key cryptography algorithms have a higher operating speed and a lower cost, which is more beneficial in resource-limited real-time IoT applications than asymmetric key cryptography strategies.

However, they incur complexity regarding key management due to providing different key pairs to IoT users. The performance of a symmetric key algorithm only depends on the particular types and requirements of the applications, such as the mandatory

level of security, the availability of resources from IoT devices, and the compatibility with the exploited MQTT implementation. It is crucial to carefully evaluate the available options to select an algorithm that offers the required security level with minimum resource consumption. Selecting lightweight symmetric key solutions based on application type and requirements is an excellent choice to improve security and efficiency.

In a study by Vaccari et al. [365], the performance of MQTT was analysed during a DoS attack communication model. The authors propose a slow-DoS attack known as SlowTT, which exploits vulnerabilities in MQTT configuration parameters and packet vulnerabilities. SlowTT can stay connected to the client for a long time and perform attacks on the IoT environment. The execution of SlowTT is related to the fact that MQTT lacks a built-in security authentication mechanism and the security specification of the MQTT proxy, indicating the need for more advanced security mechanisms to maintain secure communication in MQTT. Ahamed et al. [366] analysed the performance of MQTT in the presence of DoS attacks on the network and used AES for security configuration. Their scheme, which employs the Advanced Encryption Standard with 256-bit Key Size (AES-256) and SHA-256 Secure Hash Algorithm 256-bit (SHA-256) algorithms, demonstrated fast response times and fewer computational operations. However, it also consumed more memory than schemes based on Rivest–Shamir–Adleman (RSA) and DES algorithms. Although the study evaluated AES with various key sizes, it did not consider IoT-specific information, such as network nodes and traffic levels. Therefore, evaluating different cryptographic algorithms in various IoT scenarios is crucial to determining the most suitable scheme for MQTT.

Salim et al. [367] proposed a Lightweight Key Agreement (LKA) scheme to perform secure and efficient authentication with low power consumption and communication overhead in resource-constrained e-health systems. The scheme uses an edge network key manager to generate keys for authentication certificates, thereby reducing communication cost and latency between wearable medical devices. However, although the LKA scheme evaluates and analyses some performance indicators, it lacks a comprehensive security analysis of various attack vectors (such as replay attacks and/or MitM attacks). Secondly, there is no discussion of scalability in large-scale deployments, which is critical for

widespread adoption in healthcare systems.

Besides, many existing models lack a compelling comparative analysis of the lightweight protocols in MQTT in theory and simulation. Therefore, our work aims to evaluate the security performance of MQTT against DoS and MitM attacks on resource-constrained IoT devices using various lightweight symmetric key security protocols. Table 3.2 compares various lightweight security schemes with their advantages and limitations.

Table 3.2: Existing MQTT Security Algorithms with Advantages and Limitations.

Scheme	Description	Advantages	Limitations
[250] Lightweight cryptography protocols survey	Aims to evaluate cryptographic algorithms to analyse their suitability for different IoT platforms.	Assists in proving the suitability of symmetric key algorithms over IoT scenarios.	It analyses symmetric key cryptography on IoT platforms, but the protocols are not specified.
[251] QUIC based MQTT	Analyses the MQTT with quick UDP internet connections and improves security.	Reduces the packet processing cost and delay with improved performance.	Bandwidth utilisation problems and high energy consumption.
[262] SI-MON and SPECK	Lightweight encryption algorithms evaluated under MitM attack.	Low battery/memory usage, decent security.	No S-box, costly plaintext/ciphertext operations.
[265] ChaCha20-Poly1305 AEAD	Proposed a lightweight security scheme based on ChaCha20-Poly1305 AEAD Algorithm for MQTT/MQTT-SN.	It is high-performing and provides dual authentication and encryption.	Less secure and complex to implement.
Continued on next page			

Table 3.2 – continued from previous page

Scheme	Description	Advantages	Limitations
[267] PRESENT-MQTT	Designed a lightweight structured cryptography protocol.	Minimum level of security and high performance.	Lack of security against the latest attacks.
[268] KSA-PRESENT	Scheduling-based PRESENT algorithm.	Enhanced security reduces the avalanche effect.	High complexity and overhead.
[269] Aug-PAKE and PRESENT	Attribute-based access control with multiple authorities.	Small S-box, secure session creation.	Vulnerable long-term keys, lacks evaluation in hostile environments.
[280] MQTT-RSA-ECC	Presented ECC-based security without key revocation.	Medium security.	Poor performance, high delay and increased energy consumption.
[291] ECC-Based Scheme	Lightweight ECC Scheme with disinfection concept.	Simplified key management, privacy-preserving.	Frequent key revocation, impractical key transmission.
[293] ECC with PRESENT	Authentication system for IoMT.	Balanced security with lightweight design.	Key length trade-offs, privacy challenges in pandemics.
[307] MQTT performance evaluation	Aims to evaluate the performance of different lightweight algorithms.	Energy and memory-based analysis assist in the selection of suitable algorithms.	Lacking to analyse the algorithm performance against DoS and MitM.
[308] TLS-based MQTT	Implements TLS-based MQTT protocols for security enhancement.	High security, easy to implement, and easily integrated with existing infrastructures.	Increases complexity, high energy consumption, and high overhead due to TLS handshaking.
Continued on next page			

Table 3.2 – continued from previous page

Scheme	Description	Advantages	Limitations
[329] Keyed Hash Chains	Mutual authentication using hash chains.	Sender authentication, BAN logic validation.	No TLS, vulnerable to spoofing [330].
[338] LACO Model	Lightweight authentication and key management.	Reduced overhead, defence against attacks.	Needs further optimisation for constrained devices.
[366] AES-based MQTTj	To analyse AES-based MQTT performance under the DoS attack scenario.	Fast response and fewer computational operations.	Consumes high bandwidth resources and energy.
[366] AES-256-based MQTT security	Aims to attain secure MQTT data transmission by using AES-256 and SHA-256.	Secure MQTT data transmission through a three-layer security model.	High overhead and computational cost.
[317] TLS-based MQTT authentication	Intends to utilise the TLS authentication for MQTT security.	Improves the authentication-related security issues for hardware applications.	Not fit for providing security in the client-server communication.
[285] Lightweight ECC-based MQTT	Proposed a lightweight scheme with authentication and key management.	High security.	High computational cost of encryption algorithms.
[367] LKA Scheme	LKA-based authentication scheme uses the network key manager on the edge to build keys for each device for device validation.	Reduced operation time for key generation and lower communication costs.	Lack of scalability and security evaluation.

3.4.1 Advantages of Lightweight Cryptography

Significant advantages of the lightweight cryptography strategies currently available lie in their ability to offer adequate security levels without consuming large amounts of resources, maintaining low latency and ensuring high-performance efficiency [368]. This capability makes lightweight solutions particularly well-suited for the growing MQTT-enabled IoT landscape. The increasing demand for secure, effective, and cost-reduced solutions has heightened the need for lightweight symmetric key cryptography, which is increasingly critical to ensuring modern IoT security.

Efficient Resource Utilisation [302]: The design of lightweight cryptographic solutions enables them to operate with less computational resources, memory, and energy consumption. This property makes them ideal for resource-constrained IoT Sensor, Radio-Frequency Identification Tags (RFID), Implants, and Wearable.

Low Latency [369]: Optimised lightweight design ensures low latency in security provisioning. This lightweight security is essential in time-critical applications, such as real-time data exchange and diagnosis in Healthcare 4.0.

Strengthen Adequate Security Level [302]: Lightweight cryptography solutions can provide an adequate level of security sufficient to protect messages and information in resource-limited IoT-enabled application environments.

Performance Enhancement for Small Data Sizes [370]: Several lightweight cryptography algorithms are optimised for small data sizes in general IoT applications. The lightweight key designs of these algorithms boost the speed of encryption and decryption, making them more effective for healthcare IoT scenarios, especially in critical patient monitoring.

Cost-Effectiveness [371]: By minimising the need for powerful hardware and extensive computational resources, lightweight cryptographic strategies can minimise the cost of IoT system deployment. This approach benefits the healthcare industry, which seeks to implement security with existing infrastructure at low costs.

Ease of Integration [370]: The simplicity property of lightweight cryptography strategies allows easier integration into existing infrastructure without requiring high cost.

Their modular design means that they can be easily integrated into various available hardware and software platforms without extensive modification.

3.4.2 Shortcomings of the Lightweight cryptography

Although current lightweight cryptography solutions offer numerous benefits, their performance is limited in different aspects discussed below [372].

Limited Scalability [373]: Innumerable interconnected IoT devices evolve day by day. Here, it is vital for scalable and lightweight security solutions. Many lightweight cryptographic solutions are not scalable, particularly in MQTT-enabled Healthcare 4.0. The growing number of devices demands high computational power and energy, potentially overwhelming the entire IoT system.

Balancing Security and Performance [370]: Conventional lightweight solutions fail to balance security strength and performance efficiency, an ongoing challenge. Generally, the design of lightweight cryptographic solutions sacrifices the security level to a minimum to achieve high speed and less resource consumption, making the system vulnerable to emerging threats.

The Trade-off between Computation Cost and Efficiency [370]: In lightweight cryptography-based solutions, the primary intention is to reduce the computation cost to fit devices with restricted resources. However, shrinking computation costs frequently means using more straightforward solutions or algorithms, which can potentially weaken the security and performance of MQTT communication.

Integration with Existing Systems [374]: Although the design of lightweight cryptography is simple, integrating these solutions with traditional infrastructure poses different compatibility challenges. However, this integration is pivotal to ensure backwards compatibility and make deployment easy.

Adaptive Security Solutions [375]: Current lightweight security strategies often pose adaptability issues, and it is essential to respond quickly to evolving threats in dynamic healthcare IoT environments. The capability of adjusting security safeguards in real-time, according to the context of threats, is an active area for further exploration.

Comprehensive Evaluation Metrics [376]: The lightweight cryptography solutions lack standardised evaluation metrics, including security and efficiency, resource consumption, delay, and scalability. This process makes it tedious to evaluate and compare the effectiveness of various algorithms.

Privacy Preservation [376, 377]: It is overlooked to ensure privacy in lightweight cryptographic solutions, especially in healthcare IoT environments where sensitive information is frequently transmitted. Privacy-preserving strategies are crucial for lightweight security solutions without extra overhead.

3.5 General Research Gaps in MQTT Security and Discussion

In Sections 3.1 to 3.4, we discuss existing security schemes from several aspects, including the trade-offs between symmetric and asymmetric encryption schemes, TLS overhead and limitations, general issues in lightweight encryption schemes, and implementation limitations in the IoT/IoMT. Based on the previous review, this section outlines the general research gaps in the security of MQTT-enabled IoT and Healthcare 4.0 environments. We find several common issues below.

- **Symmetric encryption limitations:** In MQTT-driven healthcare IoT systems, most of the existing security solutions exploit symmetric key cryptography or lightweight encryption owing to resource constraints of tiny devices. At the same time, only a few include asymmetric solutions due to their high computational demands despite offering strong security [368, 378]. Although lightweight encryption guarantees adequate efficiency, it remains vulnerable to attacks such as MitM. This escalates concerns about the limitations of current solutions in protecting sensitive healthcare data and underlines the critical gap in ensuring end-to-end data confidentiality and integrity. Hence, it is necessary to examine the performance and resilience of current lightweight cryptographic algorithms under different attack scenarios to develop the most suitable security solutions for Healthcare IoT.
- **TLS and hybrid encryption challenges:** Many solutions use TLS to provide

strong encryption to consider transmission security, but are too resource-intensive for low-power devices, failing to balance efficiency with confidentiality, especially in time-sensitive applications such as Healthcare [299]. In addition, the high computational complexity or energy consumption of these conventional solutions does not comply with the lightweight nature of the MQTT protocol. Providing end-to-end security in MQTT-based data transmissions through lightweight cryptography solutions is crucial to seamlessly enhancing IoT performance.

- **Insufficient key management:** Key exchange and key management mechanisms are also essential components of MQTT security systems and allow communication devices to establish shared keys that can be used to encrypt and decrypt messages [379,380]. However, conventional solutions like CP-ABE-based authentication schemes are highly resource-intensive and struggle with scalability problems when the number of attributes grows. They are also absent to facilitate efficient key management and self-key revocation strategies, leaving MQTT-enabled healthcare IoT systems vulnerable to key compromise. In addition, the exploitation of digital certificates or biometric data increases privacy concerns, and most conventional solutions overlook authentication on the publisher side. These gaps highlight the demand for a lightweight, scalable, and CP-ABE-enabled authentication solution with dynamic key management tailored for healthcare IoT systems.
- **High computational complexity in CP-ABE:** While CP-ABE provides fine-grained access control by enforcing encryption policies over user attributes, its performance degrades significantly as the number and complexity of attributes increase. CP-ABE is challenging to implement in resource-constrained IoT devices standard in Healthcare 4.0.
- **Authorisation and access control mechanisms:** Authorisation and access control mechanisms allow users to authenticate before using a device for communication connections to verify that the user has the right permissions. Traditional access control models such as RBAC are also unsuitable for resource-constrained IoT devices common in Healthcare 4.0 [381], as they require a central authority to manage role hierarchies and permissions, adding communication and processing overhead.

In lightweight schemes, the central authority that manages roles and authorisation cannot allocate too many resources to network tracking devices. For example, granting publish permissions for specific MQTT topics becomes inefficient when relying on role-based structures that do not scale well in lightweight or decentralised systems [382]. Therefore, resource-efficient and scalable authorisation solutions are necessary to support secure data exchange in IoT-enabled medical applications.

- **Limited scalability and adaptability:** Many solutions consider transmission security, but ignore the consideration of multiple requirements regarding confidentiality, integrity, and authentication. Most solutions do not scale well or adapt well to varying device capabilities and network conditions, which are common in Healthcare 4.0 [299]. In addition, the high computational complexity or energy consumption of these conventional solutions does not comply with the lightweight nature of the MQTT protocol. Providing end-to-end security in MQTT-based data transmissions through lightweight cryptography solutions is crucial to seamlessly enhancing IoT performance.

3.5.1 Summary

Although researchers have proposed numerous general cryptographic strategies, many fall short in constrained environments like Healthcare 4.0 due to performance overhead, inflexible key management, and lack of simulation validation. These issues highlight the need for more lightweight and adaptive security schemes, especially those supporting real-time control, efficient encryption, and fine-grained access control, while feasible in real-world MQTT deployments. In addition, a detailed analysis of the performance of various MQTT lightweight cryptography algorithms under various scenarios is crucial to choosing a more efficient security algorithm to meet the application's specific requirements, network conditions, and the security needs of the IoT environment. Therefore, a comprehensive analysis of the performance of different symmetric key cryptography algorithms for MQTT is essential for different networks and attack scenarios.

3.6 Related Work on Lightweight and Attribute-Based Encryption Schemes for MQTT and IoMT

The proliferation of the IoMT has revolutionised healthcare systems, enabling real-time monitoring, remote diagnostics, and the management of Electronic Health Records (EHRs). However, the sensitive nature of medical data demands rigorous security and access control mechanisms. Traditional access control models such as RBAC and ABAC often fail in dynamic, decentralised, and resource-constrained environments typical of IoMT. To address these limitations, ABE has emerged as a powerful cryptographic tool that supports fine-grained, scalable, and secure data access. This section explores lightweight and ABE's MQTT frameworks applications within healthcare and IoMT, evaluates performance trade-offs and implementation challenges in constrained devices, and contrasts ABE with traditional access control frameworks.

3.6.1 Applications of lightweight and ABE in Healthcare and IoMT Environments

Many conventional solutions like [383] rely on ABE algorithms instead of complex security schemes. For example, Mendoza-Cardenas et al. [346] intended to evaluate the performance of MQTT with CP-ABE and its adoption in an IoT environment. Wang et al. [384] developed a Dual Policy Attribute-Based Encryption model (DP-ABE) that allows two access control strategies and ensures high security in cloud scenarios. Hence, enabling more nuanced permissions in cloud-integrated IoT architectures. However, it increases the computational cost and overhead in the network.

While DP-ABE provides a promising direction for improving granular control and adaptability in data-sharing scenarios, its integration into resource-constrained IoMT environments reveals substantial limitations. For example, using multiple policies inherently increases computational complexity, particularly during the encryption, decryption, and key generation phases. This limitation poses a significant challenge for low-power medical devices, such as wearable sensors, implants, and mobile health monitors, where energy ef-

efficiency and response time are critical to system viability. Additionally, DP-ABE schemes typically rely on more intensive cryptographic operations, often requiring pairing-based cryptography and complex key structures, increasing latency in data transactions. For critical IoMT operations, such as patient monitoring in real time, such delays could be detrimental to the effectiveness and safety of the system.

Elbanna [385] developed a secure MQTT scheme based on attribute access control supporting CP-ABE using improved ABE and chaotic synchronisation. It allows Trusted Authority (TA) to handle user credentials efficiently. However, it meets single-point failure issues in many scenarios. However, additional schemes in ABE-based MQTT may need to be better suited for devices with limited resources. Existing security schemes focus on either communication security or computational overhead. Finally, CP-ABE-based security schemes could be more efficient for large-scale IoT applications. However, the number of attributes increases the key size and resource consumption, which are unsuitable for resource-constrained IoT devices. Therefore, a lightweight authentication scheme for MQTT should be designed without sacrificing privacy and scalability while minimising computational overhead and cost.

Compared to high-complexity security schemes, the CP-ABE scheme is widely used among lightweight IoT devices [386]. However, it generally requires longer secret keys that increase the key length proportionally according to the number of involved attributes. This approach incurs higher computational overhead and demands large storage, posing vital implementation challenges in resource-constrained IoMT devices. Ling et al. [386] introduced a multi-authority CP-ABE framework to improve key management and policy enforcement. Implemented multiple authorities in place of a single attribute authority to enhance the security of MQTT. Although it improves the security of MQTT communications, this approach is unsuitable for devices with limited resources. In the multi-authority environment, limited randomness permits malicious users with similar attributes to compromise and trace the secret keys of others. Thus, it leads to significant security risks for IoMT, often requiring attribute-based access control.

It is essential to highlight that although there are numerous security measures for MQTT over IoT, more research is needed, exclusively dedicated to MQTT security for

IoMT applications. In IoMT, intelligent healthcare devices interact with each other through a communication protocol such as MQTT to share patient data.

Also, the integration of ABE into healthcare and IoMT systems is increasingly recognised for enhancing data confidentiality and access control. One of the key applications is the secure sharing of patient-centric health data across decentralised platforms. For example, Xiang et al. [387] proposed a Patient-Centric and Trusted Data Access (PCT-DA) model that utilises ABE within a blockchain-enabled IoMT framework to ensure secure and efficient data sharing among patients and healthcare providers. Their system combines cryptographic strength with decentralised verification, enabling privacy-preserving and tamper-proof data exchange.

Similarly, Guo et al. [388] developed a hybrid blockchain-edge architecture to manage EHRs, integrating multi-authority ABE and homomorphic encryption to enhance data confidentiality and patient anonymity. The architecture supports decentralised access control policies while leveraging edge computing for scalability and responsiveness. Zhou et al. [389] introduced a privacy-preserving, searchable ABE framework tailored for IoMT environments, wherein authorised users with appropriate attribute sets can search and retrieve encrypted data from distributed nodes. This system improves efficiency and usability by enabling keyword-based encrypted searches without compromising data integrity. Yang et al. [390] addressed the issue of centralised trust in ABE systems by proposing a revocable CP-ABE scheme with multiple authorities implemented via blockchain. This approach distributes trust, minimises single-point failure risks, and allows dynamic revocation and reissuance of user credentials.

These advancements demonstrate that ABE is technically feasible in IoMT settings and adaptable to real-world requirements such as patient-controlled data sharing, multi-tenant hospital environments, and decentralised trust models.

3.6.2 Performance Evaluations and Limitations of ABE Schemes in Constrained Environments

IoMT provides many conveniences to patients and medical professionals. For example, medical professionals can receive data whenever needed, and IoMT enables physicians to offer correct telemedicine to patients anywhere based on their history. Several security schemes for data sharing have previously been developed. However, only some [391] performed well in the IoMT environment due to bilinear pairings, which need more efficiency. These high-complexity operations with high resource constraints must be more lightweight and sufficient. Provisioning of data integrity while ensuring user privacy is lacking in most of the existing IoMT schemes. To overcome this problem, Shen et al. [392] have developed an efficient security scheme for healthcare to address critical vulnerabilities effectively. Meanwhile, Zhang et al. [393] present an identity-based encryption strategy for securing healthcare information with MQTT servers, which may incur additional complexities in a resource-constrained IoMT environment.

In terms of performance, CP-ABE-based solutions face considerable challenges when deployed in resource-constrained devices standard in IoMT. The primary bottlenecks stem from the exponential growth in key length and encryption/decryption time as the number of attributes increases. For example, the RSA-based CP-ABE scheme is implemented in [394] with secret keys of constant size and without using bilinear maps to reduce the complexity of RSA. However, this solution needs to address the key revocation issue. Singh et al. [395] developed SMQTT by implementing Key Policy Attribute Based Encryption (KP-ABE) and CP-ABE separately using lightweight ECC. However, it does not focus on key revocation and group Pub/Sub. The ECC scheme uses a secret key for a long lifetime, which could weaken its security in a hostile environment.

Additionally, as the number of attributes increases, the length of the private key and the communication delay increase significantly. The SMQTT incurs high overhead and costs because of the incorporation of many attributes. In addition, SMQTT fails to meet challenges such as limited processing power, memory and energy resources and inevitably encounters compatibility issues whenever implemented with the MQTT-enabled

resource-limited environment. Liao et al. [396] have developed an improved ABE and chaos synchronisation scheme for MQTT. Incorporating additional heuristic and enhancement schemes in ABE-based MQTT results in a heavy security strategy unsuitable for IoMT. In addition, the increasing number of attributes increases both key size and resource consumption, further straining the capabilities of the IoMT device.

A comprehensive evaluation by Perazzo et al. [397] assessed the computational overhead of various ABE schemes deployed on IoT devices and concluded that encryption and decryption latencies, as well as memory usage, significantly impact device usability. The evaluation highlights that conventional ABE schemes are often unsuitable for low-memory or battery-operated devices. While ABE offers expressive access control, its performance on constrained devices such as sensors, wearables, and embedded medical hardware is a critical consideration.

To mitigate such constraints, Jebrane et al. [398] developed an enhanced and verifiable lightweight authentication protocol based on CP-ABE specifically tailored for IoMT. This protocol incorporates precomputation and modular operations to offload cryptographic overhead while maintaining core security guarantees. Their simulation results confirm reduced latency and energy consumption, making the scheme viable for real-time healthcare scenarios.

Bezerra et al. [399] modelled the availability and performance of IoMT systems using stochastic Petri nets, incorporating redundancy and fault tolerance strategies. Although not an encryption scheme per se, the modelling provided insight into how ABE can be systematically integrated with redundant IoMT architectures to ensure service continuity despite partial failures. These underscore the need to optimise ABE mechanisms for the IoMT domain.

3.6.3 Comparisons Between ABE-Based Lightweight Schemes and Traditional Access Control Mechanisms

Conventional access control frameworks like RBAC and ABAC have been widely used across various sectors, including healthcare. However, these models are often static and

hierarchical, making them inflexible for IoMT environments where users, roles, and permissions may change frequently. A comprehensive survey by Ahsan and Pathan [400] confirmed that traditional models lack support for scalability, attribute delegation, and decentralised enforcement—all essential in medical applications involving numerous stakeholders.

On the contrary, ABE provides cryptographic enforcement of policies, eliminating the need for continuous communication with centralised policy decision points. In their systematic review, Imam et al. [401] outlined how ABE schemes outperform RBAC/ABAC in terms of fine-grained access, dynamic user revocation, and reduced reliance on trusted third parties. Especially in multiinstitutional healthcare networks, ABE allows secure sharing of encrypted data across boundaries without revealing identity or access policies to intermediaries.

Guo et al. [402] addressed some of the inefficiencies in ABE by introducing offline/online ciphertext separation and chameleon hashing to reduce the computational burden during attribute/user revocation. Although effective in reducing online costs, it uses the advantages of the chameleon hash function to avoid causing a network collision. Most complex processes are performed offline using third-party servers, which may not be practical in latency-sensitive IoMT or some IoMT devices with resource and cost restrictions. Similarly, Sammy and Vigila [403] used a distributed blockchain-based cryptographic text policy attribute-based encryption scheme to avoid multiple authorities in their research. By decentralising authority management in the CP-ABE system, scalability is improved. However, it is affected by the length of the ciphertext, especially when the number of attributes increases, which means that the scheme will face the challenges of expanding the ciphertext and the complexity of the policy.

Hwang et al. [404] proposed a medical data sharing system based on the CP-ABE scheme in an IoMT cloud environment to address the key misuse problem based on access control techniques. However, a key escrow problem can escalate, indicating a lack of selection of key decryption for encrypted data held in escrow, especially due to the complexities of multiauthority-based IoMT communication. Most existing projects in IoT and IoMT face issues such as ciphertext length, short- and long-key issues, the need for

third-party support, and ineffective utilisation of randomness in the key generation, key renewal, and revocation scheme.

In summary, while ABE schemes offer important advantages for secure and flexible data sharing in healthcare and the Medical IoT, they usually incur significant computational and storage overheads. Therefore, our study aims to balance the expressive power of ABE with the operational constraints of constrained IoMT devices. In Chapters 7 and 8, we design two new schemes: ICP-ABE and OCP-ABE. These frameworks meet the dual needs of privacy and performance through lightweight encryption schemes and a hybrid model that combines traditional and attribute-based mechanisms to address latency, energy efficiency, and security issues in MQTT communications.

3.7 Research Gaps in Existing ABE-Based MQTT Security Schemes in IoMT

Combining the research and analysis of lightweight symmetric encryption schemes and existing ABE schemes in the previous chapters, although lightweight symmetric encryption has been proven to be useful for MQTT security, many ABE-based methods are still too resource-intensive for practical IoMT applications. We found the following gaps:

- **Computational Overhead and Resource Constraints:** ABE schemes such as CP-ABE often involve computationally expensive operations, including bilinear pairings, complex key derivation, and policy tree evaluations. In MQTT-enabled IoMT systems, including energy-constrained sensors and wearables, these overheads result in poor latency, increased energy consumption, and longer response times. Studies have shown that key sizes and encryption/decryption times increase significantly with the number of attributes, posing practical limitations on scalability in real-time healthcare applications.
- **Lack of attribute revocation mechanism:** Most existing ABE schemes lack efficient user and attribute revocation mechanisms. Revocation processes are often centralised, and rekeying entire networks can incur significant delays. In dynamic IoMT environments, where user roles, device states, and access policies frequently

change, revocation delays propagate unauthorised data access risks. Although some schemes introduce multi-authority models or chameleon hashes, they often increase system complexity and communication overhead, undermining their suitability for constrained medical devices.

- **Lack of Topic-Level Access Control for MQTT:** Fine-grained topic control is essential in Healthcare 4.0, where different data streams (e.g., ECG, temperature, medication) should be selectively accessible based on user roles or patient context. Most CP-ABE models operate on payloads or metadata without integrating topic-based encryption or publisher verification, reducing the effectiveness of access enforcement.
- **Limited Validation Under Realistic IoMT Conditions:** Many proposed schemes, including SMQTT-ABE, DP-ABE, and others, focus on theoretical security proofs or basic simulation environments. There is a notable lack of evaluation under realistic attack conditions, such as DoS and MitM attacks specific to MQTT and TCP/IP stack vulnerabilities. Furthermore, few frameworks provide simulation-based validation using IoMT-specific tools (e.g., Cooja, Contiki-NG) or consider practical metrics such as CPU energy consumption, message throughput, and delay.
- **Security Gaps in Publisher Authentication:** ABE solutions often authenticate subscribers but neglect publishers, allowing spoofed or malicious publishing if the broker's trust assumptions are compromised. In the context of IoMT, where sensitive medical data is transmitted in real-time, the absence of publisher-side verification can lead to severe data manipulation and misinformation.
- **Balancing Security and Lightweight Performance:** While many hybrid schemes propose enhanced security features, like chaos synchronisation or multi-authority ABE, these additions often result in heavy computation, key bloat, or ciphertext expansion. For instance, multi-authority schemes increase randomness but allow key tracing by adversaries with overlapping attributes. Hence, a trade-off between strong encryption and resource efficiency remains unresolved in most ABE-based models for IoMT.

In summary, the analysis of prior ABE-based MQTT security schemes reveals that high computational complexity, ineffective revocation, inadequate topic control, limited simulation validation, weak publisher authentication, and imbalanced security-performance trade-offs all hinder their practical application in IoMT environments. These challenges motivate the introduction of novel frameworks such as ICP-ABE and OCP-ABE, which aim to combine lightweight symmetric algorithms (e.g., PRESENT, Fast-PRESENT) with fine-grained, scalable attribute-based policies. These are introduced and evaluated in Chapters 7 and 8 under realistic network scenarios and medical application constraints.

3.8 Chapter Summary

Lightweight cryptography for MQTT significantly protects devices and internet-enabled communication protocols from various security attacks. Several MQTT works exploit the advantage of lightweight cryptography in MQTT [250, 285]. However, they face several limitations, including high overhead, security trade-offs, and inefficient key management in various IoT scenarios and applications [285, 308, 405]. Firstly, existing work adds more overhead due to complex encryption and decryption strategies [308]. Secondly, conventional lightweight cryptography does not provide effective security against evolving attacks effectively [366, 405, 406]. Finally, existing works cannot manage cryptographic keys in resource-limited environments, which can be very challenging [308, 405].

This literature review on MQTT security in IoT and IoMT classified various security schemes with potential strengths and weaknesses. It discusses the comparative evaluation of lightweight MQTT security solutions and ABE schemes in IoMT environments in detail. Finally, the research gaps are summarised by highlighting the limitations of the existing security schemes.

The proposed methodology is outlined in this chapter. The research gaps in lightweight cryptographic algorithms identified in Chapter 3 are addressed to protect the MQTT protocol in Healthcare 4.0 scenarios based on IoT from attacks. Robust security solutions are proposed using lightweight cryptography algorithms suitable for resource-constrained IoT environments by offering potential contributions to address the identified shortcomings. The design of the research methodology focuses on lightweight security solutions to meet the security requirements of the MQTT protocol in Healthcare 4.0. The lightweight symmetric key algorithms are initially analysed and validated for robustness against attacks using the MQTT protocol for various performance metrics. After identifying the robust, lightweight algorithm, it proposes attribute-based encryption to preserve medical IoT users' privacy and ensure authentication with precise access control policies. Tool-based and simulation-based validation and performance evaluation strategies are analysed in detail with performance evaluation metrics and scenarios.

4.1 Need for Novel Lightweight Security Solutions for MQTT

Technological breakthroughs in IoT and smart mobile devices have outpaced adequate security solutions, which demand lightweight and robust solutions. Most current lightweight cryptographic solutions occasionally compromise the security level for better performance efficiency, struggling with integration into conventional networks [407]. Furthermore, there is a pressing need for lightweight, scalable, attack-resilient, and privacy-preserving cryptographic techniques to protect resource-constrained environments from widely emerging attacks.

4.1.1 Significance of Lightweight Cryptography Analysis

Real-time data collection and technological breakthroughs have increased the demand for IoT in Healthcare 4.0. MQTT is a simple, efficient, lightweight messaging protocol supporting many IoT applications. Notable advantages of MQTT include low power consumption, low bandwidth usage, and highly scalable and efficient communication among low-power IoT devices that need to conserve energy. However, they pose many security vulnerabilities, including confidentiality, authentication, and access control-related attacks.

Lightweight security algorithms are the most powerful methods for ensuring data confidentiality in IoT networks, as they have minimal costs, simple rounds, and robust security features [408]. Several lighter versions of cryptography methods are available, and it is crucial to analyse which lightweight security algorithm is suited to protect MQTT-enabled communication in Healthcare 4.0 [409]. Table 4.1 analyses the lightweight cryptography solutions widely used for IoT networks.

Table 4.1: Analysis of Different Lightweight Cryptography Solutions.

Lightweight Solutions	Algorithm type	Solution Type	Key Size (bits)	Block/key size (bits)	Rounds	Performance	Security Level	Healthcare Suitability	Limitations
AES-128			128	128	10	High	Strong	Low	High cost, integration issues
PRES-ENT			80/128	64	31	Very High	Moderate	High	The tradeoff between security and cost

Continued on next page

Table 4.1 – continued from previous page

Lightweight Solutions	Algorithm type	Solution Type	Key Size (bits)	Block/key size (bits)	Rounds	Performance	Security Level	Healthcare Suitability	Limitations
LED	Block Cipher	Symmetric	64-128	64	32-48	Moderate	Strong	Medium	Security trade-off issue
DES			56	64	16	Moderate	Weak	Low	Legacy issues
FBC			112/168	64	48	Moderate	Moderate	Medium	Tradeoff problems
ECC	Public-key	Asymmetric	160-256	160-256	-	High	Strong	Low	High cost and overhead
RSA			1024-2048	1024-2048	-	Moderate	Strong	Low	Very high cost and complexity
NTRU-Encrypt			128-256	128-256	-	Moderate	Strong	Medium	High overhead

4.1.2 Bridging Gaps with Novel Cryptography Solutions

Many lightweight solutions accomplish various performance levels with adequate security in MQTT-enabled healthcare IoT. Efficient analysis against different types of attacks is crucial to enhance the efficiency of specific environments. Conventional lightweight authentication exploits the self-key update and authentication scheme [410]. ABE [411] is commonly used in IoT solutions. However, traditional access control solutions require the encryption of IoT data using the CP-ABE scheme, and only users who comply with the access control policy can decrypt the encrypted information received [346]. Never-

theless, ABE schemes usually involve high computational complexity and are challenging to implement in resource-limited wireless sensors with limited power and computational capacity. Therefore, a lightweight MQTT authentication and privacy-preserving scheme for Pub/Sub communication models must be proposed.

Most secure MQTT applications specify symmetric/asymmetric encryption and a hashing algorithm to ensure data confidentiality and integrity. However, resource-constrained devices need more processing power to perform complex authentication tasks with asymmetric encryption algorithms. CP-ABE offers fine-grained topic-related tree creation and easy data access. It is one of the most commonly used security schemes in MQTT. CP-ABE encrypts the topic and its content based on its attributes. Decrypting the message is possible if the ciphertext contains the attributes. Most conventional CP-ABE systems suffer from two main issues: inefficiency with many attributes [412,413] and lack of attribute revocation mechanisms [414,415]. Additionally, while lightweight, the PRESENT algorithm [416] has limitations due to the high complexity of its S-box and key scheduling algorithms compared to other lightweight block cypher schemes. The MQTT broker generally serves as an intermediary between publishers and subscribers, facilitating communication between the two. Moreover, establishing secure and efficient lightweight security solutions is crucial to ensure safe communication with MQTT-enabled Healthcare 4.0.

4.1.3 Trade-off Between Security and Computation Cost

Computation cost represents the resources needed to perform cryptographic solution operations, including processing power, energy consumption, and memory usage. The primary intention of lightweight cryptography is to reduce computational cost and resource efficiency while ensuring an adequate level of security and higher performance [369]. The purpose is to ensure that the cryptographic strategy is strong enough to protect information and communications from unauthorised access and tampering, even though the network is subject to advanced attacks. Although the smaller key sizes of these lightweight solutions decrease computational complexity, they also minimise security. The nature of shorter keys makes them more accessible to brute-force attacks, which can seriously lower

the resistance level of lightweight algorithms to attacks. Notably, accomplishing security and computation cost trade-offs is difficult as the smaller key design saves device resources. They may not offer a significant level of security in high-risk environments [417]. This trade-off is crucial in lightweight cryptography but is a key challenge due to the careful consideration of requirements. Although the fundamental goal is to decrease the computational burden of resource-limited IoT devices, this should be balanced against the need to maintain a sufficient security level. In several cases, achieving this security and computation cost trade-off involves compromises, where future solutions scale back some security features to assure performance efficiency or where extensive resources are allocated to perform critical security functions.

4.2 Use Case for Healthcare and Medical IoT

In healthcare and medical IoT, lightweight cryptography solutions play a crucial role in providing security while minimising computation costs and maintaining the efficiency of communication performance between medical devices and systems [370]. In this context, the use case for lightweight cryptography secures communications, ensuring the integrity of information and preserving patient privacy while operating within the limited resources of medical devices.

4.2.1 Healthcare 4.0 Overview

A modern hospital uses an IoT network that allows intelligent medical devices, wearables, and implants to monitor patients in real-time. These devices range from wearable health monitors, smart insulin pumps, implants, connected heart rate monitors, smartwatches, and remote patient monitoring systems [418]. The information collected by these devices is essential for timely patient care. In addition, the data is securely transferred to the central health systems for analysis, disease diagnosis, and treatment decisions.

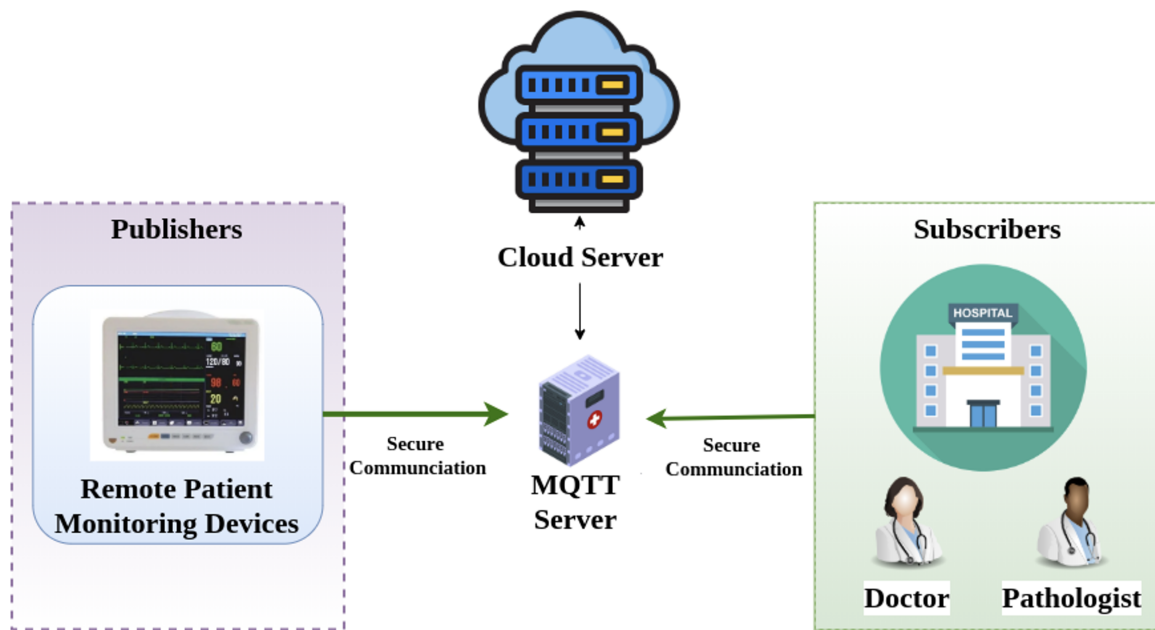


Figure 4.1: Lightweight Security and MQTT-Enabled Healthcare Monitoring System

4.2.2 Key Components of MQTT Enabled Healthcare 4.0

IoT Devices on Patient Side or MQTT Clients: Monitoring patients consists of the following smart healthcare devices: wearables, medical sensors and implants. The primary function of these healthcare devices is to continuously monitor critical signs such as heart rate, blood pressure, and physical activity of the patient and frequently send reports to healthcare providers who are physicians through the MQTT broker.

MQTT Broker: This is the core component of the MQTT communication setup that facilitates communication between healthcare devices and service providers. Data collected from MQTT clients is routed to appropriate subscribers, such as healthcare providers and cloud platforms.

Healthcare Providers or Subscribers: They use websites, mobile applications, and dashboards that assist healthcare professionals or doctors with a user-friendly interface to monitor patient data in real-time. Subscribers can receive the intended information through MQTT topics.

4.2.3 Communication Flow of Healthcare 4.0

The Healthcare 4.0 industry leverages the advantages of lightweight and efficient MQTT algorithms to ensure seamless real-time data exchange between different remote patient monitoring system components. The communication flow consists of the following steps.

- Step 1: Data Collection: Healthcare IoT devices frequently monitor patient conditions and collect information to transfer to the healthcare provider through a gateway.
- Step 2: Data Transmission: According to the MQTT steps, the publisher's healthcare devices or gateway transmit the data to the MQTT broker.
- Step 3: Data Processing: The MQTT broker forwards patient information to the remote monitoring server or cloud, where it is stored, analysed, and integrated into electronic healthcare record systems.
- Step 4: Real-Time Monitoring: Healthcare providers, such as doctors, can monitor the information through a dashboard or applications, receiving timing alerts when information crosses critical thresholds. They receive critical information based on topic subscriptions.
- Step 5: Patient Interaction: Patients receive feedback and alerts from healthcare providers through a mobile application that helps them manage their health proactively.

4.2.4 Security for Healthcare Data

The MQTT secures the information transferred across its clients, who are publishers and subscribers, by utilising lightweight cryptography strategies. However, an ingenious attacker can break the security of MQTT communication, as shown in Figure 4.2. The attacker intends to intercept and alter the healthcare information transmitted from the IoT devices attached to the patient's side to the healthcare provider, potentially leading to wrong diagnoses or treatment decisions. The health care system makes decisions based on the information transferred. Therefore, healthcare data security is paramount due to the sensitive nature of the data involved, including personal health data, treatment histories, and other confidential patient information. Ensuring security for healthcare data should address many key challenges, and it is crucial to implement lightweight, robust security

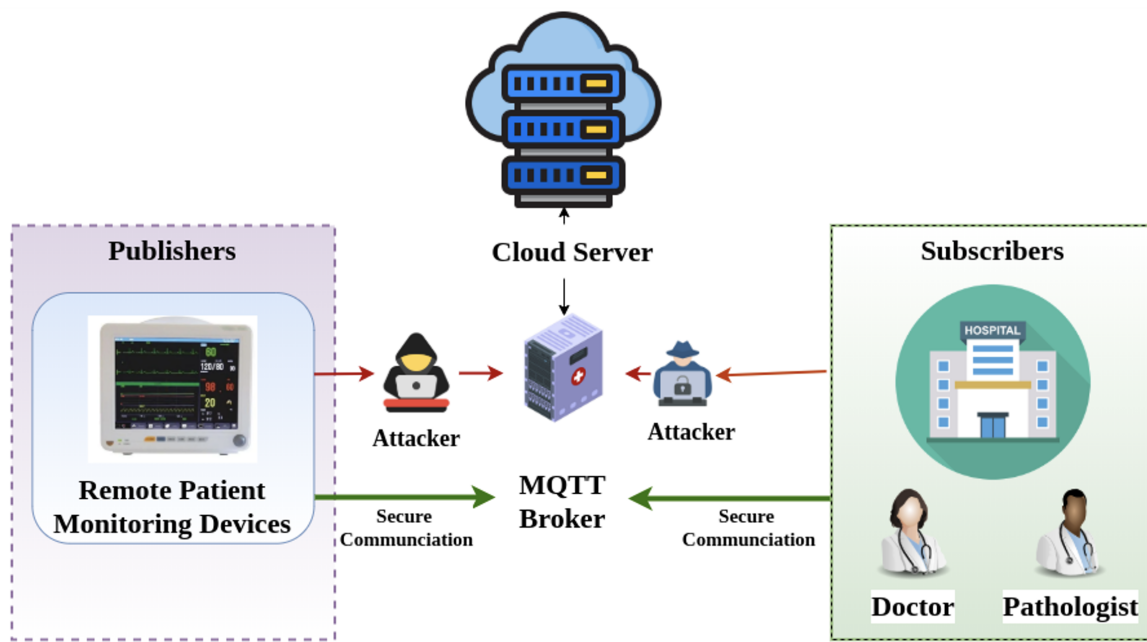


Figure 4.2: Healthcare IoT Security with Attack Scenario

countermeasures at different levels.

4.2.5 Security Challenges in Healthcare 4.0

Implementing lightweight cryptography in MQTT-enabled Healthcare 4.0 meets the following challenges.

Resource Constraints [370]: Healthcare IoT devices are small and limited in processing power, battery life, and storage. These limitations make current lightweight cryptographic strategies unsuitable in many situations.

Data Sensitivity [371]: Information transmitted by medical IoT devices is susceptible, as it incorporates personal health information and biometric information in real-time. It is crucial to ensure the confidentiality, integrity, and availability of this data to achieve seamless performance.

Real-Time Requirements [370]: Healthcare providers rely on real-time information from medical IoT devices to make decisions. Delays or incorrect readings caused by complex cryptographic operations should significantly impact timely patient care.

Capacity to Scale [419]: Smart medical IoT devices evolve daily due to technological

advances, which require secure communication. Therefore, the lightweight design must be scaled efficiently without overwhelming the security and performance of the healthcare environment.

4.3 Methodology of Lightweight Security Solutions for MQTT

This work aims to introduce three objectives to enhance security in terms of confidentiality, authentication, access control, and privacy preservation in a Healthcare 4.0-enabled environment. Proposed methodology 1 intends to protect message confidentiality against two significant attacks: DoS and MitM. Methodologies 2 and 3 propose attribute-based encryption and lightweight symmetric key cryptography-based solutions to preserve medical IoT users' privacy and ensure authentication with precise access control policies. Figure 4.3 illustrates an overview of the proposed methodology.

4.3.1 Analysis of Lightweight Symmetric Cryptographic Algorithms for MQTT against Confidentiality-related Attacks

The primary objective of methodology 1 is to analyse various symmetric key cryptography algorithms under different attack scenarios to ensure confidentiality in MQTT-enabled IoT systems. This work analyses the security performance of these algorithms using three different methods: formal security, tool-based security, and simulation-based security. Each evaluation model plays a crucial role in the comprehensive evaluation of various symmetric key cryptography algorithms. First, formal analysis verifies the encryption and decryption functions of the algorithm by formulating mathematical functions. Secondly, tool-based security analysis enables practical, real-world implementation testing, ensuring the security solution is robust against known vulnerabilities. Third, simulation-based analysis provides insight into how MQTT security measures perform in real-world scenarios by focusing on the impact and resilience of the performance. In addition, it allows one to evaluate how lightweight cryptographic algorithms affect the performance of

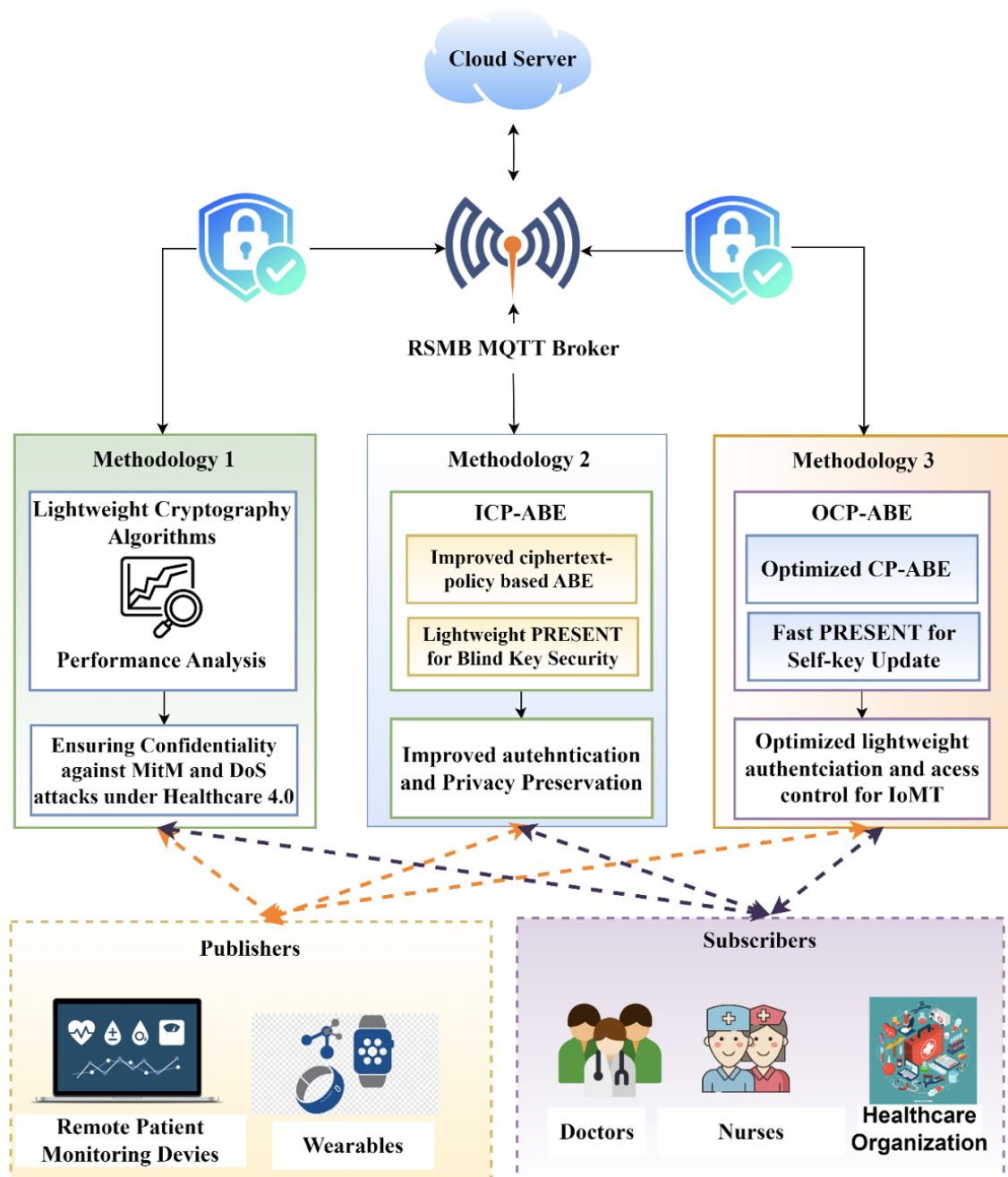


Figure 4.3: Proposed Methodology

MQTT, such as latency, throughput, and energy consumption, which is crucial in resource-constrained IoT devices. These evaluations are critical in healthcare IoT, where devices are resource-limited and more vulnerable to confidentiality-related attacks. Through this comprehensive performance evaluation, healthcare use cases can select highly adoptable security strategies to improve performance.

4.3.2 Lightweight Security Scheme for Topic Encryption and Attribute-based Authentication

Proposed methodology 2 proposes a novel lightweight security scheme to protect user privacy and ensure authentication over MQTT-enabled Healthcare 4.0. Although conventional ciphertext policy attribute-based encryption could be more effective for large-scale IoT applications, the number of attributes increases the key size and resource consumption. It makes it vulnerable to privacy attacks and is unsuitable for resource-constrained IoT devices. Therefore, methodology 2 presents a lightweight security scheme with topic encryption and attribute-based authentication that can significantly enhance the security and privacy of MQTT-based IoT environments. Topic-based encryption ensures that all messages published under a particular topic are encrypted, making the data inaccessible to unauthorised entities, including the MQTT broker. Attribute-based authentication ensures that only authorised users can access patient information by effectively defining the attributes for authentication.

4.3.3 Optimisation of Attribute-Based Lightweight Authentication Scheme

The primary objective of the proposed methodology 3 is to optimise the lightweight attribute-based authentication scheme to suit resource-constrained medical IoT environments. The MQTT broker generally serves as an intermediary between publishers and subscribers, facilitating communication between the two. Establishing a secure and efficient security method is crucial to ensure safe communication between all parties involved. To address these concerns, a proposed solution combines the optimised CP-ABE and Fast-

PRESENT schemes to improve security. Although existing methods often showcase their lightweight properties through simulations, focusing on execution time and energy consumption, they usually lack a thorough analysis of their security strength.

4.4 Validation and Evaluation Methods and Tools

To ensure lightweight security in MQTT-enabled healthcare IoT systems, various approaches are used to analyse the security strength and performance of the algorithms implemented.

4.4.1 Validation and Evaluation Methods for MQTT Security

MQTT security solutions use different methods to validate their security strength and impact on performance, and these approaches can be broadly classified into formal security analysis, tool-based security analysis, and simulation-based security analysis. Each approach has its benefits and limitations, and conventional works jointly consider these approaches for a comprehensive security assessment. Formal security analysis uses unique mathematical models and formal strategies to demonstrate the security properties of a lightweight security solution rigorously. This method is highly beneficial for proving the accuracy of the security strength of lightweight cryptographic protocols under defined assumptions. The tool-based security analysis approach uses automated tools to assess the security level of lightweight MQTT security solutions such as Scyther [420] and JcryptTool [421]. They can be used to determine general security problems such as misconfigurations, weak encryption, and vulnerable ability to known attacks. Simulation-based security analysis takes advantage of any network simulator, such as the Cooja simulator [422], to test and validate the performance of the lightweight security protocol in simulated healthcare IoT environments. Among the three methods, simulation-based analysis is instrumental as it produces results nearly equal to the real-time performance of MQTT-enabled IoT networks.

The computational complexity of cryptography algorithms depends on several factors: message length, key type, number of rounds, key management, and key length exploited

for data encryption and decryption. This work exploits the symbol O for the representation of computational complexity. The other notations used to estimate computational complexity in the proposed work are defined in Table 4.2.

Table 4.2: Notations Used for Computational Complexity Estimation.

Notations	Definition
$O(N)$	Key Generation or Sharing
N	Message Length
$O(N.t)$	Signing/Verification
$O(U.N)$	Key Revocation
$O(1)$	Encryption Cost or Decryption Cost or XOR Operation Cost or Key Scheduling Cost
H	Key Size
$1R$	One-bit Rotation Operation Cost
$1L$	S-box Table Lookup Cost in One Byte
$1M$	Two-byte Multiplication Cost
$O(31)$	Total Shifting Operation Cost of PRESENT with 31 Rounds
$O(6)$	Total Permutation Operation Cost
$O(12) \times 4$	Total Cost for 12 XOR Operations Performed on a 4×4 Matrix, Resulting in $48O$
16	Number of Transformations Executed during the Key Scheduling Process
$64M$	Extensive Cost Produced by Confusion owing to Two-Byte Multiplications of Key Schedule

4.4.2 Tools-based Validation and Evaluation

Lightweight MQTT security solutions widely use Scyther [420] and JcrypTool [421], which are more powerful in assessing security performance in resource-limited environments. Scyther provides formal verification steps to prove the MQTT security properties, while JcrypTool allows practical implementation by testing lightweight cryptographic functions. This combination ensures that the lightweight MQTT solution is analysed theoretically and practically, effectively meeting the needs of healthcare IoT, where resource limitations and security are critical considerations.

Scyther [423]: It exploits, proves, or disproves the security properties of lightweight MQTT through formal strategies. For this purpose, it precisely models the algorithm and

exhaustively verifies all possible execution paths to determine vulnerabilities, MitM, DoS, replay, and key compromise [424]. Instead of requiring actual implementation, it works with an abstract model of the lightweight algorithm. This approach allows rigorous proofs of security properties, ensuring the proposed MQTT protocol design is secure. Scyther is a powerful tool for checking properties such as secrecy, ensuring sensitive information remains confidential, and authentication verifies the identities of communicating entities. By exhaustively searching the scenario possibilities, it can determine different types of attacks, including MitM, replay, and impersonation.

JCrypTool [425]: It allows lightweight MQTT cryptographic protocols to be implemented and tested within actual software environments. This approach incorporates testing to verify the correctness of key generation, encryption, decryption, and signature verification of cryptographic operations [426]. Evaluates the security and performance of lightweight cryptographic operations in a practical context, which is essential for systems such as resource-constrained IoT in healthcare. JcrypTool ensures that lightweight MQTT cryptography implements its functions correctly, validating that encryption, decryption, and key exchange work properly in actual software. Moreover, JcrypTool allows the implementation of the lightweight MQTT protocol to be tested against practical attack vectors, such as MitM and cryptographic flaws in its code.

Finally, it is determined that the togetherness of Scyther and JCrypTool tools can provide a comprehensive model to develop and access secure, lightweight MQTT cryptographic protocols from both theoretical design and practical implementation.

4.4.3 Need for Simulation-based Validation and Evaluation for MQTT

The critical analysis and validation of the proposed lightweight MQTT security solution are paramount, and a systematic and valid approach is needed to assess its efficiency and effectiveness. The analysis mainly evaluates lightweight security solutions based on performance, such as throughput, latency, memory usage, and energy consumption. Another dimension of validation is related to security risk and threat analysis. This analysis tests

the ability to assess security solutions against possible MQTT security risks. It validates the robustness of the approach under known vulnerabilities and its resilience against these conditions during encrypted communication and message authentication. Cryptographic efficiency is a critical analysis that validates computational cost and effectiveness based on key strength and size, algorithm efficiency, and encryption operating speed. The potential approaches to validating the security solutions vary significantly depending on the influencing factors such as realism, maturity, and level of control. Based on these factors, validation and analysis can be performed using real-world and practical deployments, prototypes, and simulation environments.

Validating the practical environment in a fully operational scenario makes the validation process more reliable but highly complex. This validation offers testing against working systems in real-world settings, potentially capturing the underlying complexities of network conditions, fluctuating network parameters, unpredictable device failure and behaviour, and unexpected network dynamics. This approach leads to the risk of downtime in a live operating environment. Testing under real-world vulnerabilities with potential attackers offers valuable testing for security effectiveness. However, it is challenging due to ethical concerns and the potential risk of compromising operational integrity. Practically managing large-scale real-world deployments for testing is resource-intensive and operationally expensive. In particular, comprehensively validating security solutions requires a high cost, which is not feasible with limited budgets.

The prototype environments are small systems with limited features. It makes comprehensively validating security solutions more challenging because the simplified features do not satisfy the practical considerations and complexity. Attack modelling with the security features in a prototype environment, with possible attack vectors, limits the testing capability against various attacks. Prototypes are typically limited to specific network configurations that potentially restrict the diversity of devices, brokers, and their interoperability.

Validating lightweight security solutions in sensitive environments imposes privacy and confidentiality requirements. For example, sensitive healthcare records or industrial control systems are paramount, where strict regulatory compliance such as General Data

Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) ensures the non-disclosure of patient records. Practical validation methods that meet this legal regulation are challenging, especially validating cryptographic mechanisms against stringent legal regulations.

Simulation-based validation allows for early testing scenarios of security concepts without practical systems, significantly reducing testing and validation costs. Security validation can be performed safely using a simulator in various attack scenarios without affecting real systems or data. This approach is practical when strict regulatory compliance is mandatory, such as in healthcare. Another advantage is that extending the simulation model for large-scale environments is possible with limited cost and time. The simulator enables abstract devices and resources; this kind of modelling leads to easier validation. Despite limitations, simulation models are typically more affordable for study environments because of their accessibility than real-world deployments. It also allows for a rapid and iterative testing cycle without extensive hardware support. Considering these advantages, a simulation environment is preferred for this study.

Simulation-based security analysis can provide comprehensive scenario-based testing in simulated healthcare IoT environments, enabling critical insights into the behaviour of lightweight MQTT security solutions under various conditions, including different security attacks. Among the available simulators for the MQTT protocol, Contiki/Cooja [422] is the widely employed open-source network simulator, and the critical characteristics of the Cooja-based MQTT simulations are described below.

1. **Real-world Performance Mimic [427]:** Cooja allows the simulation of the healthcare IoT network considering all features, enabling MQTT performance testing in a controlled environment that closely mimics real-world network conditions and performance results.
2. **Controlled Environment [428]:** This property offers a controlled testing environment for various security solutions where simulations undergo repeated validations systematically under the same network conditions. The controlled environment can easily reproduce repeated results to identify the impact of MQTT security solutions,

such as authentication, message integrity checks, network overhead, and the energy consumption of lightweight algorithms. This environment potentially reduces the validation time of the simulation events.

3. **Extended Scalability [428]:** Cooja can simulate a large-scale healthcare IoT environment, making it possible to validate how MQTT security stands up to various attacks and network scenarios with some limitations.
4. **Customizability [429]:** Cooja allows a customised simulation environment in which it is possible to integrate particular network configurations, devices, and various security algorithms.
5. **Scenario-based Security Testing [430]:** Cooja can be exploited to simulate different attack scenarios, such as DoS or MitM, and to assess how the MQTT design handles these attacks in a simulated IoT environment. Permits extensive MQTT testing under diverse network scenarios, including network vulnerabilities. It is highly beneficial to analyse the performance in addition to security, assisting in assessing the trade-offs between security and network performance.
6. **Cost Effectiveness [429]:** Cooja offers a low-cost simulation environment, an attractive option for the early-stage development of IoT systems. This low cost makes the simulation model affordable for validating various security solutions before practical use. This approach allows the developer to iterate, validate, and assess the performance cost-effectively.
7. **Early Identification of Issues [431, 432]:** This approach helps identify design issues in the early stages by quickly testing security properties before entering practical environments. Early stage validation significantly reduces the potential risk of larger changes during practical deployment.

Network Settings and Configurations

The latest version of the MQTT protocol is MQTT Version 5.0 (MQTT v5) [93], designed to improve the previous versions of MQTT Version 3.1 (MQTT v3.1) and 3.1.1 (MQTT v3.1.1) [95] by adding new features and enhancements. Thus, MQTT v5 is more flexible, scalable, and efficient for innovative Healthcare 4.0 applications and their use cases. The

Eclipse Paho project provides the characteristics of MQTT clients and libraries of different programming languages, and the Paho C Client is mainly designed for IoT applications written in C. Integrating Paho C libraries with Contiki is vital to simulate the lightweight MQTT v5 security solution in Cooja-based simulations.

Mosquitto Really Small Message Broker (RSMB) [433] is extremely useful for quick proof-of-concept prototypes with simple IoT scenarios for evaluation and experimental testing. Simplified and fast setup, resource efficiency, low-cost deployment, low-cost experimentation, and less time consumption are the advantages of using the RSMB broker [434]. RSMB is a lightweight, minimalistic broker that is tightly integrated with Cooja. This broker is primarily used when simulating MQTT-SN communication in constrained IoT environments [433]. Its simplicity and low resource requirements make it suitable for small-scale simulations in Cooja. Mosquitto is capable of handling a large number of clients and managing message loads larger than RSMB. Using Mosquitto is mandatory only if a full-featured broker with advanced MQTT features is required. This is ideal for production-grade deployments. However, it is unsuitable for the Cooja simulator due to external integration outside the Cooja environment, whereas RSMB can be tightly integrated with Cooja.

When used with the MQTT v5 algorithm and a Mosquitto broker, Cooja plays a crucial role in simulating and assessing performance in resource-limited healthcare environments. The main reason is that Cooja is consolidated with Contiki OS, which supports MQTT v5 data transmission, making it highly suitable to simulate MQTT v5-enabled applications.

Realistically simulates MQTT v5 by evaluating it with various features that improve the properties of the protocol, improve error handling, and make it highly adaptable to MQTT application-specific environments. Furthermore, the Contiki OS offers precise MQTT v5 implementations by creating a controlled communication environment through advanced features. In addition, Cooja provides a platform for tuning different protocol parameters for MQTT v5 and allows one to evaluate their impact on the performance of the entire system. Moreover, Cooja allows various simulations from the data to the application layer. It is highly flexible in implementing the MQTT v5-based IoT-assisted healthcare scenario with greater effectiveness and less memory usage.

The simulation on IoT nodes is conducted using the Paho C Client Library Module, Ubuntu 18.04 LTS 64-bit, Instant Contiki-3.0, and VMware Player 17. The Paho C Client is a C language implementation of MQTT that supports MQTT v5 and its preceding versions. It can support different QoS levels like 0, 1, and 2 in message delivery. Moreover, Cooja-based simulations show the performance of the protocol in terms of various performance metrics.

Performance Metrics

The Cooja simulation-based analysis employs the following metrics to evaluate the lightweight MQTT security algorithm.

Throughput: It is the rate of data delivery measured in terms of bits/second.

Packet Delivery Ratio (PDR): The percentage of packets successfully delivered to the total number of packets generated.

Delay: The extra time is taken to deliver a packet from a source to a destination.

Execution Time: The amount of time taken to execute the operations of the lightweight MQTT algorithms.

Energy Consumption: The average energy consumed for data transmission and security operations.

CPU Energy Consumption: It is the energy consumed by the CPU. CPU Energy Consumption: This measures the average amount of energy consumed by the CPU in the MQTT client during the execution of authentication and communication processes.

Computation Overhead: It is the time taken to perform the algorithm computations.

Strength Evaluation Criteria: This is defined as the relationship between the length of the ciphertext and the plaintext, which is used to evaluate the strength of the security scheme.

Communication Overhead: This is defined as the length of the ciphertext, which includes all additional information added to the original plaintext to secure it.

Packet Loss: Packet loss is measured as the total number of lost packets over the total number of sent packets.

4.5 Chapter Summary

This chapter proposed a novel research methodology to secure the MQTT protocol using lightweight cryptography against attacks in intelligent Healthcare 4.0 to solve the state-of-the-art research gaps. The proposed method overcomes the shortcomings by analysing the challenges in the IoT-based MQTT protocol environment to accomplish the security requirements. To this end, the research presents significant contributions and proposes a methodology to overcome the challenges. Robust and lightweight algorithms are identified to develop attribute-based authentication and access control policies to protect Healthcare 4.0 against security vulnerabilities. In addition, this chapter outlines the validation methods and software requirements for implementing the proposed model and specifies performance metrics to evaluate its effectiveness.

Performance Validation for MQTT in Healthcare 4.0

Lightweight security algorithms are the most powerful methods for ensuring data confidentiality in IoT networks. This chapter validates the security of five lightweight symmetric key algorithms for the MQTT protocol. In this context, analysing the security features and robustness of lightweight security algorithms appropriate for the data confidentiality of MQTT-enabled communication in Healthcare 4.0 under MitM and DoS attack scenarios is crucial. Therefore, this chapter exploits two ways to validate and analyse MQTT security issues: formal security analysis and tool-based security analysis, that is, Scyther [420] and JcryptTool [421]. The initial method exploits mathematical strategies to prove the security issues of MQTT under MitM and DoS attacks. The remaining techniques use the tool to show the performance of five lightweight symmetric cryptography algorithms in the MQTT-enabled IoT environment for Healthcare 4.0.

5.1 Introduction for MQTT Lightweight Symmetric Key Cryptography

Integrating advanced technologies creates notable breakthroughs in IoT, enabling intelligent healthcare systems to improve hospitalisation quality. In this context, Healthcare 4.0 has contributed significantly to the advancement of next-generation healthcare services using emerging technologies that can create superior healthcare solutions. It enables patient monitoring using advanced wearable healthcare devices that collect data to improve quality exponentially. As an extension of IoT, IoMT simplifies healthcare applications and improves the overall system [435]. Wireless Body Area Network (WBAN) comprises intelligent sensor technology capable of capturing the information detected from medical

sensor devices embedded in the patient's body to form a network [436]. This WBAN network exchanges information on the wireless communication system, collects healthcare data on the network edge, and stores it on the server side for further analysis. WBAN enables the continuous monitoring and diagnosis of patients affected by chronic diseases in real time. It allows healthcare professionals to monitor better, diagnose patients, generate reports, and improve the quality of healthcare. Despite advances, security and privacy are the main aspects, while the focus is on efficient, cost-effective, and reliable real-time healthcare systems [437].

IoT technology allows seamless internet connectivity among patients and physicians using different communication technologies, such as RFID, Zigbee, Bluetooth, 6LoWPAN, and Wi-Fi. Pushing and polling protocols are employed by the IoT to enable such communications. Due to high productivity and lightweight sensors, push protocols like Extensible Messaging and Presence Protocol (XMPP) [438], Advanced Message Queuing Protocol (AMQP) [439], and Message Queuing Telemetry Transport (MQTT) [440] are highly adaptable for IoT environments [441]. The MQTT-assisted Healthcare 4.0 opens up new ideas due to the accurate and frequent updating of monitored healthcare information to the central server. Despite the advantages, security is a critical constraint in the MQTT-enabled Healthcare 4.0 scenario, leading to security breaches, inaccurate diagnosis, and inopportune hospitalisation. Lightweight cryptography ensures the confidentiality of the data on the MQTT protocol in IoT applications.

Confidentiality protects sensitive data from unauthorised access, tampering, and disclosure. In MQTT-based data transmission, confidentiality ensures that only authorised individuals or systems can access the particular data. Thus, it prevents unauthorised parties from gaining knowledge of sensitive information. Lightweight cryptography helps mitigate security risks and ensure low-power IoT network settings by providing a lightweight and efficient way to encrypt and authenticate data. However, MQTT performance is limited in the presence of Man-in-the-Middle (MitM) [442] and Denial-of-Service (DoS) [443] attacks. The DoS attacker intends to deny MQTT services to genuine users by flooding unwanted control traffic into the network. Thus, it temporarily disturbs the MQTT server's service.

Consequently, the MitM attacker secretly modifies the MQTT packets transmitted between clients, while the clients believe that communicating through the MQTT server is secure. Hence, accurate analysis is essential to select a suitable security algorithm that can provide strong confidentiality and improve communication efficiency, resulting in precise and timely hospitalisation services. Therefore, an efficient security algorithm is needed to provide strong security against such attacks. Many lightweight cryptography algorithms are available, and it is essential to carefully evaluate available options to decide on an algorithm that provides the necessary level of security within the resource requirements.

Several familiar symmetric key cryptography algorithms can be used to secure MQTT, including Advanced Encryption Standard (AES) [444], Data Encryption Standard (DES) [445], Triple DES [445], PRESENT [267], Twofish [446], Blowfish [447], Rectangle [448], SIMON [263], Light Encryption Device (LED) [449], and Fast Bitslice Cipher (FBC) [450]. Therefore, this chapter performs an adequate analysis of security and performance, which is crucial among algorithms to determine which type of lightweight algorithm is more efficient and secure for resource-limited, MQTT-enabled Healthcare 4.0. Thus, it motivates an in-depth analysis of the performance of various lightweight cryptography models in MQTT-enabled Healthcare 4.0, specifically against MitM and DoS attacks. Symmetric key algorithms can address the constraints in resource-limited MQTT environments by enabling suitable encryption and decryption strategies with appropriate keys. The intrinsic motivation of the proposed model is to analyse and validate the security of five lightweight cryptography algorithms, such as LED, PRESENT, AES, DES, and FBC, in both theoretical and tools simulation, such as Scyther [420], and JCrypTool [421] aspects under the presence of data confidentiality-related DoS and MitM attack scenarios in MQTT-IoT enabled innovative Healthcare 4.0.

This chapter investigates the performance validation of lightweight symmetric encryption algorithms within MQTT-based Healthcare 4.0 environments. It incorporates domain-specific requirements of Healthcare 4.0 to ensure that selected algorithms provide robust security and are viable under the stringent constraints of medical IoT systems.

5.2 Research Gaps and Motivation

Healthcare 4.0 demands security schemes that are optimised for real-time responsiveness and low overhead. Traditional cryptographic techniques often fail in constrained environments. This motivates the exploration of lightweight encryption algorithms tailored for medical IoT devices.

The performance evaluation in this chapter builds upon the literature review of symmetric cryptographic algorithms provided earlier in Chapter 3. These lightweight algorithms—AES, DES, PRESENT, LED, and FBC—are specifically revisited in secure communication for MQTT-enabled Healthcare 4.0 environments. This section highlights the limitations of these schemes when applied under real-world healthcare constraints and motivates the validation approaches used.

Lightweight cryptography for MQTT significantly protects devices and Internet-enabled communication protocols from various security attacks. Several MQTT works exploit the advantage of lightweight cryptography in MQTT [250, 285]. However, they face several limitations, including high overhead, security trade-offs, and inefficient key management in various IoT scenarios and applications [285, 308, 405]. Firstly, existing work adds more overhead due to complex encryption and decryption strategies [308]. Secondly, conventional lightweight cryptography does not provide adequate security against evolving attacks effectively [366, 405, 406]. Finally, existing works lack the ability to manage cryptographic keys in resource-limited environments, which can be very challenging [308, 405].

Research gaps in Chapters 3 and 4 identified broad deficiencies in MQTT security frameworks which highlighting gaps in access control, key management, and end-to-end security and the lack of methodological rigour in validating lightweight algorithms, respectively. The research gaps in this chapter are uniquely domain-specific, which is the evaluation of symmetric key cryptographic protocols in Healthcare 4.0 scenarios. This chapter focuses on the performance shortcomings of lightweight symmetric key algorithms, specifically under Healthcare 4.0 constraints, such as real-time data processing, low latency requirements, and resilience against DoS and MitM attacks. Unlike the previous chapters, it emphasises the need for integrated formal, tool-based, and simulation-based

validation to assess the feasibility of these algorithms in IoMT environments.

Although different work evaluates the performance of lightweight symmetric encryption algorithms in MQTT-enabled healthcare IoT, there is still a lack of adaptive validation that can autonomously adjust encryption strength according to varying medical data rates, patient-criticality levels, and resource profiles of healthcare devices. Most existing approaches exploit static configurations for validation that overwhelm resource-limited devices or fail to protect sensitive data sufficiently. In addition, these solutions lack comprehensive validations against evolving dynamic security threat scenarios, particularly MitM and DoS, which are highly relevant in real-time healthcare IoT applications. Hence, evaluating the performance efficiency and security resilience of different symmetric key cryptography algorithms in dynamic healthcare IoT scenarios is crucial.

Therefore, the proposed work aims to comprehensively analyse the performance of different symmetric key cryptography algorithms for MQTT through Cooja-based simulation with different networks and attack scenario creations. This comprehensive performance analysis of five algorithms using Contiki/Cooja simulation helps inform informed decision making regarding security, performance, and resource constraints across IoT applications, ultimately leading to more secure, robust, efficient and scalable MQTT-enabled IoT systems.

5.3 Preliminaries of MQTT and Security Vulnerabilities

MQTT brokers and MQTT clients are two fundamental components of MQTT communication. The MQTT broker plays a crucial role in the MQTT architecture, acting as a mediator between MQTT clients. It receives all the messages published by the clients and distributes them to the relevant subscribers. In addition, the broker manages the clients' connections, ensuring that they remain active and messages are delivered correctly and on time. The broker may perform tasks such as message filtering, access control, and Quality of Service (QoS) management. MQTT clients can send messages and receive messages from the broker. It refers to sending a message to a client as “publishing” a message.

This work incorporates MQTT v5 to introduce numerous enhancements and additional features compared to the predecessor version 3.1.1. The importance of the topic is a fundamental concept in MQTT v5. Introduces the concept of shared subscriptions, in which multiple MQTT subscribers can share a similar subscription by employing a subscription identifier. Thus, it is very beneficial for scenarios that comprise multiple subscription instances. The topic's importance is explained using various added features like hierarchical structure and topic filter.

Hierarchical Structure: The topics in MQTT v5 enable the users to maintain a hierarchical structure which allows logical organisation and message categorisation. Therefore, the hierarchical structure of MQTT v5 is highly valuable for effectively organising and managing data transmission between devices or systems.

Topic Filter: It plays a significant role in MQTT v5, enabling subscribers to integrate their particular interests in receiving messages according to the topic patterns. This filter allows subscribers to select message subscriptions, reducing the amount of irrelevant information obtained by an MQTT subscriber. Therefore, the topic filter is significant for appending MQTT v5 in the specific application environment.

Moreover, the topic offers a structured and organised way for managing communication between resource-limited devices. The additional features introduced in MQTT v5, such as shared subscriptions, hierarchical structure, and topic filter, contribute significantly to the flexibility, effectiveness, and reliability of data transmission in MQTT-enabled applications.

5.3.1 System Architecture of MQTT Enabled Smart Healthcare 4.0

The system architecture reflects a smart healthcare environment, incorporating wearable medical sensors, body area networks, and real-time patient monitoring systems. MQTT acts as the lightweight communication backbone for this architecture.

The innovative healthcare environment consists of many intelligent wearable and non-wearable sensing devices to enable timely patient hospitalisation services. The smart

healthcare scenario exploits intelligent objects and devices to continuously collect data such as temperature level, sugar level, blood pressure, pulse rate, and movements in the body to know about patients' health conditions. In addition, the collected information will be uploaded to the remote server so that healthcare professionals can assist in timely hospitalisation services to patients through efficient monitoring and communication strategies. In a smart hospital scenario, it is crucial to reduce the negative impact of hospitalisation to improve patient life quality and allow patients to walk around hospital rooms or environments with effective sensor-based monitoring strategies. Thus, it provides high-quality and stress-free medical services to patients.

Figure 5.1 demonstrates an MQTT communication-based smart healthcare scenario constructed using IoT with sensors and the MQTT protocol. The architecture comprises three layers: hospitalisation, MQTT communication, and sensing. The doctors or nurses in hospitalisation are called caretakers, and the patients in the sensing layer are called subscribers. Initially, it divides the hospital area into various monitoring areas or rooms and allows for a separate MQTT broker for each area or room, resulting in continuous monitoring of patients. For example, the temperature sensor is used to measure the temperature level, and the accelerometer sensor is used to measure the movement of a patient's body. These sensors monitor patient-related information and use the MQTT protocol to communicate with the server.

Remote Patient Monitoring System: In this work, secure MQTT-based communication is used to monitor patient health data remotely. This system comprises different wearable sensor devices to construct the IoT data layer. It transmits the monitored information to the remote or server location using a secure MQTT protocol. Therefore, MQTT uses a lightweight cryptography algorithm for security. Thus, medical sensors attached to the bodies or rooms of patients act as publishers and are limited in resources such as memory, bandwidth, and energy. The MQTT broker is stationary and equipped with adequate memory, communication bandwidth, and energy resources. Then, the MQTT server connects to the MQTT broker to transfer the information to the corresponding subscribers through the Internet.

Generally, the MQTT broker exploits lightweight cryptographic protocols to enhance

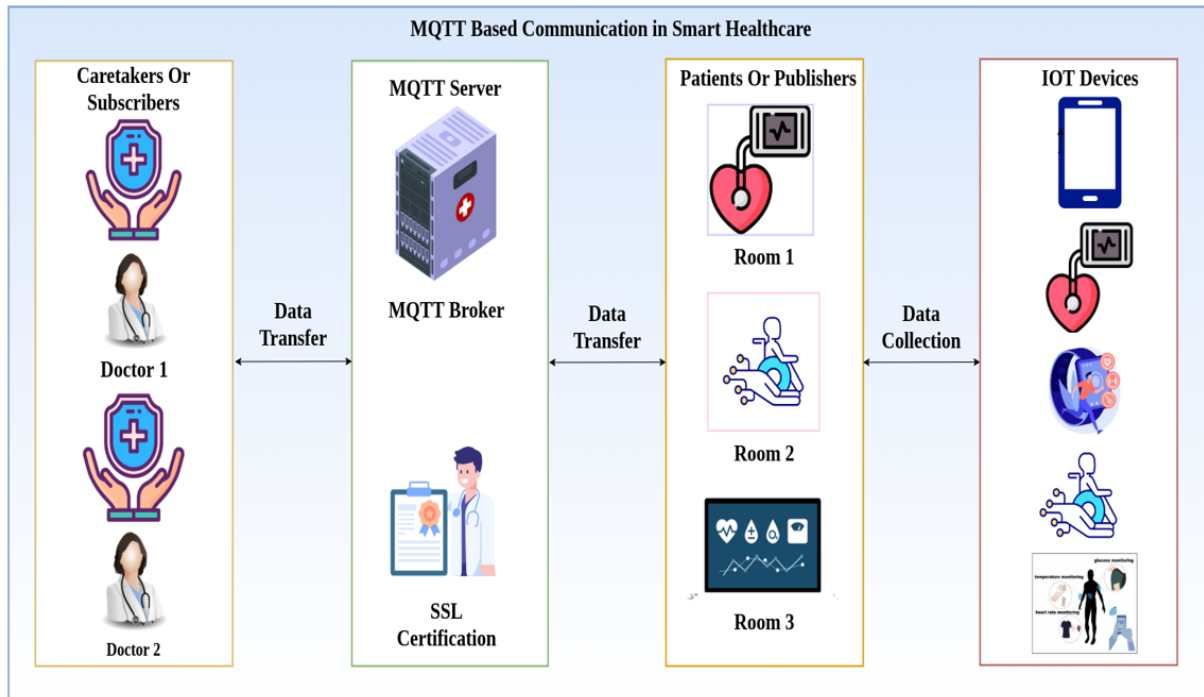


Figure 5.1: MQTT Communication for Smart Healthcare Scenario

the security level of patient data transmission and improve hospitalisation quality. Therefore, MQTT security involves managing the patient database with secured log-in methods by creating a username and password to access the necessary patient information. Hence, each patient record has a unique user ID per user. In short, using cryptography-based security methods can strengthen MQTT, making it desirable for timely hospitalisation. Finally, patient-related information is displayed using any mobile app or web page. In such a scenario, subscribers are resource-limited devices, and efficient, lightweight security schemes are essential for MQTT communication.

5.3.2 DoS Attack and Impact on MQTT

Validation includes security testing against DoS and MitM attacks, as these pose significant threats in healthcare scenarios, potentially interrupting life-critical data transmissions.

The main objective of a DoS attacker is to disrupt MQTT broker services, interrupting the coordination and dissemination of messages among MQTT clients. There are three forms of DoS attacks in MQTT: TCP-based, payload-based, and QoS-based [451]. In the

first type, the attacker can exhaust the MQTT broker's bandwidth by exploiting TCP security breaches. In the second type, the attacker creates messages with a payload size of more than 256 MB, which can drain the energy level of the MQTT nodes, leading to payload-based DoS attacks. In the third type, the attacker sends many QoS level 2 messages that require more resources than QoS levels 0 and 1, which can exhaust the MQTT broker's resources, resulting in QoS-based DoS attacks. Providing security against DoS attackers is crucial to improving MQTT performance in resource-limited IoT environments.

5.3.3 MitM Attack and Impact on MQTT

In a MitM attack, an attacker intercepts the MQTT connection by altering the messages between two entities. This attack is also known as a hijack attack or a cyber attack. It is a severe threat to MQTT, as MitM attackers can read and write messages and have a high degree of freedom to manipulate MQTT messages within an IoT network [27]. Detecting MitM attacks can be challenging compared to other MQTT attacks, as attackers have in-depth knowledge about network functionalities. Detecting and preventing MitM attacks is crucial to improving the performance of MQTT in IoT-based networks.

5.3.4 Lightweight Cryptography based Security for MQTT

Security algorithms are evaluated regarding their applicability to healthcare constraints—real-time communication, low computational overhead, and energy efficiency.

In general, lightweight cryptographic algorithms are characterised by short key lengths, simple rounds, and low energy and memory costs. These properties make lightweight algorithms highly adaptable for MQTT-enabled IoTs. Existing TLS protocols require significant memory [452] and energy overhead [30] to ensure security between MQTT servers and clients. To address these issues, lightweight cryptographic algorithms use improved methods to secure data without increasing asset rate and power costs, thus maximising the throughput and packet delivery ratio of MQTT with minimum cost and overhead. Lightweight cryptographic algorithms are particularly suitable for resource-

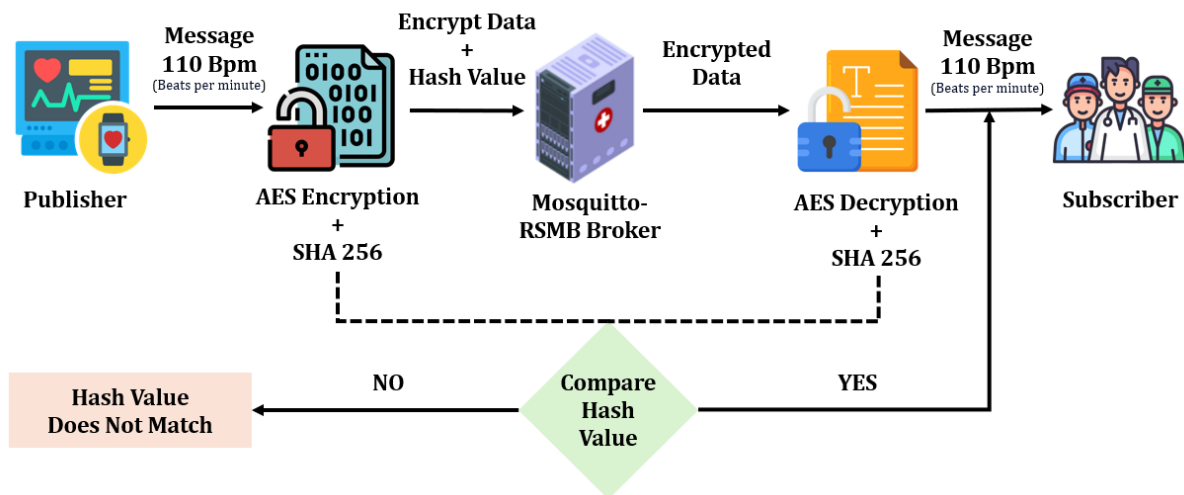


Figure 5.2: MQTT-based AES Algorithm for Smart Healthcare Scenario

limited IoT environments with small devices. However, given the numerous lightweight cryptographic algorithms available, it is crucial to select the most appropriate algorithm to secure MQTT-enabled IoT environments.

5.3.5 Symmetric Cryptographic Algorithms

Lightweight cryptography algorithms can be broadly classified as asymmetric and symmetric. Asymmetric cryptography algorithms use different keys for encryption and decryption, while symmetric cryptography algorithms use the same key. Figure 5.2 shows the operation case of this approach using the AES algorithm as an example in the IoMT.

Taking the AES algorithm as an example, MQTT communication is divided into the following six main parts:

1. Connection Establishment

- A client (either a Publisher or Subscriber) establishes a connection to an MQTT RSMB broker.
- The connection can be secured using SSL/TLS before any data transfer occurs.

2. Topic Subscription

- Subscribers express interest in specific topics by subscribing to them.
- Topics act as channels for messages.

- Subscribers indicate which topics they want to receive messages from.

3. Publishing Messages

- Publishers create messages and send them to the broker.
- Each message includes a topic and a payload.
- The broker, upon receiving a message from a Publisher, routes it to the appropriate Subscribers based on their subscriptions.

4. Encryption

- The publisher publishes a hello message, encrypts it using AES along with the SHA-256 hashing algorithm for integrity verification, and then sends the AES-encrypted message along with the SHA-256 hash value to the broker.
- The broker receives the AES-encrypted message along with the hash value.

5. Decryption

- When the Subscriber intends to decrypt the message, it initiates an AES decryption process along with SHA-256 verification.
- The broker then publishes the decrypted message along with its SHA-256 hash, reencrypts it using AES, and sends it to the subscriber.
- The Subscriber receives the AES-encrypted message and processes it.

6. SHA-256 Hashing

- The Subscriber will only decrypt the message if the received SHA-256 hash value matches the calculated hash value.
- In other words, successful authentication of the hash value is a prerequisite for decryption.
- If the hash value does not match, the subscriber cannot proceed with message decryption.
- This ensures that the message's integrity is preserved throughout the communication process and that only correctly authenticated messages are decrypted and processed.

Concerning speed and energy consumption, symmetric key algorithms typically per-

form better than asymmetric ones. Therefore, the performance of MQTT is evaluated using symmetric key algorithms such as AES, DES, PRESENT, LED and FBC. Compared to symmetric key algorithms, AES, DES, PRESENT, LED, and FBC provide excellent security features with minimal cost and overhead. Table 5.1 compares these five symmetric key cryptography algorithms based on various factors.

Table 5.1: Comparison of AES, DES, PRESENT, LED, and FBC Algorithms.

Factors	Algorithms				
	AES	DES	PRESENT	LED	FBC
Algorithm Structure	Substitution and Permutation Network	Fiestel Network	Substitution and Permutation Network	Substitution and Permutation Network	Bit Slicing
Key Used	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric
Key Length (bits)	128/192/256	56	80/128	64/128	128
Block size (bits)	128	64	64	64	128
Rounds	10,12, and 14	16	31	32 and 48	64
Cycles Per Block	1032	144	547	1872	144
Speed	Fast	Slow	Slow	Slow	Fast
Tunability	No	No	No	No	No
Vulnerabilities	Nil	Brute Force and MitM	Nil	Key Related Attacks	Nil
Security	Excellent	Moderate	Good	Good	Excellent
Energy Consumption	Low	Low	Moderate	Moderate	Low
Cost	Minimum	High	High	High	Minimum
Memory	Moderate	Less	Moderate	Less	Less
Efficiency	Efficient in both software and hardware	Slow	Good in software	Slow	Efficient in both software and hardware
Implementation	Simple	Complex	Simple	Complex	Simple
Flexibility	Yes	No	No	No	Yes

5.4 Analysis

This section provides an attack and computation cost analysis of five lightweight symmetric key cryptography algorithms to show their efficiency. Different algorithms are vulnerable to various attack activities and require distinct computations to ensure security, which are discussed in this section in detail.

5.4.1 Attack Analysis

Attack resilience is assessed in healthcare-relevant use cases. For example, a MitM attack simulates intercepted ECG data, and DoS scenarios mimic the resource exhaustion of wearable devices.

DoS and MitM attacks are common in MQTT communication, where DoS floods control messages and wastes network resources. MitM attackers secretly modify MQTT packets sent between clients who believe they communicate securely over the MQTT server. MitM attacks can be performed by brute-forcing any encryption mechanism. Strong encryption-based implementations with large keys ensure safe operation, but the high computation costs associated with large key sizes are not feasible in an IoT environment. Attackers can steal secret information, such as login credentials or personal information, without the need for key-breaking algorithms. However, generating false packets for network damage, credential tracing, and data integrity violations in communication requires tracking the secret key associated with the selected encryption scheme [453]. Distributed and Internet-connected devices perform botnet attacks, but symmetric encryption schemes employ lightweight cryptography to prevent malicious activities with minimal cost and overhead. Table 5.2 compares the attack prevention capacity of selected symmetric encryption algorithms.

Table 5.2: Attack Prevention Capability of Symmetric Encryption Algorithms.

Encryption Scheme	Attack	Attack Prevention Possibility	Reason
AES	DoS	High	MQTT server identifies the same message using the timestamp.
	MitM	Less	Generated words using the original key are related and can be traced easily.
	Spam	High	Only pub/sub-messages can be transmitted.
	Poisoning	High	High-complexity processes improve key strength.
PRESENT	DoS	High	MQTT server identifies the same message using the timestamp.
	MitM	Less	Biased inputs in the key space of PRESENT tend to attack the possibility.
	Spam	High	Only pub/sub-messages can be transmitted.
	Poisoning	High	High-complexity processes improve key strength.
LED	DoS	High	MQTT server identifies the same message using the timestamp.
	MitM	Less	The brute forcing approach can trace secret keys due to the usage of fewer rounds.
	Spam	High	Only pub/sub-messages can be transmitted.
	Poisoning	Less	The one-to-one mapping property of the XOR function creates open spaces for an adversary.
FBC	DoS	High	MQTT server identifies the same message using the timestamp.
	MitM	Less	Poor key strength.

Continued on next page

Table 5.2 – continued from previous page

Encryption Scheme	Attack	Attack Prevention Possibility	Reason
DES	Spam	High	Only pub/sub-messages can be transmitted.
	Poisoning	Less	Using most of the traced encrypted plaintext, the key can be identified.
	DoS	Less	2^{47} known plaintexts are enough to break DES.
	MitM	Less	Small key size.
DES	Spam	High	Only pub/sub-messages can be transmitted.
	Poisoning	Less	2^{47} known plaintexts are enough to break DES.
	DoS	Less	2^{47} known plaintexts are enough to break DES.
	MitM	Less	Small key size.

5.4.2 Computational Cost Estimation

Metrics such as encryption time, CPU cycles, and energy consumption are selected due to their direct impact on the feasibility of deploying encryption in energy-sensitive medical devices.

The computational complexity of lightweight algorithms depends on factors such as key type, number of rounds, and key length used to encrypt and decrypt data. Among the five protocols considered, AES has the highest computation cost in MQTT because it uses different key lengths and rounds to ensure security against DoS and MitM attacks. We use O to represent computational complexity. Table 5.3 summarises the computational costs of symmetric key cryptography algorithms.

In general, block cypher algorithms have fixed block size values, such as the AES 128 algorithm, which works based on a block size of 128 bits, resulting in a computational cost of $O(1)$ for message encryption and decryption. However, for longer messages, the cost value increases and is defined as $O(N)$, where the term N depends on the length of the message. The computational complexity values of DES and FBC are decided based on the confusion metric. The AES, PRESENT, and LED follow the substitution permutation

network model, while the DES and FBC follow the Feistel network model. The AES, PRESENT, and LED offer flexibility to IoT devices to select suitable encryption and decryption methods based on the length of the message. At the same time, FBC and AES provide robust security features with minimum cost values for different IoT applications.

Table 5.3: Computational Cost Analysis of Cryptography Algorithms.

Light-weight Algorithms	Key Length (Bits)	Encryption and Decryption Cost	Sub-bytes	Shift rows	Mix-columns	Confusion Matrix	Computational Complexity
AES	128	$O(1)$	$8L$	$O(2)$	$8M \times 4$ & $O(4) \times 4$	-	$O(AES(128)) = 8L + O(13) + 16M$
	192	$O(2)$	$12L$	$O(4)$	$12M \times 4$ & $O(8) \times 4$	-	$O(AES(192)) = 16L + O(27) + 32M$
	256	$O(3)$	$16L$	$O(6)$	$16M \times 4$ & $O(12) \times 4$	-	$O(AES(256)) = 32L + O(54) + 64M$
DES	56	$O(1)$	-	-	-	$O(32)$	$O(DES(56)) = O(55)$
PRESENT	80	$O(1)$	$4L$	$O(1)$	$4M \times 4$ & $O(1) \times 4$	-	$O(PRESENT(80)) = O(42) + 32M$
	128	$O(2)$	$8L$	$O(2)$	$8M \times 4$ & $O(2) \times 4$	-	$O(PRESENT(128)) = O(85) + 64M$
LED	64	$O(1)$	$2L$	$O(1)$	$4M \times 4$ & $O(2) \times 4$	-	$O(LED(64)) = 8L + O(27) + 32M$
	128	$O(2)$	$8L$	$O(2)$	$8M \times 4$ & $O(4) \times 4$	-	$O(LED(128)) = 16L + O(54) + 64M$
FBC	128	$O(1)$	-	-	-	$O(12)$	$O(FBC(128)) = 32L + O(44)$

5.5 MQTT Protocol Verification

The methodology includes two methods to analyse the efficiency of MQTT performance in smart healthcare enabled by IoT in the DoS and MitM environments. The initial method is a formal security analysis using mathematical strategies to validate the performance of MQTT with five algorithms. The second method is tool-based security analysis, in

which Scyther and JCrypTool are used to evaluate the performance of algorithms for MQTT-enabled IoT.

5.5.1 Formal Security Analysis

Mathematical models confirm that the selected algorithms meet confidentiality and integrity requirements crucial for sensitive healthcare data.

The formal security method evaluates the performance of MQTT under three cryptography algorithms using mathematical theorems. Table 5.4 lists the symbols used in security analysis with descriptions.

Table 5.4: Lists the Symbols Used in Security Analysis.

Symbol	Description
h	Number of healthcare devices
$Dev = \{d1, d2, \dots, d_h\}$	Healthcare device set
Pub	Publisher
Sub	Subscriber
X	Plaintext
Y	Ciphertext
E	Encryption function
E_{key}	Encryption function with a key
D	Decryption function
D_{key}	Decryption function with a key
X_i	Possible plaintexts
$P(\frac{X}{Y})$	Confidentiality
t	Time
$PS(t)$	Success probability of key tracing of a cryptographic algorithm
$PF(t)$	Failure probability of the key tracing or strength of the secret key
$K_x(t)$	Number of keys an attacker tries to process until time
2^n	Total number of possible keys
R_K	Rate of keys
m	Number of bits in its secret key
n	Number of bits
Key_{RF}	Impact of related cipher texts, biased input, and the number of rounds on key tracing

The MQTT protocol has no data confidentiality or security features in its design, and hence it is crucial to provide security for MQTT with cryptographic primitives. The IoT network comprises h number of healthcare devices, which are defined as a set $Dev = \{d1, d2, \dots, d_h\}$ in which Pub wants to establish communication with Sub to share patient information through lightweight MQTT communication. MQTT guarantees data confidentiality by employing the encryption and decryption strategies of lightweight symmetric key cryptography algorithms. Duplicate keys are used to encrypt and decrypt the original data. Consider that the term X denotes the plaintext and the term Y denotes the ciphertext.

Encryption: The encryption process of lightweight symmetric key cryptography can be expressed in the following equation 5.1.

$$E = E_{key}(X) \quad (5.1)$$

Here, the term E converts the plaintext X into the ciphertext Y .

Decryption: The decryption of lightweight symmetric key cryptography can be expressed in equation 5.2.

$$D = D_{key}(Y) \quad (5.2)$$

Here, the term D converts the ciphertext Y into plaintext X . In symmetric key cryptography, E_{key} and D_{key} are always similar, and the sender and receiver know a common relationship between those secret keys.

Confidentiality: Confidentiality is defined as a set of rules that restrict data manipulation and exposure by hackers during communication. In mathematical terms, the confidentiality of the data is expressed in cryptography. Figure 5.3 illustrates the encryption and decryption process in symmetric key cryptography.

For example, perfect key secrecy is primarily related to confidentiality in lightweight encryption. A ciphertext can achieve an ideal confidentiality level when the conditional probability value of a specific plaintext of a ciphertext is similar to the unconditional probability of the corresponding plaintext. It applies to every possible pair of plaintext

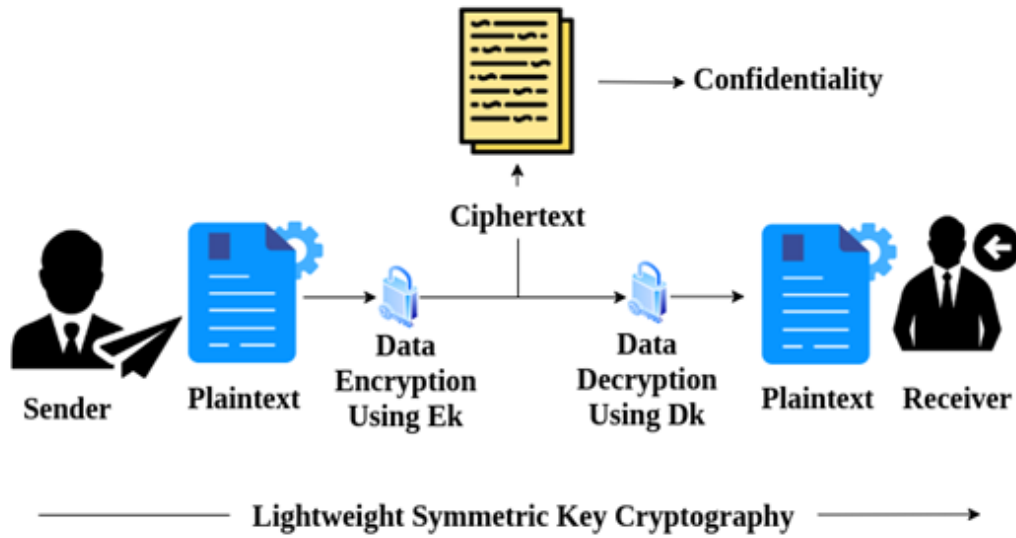


Figure 5.3: Lightweight Symmetric Key Cryptography

and ciphertext generated from lightweight cryptography in the MQTT communication environment. It can be expressed in equation 5.3.

$$P\left(\frac{X}{Y}\right) = - \sum_{x_i} p\left(\frac{x_i}{Y}\right) \log_2 p\left(\frac{x_i}{Y}\right) \quad (5.3)$$

Here, the term x_i is a possible plaintext taken to measure confidentiality in terms of $p(\frac{X}{Y})$. Maintaining confidentiality has a high conditional probability, meaning that the attacker can obtain little knowledge of the plaintext through the ciphertext. In MQTT communication based on lightweight cryptography, the strength of the keys can protect the confidentiality of the MQTT data, and the confidentiality formula could be changed depending on the strength of the encryption algorithm to prevent an attacker from deducing the plaintext from the ciphertext. The way of tracing the key of lightweight cryptography algorithms generally causes data security breaches. Hence, the fundamental keys protect the confidentiality of the data during data transmissions.

Moreover, the key-tracing property in lightweight cryptography can have severe implications for ensuring data confidentiality. Therefore, this work performs a formal security analysis of MQTT, which involves evaluating the security properties of lightweight cryptography methods for potential data security vulnerabilities by exploiting formal methods. Formal methods employ different mathematical strategies to model and analyse the

behaviour of the system and assist in determining the weaknesses by ensuring that the lightweight cryptography meets specified security requirements. In this work, formal analysis is performed in two ways. First, it assesses the confidentiality of transmitting data over MQTT by effectively analysing the encryption mechanisms of various lightweight cryptography algorithms. The formal methods can verify the strength of lightweight algorithms concerning their key generation and encryption/decryption processes. Estimates the level of data exposure due to attack behaviours and analyses the strength of the algorithm. Second, it assesses the susceptibility of MQTT data transmission to MitM attacks. Formal analysis can help identify weaknesses in the lightweight cryptography algorithm that could allow attackers to intercept and manipulate data.

Typically, attackers can trace security keys from lightweight algorithms and attempt MitM, Botnet, and DoS attacks. Botnets can cause spam and poisoning attacks. Replaying previously sent packets does not require secret key-breaking algorithms, but generating false packets for network damage, credential tracing, and data integrity violation requires tracing the secret key of the chosen encryption scheme [453]. At time t , the probability of key tracing $PS(t)$ is equal to the complement of the probability of key discovery by brute force attack. The probability of failure in key tracing or secret key strength $PF(t)$ is computed using the following equation 5.4.

$$PF(t) = 1 - PS(t) \quad (5.4)$$

To determine the value of $PS(t)$, it is essential to estimate the value of $PF(t)$. For a key with Ks bits, the probability of secret key tracing, $PS(t)$, can be estimated using the equation 5.5. By plugging the value of $PS(t)$ into equation 5.6, the value of $PF(t)$ can be estimated.

$$PS(t) = \frac{K_x(t)}{2^n} \quad (5.5)$$

$$PF(t) = 1 - \left(\frac{K_x(t)}{2^n}\right) \quad (5.6)$$

In equation 5.5, $PS(t)$ is estimated using $K_x(t)$, by the total number of possible keys, 2^n . Furthermore, $K_x(t)$ is calculated by multiplying the rate of keys R_K processed by an attacker over time t . Therefore, the equation can be rewritten as shown in equation 5.7.

$$PF(t) = 1 - (R_K \cdot \frac{t}{2^n}) \quad (5.7)$$

The security strength of a network is determined by the number of bits in its secret key, denoted n . To test all possible keys for a key with n bits, an attacker would require $2^n \times R_K^{-1}$ time units. The probability of a key of the network security strength getting zero within R_K^{-1} time units. The probability of a key with network security strength being traced within R_K^{-1} time units is denoted by $PF(t)$ and is defined by equation 5.8.

$$PF(t) = \begin{cases} 1 \text{ or } 1 - (R_K \cdot t/2^n) & \text{if } t \leq (2^n/R_K) \\ 0 & \text{otherwise} \end{cases} \quad (5.8)$$

Equation 5.8 shows that $PF(t)$ decreases as time passes, as more keys can be tested at each unit. Increasing the value of n can extend the duration of the security strength remaining at a level of one or more than zero. The measurement of cryptographic security strength is crucial for changing the key in the time $2^n/R_K$ to ensure the desired level of protection for the network and data. This metric can be used to compare different cryptographic algorithms. Therefore, the proposed work organised the security algorithms, such as PRESENT, FBC and LED, according to their security strength or key size. Moreover, formal analysis is essential to ensure the effectiveness and robustness of the MQTT protocol's security measures, especially in the Healthcare 4.0 environment, where data confidentiality and overall system security are critical. The above-mentioned security strength measurement is a universal measure for all cryptographic algorithms. However, even though security strength is an essential metric for determining the appropriate cryptographic algorithm, some attacks use the specific characteristics of the cryptographic algorithm to trace the secret key.

The FBC lightweight security schemes use small key sizes, making their $PF(t)$ values small [454]. This vulnerability makes them susceptible to MitM attacks, spam, and poi-

soning attacks. The secret keys of PRESENT algorithms can also be traced by analysing biased inputs in the key space and relating the generated ciphertexts. The number of rounds used in lightweight security schemes also affects key tracking. The iterative blocks in these schemes are considered as rounds, which execute three or more subfunctions and work effectively against linear and differential cryptanalysis in MQTT communication. The impact of related cypher texts, biased input, and the number of rounds on key tracing is noted as Key_{RF} . Due to fewer rounds, the LED is susceptible to brute-force attacks, resulting in a modification of the $PF(t)$ value as shown in Equation 5.9.

$$PF(t) = \begin{cases} 1 \text{ or } 1 - (R_K \cdot t \cdot Key_{RF}/2^n) & \text{if } t \leq (2^n/R_k \cdot Key_{RF}) \\ 0 & \text{otherwise} \end{cases} \quad (5.9)$$

Equation 5.9 shows that the time interval for key renewal in MQTT communication over IoT networks is determined by R_K and Key_{RF} to ensure security. Additionally, it should be noted that the likelihood of key tracing is higher for LED and FBC algorithms compared to PRESENT because of the involvement of less complex operations. In fact, for the LED and FBC algorithms, the value of $PF(t)$ decreases before reaching the time unit t , which indicates a higher risk of key tracing.

5.5.2 Tool Based Security Analysis

We verify and evaluate the five aforementioned symmetric algorithms in MQTT communication using security analysis and the following simulation tools Scyther [420], JCrypTool [421].

Scyther and JCrypTool are used for security analysis and algorithm verification. The tool-based security analysis model aims to analyse the security level of the LED, PRESENT, and FBC cryptography algorithms over MQTT to enable smart healthcare scenarios using JCrypTool and Scyther. Security properties are validated using JCrypTool to simulate encryption within MQTT message exchanges in a healthcare context. Scyther offers some specific feature-based analysis that JCrypTool does not provide, such as unbounded verification and in-depth security analysis regarding claims and trace pat-

terns. It assists in analysing possible attacks and protocol behaviour. On the other hand, JCrypTool includes many cryptographic mechanisms, including symmetric and asymmetric encryption, hash functions, analysis tools, visualisations, and crypto games, and helps verify cryptographic algorithms. In general, combining these tools provides a comprehensive evaluation of the security and performance of MQTT and the selected cryptographic algorithms in various attack scenarios.

Scyther Tool-based Analysis

Scyther is a tool that analyses possible attack classes and the behaviour of the protocol. Cryptographic algorithms, namely the PRESENT, LED, FBC and MQTT descriptions, are specified in the Security Policy Definition Language (SPDL), which serves as input for Scyther's verification. The Scyther tool automatically proves the security properties or provides knowledge of the presence of an attack on a cryptographic protocol. It offers an easy way to model security properties such as secrecy and authentication. Scyther follows the Dolev-Yao intruder model [455], and this model abstracts the messages using term algebra and formalises the cryptographic primitives used in the cryptographic algorithm. In addition, it designs a perfect encryption hypothesis, and a malicious node can decrypt an encrypted message if an attacker knows the associated secret key. If a cryptographic algorithm is secure against such an attacker, it can generate traces of attack patterns. The descriptions of the MQTT protocol with selected cryptographic algorithms are given in the SPDL language. The descriptions are called claims. The following descriptions are written in SPDL for all cryptographic algorithms.

- (1) The initiator sets up the secret parameters required for secure communication.
- (2) The server performs all the required operations based on the initiator functions.
- (3) Once the server finishes its operation, it sends the output to the receiver.
- (4) The receiver receives all the outputs and displays the final commit on the receiver side if any attacks are present on the network.

The Scyther tool-based verification and evaluation for FBC algorithms is explained using three steps: input creation, algorithm verification, and output results.

- a) **Input creation:** Initially, the Scyther tool window is opened for evaluation, and the Scyther tool sets the necessary FBC parameters to initialise the verification process. It uses the SPDL language to take the description and use the FBC algorithm related to coding. Initially, Scyther divides the MQTT message into two parts: left and right. In addition, it starts the verification process.
- b) **FBC Verification:** The protocol verification uses four criteria: claim, status, comments, and patterns.
- *Claim:* After setting the main parameters, the verification process is started by clicking the Verify/Verify protocol (F1). A window appears, and each row in such a window represents one claim. The claim menu shows the individual processes of the initiator, receiver, and server.
 - *Status:* As per the cryptographic algorithm, the secret parameters are varied. The status shows the presence of attacks in terms of OK and FAIL.
 - *Comments:* The comments option is used to show the comments of the attack. There are two types of comments: no attacks within bounds, and if there is an attack. In addition, the comments are classified into three types as follows.
 - (i) At least n attacks.
 - (ii) At most n attacks.
 - (iii) Exactly n attacks.
 - *Pattern:* The pattern menu is used to show the attack pattern of a network. Different attack patterns are generated on the receiver side according to the cryptographic algorithm and its secret parameter verification. The FBC algorithm obtains 12 attack patterns during the Scyther tool-based verification process. Finally, the role of characterisation is used to show the presence of attack trace patterns on the initiator, receiver, and server sides.
- c) **Output Results:** The trace pattern shows whether the claim status is reachable. The attack patterns for DoS and MitM attacks obtained for the FBC algorithm are shown in the following figure. Figure 5.4 shows that the result of the DoS attack pattern is obtained according to the time synchronisation between patients and

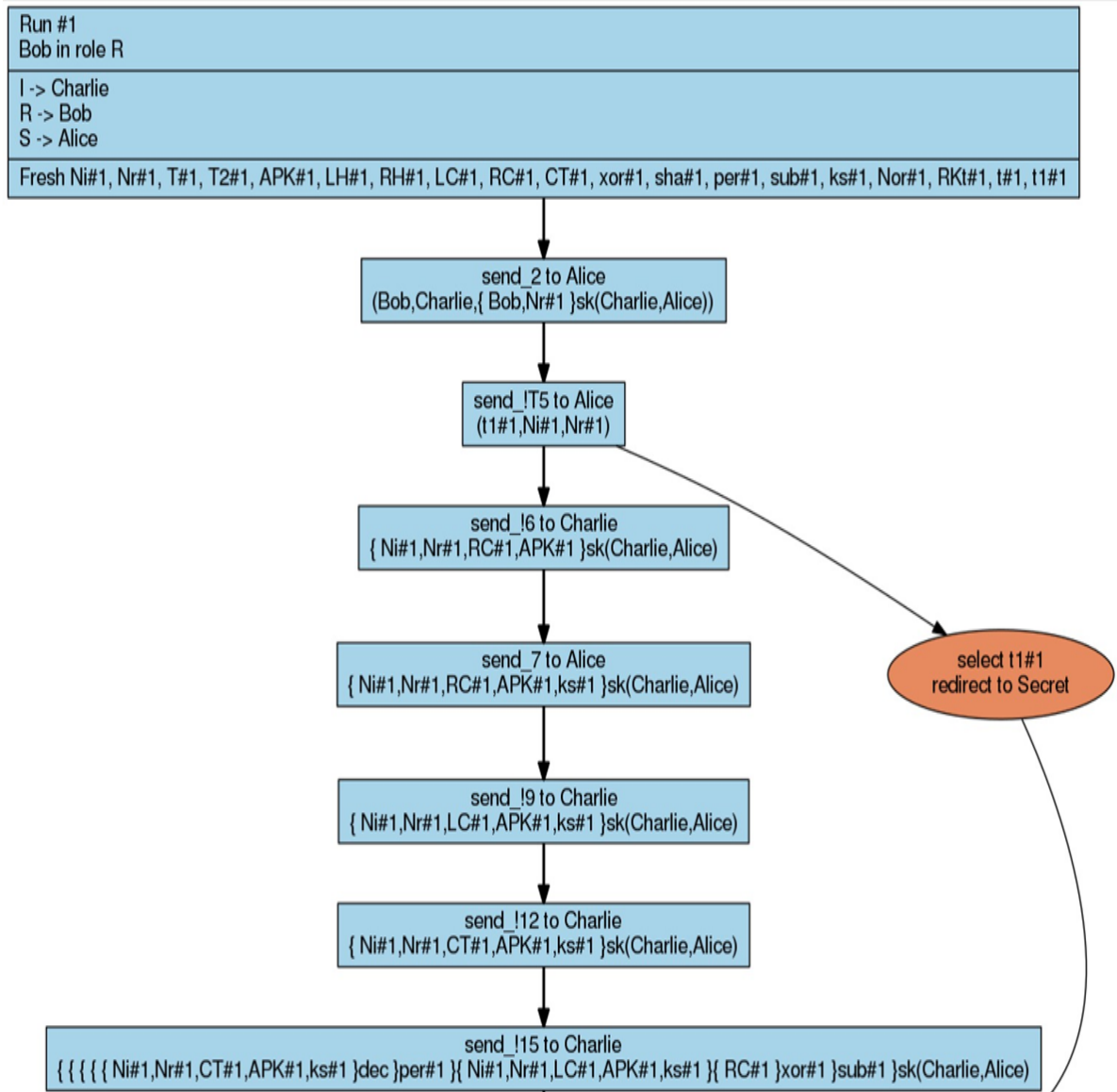


Figure 5.4: Attack Pattern Result of DoS Attack

remote locations. Second, Figure 5.5 shows that the MitM attack traces decrypt the left-hand-side ciphertext message of the MQTT communication initiated between the sender and the receiver.

Similarly, the descriptions of the cryptographic algorithms PRESENT and LED are provided as input to the Scyther tool for the verification process. These algorithms obtain twelve and seven attack patterns on the receiver side, respectively. Hence, the attack patterns show the security strength of the algorithms against DoS and MitM. The attack pattern depends on the security parameters used in the specific cryptographic

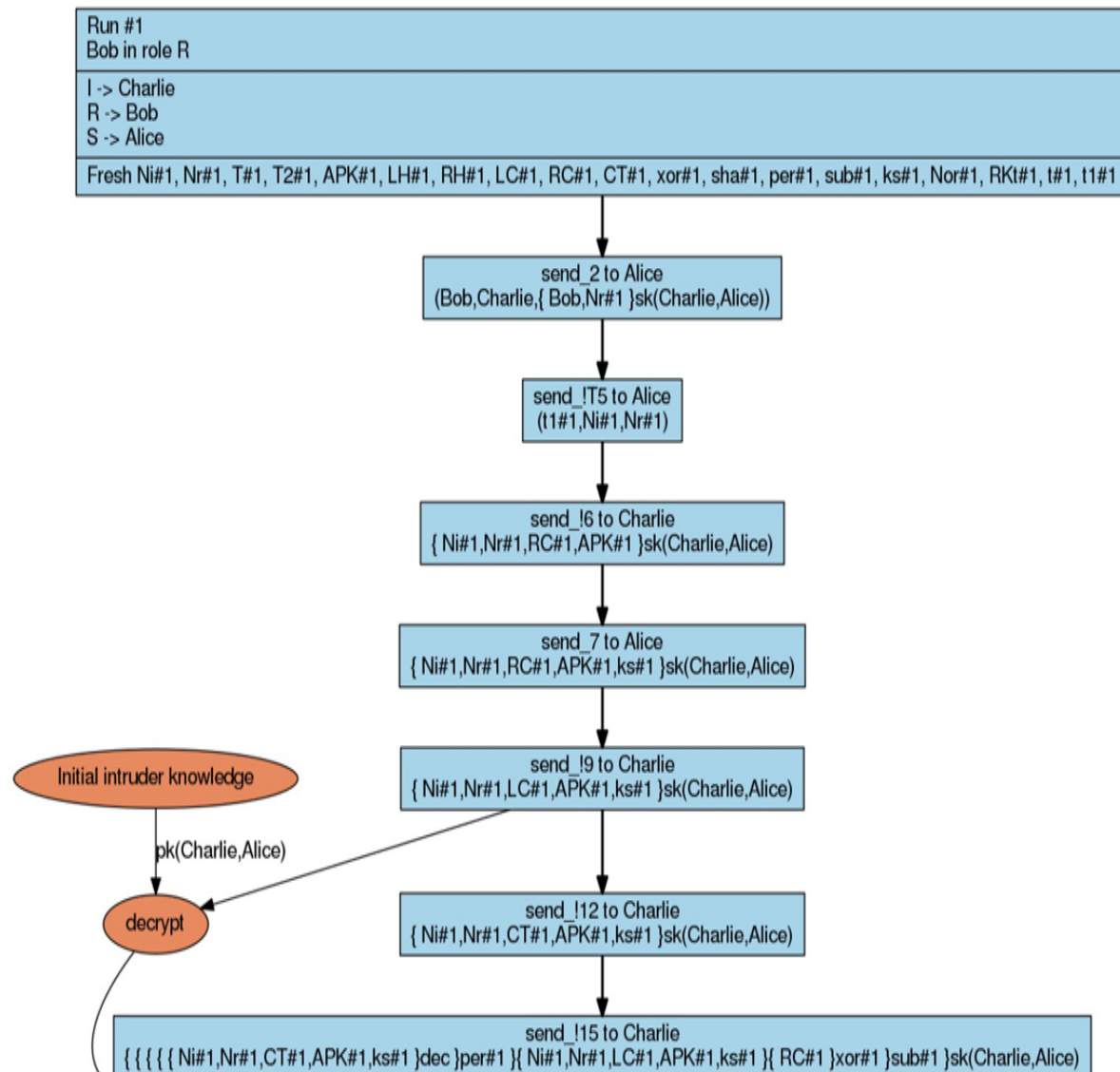


Figure 5.5: Attack Pattern Result of MitM Attack

algorithm. The high number of attack patterns increases the possibility of key tracing and reduces the security strength of the FBC, LED, and PRESENT cryptographic algorithms. Therefore, the Scyther results show that the LED has fewer attack patterns than FBC and PRESENT.

JCrypTool-based Analysis

Another tool for validating cryptographic algorithms is JCrypTool. All lightweight FBC, LED, and PRESENT are implemented in JCrypTool. According to JCrypTool, lightweight cryptographic algorithms such as PRESENT, LED, and FBC are simulated using the Cooja simulator, and the mote output files from the simulator are used as input for security analysis.

The following process is applied to each algorithm in JCrypTool:

- (1) Multiple ciphertexts are analysed to determine the possibility of traceback to the corresponding plaintext or secret key.
- (2) The frequency of characters in the ciphertexts is analysed to determine the possibility of tracing the plaintext.
- (3) Transposition analysis is performed to determine the possibility of tracing back to the corresponding plaintext or secret key.

The steps for JCrypTool-based analysis are input selection, encryption process, decryption process, and output.

- (1) Multiple ciphertexts are analysed to determine the possibility of traceback to the corresponding plaintext or secret key.
- (2) The frequency of characters in the ciphertexts is analysed to determine the possibility of tracing the plaintext.
- (3) Transposition analysis is performed to determine the possibility of tracing back to the corresponding plaintext or secret key.

The steps for JCryp tool-based analysis are input selection, encryption process, decryption process, and output.

- (a) **Input Selection:** The MQTT-based Healthcare 4.0 message is an input to Jcryp-

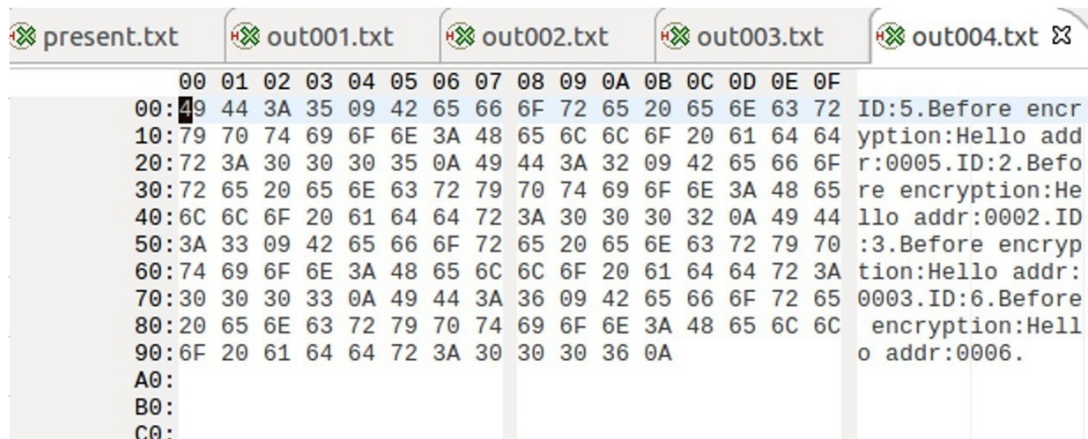


Figure 5.6: JCryptTool Output Results of PRESENT-based MQTT

Tool. The message is encrypted and decrypted using the PRESENT cryptography algorithm. Therefore, the PRESENT cryptography-related information of the corresponding MQTT message is provided as input to the tool. The double-click menu is used to select the files from the repository. Substitution and permutation are applied for data encryption and decryption.

- (b) **Encryption Process:** In this step, messages are encrypted using the encryption key provided by the PRESENT algorithm. Therefore, each patient has a unique ID with a password. Initially, JCryptTool selects an XOR with 64 characters for alphabet selection. Before encryption, a pre-operation transformation is used to alter the text before the encryption operation is applied to the text file. Consequently, it manually enters the password into the encrypted plaintext file. It is displayed in the JCryptTool.
- (c) **Decryption Process:** It is the reverse process of the encryption process. Original plaintext information is obtained using the same password and key used for encryption. The same alphabet selection and permutation operation is performed for decryption.
- (d) **Output:** A pre-operation transformation is used to alter the information to analyse the security strength of the PRESENT-based MQTT. The final results demonstrate the security level of the PRESENT algorithm. The output is shown in the following Figure 5.6.

Similarly, the other two algorithms, LED and FBC, are verified using JCrypTool. According to the results of JCrypTool, the PRESENT algorithm provides high security against DoS attacks, and FBC offers high security against MitM attacks. Finally, using formal and tool-based security analysis methods, the proposed work arranges lightweight security algorithms in PRESENT, LED, and FBC according to computational complexity and security level.

5.6 Chapter Summary

This research applies lightweight symmetric encryption algorithms to Healthcare 4.0 scenarios based on MQTT communication. We also integrate the unique performance and security needs of Healthcare 4.0 into validating symmetric encryption algorithms. However, these algorithms still have potential security vulnerabilities and may not be suitable for specific resource-constrained environments. The proposed approach performs an in-depth analysis of various cryptography algorithms under diverse network conditions, resulting in broad validation techniques with diverse metrics and scenarios. This work included two methods for comprehensive analysis: formal security analysis and tool-based security analysis. Using mathematical strategies, the formal method effectively analyses the security strength of five symmetric key algorithms integrated with MQTT under MitM and DoS attacks. Tool-based analysis using Scyther and JCrypTool demonstrated the security strength of five algorithms by rigorously verifying their resilience to diverse attacks and validating their cryptographic robustness. Metrics and scenarios used throughout the chapter were selected to reflect operational realities in innovative healthcare systems. The findings highlight the viability of lightweight algorithms such as PRESENT and FBC in maintaining data confidentiality and system efficiency under healthcare-specific constraints. PRESENT and FBC are also applicable in resource-constrained IoMT environments. To evaluate the security strength and performance of encryption algorithms in different attack modes, we will use the Cooja simulator for the evaluation analysis in the next chapter.

Performance Evaluation for MQTT in Healthcare 4.0

Based on the verification analysis in the previous chapter, this chapter evaluates the robustness of five lightweight symmetric key algorithms in various attack scenarios using the Cooja simulator. At the same time, combined with the simulation results, the performance of the five lightweight symmetric key security algorithms is compared and analysed in Healthcare 4.0 under MitM and DoS attacks.

6.1 Performance Evaluation using Cooja Simulator

This section evaluates the performance of basic MQTT v5 with and without symmetric encryption algorithms. The simulation on IoT nodes is conducted using the Paho C Client Library Module, Ubuntu 18.04 LTS 64-bit, Instant Contiki-3.0, and VMware Player 17. The Paho C Client can be configured by exploiting SSL/TLS secure connections. Using the Paho C client module in the Contiki/Cooja simulation setting is not feasible. Therefore, the simulation incorporates the MQTT library source code by placing it in the relevant directory with Paho C. This incorporation is a crucial component of the Paho C Client structure, as it houses the actual source code responsible for the functionality and security features like data confidentiality provided by the MQTT v5 library. In addition, it uses an updated Makefile to construct the project with the MQTT v5 configuration.

Finally, the evaluation is conducted. Many researchers use the Cooja simulator to simplify the research environment required to achieve their findings [422, 427, 456–460]. The Cooja is an open-source network simulator suited to simulate resource-constrained IoT networks. The Cooja simulation environment allows us to realistically model the resource-limited devices of MQTT, including different aspects, such as energy consumption, low-power modes, and radio communication. This realism is crucial for assessing the

impact of MQTT communication on resource-limited devices.

Therefore, this work selects a Cooja-based simulation environment to analyse the performance of MQTT v5 with different lightweight symmetric key cryptography algorithms. The main reason is that Cooja is consolidated with Contiki OS, which supports MQTT v5 data transmission, making it highly suitable to simulate MQTT v5-enabled applications. Realistically simulates MQTT v5 by evaluating it with various features that improve the properties of the protocol, improve error handling, and make it highly adaptable to MQTT application-specific environments. Furthermore, the Contiki OS offers precise MQTT v5 implementations by creating a controlled communication environment through advanced features.

In addition, Cooja provides a platform for tuning different protocol parameters for MQTT v5 and allows one to evaluate their impact on the performance of the entire system. Moreover, Cooja allows various simulations from the data to the application layer. It is highly flexible in implementing the MQTT v5-based IoT-assisted healthcare scenario with more effectiveness and less memory usage. For evaluation, the performance of MQTT v5 under AES, DES, PRESENT, LED, and FBC is compared with the basic MQTT-v5 (B-MQTT) protocol. The performance of MQTT v5-IoT is evaluated using throughput, packet delivery ratio, delay, execution time, energy consumption, and CPU energy consumption metrics. The simulation parameters are given in Table 6.1.

6.1.1 Network Scenario Creation

A 31-node topology is created during simulation using different sensor devices and MQTT brokers with one server, as shown in Figure 6.1. The nodes include one border router, 15 publishers, and 15 subscribers. Hence, the publishers are IoT sensing devices, servers, and doctors. The nodes from the identity of 2 to 16 and 16 to 31 act as publishers and subscribers, respectively. Increasing the number of nodes in the simulation changes only the metric values; however, the entire experiment is the same for all algorithms to maintain the scalability of the network. Thus, the proposed work takes only 31 node topologies for performance evaluation. In addition, the network communication range is

Table 6.1: Simulation Model.

Parameters	Values
Application Layer Protocol	MQTT version 5.0
Total Number of Nodes	30 and 60 Router - 1 Publisher - 15 and 30 Subscribers - 15 and 30
IoT Devices	Temperature sensor, Pulse rate sensor, accelerometer sensor, and analogue device
DoS attack Nodes	1 to 2
MitM attack Nodes	1 to 2
Data Rates (bits/second)	64, 128, 192
Simulation Area	$100m \times 100m$
Transmission Range	50m
Simulation Time	5 Minutes
MQTT Broker	Mosquito-RSMB Broker-1.3.0.2
Algorithms	AES, PRESENT, LED, FBC, DES

fixed at 50m by considering the allocation of battery resources to the nodes. For the same reason, the simulation time is set to 5 minutes, which can prevent network slowdown. Moreover, the message size is denoted in bytes, which varies for each algorithm.

6.1.2 Attack Scenario Creation

Our network model assumes that nodes 22 and 23 act as DoS attackers, attempting to compromise the generated messages. These malicious nodes cannot be predicted in advance. To mitigate the threat, lightweight encryption schemes apply cryptographic operations to messages to avoid potential attackers. We evaluated both ideal and malicious environments. Figure 6.2 illustrates the attack scenario, with different colours representing the various roles:

- The green node represents the border router.
- The orange nodes represent the publishers.
- The pink nodes represent the subscribers.

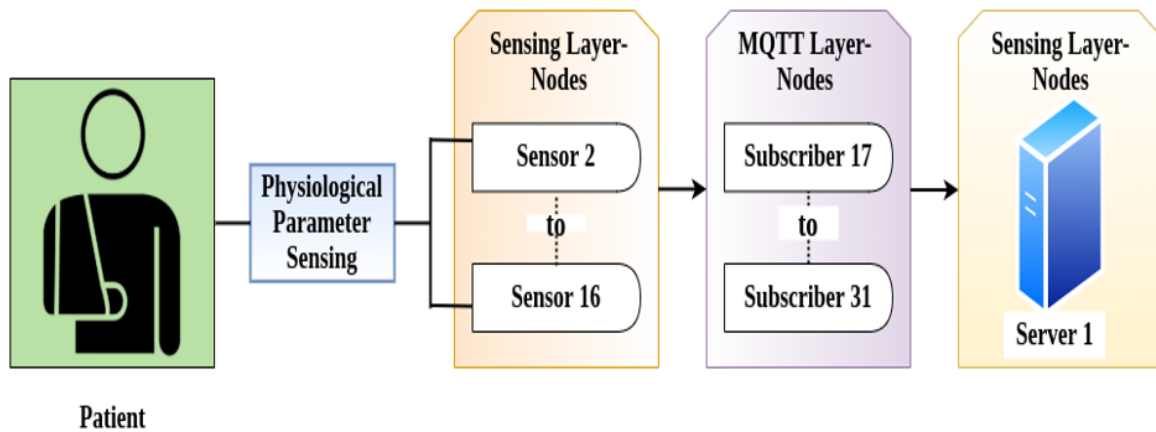


Figure 6.1: Simulation Scenario of Remote Patient Monitoring System

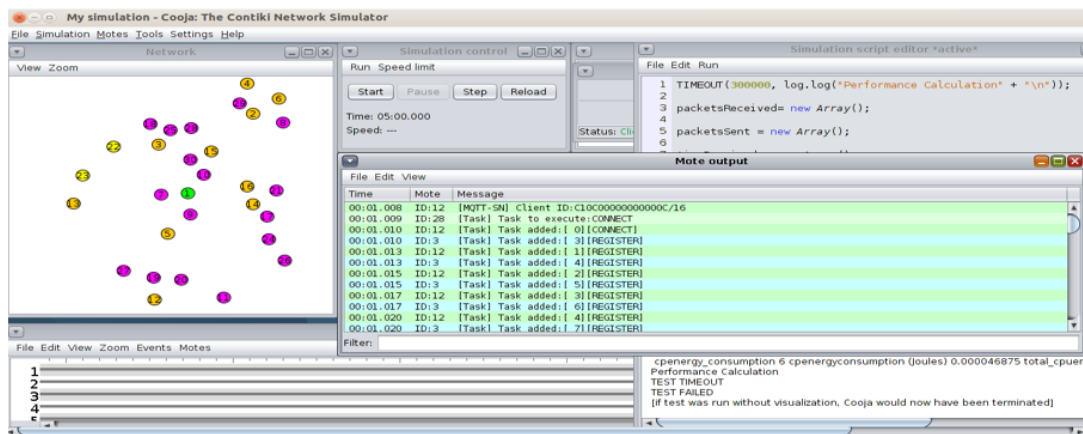


Figure 6.2: Attack Scenario

d. The yellow nodes represent the attacker nodes.

For simulation, we used the Z1 [456] mote type as the hardware type for the nodes.

DoS Attack

The client determines the keepalive timeout value during its connection to the broker. A standard value for this parameter is 60 seconds, meaning the client should send an MQTT packet at least once every 60 seconds. In cases where no other packets are sent, a PINGREQ packet can be used, and the broker is expected to respond with a PINGRESP packet. This mechanism aims to detect “dead” TCP connections that can sometimes accept writes for some time, even after the connection has been terminated.

MitM Attack

This type of attack is known as a MitM attack, where the attackers intercept and manipulate the communication between two parties, pretending to be one or both. This attack allows the attackers to eavesdrop on the conversation, steal sensitive information, or even manipulate the communication in real-time. The MitM attack is considered a severe threat to the security of communication systems and is commonly prevented using encryption and authentication techniques.

6.2 Simulation Results

To analyse the efficiency of various symmetric key algorithms in different attack scenarios, we performed simulations with varying numbers of nodes, attacks and data rates.

6.2.1 MQTT without Attack Scenario

To effectively analyse the performance efficiency of MQTT with five various symmetric key algorithms, this section observes the performance results of the scenario without attackers by creating two scenarios. The first scenario varies the number of nodes from 30 to 60. In the second scenario, the data rate varies from 64 to 192 bits/second.

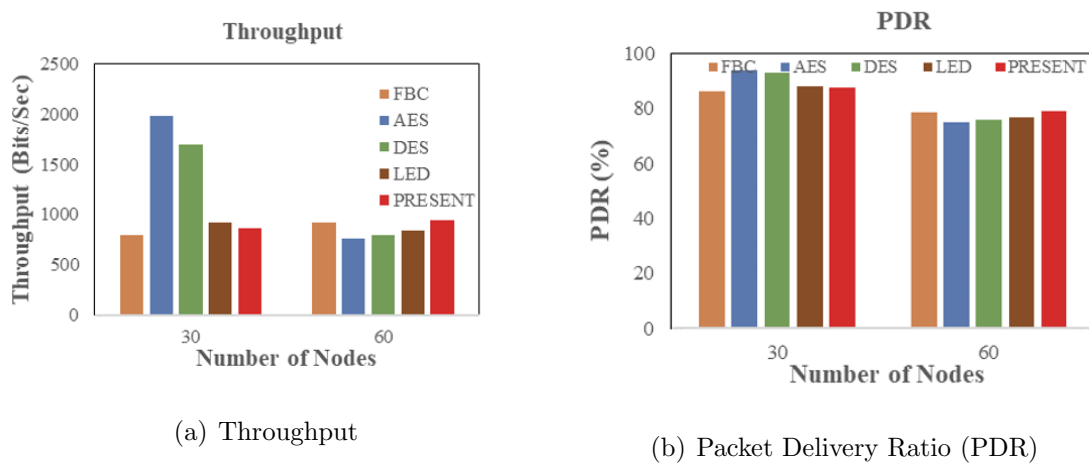


Figure 6.3: Throughput and PDR Under Number of Nodes

Figure 6.3 shows the throughput and PDR results of MQTT without attack scenarios. The results are obtained by varying the number of nodes from 30 to 60. The figure

demonstrates that throughput and PDR are reduced by increasing the number of nodes from 30 to 60. The main reason is that some packets are lost due to competition between nodes for link access. For instance, the AES and DES achieve higher throughput by 1979.73 bits/sec and 1698.13 bits/sec for the 30-node scenario, but it is varied by 755.2 bits/sec and 793.6 bits/sec, respectively, for the 60-node scenario. Similarly, the PDR of AES is 94.11%, which is high in 30 nodes, while it achieves 75% PDR in the 60 nodes scenario. The PDR of AES is reduced by 19.11% when the number of nodes varies from 30 to 60.

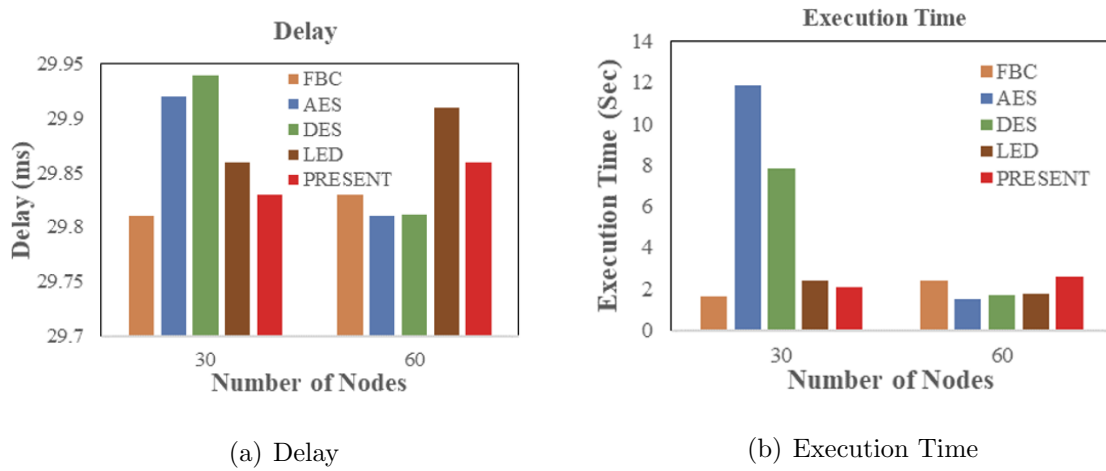


Figure 6.4: Delay and Execution Time for Different Number of Nodes Scenario

Figure 6.4 shows the results of delay and execution time comparison of five symmetric key cryptography algorithms obtained for different numbers of node scenarios. Different algorithms achieve different delay and execution time results. The reason is that the key distribution and management of various lightweight symmetric key cryptography algorithms, such as FBC, AES, DES, LED and PRESENT, differ. For example, DES accomplishes a high delay of 29.94 milliseconds under 30 node density scenarios, but it varies by 29.812 milliseconds in 60 node scenarios, respectively. Similarly, the execution time of AES is high by 11.89s in the 30-node scenario, but it is varied by 1.48 sec under a high number of nodes scenario like 60.

Figure 6.5 compares the energy and CPU consumption of FBC, AES, DES, LED, and PRESENT obtained for 30- and 60-node scenarios. Generally, energy consumption increases as the number of nodes varies from low to high. The reason is that the number

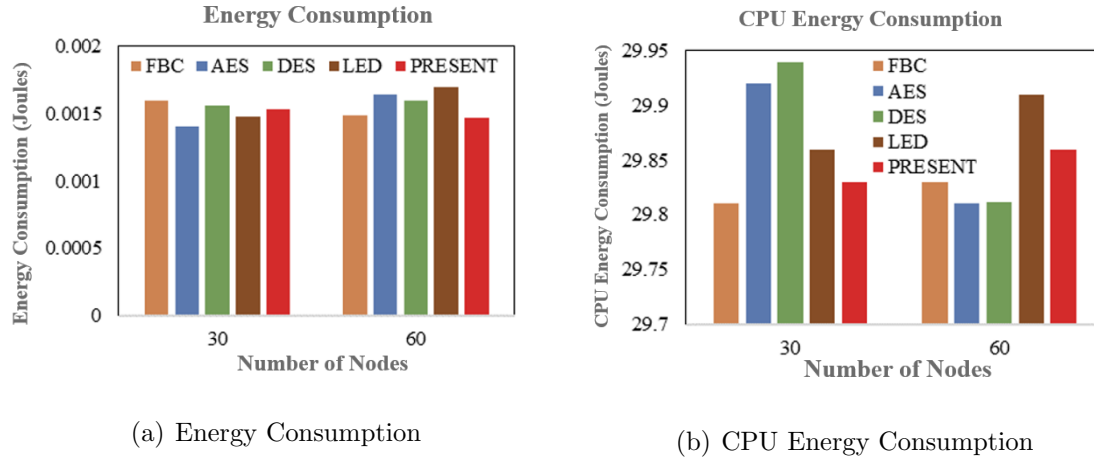


Figure 6.5: Energy Consumption and CPU Energy Consumption for Various Numbers of Nodes

of key distributions that occur in many nodes in high-density scenarios also increases energy consumption. For example, the energy consumption of FBC is high in the 30-node scenario, and the energy consumption of LED is high in the 60-node scenario, respectively. The CPU energy consumption of DES is for 30 and 60-node scenarios, respectively.

Figure 6.6 presents a comparison of the throughput and packet delivery ratio of six algorithms, namely B-MQTT, FBC, AES, DES, LED and PRESENT, at varying data rates from 64 to 192 bits/second under the no-attack scenario for both 30 and 60 nodes.

The results indicate that the B-MQTT protocol outperforms other MQTT protocols that use symmetric key algorithms regarding both throughput and packet delivery ratio. The MQTT protocol has no built-in security mechanisms and is simple to implement, making it suitable for IoT environments. However, security algorithms are required to ensure security in MQTT-based IoT environments, which can result in some packet losses on the network. For example, in the absence of an attack scenario, B-MQTT achieves a throughput of 2542.9 bits/sec and a packet delivery ratio of 97.7% at a data rate of 64 bits/second for 30 numbers of node scenarios, respectively. Among the remaining algorithms, FBC uses simple encryption and decryption schemes to achieve security in MQTT. In contrast, the LED and PRESENT algorithms involve more complex encryption and decryption schemes, resulting in poorer throughput and packet-delivery ratio results. For example, AES and DES achieve nearly equal packet delivery ratio performance under the 64 bits/second data rate scenario of Figure 6.6(c). Both methods integrate strong

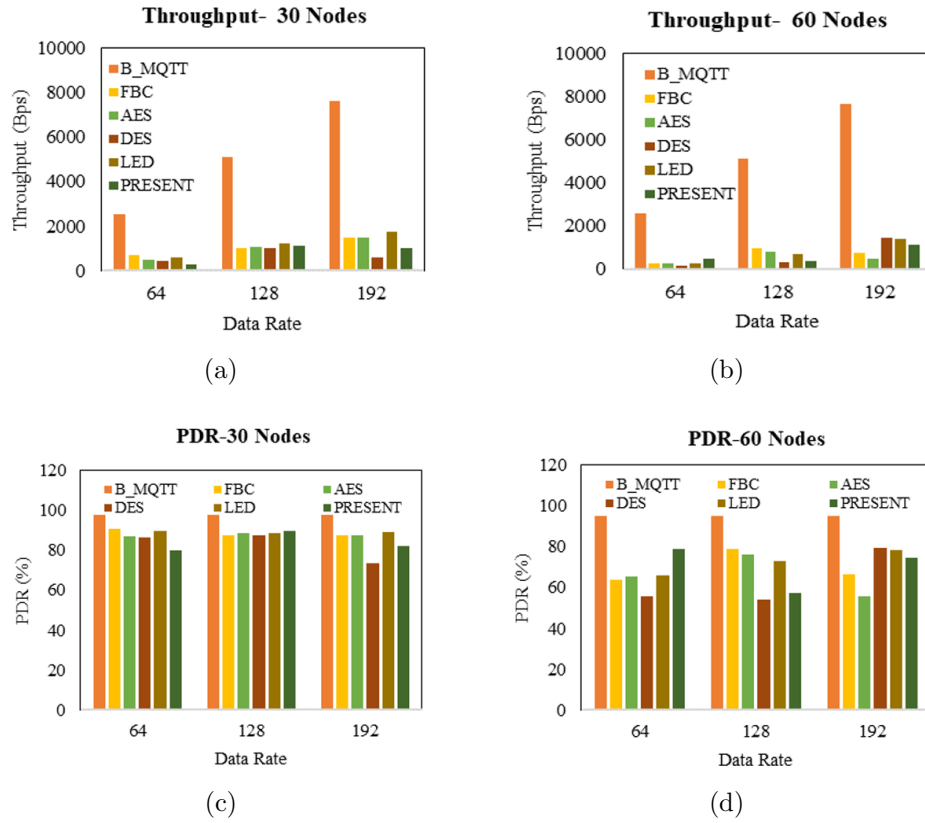


Figure 6.6: Throughput and Packet Delivery Ratio under Various Data Rates

encryption and decryption schemes to achieve security in MQTT. For example, AES and DES achieve an 87% packet delivery ratio for 64 data rate scenarios and 30 node scenarios, respectively.

Figure 6.7 compares the delay and execution time of B-MQTT, FBC, AES, DES, LED and PRESENT, obtained by adjusting the data rate from 64 to 192 bits/second and varying the number of nodes from 30 to 60. Each algorithm increases the delay and execution time, changing the data rate from low to high. This process is caused by the MQTT server providing many keys and increasing the number of rounds of keys for different data rates. Thus, it increases some delay and execution time, as shown in Figures 6.7(a) and 6.7(c). However, the delay of base MQTT is reduced compared to other algorithms. The reason is that the base-MQTT does not have encryption and decryption features, resulting in a minimum delay. For example, the base-MQTT accomplishes 29.9 milliseconds and 1.94 seconds of delay and execution time for a 64 bits/ second data rate scenario, as shown in Figures 6.7(a) and 6.7(c). Figure 6.7(a) demonstrates that

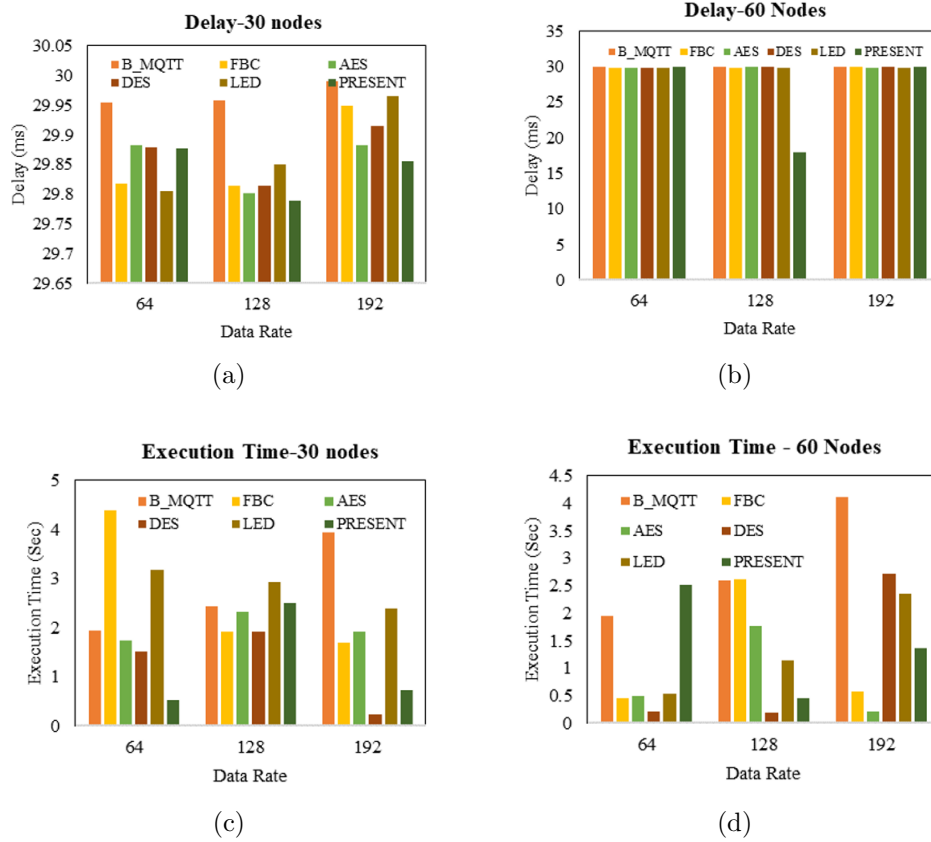


Figure 6.7: Delay and Execution Time under Various Data Rates

PRESENT accomplishes the minimum delay. The PRESENT includes straightforward encryption and decryption processes, reducing the delay under a high data-rate scenario. For example, the MQTT base and FBC achieve the same delay value of 29.85 milliseconds under a 192 bits/second data rate scenario for the 30-node density, as shown in Figure 6.7(a).

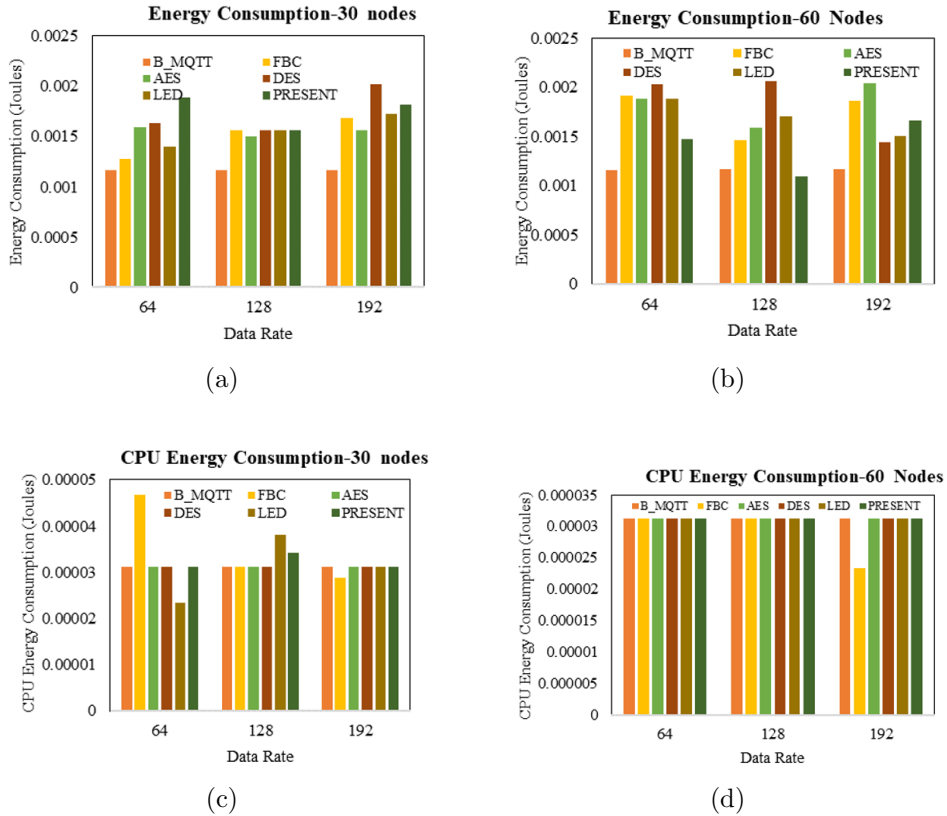


Figure 6.8: Energy Consumption and CPU Energy Consumption Results for Various Data Rates

Figure 6.8 presents a comparative evaluation of the energy consumption and CPU energy consumption of six algorithms, namely B-MQTT, FBC, AES, DES, LED, and PRESENT, for three different data rates (64, 128, and 192 bits/second) under a 30 and 60 node topology. As shown in Figure 6.8(a), the B-MQTT algorithm results in a lower energy consumption in the network than the other five algorithms. B-MQTT does not employ cryptographic data encryption and decryption, which reduces energy consumption. For example, B-MQTT attains an energy consumption rate of 0.00116 joules for 64 bits/second and 192 bits/second data rate scenarios for 30 node typologies. This is

because changing the data rate does not affect the node energy consumption level.

Similarly, the base-MQTT accomplishes 0.00003125 and 0.00003123 joules of CPU energy consumption for 64 and 192 bits/second data rate scenarios under 30-node scenarios. The DES consumes more CPU energy than the other five algorithms. The simple key structure followed by FBC significantly decreases energy consumption. In MQTT with various symmetric encryption algorithms, the difference in energy consumption is mainly due to the size of the key and the number of operations performed during encryption. An attacker who can figure out the overall key can cause unnecessary energy loss in IoT devices.

6.2.2 Performance Evaluation of Different Nodes under MitM or DoS Attacks Scenarios

The section simulates two attacks, DoS and MitM, under environments with varying numbers of nodes. The purpose is to analyse the impact of these DoS and MitM attacks on network performance metrics such as PDR, throughput, latency, overhead, and energy consumption. By exploring distinct node configurations of network scenarios, ranging from low-density to high-density, this section evaluates how the attacker's presence and behaviour influence MQTT communication reliability and security. This section creates scenarios with 30 nodes, two attackers, 60 nodes, and four attackers to simulate DoS or MitM attacks. The data rate was fixed at 128 bits to compare the performance of the algorithms.

Figure 6.9 presents the results of the throughput comparison of FBC, AES, DES, LED, and PRESENT under different numbers of DoS and MitM attack scenarios and various numbers of nodes. In a scenario with a high number of nodes, the impact of an attacker may be even more pronounced than in a scenario with a low number of nodes. Increasing node density can create more attack opportunities and disrupt communication. Conversely, the attacker effect may be less severe in low-node density scenarios but still significant enough to impact network efficiency. Moreover, the presence of DoS and MitM attackers typically creates a noticeable degradation in network throughput,

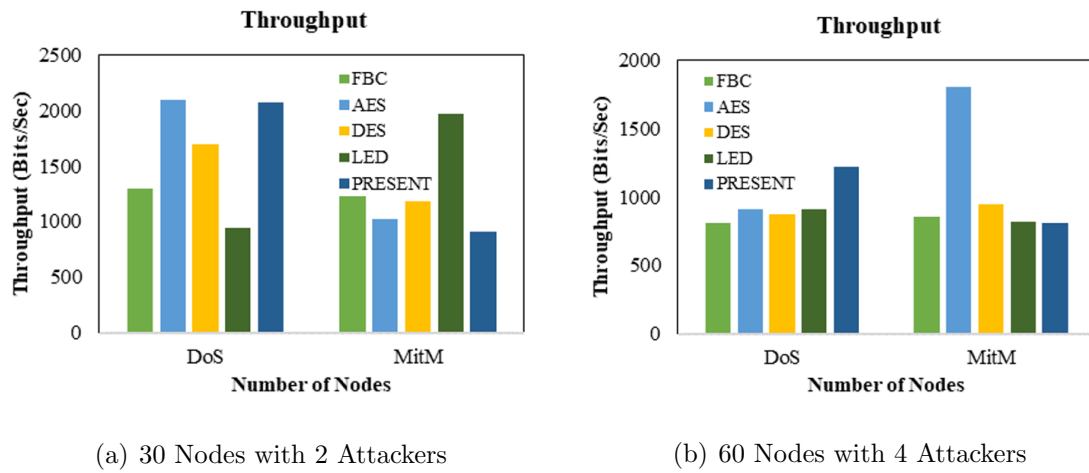


Figure 6.9: Throughput Results for Different Nodes and Attacker Scenarios

affecting efficiency and reliability.

Figure 6.9(a) shows the throughput results for 30 nodes with two attacker scenarios. Figure 6.9(b) illustrates the results for 60 nodes with four attacker scenarios, with a fixed data rate of 128 bits/sec. We observed that different lightweight algorithms achieve various throughput values when the node density increases from 30 to 60. For example, the FBC achieves 1297.06 and 806.4 bits/second throughput values for 30 nodes with two attackers and 60 with four attackers, respectively. The AES and DES are superior to FBC, LED, and PRESENT under the 30-node density scenario. The reason is that AES and DES include a fast and complex key structure to attain strong security against DoS and MitM attackers, resulting in high throughput even when the number of attackers increases. For example, the AES achieves 1809.6 bits / second throughput under MitM attack for 60 nodes with four attacker scenarios, whereas the DES, LED, FBC and DES achieve 947.2, 819.2, 853.33, and 806.4 bits / second throughput for the same scenario.

Figure 6.10 presents the results of the PDR comparison of various symmetric key lightweight cryptography algorithms under different attack and node scenarios. Attackers significantly impact performance, such as PDR, by disrupting communication. Compared to a low node density scenario, a high node density scenario is highly vulnerable to attack activities. The DoS attacker can reduce PDR by flooding unnecessary message traffic into the communication. Figure 6.10(a) shows the PDR results of FBC, AES, DES, LED, and PRESENT in two DoS and MitM attacker scenarios with 30 nodes. It is observed that

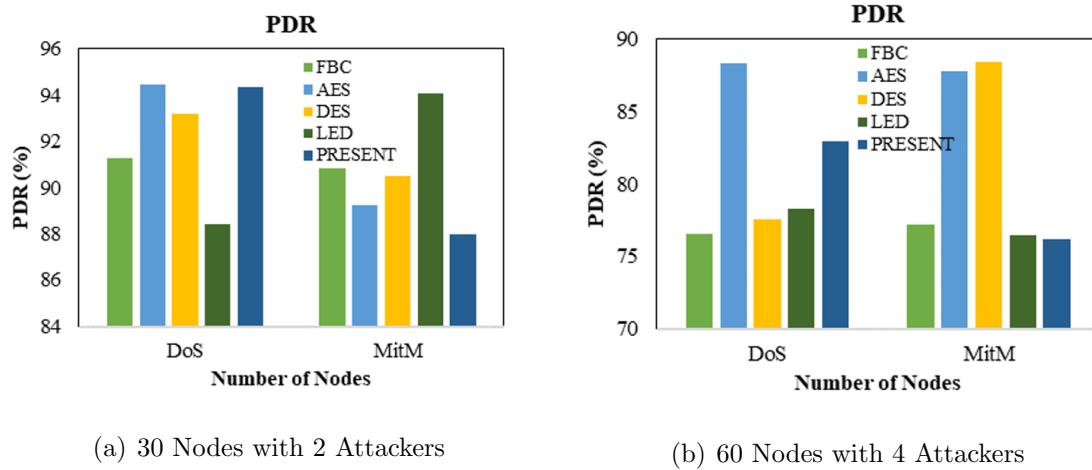


Figure 6.10: PDR Results for Different Nodes and Attacker Scenarios

AES achieves better PDR results than the other four protocols in the DoS scenario. The reason is that AES is highly flexible by rapidly providing a robust security level with a 128-bit key and a simple encryption and decryption model. For example, AES achieves 94.44% and 88.38% of PDR for DoS attack scenarios for 30 and 60-node scenarios, respectively, in Figure 6.10(a). Likewise, the FBC accomplishes 91.29% and 76.51% of PDR values for the scenario in Figure 6.10. The PDR results decrease with the number of nodes and attackers varying from 30 to 60 and from two to four. The reason is that the nodes compete to access the links under a high node density scenario, thus minimising the PDR rate.

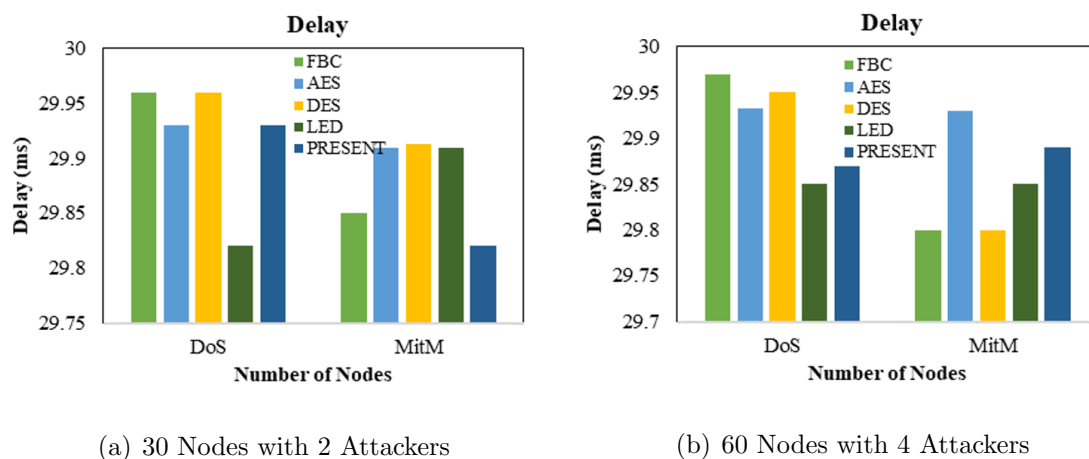


Figure 6.11: Delay Results for Different Nodes and Attacker Scenario

Figure 6.11 presents the delay results of the FBC, AES, DES, LED, and PRESENT

algorithms obtained under scenarios with 30 nodes, two attackers, and 60 nodes and four attackers. Delays are generally improved in low-node density scenarios compared to high-node density scenarios. The main reason is that nodes have to perform security operations to protect against DoS and MitM attacks. However, a countermeasure against these attack activities significantly minimises the entire latency caused by attackers. The DDoS attacker increases the packet delay by flooding unnecessary traffic into the network. In contrast, the MitM attacker escalates the delay by intercepting or manipulating the communication between publishers and subscribers. For example, in the DoS attack scenario, the FBC, LED, and PRESENT algorithms achieve delay times of 29.8, 29.93, and 29.95 milliseconds, respectively, in Figure 6.11(a). However, in Figure 6.11(b), the delay times vary to 29.97, 29.85, and 29.87 milliseconds. However, the FBC algorithm performs better in both figures than the other four in the DoS scenario. The FBC employs a 128-bit encryption and decryption algorithm, reducing the data transmission delay.

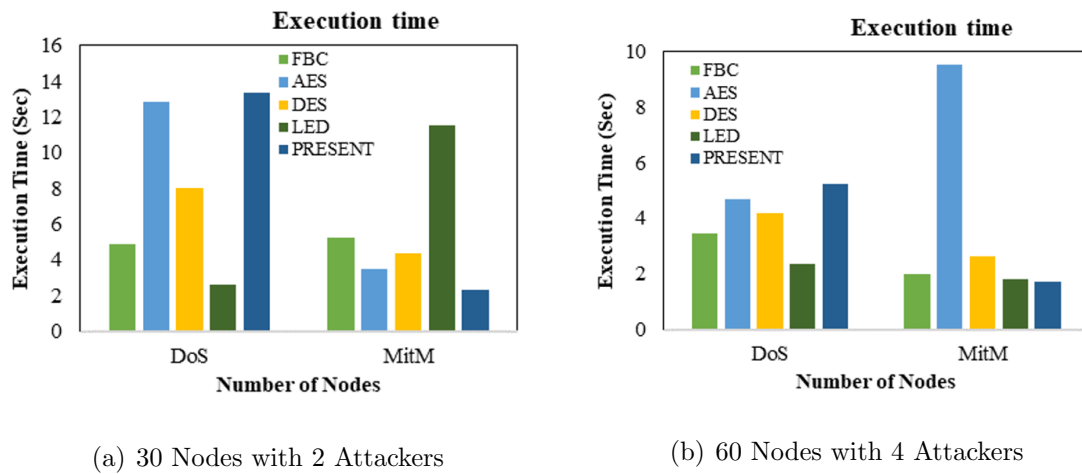


Figure 6.12: Execution Time Results for Different Nodes and Attacker Scenario

Figure 6.12 compares the execution time results for the FBC, AES, DES, LED, and PRESENT algorithms. DoS attackers overwhelm the network resources through unwanted traffic flooding, and the MitM attacker alters the communication among the entities, resulting in frequent packet retransmissions that incur a higher execution time. Therefore, results are obtained for two scenarios where the number of nodes and attackers varies, as shown in Figures 6.12(a) and 6.12(b).

The results show that the PRESENT algorithm exhibits execution times higher than

the other four algorithms, namely FBC, AES, DES, and LED. This is because PRESENT employs a substitution and permutation network structure and considerable key lengths to ensure a high level of security for MQTT, which requires a significant amount of execution time. For example, PRESENT achieves 13.37 seconds of execution time in the DoS attack scenario in Figure 6.12(a). In contrast, it differs in a range of 5.25 for the same DoS scenario in Figure 6.12(b). Moreover, the FBC achieves better execution time results than the other four algorithms in both scenarios due to its simple structure. For example, the execution time results of FBC are 4.86 and 5.26 seconds, respectively, for the scenario of 30 nodes with two attackers.

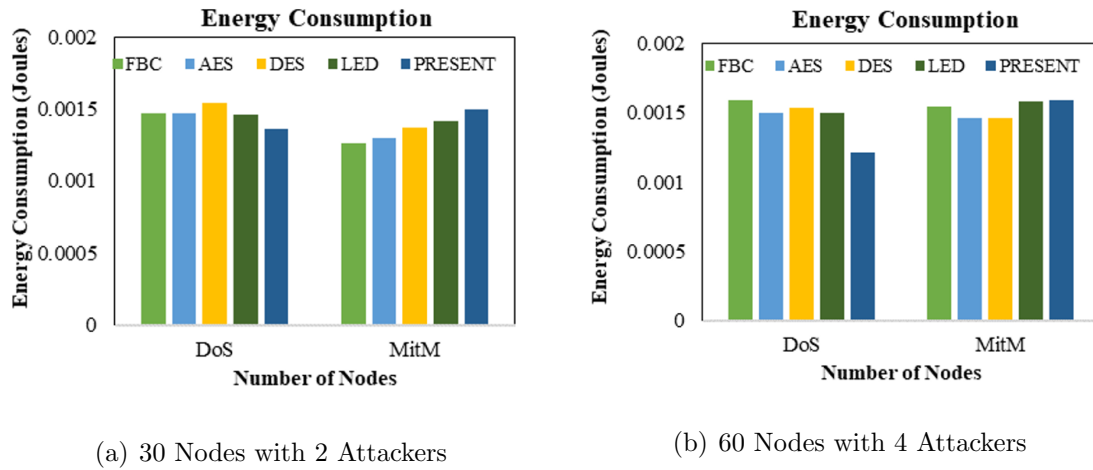


Figure 6.13: Average Energy Consumption Results for Different Nodes and Attacker Scenario

The impact of DoS and MitM attacks on average energy consumption can be significant, particularly in resource-limited IoT-enabled healthcare systems. Both attacks consume a higher average energy in an IoT by forcing MQTT entities to handle excess traffic and engage them in additional processing, like rerouting. In addition, the implementation of security solutions leads to increased levels of energy consumption. Moreover, this increased energy usage due to attack behaviour can shorten network lifetime and decrease overall performance.

Figure 6.13 illustrates the energy consumption results of the FBC, AES, DES, LED, and PRESENT algorithms for different node and attacker scenarios. The energy consumption rate of FBC and AES increases for the MitM attacker scenario compared to the DoS attacker scenario. The main reason is that the security process against a MitM

attack is more complex than against a DoS attack. For example, the FBC and AES algorithms require 0.00159 and 0.0015 joules of energy to detect the DoS attacker, as shown in Figure 6.13(a). However, because of their simple encryption and decryption structure, the FBC and AES algorithms consume less energy than the DES, LED, and PRESENT algorithms.

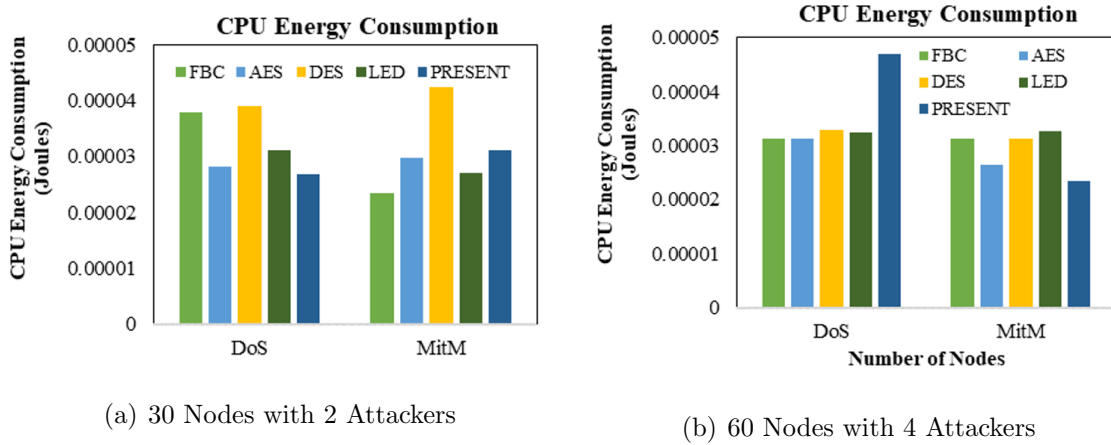


Figure 6.14: CPU Energy Consumption Results for Different Nodes and Attacker Scenario

Figure 6.14 illustrates the results of CPU energy consumption of five lightweight symmetric key cryptography algorithms: FBC, AES, DES, LED and PRESENT. The results are based on scenarios with 30 nodes, two attackers, 60 nodes, and four attackers to demonstrate the superiority of these five algorithms under various conditions. The DoS attacker bombards the network with a high volume of unnecessary or malicious traffic, resulting in high CPU energy utilisation. The CPU burden increases in attack scenarios, as it needs to counteract attacks using security algorithms and reroute traffic. All algorithms require some CPU energy to perform encryption and decryption functions. Due to its fast and straightforward bit-slice structure, the FBC algorithm achieves better CPU energy consumption results than the AES, PRESENT, LED, and DES algorithms. For example, the FBC achieves 0.0000380 and 0.00003126 joules of energy, respectively, for the DoS attack under 30 and 60 nodes present. However, it varies by 0.00002343 and 0.00003126 joules for MitM attackers in the same scenario.

6.2.3 Performance Evaluation of Various Data Rates under DoS or MitM Attacks Scenarios

In this section, we analyse the impact of lightweight encryption algorithms on the performance of the MQTT protocol for different data rates of 64, 128, and 192 bits using a 30-node and 60-node topology with 2 and 4 attacks, respectively. Typically, the DoS attacker floods massive amounts of malicious or irrelevant traffic into the communication. This overwhelming data influx limits the available bandwidth to establish communication, making the data rate for legitimate users shrink because the traffic produced by attackers saturates the network. The results of throughput and packet delivery ratio for the FBC, AES, DES, LED, and PRESENT algorithms are presented in Figure 6.15.

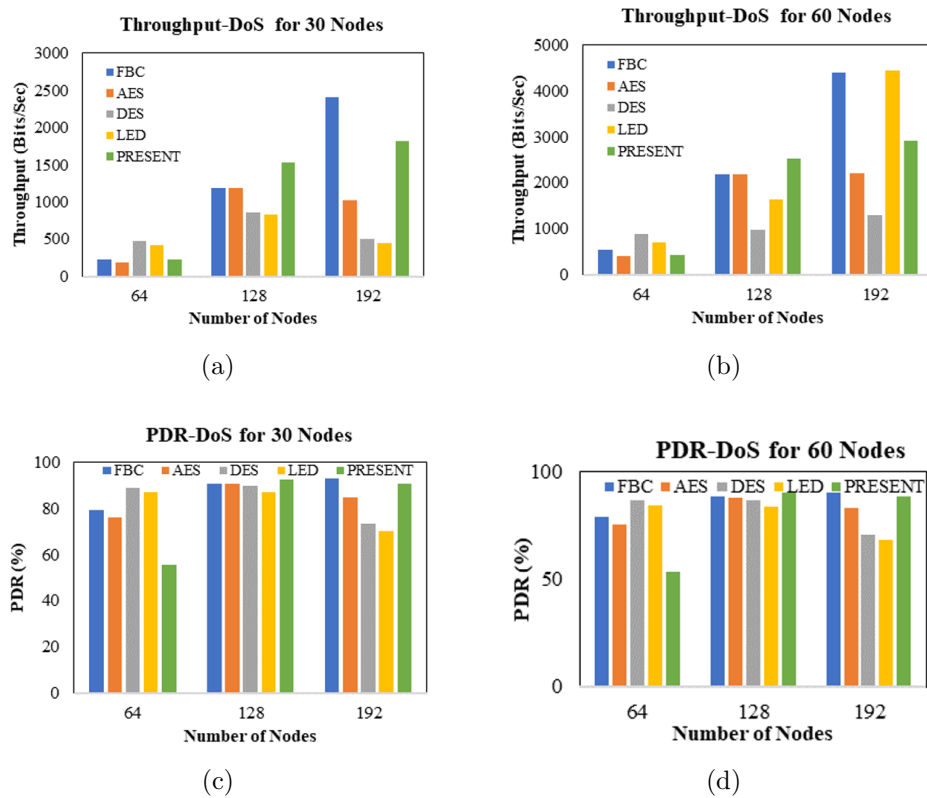


Figure 6.15: Throughput and Packet Delivery Ratio with DoS Attack under Various Data Rate Scenarios

The excessive or irrelevant traffic generated by DoS attackers can significantly cause packet loss in the network, where many legitimate data packets are dropped because the network cannot effectively manage the load. This packet loss minimises the data rate

utilisation efficiency, as fewer packets reach the desired destination successfully. Frequent retransmissions are required for further successful transmissions, increasing bandwidth consumption. Different algorithms accomplish different PDR in the presence of DoS and MitM attacks.

Figure 6.15(b) shows that FBC achieves a high packet delivery ratio and maintains it for different data rates. The reason behind this is the simple cryptography confusion function of FBC, which is suitable for resource-limited IoT nodes. Although AES has a lower PDR than FBC, it performs better than the DES, LED, and PRESENT algorithms. This can be attributed to the relatively simple key and block structure of AES.

On the other hand, Figure 6.15(a) shows that FBC outperforms the different algorithms in terms of throughput. The other algorithms, including AES, have lower throughput values than FBC. PRESENT has the lowest throughput, which can be attributed to its inefficient cryptographic structure. Overall, these results indicate that FBC is a suitable lightweight encryption algorithm for resource-limited IoT nodes, and it can achieve good performance in terms of both packet delivery ratio and throughput for different data rates. For example, the packet delivery ratio of FBC is 79.41% and 93.07% for 64 and 192 bits/second data rate scenarios under 30 nodes, respectively.

Figure 6.16 compares the delay and execution time for lightweight encryption algorithms under various data rates and DoS attack scenarios. All algorithms show an increase in the delay and execution time with an increase in the data rate. For example, FBC achieves 29.9 milliseconds and 29.95 milliseconds of delay for 64 and 192 bits/second scenarios, respectively. Generally, increasing the data rate leads to a reduction in the delay of data transmission.

However, in the presence of a DoS attacker, the delay increases with high data rates as the attacker drops or injects unwanted traffic into the network. Network entities under a DoS attack may experience significant processing delays as they attempt to handle overwhelming requests. These delays slow the transmission of legitimate data, reducing the data rate as the network struggles to keep up with malicious and legitimate traffic. Cryptographic algorithms such as LED and PRESENT take longer than FBC to encrypt and decrypt packets. For example, FBC, AES, DES, LED, and PRESENT attain 29.96,

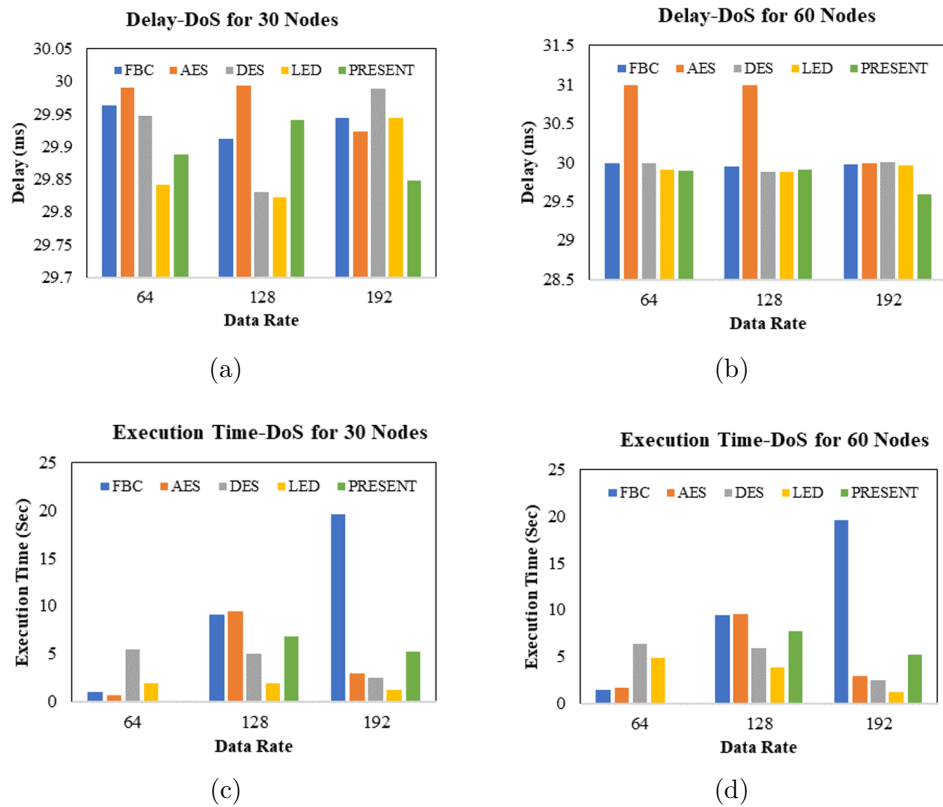


Figure 6.16: Delay and Execution Time with DoS Attack under Various Data Rate Scenarios

29.9, 29.94, 29.84, and 29.88 milliseconds of delay values for 64 bits/second data rate scenarios, respectively.

Figure 6.17 displays the CPU and energy consumption of the results of symmetric key algorithms for various data rates and DoS attack scenarios. The presence of DoS and MitM attackers significantly increases the consumption of CPU energy. Figure 6.17 shows that unwanted flooding and frequent message retransmission require significant CPU energy. Figures 6.17(c) and 6.17(d) depict that the PRESENT achieves better results in CPU energy consumption under many scenarios than the other five, such as the AES, DES, LED and PRESENT algorithms. The reason is that the PRESENT follows a fast and simple confusion matrix structure to perform encryption and decryption on the network. Thus, it incurs less energy than complex cryptography models. For example, PRESENT achieves 0.0000234 joules of energy in 64 and 192 bits/second scenarios to ensure the security level of IoT nodes. However, DES and LED achieve nearly equal performance in CPU energy consumption in all data rate scenarios, as shown in Figure 6.17(c). The

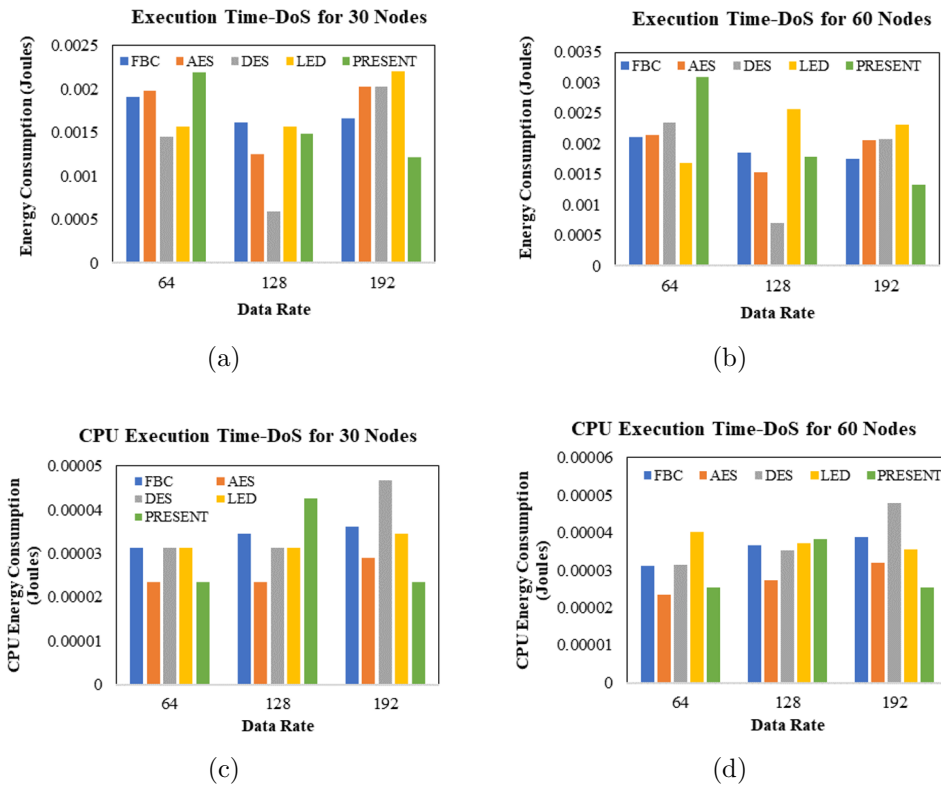


Figure 6.17: Average Energy Consumption and CPU Energy Consumption with DoS Attack under Various Data Rate Scenarios

reason is that both algorithms follow strong cryptography keys to ensure higher security in MQTT-based IoT applications.

Figure 6.18 compares the throughput and PDR of five symmetric key algorithms while adjusting the data rate from 64 bits/second to 192 bits/second and considering the MitM attack scenario. All algorithms exhibit an increase in throughput by elevating the data rate value from low to high. This is because the number of packets is reduced and delivered quickly during high-data-rate scenarios. Additionally, the impact of attackers on throughput and PDR is significant, as many genuine packets are lost, reducing the successful packet delivery rate in the network. Each algorithm provides a substantial level of security for countermeasures against these attacks and improves the performance efficiency of MQTT. However, the FBC model performs more than the other four algorithms, utilising a fast and straightforward confusion matrix structure to ensure protocol security under a high data rate scenario of 192 bits/second. Figures 6.18(c) and 6.18(d) show the packet delivery ratio results of five algorithms. The FBC and LED attain nearly equal

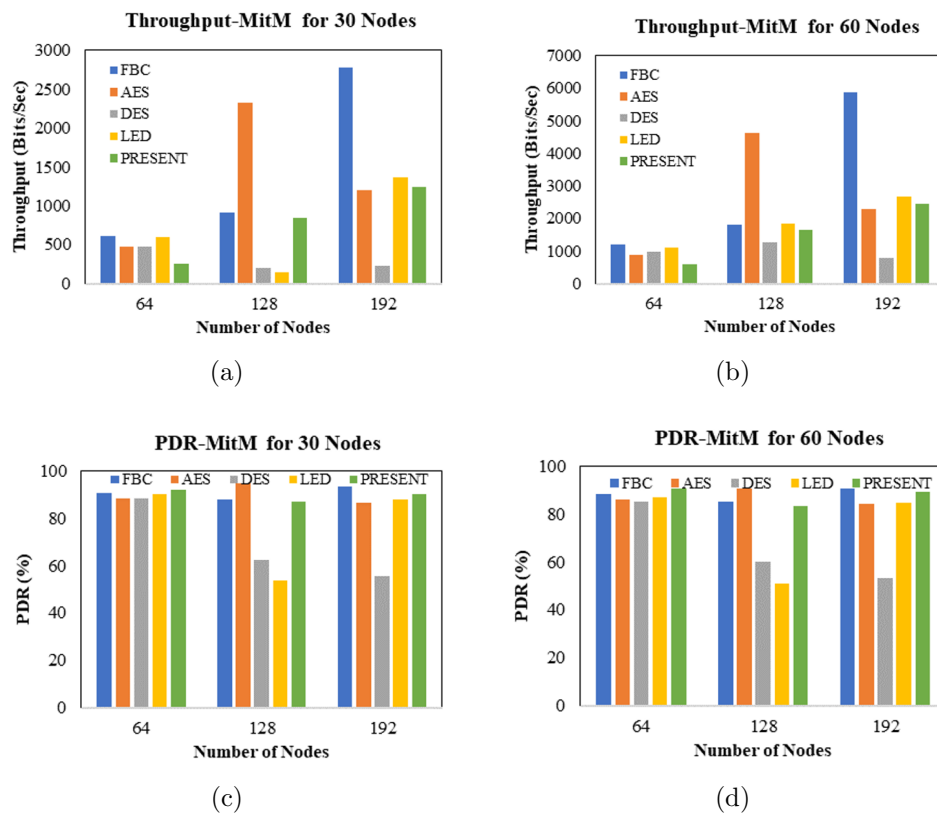


Figure 6.18: Throughput and Packet Delivery Ratio with MitM Attack under Various Data Rate Scenarios

PDR values for the 64 bits/second data rate scenario. The reason is that both models successfully detect MitM attacks using robust and straightforward cryptography methods, respectively. For example, FBC and LED obtained 90.8% and 90.5% of PDR under the 64 bits/second data rate and 30-node scenario.

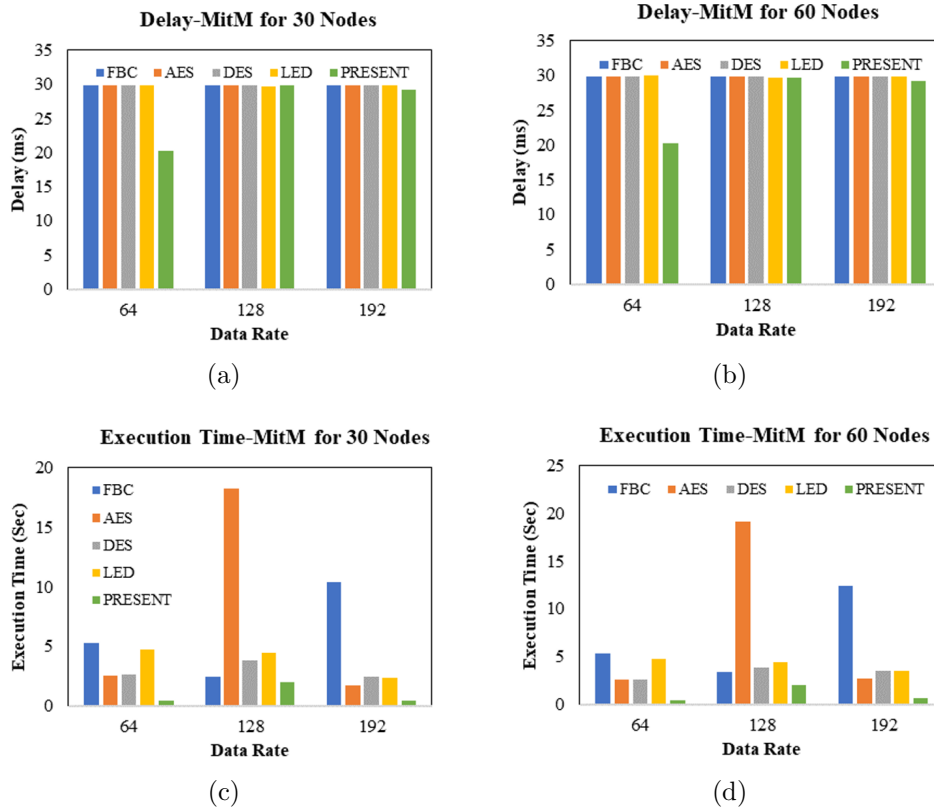


Figure 6.19: Delay and Execution Time with MitM Attack under Various Data Rate Scenarios

Figure 6.19 compares the delay and execution time of five symmetric key algorithms, FBC, AES, DES, LED and PRESENT, in the MitM attack environment and various data rate scenarios. Creating an attack environment, including DoS and MitM, incurs significant delays in data transmission. Although implementing lightweight security algorithms decreases the delay due to attack effects, these algorithms need some time to perform the security operations.

Therefore, all protocols experience an increase in delay and execution time as the data rate increases from low to high. This is because algorithms require more time to encrypt and decrypt packets, and high data-rate packets require extra time compared to low data-rate packets. For example, in 64 bits/second data rate scenarios, FBC, LED, and

PRESENT experience delays of 29.84, 29.89, and 20.23 ms, respectively. However, FBC exhibits a lower delay and execution time than the other four algorithms under all data rate scenarios. The simple confusion matrix structure of FBC provides robust protection against MitM attacks while incurring a minimum delay.

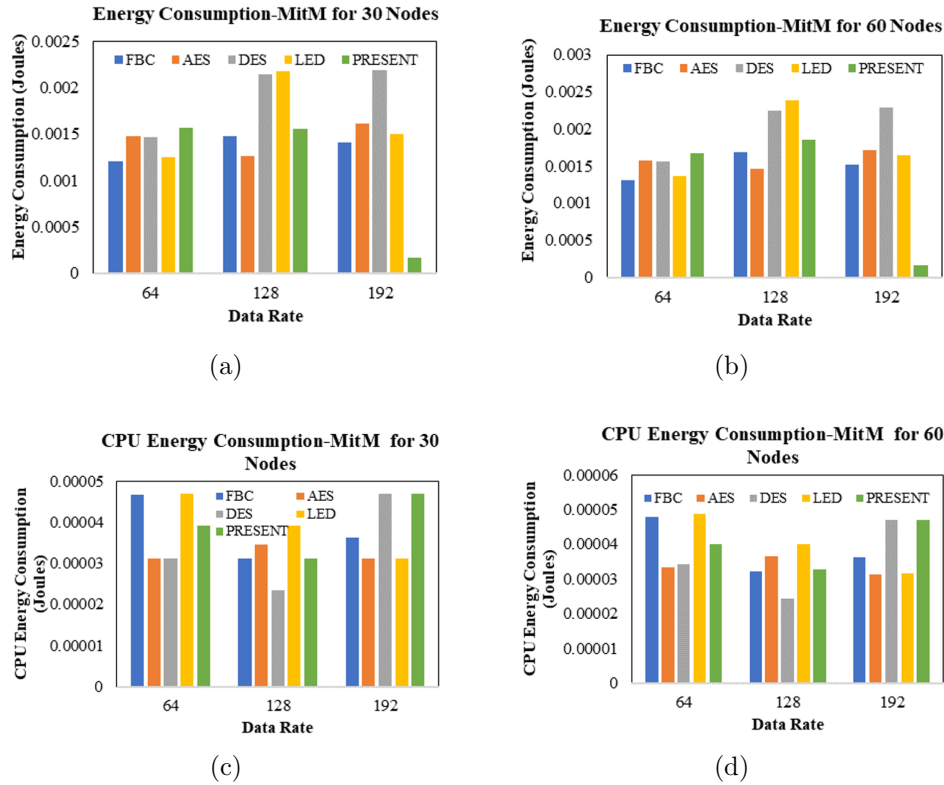


Figure 6.20: Energy Consumption and CPU Energy Consumption with MitM Attack under Various Data Rate Scenarios

Figure 6.20 shows the comparative results of the energy consumption and CPU energy consumption of the FBC, AES, DES, LED and PRESENT algorithms. The results are obtained for MitM and different data rate scenarios, as the attackers target the network resources to shorten the network lifetime. The effects of attackers on energy consumption are also included in the results in Figure 6.20. The energy consumption results of the FBC, LED and PRESENT algorithms are relatively different, as they are all lightweight algorithms. For example, the FBC, LED and PRESENT algorithms consume 0.001414, 0.0015, and 0.00164 joules of energy to transmit the data under a 192 bits/second data rate and a 30-node scenario. However, the FBC performs better than other algorithms due to its simple and fast cryptography structure. Figure 6.20(c) shows that DES and

AES consume the same energy to protect the MQTT from MitM attacks. For example, the DES and AES obtain 0.00003125 respectively for a 64 bits/second data rate and a 30-node scenario.

6.2.4 Performance Evaluation of Different Nodes under DoS and MitM Simultaneous Attack Scenarios

This section creates two scenarios to show performance in a simultaneous attack environment. During the DoS attack, the network buffers overflowed due to the excessive communication load, and more packets were dropped, further diminishing the performance efficiency. MitM attackers delayed or reordered the packets during interception or alteration, and they may receive out-of-sequence, creating problems in the reassembly of packets at the destination, further degrading the MQTT performance. To show these effects in results, the first scenario comprises 30 nodes with a DoS attacker and one MitM attacker. The second scenario includes 60 nodes with two DoS and two MitM attackers. In addition, the data rate is fixed at 128 bits to analyse the superiority of the algorithms in various numbers of node scenarios.

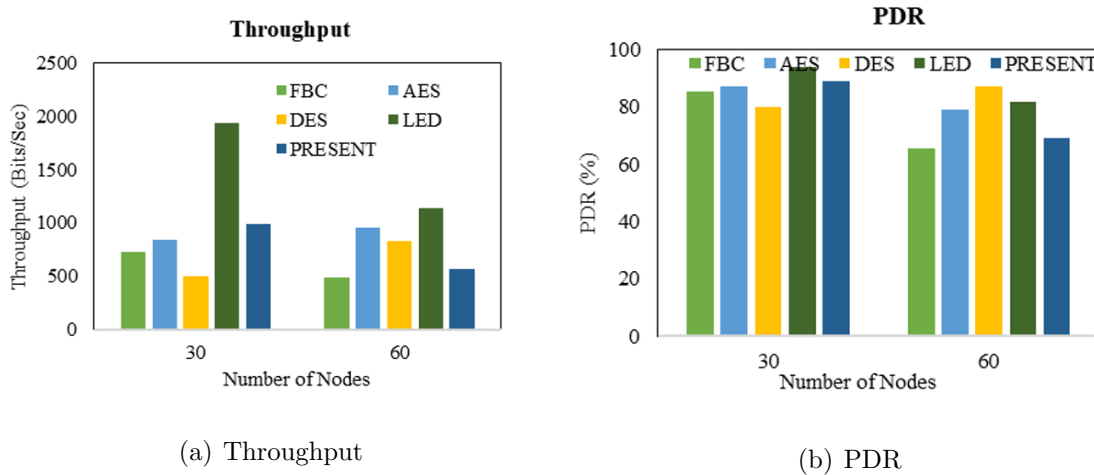


Figure 6.21: Throughput and PDR for Different Number of Nodes Scenario

The throughput and PDR results are obtained by varying the number of nodes and attackers. The reason is that the increase in the density of the attacker and the nodes significantly affects PDR and throughput. The DoS attacker increases packet loss, and

MitM escalates packet retransmissions, reducing successful packet delivery on the network.

Figure 6.21 illustrates the comparison results of throughput and PDR of five symmetric key cryptography algorithms obtained by varying the number of nodes from 30 to 60. The results demonstrate that the LED achieves better results than the other four algorithms in all scenarios. For example, the LED achieves 1941.33 bits/seconds of throughput and 94% of PDR for 30 nodes with an attack scenario. Figure 6.21(b) shows that FBC and AES achieve nearly equal PDRs of 85.5% and 87.1%, respectively, compared to the other three algorithms.

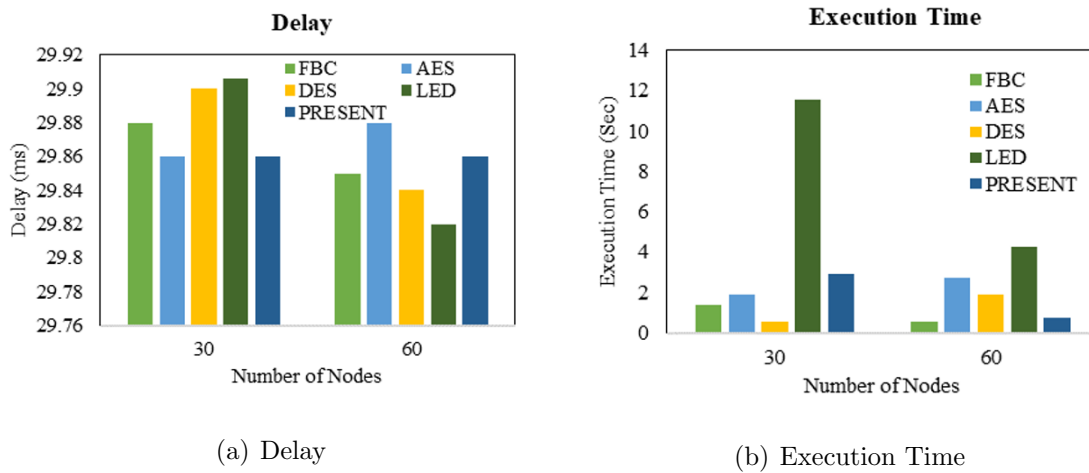


Figure 6.22: Delay and Execution Time for Different Number of Nodes Scenario

MitM and DoS attackers impact delay and execution time by flooding unnecessary traffic into the network and making packet reassembly tedious at the destination, respectively. To depict this impact on network performance, Figure 6.22 shows the delay and execution time results of the FBC, AES, DES, LED, and PRESENT algorithms obtained for scenarios with 30 and 60 nodes. To counteract a DoS and MitM attack, MQTT can include any of these five lightweight symmetric key cryptography protocols, which adds to processing time and escalates the delay. The results show that the LED achieves high delay and execution time for the 30-node scenario. For example, the FBC, AES, DES, LED, and PRESENT algorithms achieve 29.85, 29.88, 29.84, 29.82, and 29.86 seconds of delay for 60-node density scenarios. Similarly, they achieve 0.52, 2.69, 1.87, 4.24, and 0.74 ms of execution time for the same scenario.

The energy consumption and CPU energy consumption results of five symmetric key

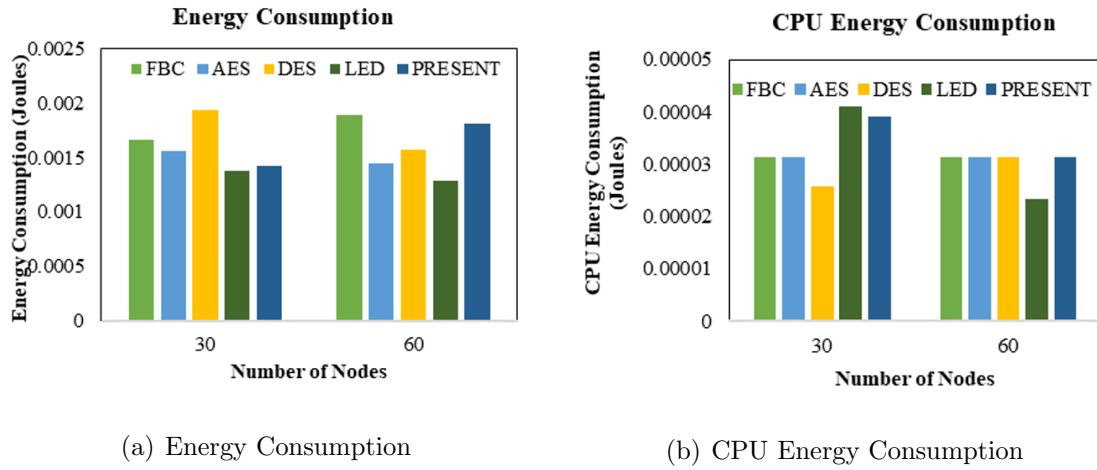


Figure 6.23: Energy Consumption and CPU Energy Consumption for Different Number of Nodes Scenario

cryptography algorithms are shown in Figure 6.23. The results are plotted for two node density scenarios, 30 and 60. MQTT entities may need additional energy to reprocess dropped, intercepted, and altered packets, particularly if they require integrity verification, message decryption, or other lightweight security-related operations. This approach escalates the overall energy consumption in MQTT-enabled healthcare IoT networks. Due to its complex structure, the DES needs a higher energy level than the other four protocols. For example, the DES incurs 0.00194 joules and 0.00157 joules of energy for 30- and 60-node density scenarios. However, the CPU energy consumption of DES is minimal by 0.00002578 compared to the other four protocols under 30-node density scenarios.

Figure 6.24 plots the results of the PDR and the throughput comparison of five lightweight symmetric key cryptography algorithms under the MitM and DoS attack scenarios. Including these attacks in evaluating throughput and PDR is crucial as they can minimise successful packet delivery through attack activities such as packet dropping, alteration, and communication interruptions. The results are obtained by varying the data rates from 64 to 192 bits/second for 30- and 60-node scenarios, respectively. In Figure 6.24(a), the FBC accomplishes better throughput than the other four algorithms: AES, DES, LED, and PRESENT in the 30-node density scenario and the 64-bit/second data rate scenario. The FBC includes a robust and straightforward key structure that provides security against DoS and MitM attacks and improves throughput. For example,

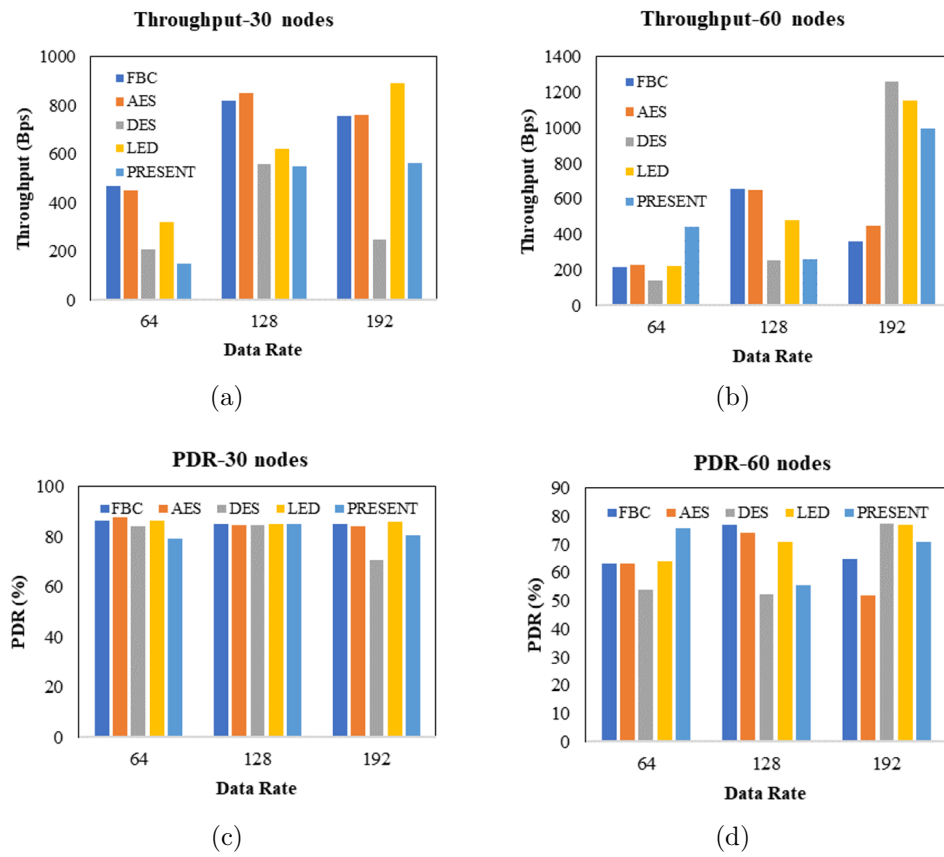


Figure 6.24: Throughput and Packet Delivery Ratio under Various Data Rates

FBC achieves 470 bits / second throughput. In contrast, the other algorithms, AES, DES, LED, and PRESENT, obtain a throughput value of 450.13, 208.24, 321.84, and 150.28 bits / second, respectively, when the data rate is 64 bits / second and the number of nodes is 30. Similarly, in the same scenario, the PDR of FBC, LED, and PRESENT is 86.57%, 86.5%, and 79.118% respectively.

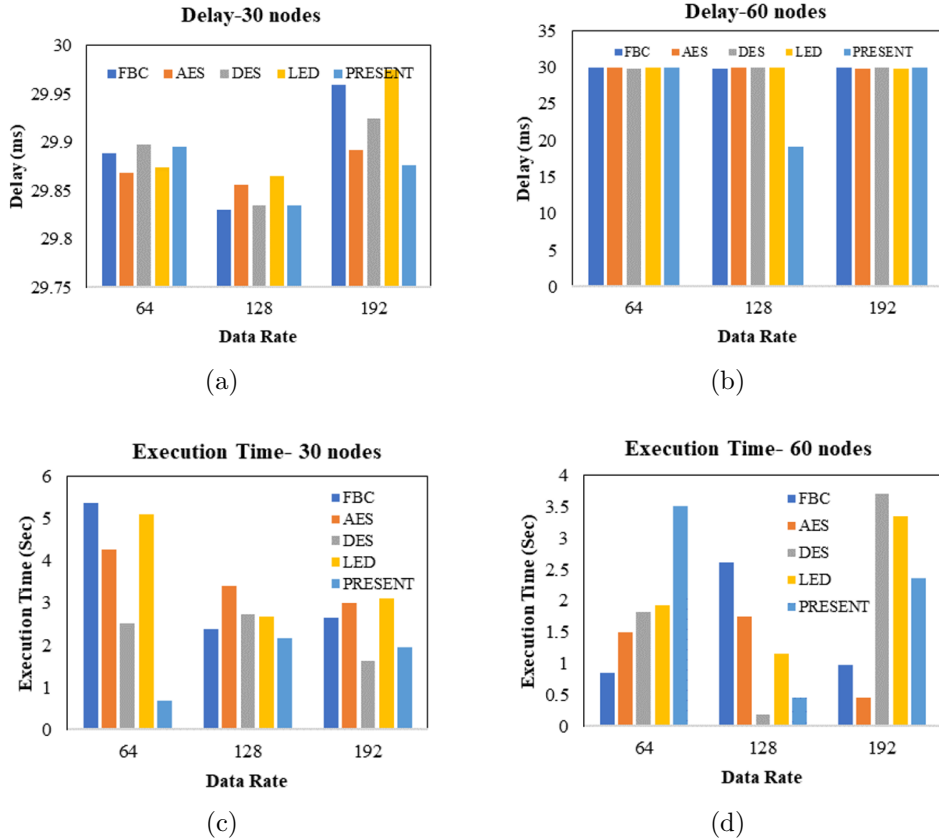


Figure 6.25: Delay and Execution Time under Various Data Rates

Figure 6.25 demonstrates the results of comparing the execution time and delay of five lightweight algorithms. To analyse the efficiency of such algorithms under various scenarios, the data rate is adjusted from 64 bits/second to 192 bits/second for both 30- and 60-node scenarios in the presence of DoS and MitM attacks. Implementing countermeasures against attacks also escalates the entire network delay and execution time. Under the 128-bit/second data rate scenario, the FBC achieves better delay results by 29.34 milliseconds, whereas the DES and PRESENT attain 29.35 milliseconds of delay compared to AES and LED. The delay of PRESENT is better in the 30 nodes and 192 bits/second data rate scenario compared to the other four algorithms. The reason is that

all algorithms use different key structures to protect the network from DoS and MitM attacks.

Finally, Figures 6.25(c) and 6.25(d) show that the FBC needs a high execution time to deliver all packets on the network. For example, the FBC obtains 0.8452 and 5.3805 seconds of execution time for 64 bits/second data rate scenarios under 30 and 60 node densities.

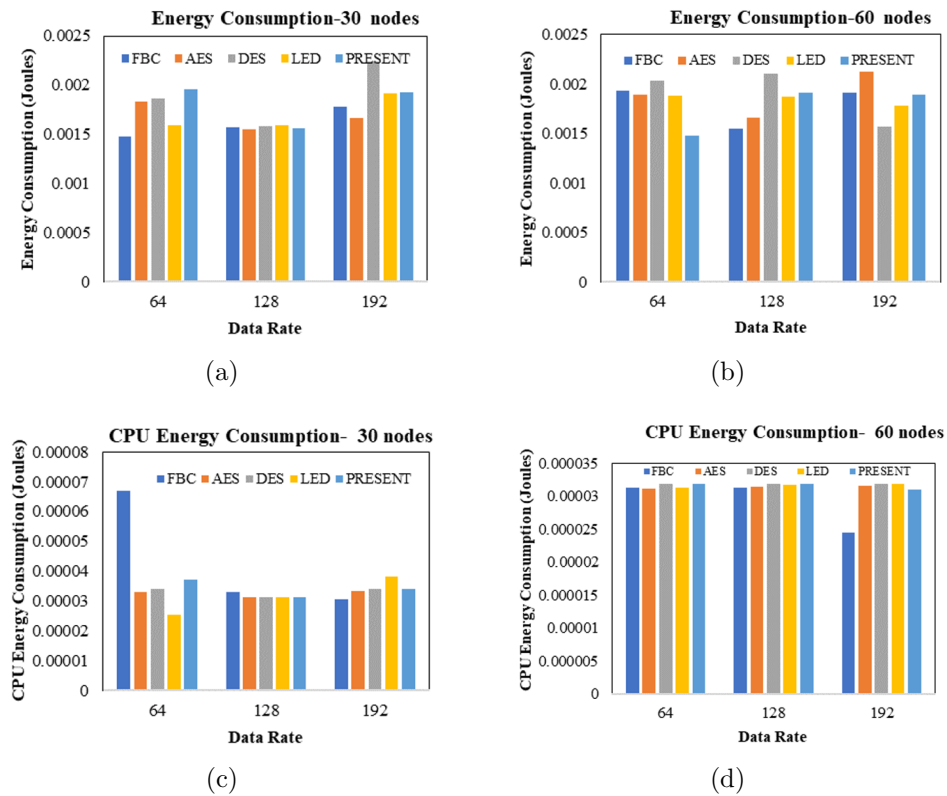


Figure 6.26: Energy Consumption and CPU Energy Consumption Results for Various Data Rates

In Figure 6.26, the energy consumption and CPU energy consumption results of FBC, AES, DES, LED, and PRESENT are compared by varying the data rate from low to high. In addition, the number of nodes varies from 30 to 60 in attack scenarios for better analysis. The MitM and DoS attacker may increase legitimate data lost in the network by dropping and altering, and here entities may necessitate re-transmitting packets frequently, dissipating extra energy each time.

In Figure 6.26(b), it is observed that the energy consumption of DES is high under 64 and 128 bits/second data rate scenarios. In contrast, the AES achieves high energy

consumption under 192 bits/second data rate scenarios. This is caused by AES and DES using complex key structures to ensure network security, resulting in high energy consumption. For example, the DES consumes 0.002035 and 0.0021066 joules of energy for 64 and 128 bits/second data rate scenarios, respectively, when the number of nodes is 60. For example, the FBC achieves 0.0000668 and 0.000031245 joules of energy consumption, respectively, for 30 and 60 node density scenarios when the data rate is 64 bits/second.

6.3 Results Discussion

The discussion of the results presented in this work begins by evaluating the performance of five symmetric key cryptography algorithms, including AES, PRESENT, LED, FBC and DES, in different scenarios, such as nodes, attackers and data rates. The performance of algorithms is compared with the base MQTT protocol in ideal and attack environments. The results of the ideal environment show that the base MQTT protocol outperforms MQTT integrated with AES, PRESENT, LED, FBC, and DES algorithms. The base-MQTT protocol does not include encryption and decryption mechanisms, leading to a lower delay, energy consumption, and higher PDR.

Furthermore, the base MQTT protocol achieves high performance results in various data rate scenarios. However, its performance significantly deteriorates in the presence of DoS and MitM attackers. The reason is that attackers impact network efficiency through PDR, throughput, delay, execution time, and energy consumption through lost legitimate packet transfer and data modifications. Therefore, the study extends the evaluations to an attacker scenario using the AES, PRESENT, LED, FBC, and DES algorithms. Tables 6.2 and 6.3 present the results of the evaluations, comparing the performance of the base MQTT protocol and the five symmetric key cryptography algorithms with MQTT.

According to Tables 6.2 and 6.3, each algorithm performs differently with respect to throughput, PDR, delay, execution time, energy consumption, and CPU energy consumption. The results show that FBC is the superior algorithm in terms of all metrics in various numbers of nodes and attacker scenarios. AES follows with better performance in throughput and PDR. However, it may not be appropriate for an IoT-enabled MQTT environment at the expense of higher energy consumption due to the more robust en-

Table 6.2: Results and Discussion of MQTT.

Metrics	Scenarios	Performance Efficiency					
		Base-MQTT	FBC	AES	DES	LED	PRESENT
Throughput	Various Data Rate without Attacker Scenario	Very high	High	Mode-rate	Mode-rate	Low	Poor
	30 nodes with two attackers		Very high	High	Mode-rate	Mode-rate	Low
	60 nodes with four attackers		High	High	Mode-rate	Low	Low
	Various Data Rates with DoS and MitM Attacker		High	High	High	High	Moderate
PDR	Various Data Rate without Attacker Scenario	Very high	High	Mode-rate	Mode-rate	Mode-rate	Moderate

Table 6.2 – continued from previous page

Metrics	Scenarios	Performance Efficiency					
		B-MQTT	FBC	AES	DES	LED	PRESENT
	30 nodes with two attackers		High	Mode-rate	Mode-rate	Low	Low
	60 nodes with four attackers		High	High	Mode-rate	Mode-rate	Moderate
	Various Data Rates with DoS and MitM Attacker		Very high	High	Mode-rate	Low	Poor
Delay	Various Data Rate without Attacker Scenario	Very high	High	High	Mode-rate	Mode-rate	Poor
	30 nodes with two attackers		High	Poor	Poor	Mode-rate	Low
	60 nodes with four attackers		Very high	High	Mode-rate	Mode-rate	Low
	Various Data Rates with DoS and MitM Attacker		Very high	High	Mode-rate	Low	Poor
Execution Time	Various Data Rate without Attacker Scenario	Mode-rate	Very high	Mode-rate	Poor	Mode-rate	Low
Continued on next page							

Table 6.2 – continued from previous page

Metrics	Scenarios	Performance Efficiency					
		B-MQTT	FBC	AES	DES	LED	PRESENT
	30 nodes with two attackers		Very high	High	Mode-rate	Low	Poor
	60 nodes with four attackers		Very high	High	Mode-rate	Low	Poor
	Various Data Rates with DoS and MitM Attacker		Very high	High	Mode-rate	Low	Poor
Energy Consumption	Various Data Rate without Attacker Scenario	Very high	High	Low	Mode-rate	High	Poor
	30 nodes with two attackers		High	High	Mode-rate	Mode-rate	Low
	60 nodes with four attackers		Very high	High	Mode-rate	Mode-rate	Low
	Various Data Rates with DoS and MitM Attacker		Very high	High	Mode-rate	Mode-rate	Low
CPU Energy Consumption	Various Data Rate without Attacker Scenario	Very high	High	Low	Mode-rate	High	Poor
Continued on next page							

Table 6.2 – continued from previous page

Metrics	Scenarios	Performance Efficiency					
		B-MQTT	FBC	AES	DES	LED	PRESENT
	30 nodes with two attackers		Very high	High	Mode-rate	Low	Poor
	60 nodes with four attackers		Very high	High	Mode-rate	Low	Poor
	Various Data Rates with DoS and MitM Attacker		Very high	High	Mode-rate	Mode-rate	Poor

This work applies the FBC, which is highly suited to the proposed remote patient monitoring system. The performance results of FBC-based MQTT over the Healthcare 4.0 remote patient monitoring system against DoS and MitM attacks are shown in Table 6.3.

Table 6.3: FBC-based MQTT for Remote Patient Monitoring System.

FBC-based MQTT for Remote Patient Monitoring System				
Dos Attack Scenario	Number of Patients	Communication Latency (ms)	Energy Consumption (Joules)	Computational Overhead
	5	8.6	0.0016	0.15
	10	13.2	0.0021	0.22
	15	17.5	0.0027	0.295
MitM Attack Scenario	5	7.4	0.0014	0.125
	10	12.6	0.0020	0.182
	15	18.4	0.0029	0.022

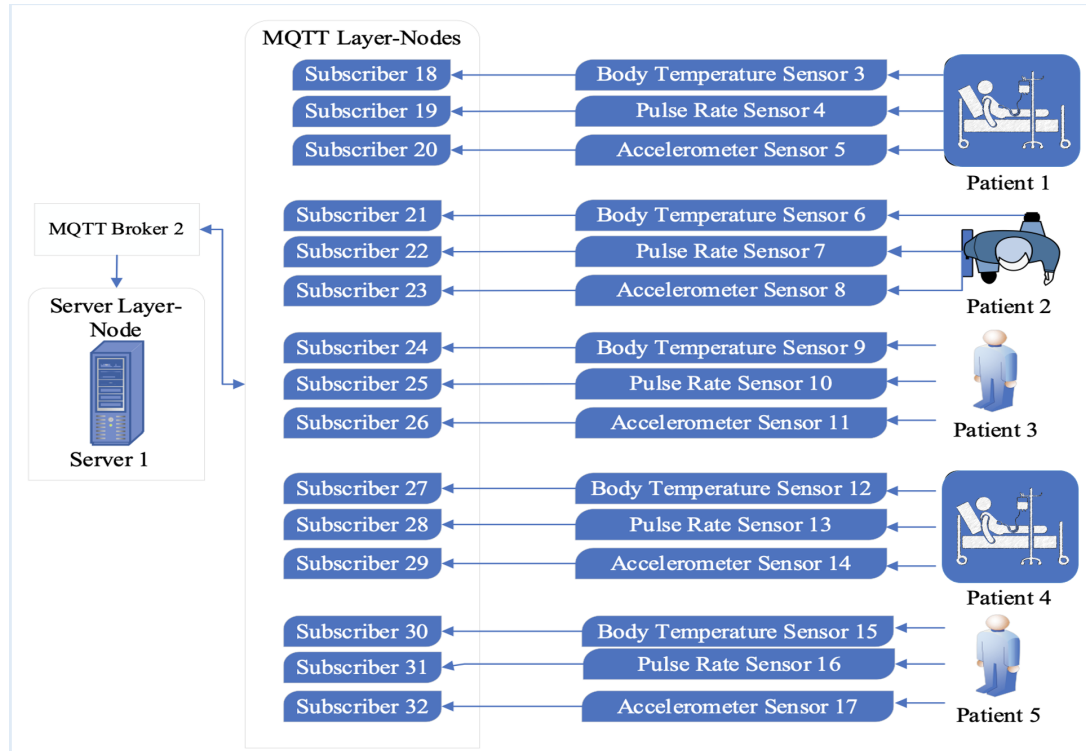


Figure 6.27: MQTT-based Communication for Remote Patient Monitoring System

6.3.1 Balance of Performance and Security

Balancing the performance and security of cryptographic algorithms is critical for applications in resource-constrained medical IoT environments. Based on previous research work and simulation results, we created a new remote patient monitoring system to compare the encryption and decryption times of the FBE, LED, and PRESENT algorithms. AES is secure and efficient on systems with hardware acceleration, but may not be suitable for constrained IoMT devices due to its relatively high resource requirements [42]. DES is less secure against brute-force attacks, whereas the newer 3DES requires higher computational costs and is unsuitable for remote IoMT environments [461].

In the remote patient monitoring system, patients wear different sensors to sense the level of body temperature, pulse rate, and other physiological parameters. Therefore, the network scenario is constructed with 32 nodes in which 15 wearable devices are presented as publishers, 15 MQTT subscribers, an MQTT broker, and a server, as shown in Figure 6.27. The sensor-monitored data are transmitted securely through FBC, LED and PRESENT-based MQTT protocols.

The simulation parameters are shown in Table 6.4.

Table 6.4: Remote Patient Monitoring System Simulation Parameters.

Parameters	Values
Application Layer Protocol	MQTT
Simulation Area	$100m \times 100m$
Total Number of Nodes	31 Server - 1 Broker - 2 MQTT subscribers – 18 and 32 Wearable Sensors as publishers – 3 and 17
IoT Devices	Temperature Sensor, Pulse Rate Sensor, Accelerometer Sensor, and Analog Device
MQTT Broker	Mosquitto-rsmb broker-1.3.0.2
DoS attack Nodes	1 (Sensor 20)
Data Rates (bits/second)	64
Transmission Range	50m
Simulation Time	5 Minutes
Algorithms	PRESENT, LED, FBC

Figure 6.28 shows the simulation window for the remote Healthcare 4.0 scenario. In the screenshot, the green, orange, pink, yellow and blue nodes refer to the Broder Router, the MQTT broker, the MQTT subscriber, the MQTT publishers and the DoS attacker, respectively. MQTT uses the lightweight cryptography algorithms FBC, LED, and PRESENT to transfer data between publishers and servers. Two metrics are used in this work scenario to analyse the efficiency of various cryptography algorithms over MQTT: encryption time and decryption time.

Encryption Time: The amount of time required by lightweight cryptography algorithms such as FBC, LED, and PRESENT to encrypt the data sent by the publishers.

Decryption Time: The amount of time required by lightweight cryptography algorithms such as FBC, LED, and PRESENT to decrypt the data on the server.

Figure 6.29 shows the results of the encryption and decryption time of the lightweight cryptography algorithms FBC, LED, and PRESENT implemented for the MQTT-enabled remote patient monitoring system of Healthcare 4.0. The results are obtained for the attack scenario. Generally, remote patient monitoring systems devices are battery-powered

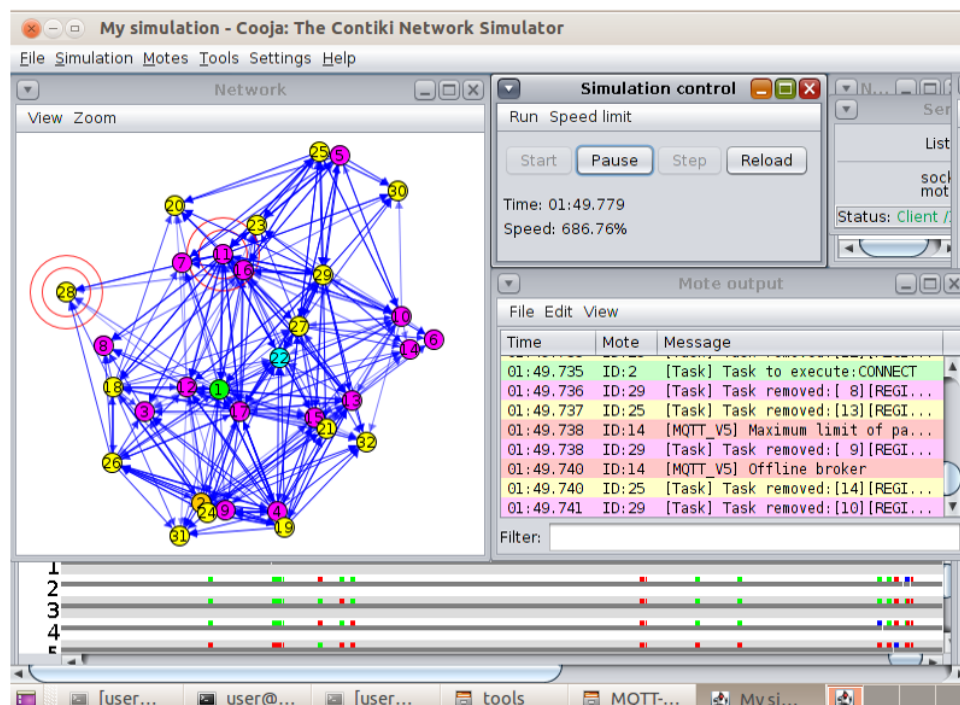


Figure 6.28: Remote Patient Monitoring System Using Lightweight MQTT

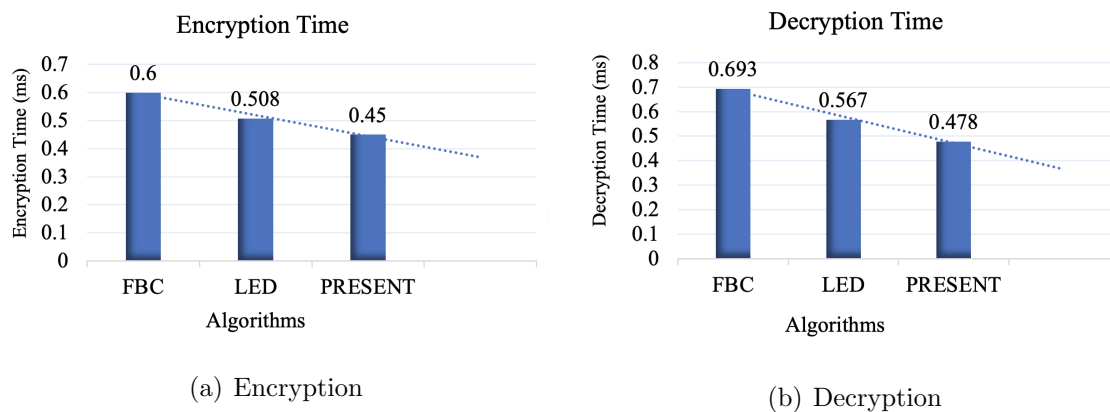


Figure 6.29: Encryption and Decryption Time

and require lightweight computation-based cryptography for data transmission. The encryption and decryption time also reflects the speed of the algorithms. The FBC and PRESENT algorithm exploits 64-bit and 80-bit key lengths to secure data transmission among the MQTT server and clients, whereas the LED utilises 64-bit to ensure security against attacks. The results show that the PRESENT algorithm achieves minimum encryption and decryption time results of 0.45 and 0.478 ms, respectively. The reason is that the PRESENT algorithm uses an adequate key length size, which provides the desired level of security against attacks in resource-constrained remote patient monitoring system scenario settings. Compared to PRESENT, the other two algorithms are FBC and LED, which use 64-bit data encryption and decryption keys. Hence, they lack a better balance between security and resource use of Healthcare 4.0. Therefore, the proposed work rearranges the lightweight cryptography algorithms PRESENT, LED, and FBC to provide security against DoS attack scenarios without impacting the actual MQTT data transmission performance. The PRESENT achieves high encryption and decryption speeds in terms of time compared to others. Although it may not offer the same level of security as AES, it is suitable for low-power devices and resource-constrained environments.

6.3.2 Results Summary

This work focuses mainly on confidentiality in MQTT data transmission. Measuring confidentiality in a simulation environment involves assessing how sensitive information is protected from tampering and unauthorised access or disclosure. In the simulation context, especially in MQTT security and data transmission, confidentiality can be evaluated by exploiting different metrics. The proposed model creates different scenarios using MitM and DoS attackers to effectively analyse the confidentiality performance of the five symmetric key cryptography algorithms. The lightweight symmetric key cryptography algorithms prevent unauthorised parties from understanding message content by encrypting the MQTT data using a shared secret key on the sender. Thus, they ensure security and confidentiality in the MQTT environment. Performance evaluation effectively shows the security trade-offs of lightweight cryptography algorithms. By evaluating performance in

terms of throughput, PDR, delay, execution time, energy consumption, and CPU energy consumption, the proposed model determines the balance between the security of MQTT and resource consumption without sacrificing performance level. In a simulation setting, integrity is also added to evaluate performance. Integrity shows the truthfulness of the messages received on the receiver side. Confidentiality is the primary focus in a simulation setup of MQTT v5, but the term integrity is also considered. The simulation environment can design and execute various scenarios where lightweight solutions preserve data confidentiality and maintain transmitted data integrity.

This work focuses mainly on confidentiality in MQTT data transmission. Measuring confidentiality in a simulation environment involves assessing how sensitive information is protected from tampering and unauthorised access or disclosure. In the simulation context, especially in MQTT security and data transmission, confidentiality can be evaluated by exploiting different metrics. The proposed model creates different scenarios using MitM and DoS attackers to effectively analyse the confidentiality performance of the five symmetric key cryptography algorithms. The lightweight symmetric key cryptography algorithms prevent unauthorised parties from understanding message content by encrypting the MQTT data using a shared secret key on the sender. Thus, they ensure security and confidentiality in the MQTT environment. Performance evaluation effectively shows the security trade-offs of lightweight cryptography algorithms. By evaluating performance in terms of throughput, PDR, delay, execution time, energy consumption, and CPU energy consumption, the proposed model determines the balance between the security of MQTT and resource consumption without sacrificing performance level. In a simulation setting, integrity is also added to evaluate performance. Integrity shows the truthfulness of the messages received on the receiver side. Confidentiality is the primary focus in a simulation setup of MQTT v5, but the term integrity is also considered. The simulation environment can design and execute various scenarios where lightweight solutions preserve data confidentiality and maintain transmitted data integrity.

6.4 Chapter Summary

This work evaluates the performance and security of five lightweight symmetric encryption algorithms. Simulations were conducted in ideal and non-ideal environments under distinct node density and data rates to comprehensively assess their performance and security. The results of Cooja simulations indicate that FBC is suitable for resource-constrained healthcare IoT devices in ideal environments due to its simple yet strong cryptographic operations. PRESENT and AES, while more complex, still offer high security and acceptable performance in non-ideal circumstances. The study concludes that PRESENT is highly suitable for secure MQTT communication, particularly in Healthcare 4.0 scenarios.

Lightweight Security Scheme for Topic Encryption and Attribute-based Authentication

The preceding chapter analyses the security of MQTT in detail using five different lightweight symmetric key cryptography algorithms. It investigated the performance of lightweight cryptography solutions over MQTT to focus only on the confidentiality of data transmission. This chapter proposes a novel security strategy using Improved Ciphertext-Policy Attribute-Based Encryption (ICP-ABE) for a Healthcare 4.0 scenario. Provides efficient authentication, enforces fine-grained access control, and improves the confidentiality of the Pub/Sub-based communication model for MQTT data transmissions. The proposed approach simplifies the complex attribute-based key management of existing approaches in a dynamically and frequently changing MQTT-based Pub/Sub environment. The computational overhead of CP-ABE is reduced by applying a proven and lightweight PRESENT algorithm and self-key revocation schemes without compromising security.

Therefore, this work focuses on authentication and access control while indirectly satisfying confidentiality. The proposed model includes topic-based encryption and attribute-based access control to accomplish its intention. Within the context of MQTT, the term authentication not only computes the legitimacy of subscribers and publishers but also indirectly supports the confidentiality of data transmission by ensuring that only authenticated MQTT publishers or subscribers can access encrypted data and join the sessions.

The proposed ICP-ABE strategy employs key components, such as CP-ABE-based authentication using a lightweight PRESENT algorithm, a self-key revocation mechanism, and the Pub/Sub Secure Data approach, to achieve the objectives. The security, computational cost, and complexity analysis are performed to demonstrate the effectiveness of the proposed strategy. The security validation reveals the security strength of the

strategy. The results of the Cooja simulation illustrate the efficiency of the ICP-ABE strategy compared to existing approaches by evaluating throughput and PDR, execution time and delay, average energy consumption, computation and communication overhead, and strength evaluation criteria.

7.1 Introduction for Improved CP-ABE

Operating an extensive IoT network poses numerous hurdles, including device authorisation, safeguarding measures, and platform service infrastructure that could burden network bandwidth and communication protocols. MQTT is a message and information exchange protocol that provides bidirectional and ordered connections and can meet the needs of resource-constrained IoT applications [462]. In MQTT, a Pub/Sub communication model is used. With a single server, thousands of clients can be connected, each serving as a publisher and a subscriber. With the notable increase in the count of IoT devices, MQTT has faced particular challenges, including security vulnerabilities. The critical security issues MQTT needs to address are authentication and access control against different attacks. The broker-based Pub/Sub model makes MQTT communications more vulnerable, as malicious nodes can spoof the identities of legitimate publishers and communication participants by sending unauthorised fake data. DoS and MitM attacks can cause serious problems in the operation of IoT protocols, including MQTT. The goal of a DoS attack is to waste network resources, for example, by flooding the network with unnecessary control messages.

To prevent DoS attacks, MQTT exploits the KeepAlive parameter. The latter defines the time that MQTT connections remain alive without message exchanges. However, SlowDoS [463] has been specifically developed to attack MQTT by manipulating the KeepAlive parameter. This attack is possible because MQTT allows the client nodes to configure the server's behaviour using the KeepAlive parameter. The following issues have been identified:

- KeepAlive is defined on the client side.
- KeepAlive is transmitted through the CONNECT message, so an attacker can use

it to occupy the connection as long as possible and perform a MitM attack, which may result in a DoS attack.

The security issues associated with MQTT [308] have hindered the popularity and implementation of IoT protocols in various innovative applications. In particular, the current MQTT implementation has security flaws related to data privacy preservation, authentication, and data access regulations. Therefore, IoT applications usually implement MQTT on top of TCP because it can be secured with TLS. Running MQTT over TLS addresses the problem of insecure client connections, and it is easily scalable from a single device to many. Although the combination of MQTT and TLS improves communication security, it may not be suitable for devices with limited resources due to its high computational, storage, and communication costs.

Authentication has been integrated with key update mechanisms in multiple research studies to enhance MQTT security [464, 465]. However, frequent key updates can degrade the performance of MQTT in terms of overhead and energy. Therefore, it is essential to design lightweight and efficient security mechanisms for MQTT to address security concerns without compromising performance. In general, the growth of IoT devices has introduced significant challenges to communication protocols, communication security, and platform service architecture. As the preferred protocol for IoT communication, MQTT must address the security concerns arising from its broker-based Pub/Sub model. The proposed security mechanisms for MQTT must be lightweight and efficient, ensuring secure and reliable communication without compromising performance.

Lightweight authentication uses the self-key update and authentication scheme [466]. Attribute-based encryption (ABE) [467] is commonly used in IoT solutions. However, traditional access control solutions require the encryption of IoT data using the Cyphertext Policy-Attribute-Based Encryption (CP-ABE) scheme [468], and only users who comply with the access control policy can decrypt the encrypted information received [395]. However, ABE schemes usually involve high computational complexity and are challenging to implement in resource-limited wireless sensors with limited power and computational capacity. Therefore, a lightweight MQTT authentication and privacy-preserving scheme

must be proposed for Pub/Sub communication models. Most secure MQTT applications specify symmetric/asymmetric encryption and a hashing algorithm to ensure data confidentiality and integrity. However, devices with limited resources need more processing power to perform complex authentication tasks with asymmetric encryption algorithms. CP-ABE offers fine-grained topic-related tree creation and easy data access [469]. It is one of the most commonly used security schemes in MQTT. CP-ABE encrypts the topic and its content based on its attributes. Decrypting the message is possible if the ciphertext contains the attributes [470]. Most conventional CP-ABE systems suffer from two main issues: inefficiency with many attributes and a lack of attribute revocation mechanisms.

This work proposes an ICP-ABE scheme, a lightweight encryption and secure access control scheme for MQTT in resource-constrained healthcare IoT networks. The scheme ensures privacy-preserving access to IoT data on the MQTT server. The KeepAlive setting and the self-key revocation scheme provide effective and secure communication in the IoT.

7.1.1 Motivation to Propose ICP-ABE

Intelligent IoT applications have recently been characterised by limited processing power, limited energy, and less memory. Conventional authentication strategies are designed mainly for powerful devices with high energy, memory, and processing capabilities. Directly applying such authentication schemes could be impractical or unrealistic for resource-limited IoT devices. Therefore, a novel lightweight authentication strategy with minimum burden is tailored to address the constraints of intelligent IoT devices. In MQTT-enabled resource-constrained environments, lightweight authentication is essential to enhance security without sacrificing performance efficiency. Generally, lightweight authentication strategies exploit the self-key update mechanism and authentication method. ABE is the most common strategy employed in the MQTT-enabled IoT environment. In particular, this work incorporates a CP-ABE strategy to encrypt IoT information. According to Chapter 3, since ABE schemes usually incur a higher level of computational complexity, they cannot fit precisely into MQTT-enabled resource-restricted IoT environments. Hence, a lightweight MQTT authentication strategy is paramount to secure the publisher

and subscriber communication model. Therefore, this work proposes an ICP-ABE scheme with a lightweight authentication design to make it highly adaptable to MQTT-enabled IoT environments with limited resources. The proposed model mainly addresses authentication in MQTT data transmission between publishers and subscribers.

7.2 The Proposed ICP-ABE

Unlike the previous work, the proposed ICP-ABE focuses on the security parameters of authentication and access control while providing adequate confidentiality. Adding novel confidentiality and integrity solutions increases latency and computational complexity, which could increase overhead for a resource-constrained Healthcare 4.0 environment. The proposed model neglects this issue by indirectly offering adequate confidentiality in MQTT data transmission through topic-based encryption and decryption. The primary mechanisms of the proposed ICP-ABE are topic-based encryption and decryption with PRESENT-based security keys and attribute-based access control policy. In this work, the new scheme separates the roles of attribute auditing and key extraction. By using a blind key, MQTT servers verify the identity of sender nodes without knowing the sender's attributes. The PRESENT algorithm enables secure sharing of blind keys among clients. The efficiency of ICP-ABE is evaluated using provable security and formal methods. In general, the proposed lightweight security framework for MQTT addresses its vulnerabilities. Ensure secure communication in a resource-constrained environment, making it a promising solution for IoT applications in emerging markets.

7.2.1 Overview of ICP-ABE

The ICP-ABE aims to propose new authentication and access control strategies for the MQTT protocol with the assistance of CP-ABE and PRESENT algorithms. The primary components of ICP-ABE are topic-based encryption, attribute-based access control, blind key-based PRESENT, self-key revocation strategy, and secure data publishing/subscribing approach. According to the ICP-ABE design, the MQTT broker should be authenticated to both MQTT clients who are publishers and subscribers during the

topic-based data publishing/subscribing process. The primary operations of the proposed ICP-ABE are shown in Figure 7.1.

The client can be a publisher or subscriber in an enabled IoT environment. The broker is significant in establishing a publish/subscribe communication model using MQTT. It is responsible for maintaining the topics published by the MQTT clients. The MQTT broker registers the MQTT clients before the establishment of communication. Figure 7.1 demonstrates the MQTT ICP-ABE communication scheme. And Figure 7.2 shows the communication process of ICP-ABE. The MQTT clients, which are publishers, send the monitored healthcare data to the subscribers through topic-based encryption and attribute-based access control mechanisms. Initially, an MQTT client or publisher wants to send data to the subscriber. Therefore, it initialises a session with the broker. The MQTT broker verifies the client and provides user access control. After that, the MQTT client establishes a secure communication in which data encryption and decryption are performed using a lightweight symmetric key PRESENT algorithm with a self-key revocation strategy. This approach enables additional security by key revocation according to the dynamic attribute sets. Finally, lightweight attribute-based access control ensures a secure session login service, and topic-based encryption/decryption ensures safe data transmission to the MQTT publish/subscribe communication model with minimum computational overhead and latency.

This section describes the proposed solution that relies on an improved CP-ABE scheme for lightweight MQTT authentication. An ABE scheme performs both encryption and decryption on the client's attributes. A secret key to a set of attributes is generated that encrypts the data using an access control structure. Data are only decrypted by the receiving client when attributes related to that client satisfy the access control policy. The scheme performs authentication using CP-ABE in combination with the PRESENT algorithm [267], self-key agreement using previously accessed topics, and secure data publishing/subscribing. A block diagram for the proposed scheme is shown in Figure 7.3. Communication over MQTT is encrypted, and the broker authenticates all types of MQTT clients, including publisher and subscriber clients, during data publication and subscription. In particular, the primary intention of the ICP-ABE is to ensure

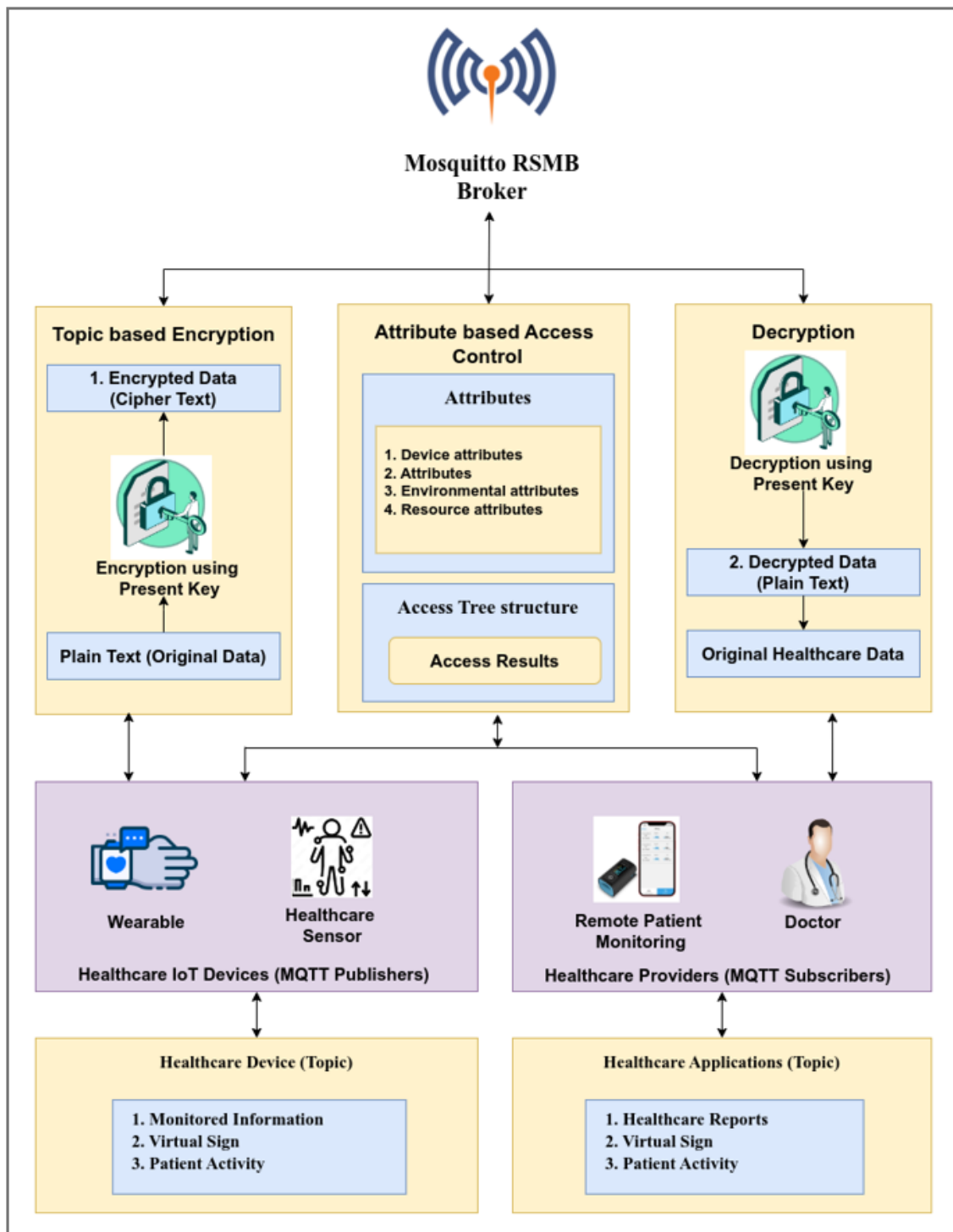


Figure 7.1: Architecture for MQTT Secure Data Transmission

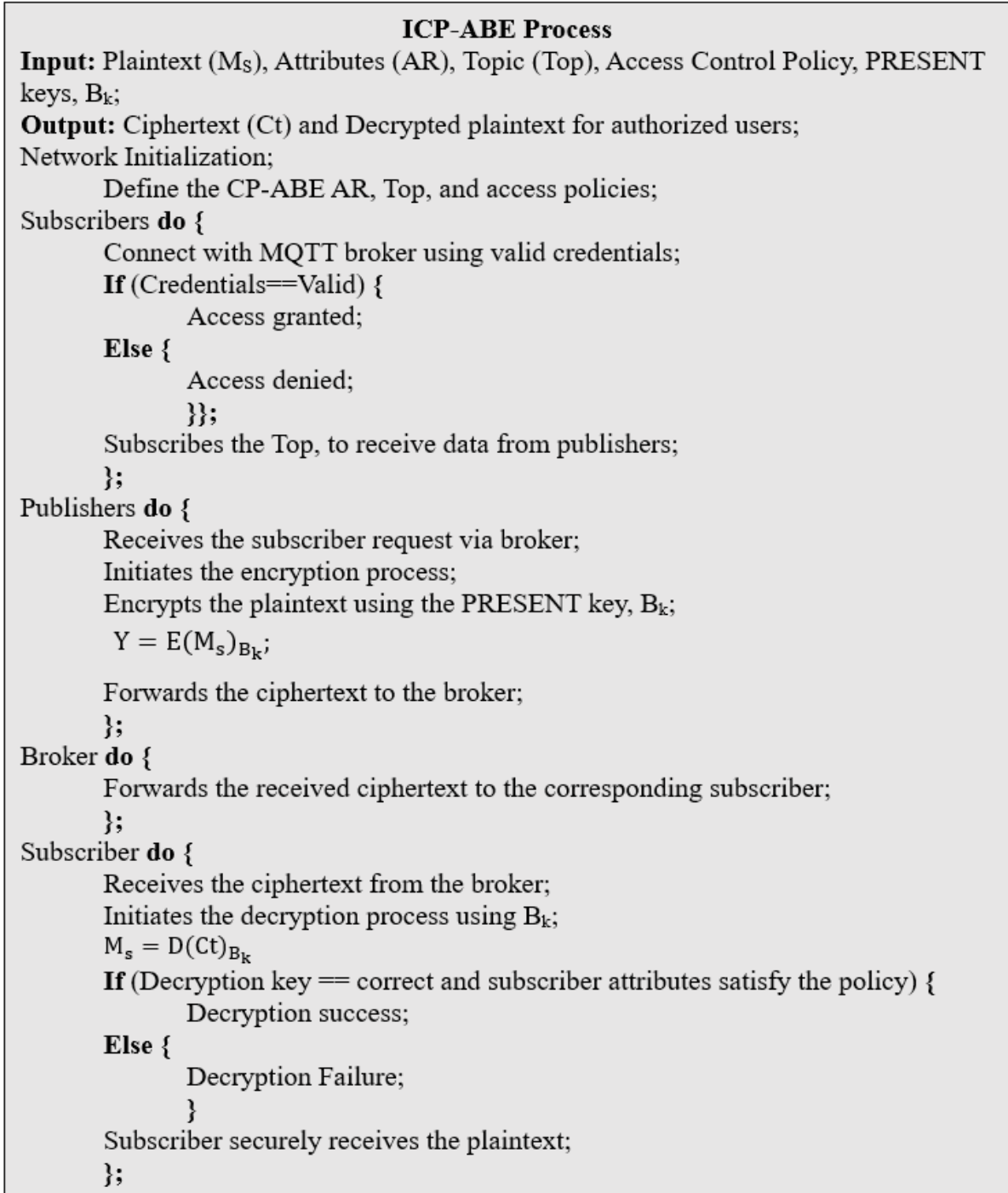


Figure 7.2: Communication Process of ICP-ABE

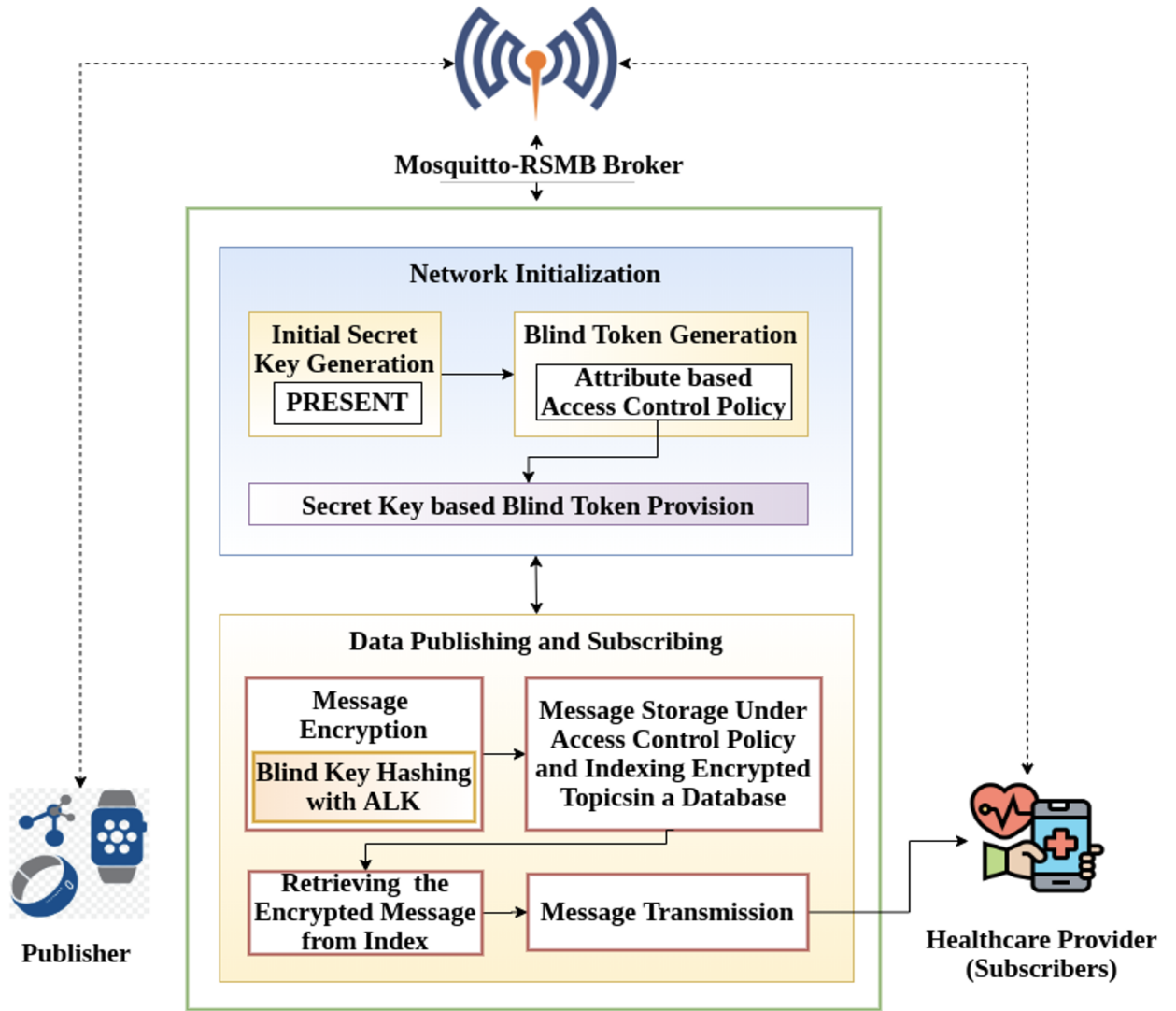


Figure 7.3: Block Diagram for Proposed Lightweight Authentication Scheme in MQTT

the transmission to MQTT subscribers. Publishers receive a blind token from subscriber devices using their unique identifier, such as a Universal Resource Identifier (URI) and its attributes to obtain the blind token from the publisher.

Moreover, the devices are installed with a secret key generated via the PRESENT algorithm. Using such a key, the blind token can be securely shared with subscribers. A blind key is generated on the subscriber devices using the blind token. Publishers secure the data and topics using this key. In particular, publishers use hashing for encrypted data using the attribute length h . This process is known as Attribute Length Key (ALK). The subscriber also performs the ALK process of hashing encrypted data while subscribing to specific topics on the MQTT server. The blind key is updated using the previously

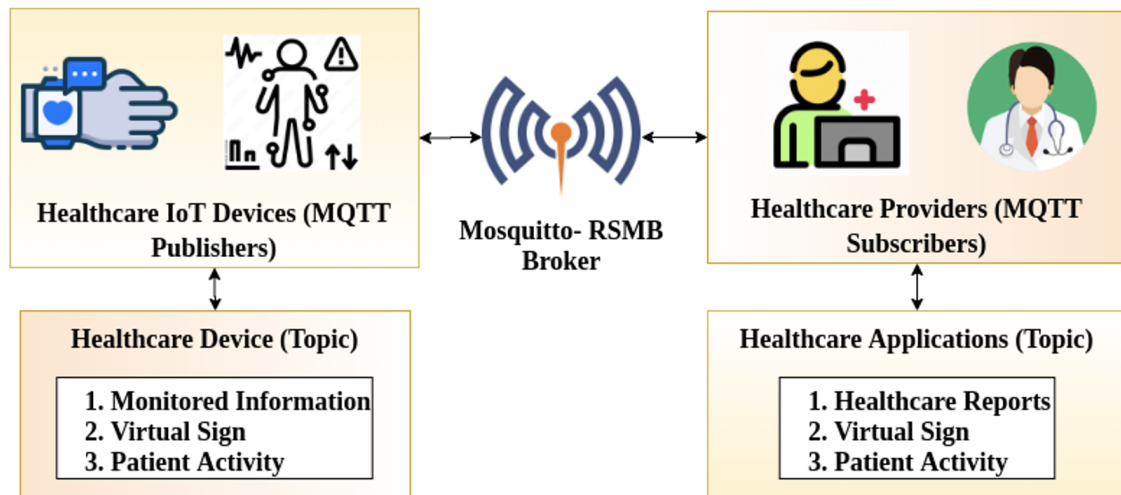


Figure 7.4: System Architecture of ICP-ABE

accessed topic, the previously used blind key, and ALK. Security is ensured by preventing key sharing between devices. MQTT clients can only store the previous topic, the blind key, and ALK due to limited battery resources.

7.2.2 System Model

Let the Pub/Sub network model, Q , be attribute-based and consist of h number of clients, including a set of publishers and subscribers. Note that a client may be both a publisher and a subscriber. The critical elements of the network model Q are topic Top and attribute AR . In the proposed ICP-ABE, a Healthcare 4.0 scenario is considered a running model to explain the working steps of ICP-ABE. The Healthcare 4.0 scenario comprises different types of wearable sensor devices to construct the IoT data layer, and they transmit the monitored information to the remote location or server location with high authentication and access control using a secure MQTT protocol. Since authorised users can only access the information with high security, subscribers request their topics of interest Top , and receive messages through the broker. Publishers send a response to the broker and include the requested topic. Subscribers receive messages if subscription requests are met. The architecture of the ICP-ABE system is shown in Figure 7.4.

7.3 Attributes-based Access Control and Topic-based Encryption Using PRESENT

This section explains the proposed ICP-ABE scheme using mechanisms such as attribute-based access control and the self-key revocation scheme. Table 7.1 describes the list of symbols used in the work.

7.3.1 Attributes-based Access Control

Attribute-based access control is a security model that grants access to data and resources based on attributes associated with a user. In Healthcare 4.0, sensitive patient information and critical healthcare systems are integrated, as attribute-based authentication can maximise security and ensure that only authorised personnel access specific information. Consequently, the steps involved in attribute-based authentication of ICP-ABE in a hospital 4.0 setting are defining the attributes, defining the XOR Policies for key generation, enforcing the access policies, and dynamic updates. The Pub/Sub network model Q considers the message's topic or content. Subscribers indicate their interest in receiving particular events regarded as Top or values of AR in the event. One of the applications for Q with the hospital 4.0 scenario, also known as subscription healthcare monitoring, offers various application services, such as remote healthcare data monitoring and emergency assistance. Doctor U subscribes to obtain patient data from a specific location and requests to receive it since authentication uses different attributes. To protect against malicious activities, it is necessary to encrypt the content before transmitting it to subscribers. Therefore, an encryption scheme is used in the access policy. For example, patient data at a particular location is encrypted by attributes of units such as cardiology and radiology and "location" before sending to the subscriber U . The following section explains relevant concepts and provides valuable definitions.

Definition 1 (Setup and Extract (ID ; AR)): The setup algorithm installs clients with identity ID , and the names of attributes AR . Three types of attributes are included in the proposed ICP-ABE.

Table 7.1: Lists the Symbols.

Symbol	Description
Q	Pub/Sub network model
Top	Topic
AR	Attribute
U	Doctor
ID	Identity
U_{AR}	User attributes
R	Role
Dep	Department
R_{AR}	Resource attributes
L	Location
Ty	Type
E_{AR}	Environmental attributes
t	Time
Nl	Network location
S_k	Secret key
B_t	Blind token
B_k	Blind key
Ms	Message
AS	Adversary
a_0	Number of zeros in the n-bit sequence of secret key generation of PRESENT algorithm
a_1	Number of ones in the n-bit sequence of secret key generation of PRESENT algorithm
N_{AR}	Number of Attributes
tn	Number of attributes
k	The average size of an attribute
l	Number of rows in the access structure
$R_q R_s$	The sum of the round-trip time taken by topic requests R_q and the response from the server R_s
$Time_{Comp}$	Time complexity
R_q	Request
R_s	Response
n	Number of input bits
m	Number of output bits
2^n	Total number of possible keys
A	Attack event
$(A_0 A_1)$	Conditional probability of A_0 given A_1

$$AR = \{U_{AR}, R_{AR}, E_{AR}\} \quad (7.1)$$

Where,

- U_{AR} - User attributes (Role (R) and Department (Dep))
- R_{AR} - Resource attributes (Location (L) and Type (Ty))
- E_{AR} - Environmental Attributes (Time (t), and Network Location (Nl))

$$U_{AR} = (R \{r_1, r_2, \dots, r_n\}, Dep \{dep_1, dep_2, \dots, dep_m\}) \quad (7.2)$$

$$R_{AR} = (L \{l_1, l_2, \dots, l_o\}, Ty \{ty_1, ty_2, \dots, ty_p\}) \quad (7.3)$$

$$E_{AR} = (T \{t_1, t_2, \dots, t_o\}, Nl \{nl_1, nl_2, \dots, nl_p\}) \quad (7.4)$$

Attributes related to users U_{AR} : Different people in a hospital have various roles, such as doctors, nurses, and administrative staff, in different departments, such as cardiology, emergency and radiology. Hence, the level of authentication for each user is different. In the proposed ICP-ABE, access can be based on the unit, such as the various departments a person belongs to, such as cardiology, radiology, and administrative. Thus, it ensures that authorised individual parties only access information according to their work. The ICP-ABE exploits the roles that can serve as attributes for attribute-based authentication, finding the level of access each person has. For example, a nurse can read the information, whereas a doctor has read and write access permissions to patient data.

Attributes Related to Devices or Resources R_{AR} : In Healthcare 4.0, the attributes related to devices or resources are the type of medical device and the location. For example, a heart rate monitor has attributes like “type=monitor” and “location = ICU or patient room”.

Attributes Related to Environment E_{AR} : The ICP-ABE includes environmental attributes such as access time, location of the network, or other environmental conditions. For example, the patient’s healthcare record can be generated during the day or at night, but it is generated during the working hours of hospitals. Access should be restricted

during non-work hours or from particular IP addresses in non-emergency cases.

In addition, ICP-ABE generates the secret key S_k using the PRESENT encryption scheme and the blind token B_t , according to the access control of a particular subscriber. B_t is shared between the publisher and the subscriber in encrypted format using S_k . Using ID and B_t , a blind key B_k is generated on the subscriber side by performing the XOR operation:

$$B_k = B_t \oplus ID \quad (7.5)$$

Function 1 (Secret Key Sharing): By exploiting the blind key generation process, ICP-ABE generates a secret key. In addition, it shares secret keys between the publisher and the subscriber with a lightweight encryption/decryption strategy. Here, the primary functions are the client installation with required attributes and key sharing.

Definition 2 (Access Tree Creation (AR, ID)): Considering a set of attributes as $\{AR_{t1}, AR_{t2}, \dots, AR_{tn}\}$, the correct representation of the number of different access policies provides a set of attributes that is equal to $2^{\{AR_{t1}, AR_{t2}, \dots, AR_{tn}\}}$. The access tree structure AR is a non-empty set obtained from the set of attributes defined as $AR \subseteq 2^{\{AR_{t1}, AR_{t2}, \dots, AR_{tn}\}} \setminus \{\emptyset\}$. Here, $\{\emptyset\}$ is not included as an access structure cannot be empty and should comprise at least one subset of attributes. Figure 7.5 illustrates an example of the access tree structure.

Function 2 (Access Tree): Identify attribute-based access controls to specify the particular rules for access policies. All possible sets in AR are called authorised sets. A set of attributes to create an access tree. Different sets of attributes are used. For example, a nurse might have read-only access to patient records, while a doctor may have read-and-write-only access. Each non-leaf node of Γ is a gate function, either AND or OR. According to the attribute sets, each publisher can obtain a blind key from the publisher for suitable access.

Definition 3 (Encrypt ($Ms; AR; B_k$)) and (Decrypt ($Ct; B_k$)): The encryption algorithm considers the message Ms and an access tree structure AR in the universe of attributes. It outputs the ciphertext Ct using B_k , and the publisher takes the hash of the

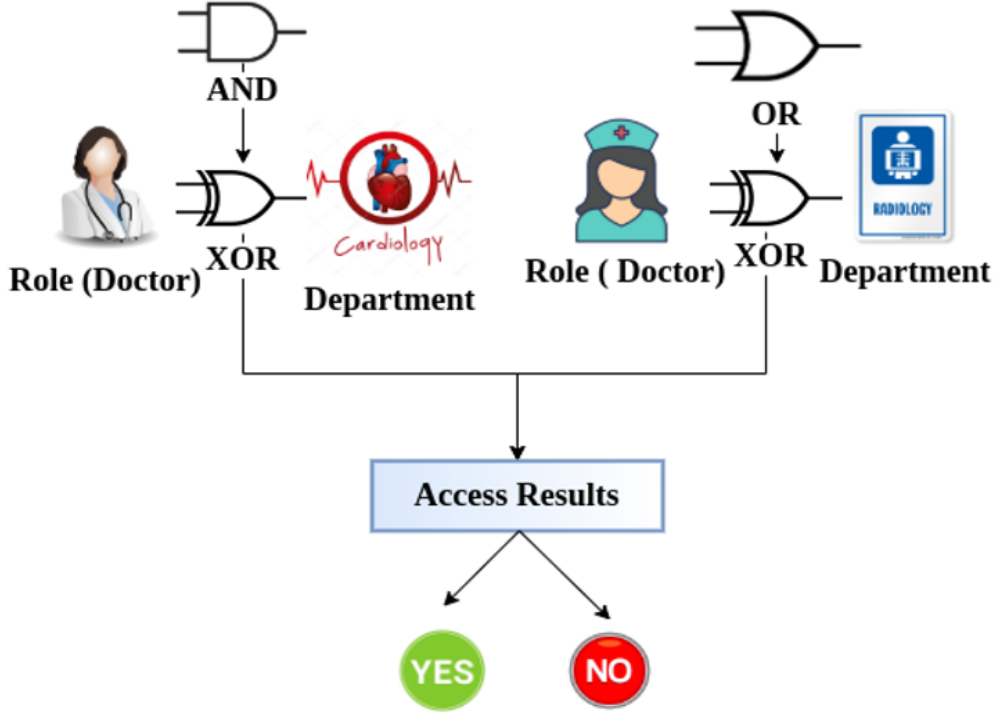


Figure 7.5: Example of Access Tree Structure

length of the attribute using B_k . ALK is used to hash the encrypted message Ms . This process is also executed on the subscriber Sub while subscribing to a particular topic Top on the server. The decryption algorithm at Sub takes Ct as input and the blind key B_k of a client. Only outputs the requested content Ms if the request satisfies B_k . Otherwise, the output is \perp .

Function 3 (Encryption and Decryption): The data is securely transmitted using lightweight cryptography from the publisher and subscriber.

Encryption: During encryption, the ICP-ABE takes the plaintext of the publisher with the corresponding key as cryptographic input. In addition, it exploits mathematical encryption operations on plaintext to convert it into ciphertext.

$$Ct = E(Ms)_{B_k} \quad (7.6)$$

The ciphertext is transmitted to the subscriber through a broker using lightweight cryptography.

Decryption: The cyphertext with its corresponding key is input into the decryp-

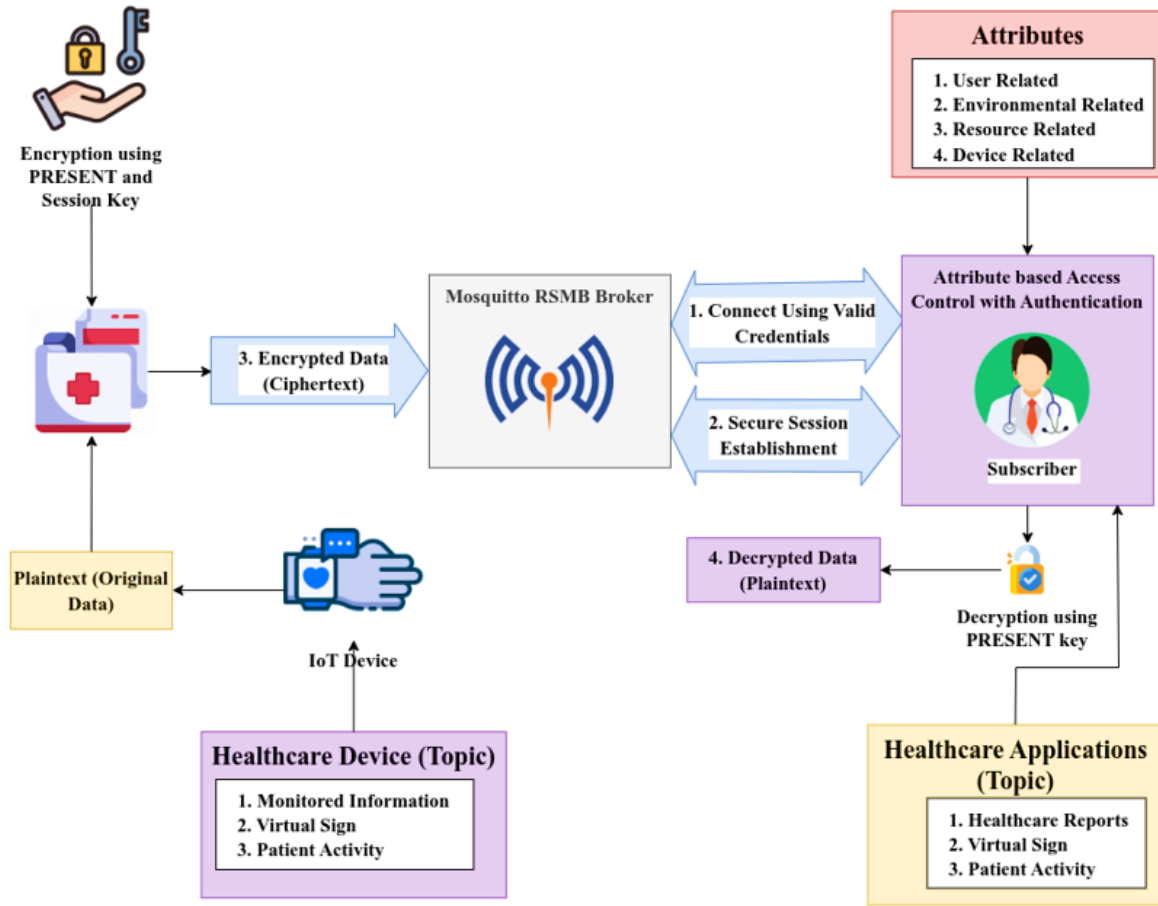


Figure 7.6: Encryption and Decryption of ICP-ABE

tion model on the subscriber side. In addition, it exploits the mathematical function of cryptography used to encrypt the ciphertext to perform the reverse operation of such encryption. Moreover, the reversed encrypted data is the original plaintext, available only to authorised users.

$$Ms = D(Ct)_{B_k} \quad (7.7)$$

Figure 7.6 shows the encryption and decryption process of the ICP-ABE scheme.

Table 7.2 illustrates various MQTT topics for the Healthcare 4.0 environment and the corresponding access policies for both the ICP-ABE and optimised CP-ABE schemes. The topics are used to access sensitive patient information. This table helps to combine topic-based access control policies based on roles and attributes.

Table 7.2: Access Policies for ICP-ABE and Optimised CP-ABE..

Topic	Department	Access Policies for ICP-ABE and Optimised CP-ABE
Heart Rate	Cardiology	Cardiologists and Nurses in on-duty
Blood Pressure	General	General Practitioners and Caregivers in on-duty
Patient History	General and Specialist	General physician and Specialists in on-duty
Telehealth	General	General physician in on-duty
Glucose Level	Diabetic	Diabetologist in on-duty
Body Temperature	General	Doctor or Nurses in on-duty
Blood Oxygen Saturation	General	Pulmonologists, Doctors, and Nurses in on-duty
Emergency Alerts	General	Emergency Response Team and Doctors in on-duty
Medication	General	Caregiver, Pharmacist, and Doctor
Prescription Detail	General and Specific	Doctor, Prescriber, and Pharmacist

7.3.2 Security Model

The security model describes communication between an adversary AS and the broker.

Definition 4 (Adversary (AS and $A_0 \mid A_1$)): An adversary AS trace a bit of ciphertext policies $A_0 \mid A_1$ as a trace of ciphertext, and it can connect to the publisher Pub if AS is satisfied. An adversary's probability of winning the game is directly proportional to the key size and algorithm complexity and indirectly proportional to the KeepAlive parameter. Where $(A_0 \mid A_1)$ indicates the conditional probability of A_0 given A_1 . The term A denotes the attack event.

$$P(A) = Probability(A_0 \mid A_1) = A \propto (S_{ksize} \wedge operations) \quad (7.8)$$

$$\&A \frac{1}{\propto} (KeepAlive \ Value).$$

Consequently, MitM attacks are prevented by using large key sizes and complex encryption operations, and SlowDoS attacks are mitigated by using smaller KeepAlive parameters.

Definition 5 (Key Tracing): Below are the steps necessary to distinguish between

a malicious and a legitimate MQTT request.

Initialization: The legitimate *Pub* submits two ciphertext policies, A_0 and A_1 , to the server. The server runs the setup algorithm and outputs the public key to the publisher. *AS* queries the server in polynomial time.

Guessing: *AS* attempts to output a B_k by tracing and analysing one bit of the ciphertext; it wins the game if B_k is satisfied. Otherwise, *AS* repeats the same process.

Key Generation: *AS* sends an attribute list A^* to the server; the latter verifies whether $(A^*| = A_0 \wedge A^*| = A_1)$ or not.

Hashing: *AS* generates ciphertext using B_k and applies a hash algorithm using ALK if $(A^*| = A_0 \wedge A^*| = A_1)$.

By obtaining the secret key, adversaries can gain access to the server. Key tracing is mainly dependent on the complexity of the system. The strength of the security scheme is adversely affected by minor secret keys and simple operations. As a result, the proposed project aims to improve system security in various ways. This approach includes the introduction of blind tokens, the implementation of attribute auditing processes, and the implementation of ALK for hashing operations.

7.3.3 Topic-based Encryption and Decryption using PRESENT Keys

This strategy mainly focuses on encrypting and decrypting messages of MQTT clients who are publishers or subscribers based on specific topics, where each topic is assigned a particular key to improve security and message compartmentalisation. The topic-based encryption and decryption strategy ensures less computational requirements and energy efficiency by utilising PRESENT keys. Thus, the scheme offers robust encryption and decryption while meeting the specific requirements of low-power healthcare IoT devices. Moreover, this strategy ensures that only authorised subscribers with the correct decryption key can access and decrypt data, thus indirectly enhancing data confidentiality and integrity while maintaining significant performance suitable for resource-limited healthcare IoT networks.

7.3.4 Secret Key Sharing Using the PRESENT Algorithm

In most existing ABE schemes, the MQTT server generates secret keys based on attribute information about each user, which may compromise client privacy. The proposed scheme addresses this issue by separating the attribute audit and key extraction functions. ICP-ABE allows the publisher to act as an attribute audit centre. By limiting the secret key length using the key register rotation and attributes-based self-key revocation, the proposed ICP-ABE model minimises the overhead and resource consumption at the publishers. The overhead of publishers is nearly 64 bits, a minimum of the conventional work.

Publishers and subscribers know the blind token and its associated attributes, while brokers are only authorised to know them without their corresponding attributes. Clients can submit their attributes and identities to the publisher through the broker during the data request. Publishers are responsible for auditing blind tokens according to attributes and providing subscribers with encrypted data with a signature. Data encryption uses a blind key, and hashing is performed using ALK. Moreover, the blind token offers evidence that the client has specific attributes while keeping the attributes or contents of the token confidential. In particular, the blind token is initially shared using the secret key generated using the PRESENT algorithm.

Figure 7.7 illustrates this process. The symbol \oplus in the figure denotes the XOR operation. The key register field stores keys for specific tokens, which rotate every round. The private keys of the subscribers are stored in the key register for a particular interval of time. The “all-rounder” encompasses the key sets used to add the keys in every state. The S-box layer replaces the bytes in the array or sub-type forms, each consisting of 4 bits. Initially, the subscriber sends a blind token subscriber content request to the publisher using MQTT for network initialisation. The subscriber requests the publisher through the broker to send the content of interest, and the publisher is responsible for verifying the legitimacy of the blind token. The requested content is sent if the token is valid; otherwise, the request is aborted.

The subscriber and publisher store their attributes and relevant blind keys in the at-

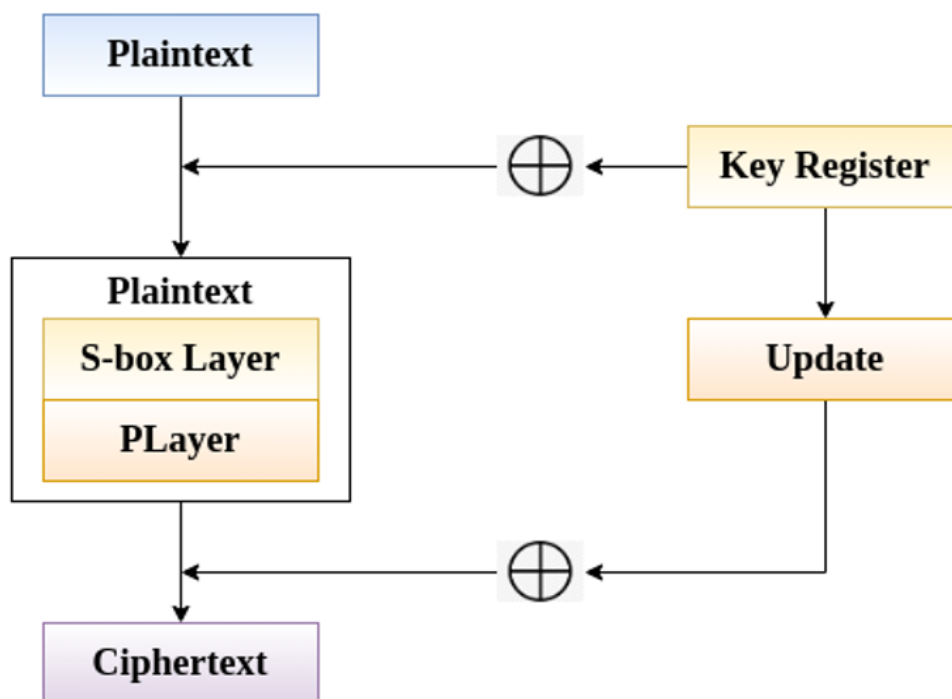


Figure 7.7: Functions in PRESENT Algorithm

tribute set. The publishers audit the blind keys and associated attributes and, depending on the attribute policy, send the same blind key to the broker to deliver the requested content to the subscriber. The blind key is securely sent using PRESENT encryption [267], which generates a secret key by running 31 rounds of the XOR operation with a key size of 80 bits. PRESENT consists of the following functions:

1. AddRoundKey, which adds a round key to each state.
2. The S-box layer replaces each byte in an array with a subbyte (the S-box is of type 4-bit to 4-bit).
3. P-Layer, which permutes each state into the predefined position.

The user-provided secret keys are stored in the key schedule, and a key register is rotated 61 positions to the left. In the S-box, the leftmost four bits constitute the key. When requested content is sent, the publisher and the subscriber modify the blind key by performing the XOR operation between the previous blind key, the blind token, and the previously accessed topic.

7.3.5 Attribute-Based Signature Scheme and Self Key Revocation Scheme using Attributes

After encrypting the data, the publisher calculates the length of the attribute using ALK, which is also used to hash the secured data. This process is also executed on the subscriber when subscribing to a particular topic on the MQTT server. The second component of the scheme is called the self-key agreement, which both the publisher and the subscriber perform. The publisher stores the published encrypted data on the server with the index of the encrypted topic. Subscribing devices send a request message and an encrypted topic using a blind key to access a particular topic. When the server finds a match in the index, it sends a message to the subscriber associated with the encrypted topic. To decrypt the received message, the subscriber generates the ALK.

The proposed scheme updates the ALK using the previously accessed topic and key. This approach avoids secure key sharing between devices and ensures the security of the system. To accommodate devices with limited battery resources, the proposed scheme allows MQTT clients to store only the previous topic, blind key, and ALK. This method will enable them to generate the security key on their own. The KeepAlive parameter alone cannot prevent SlowDoS specific to the MQTT protocol. Therefore, the proposed scheme disables client nodes from defining the KeepAlive parameter on the server. Instead, the broker represents a standard KeepAlive parameter value for all client nodes. This approach ensures lightweight authentication and privacy protection for IoT clients.

7.4 Security and Computational Cost Analysis

In this section, we will perform a security and computational cost analysis of the proposed scheme, which aims to address some challenges associated with applying CP-ABE schemes over MQTT using lightweight schemes.

7.4.1 Security Analysis

We consider the following types of attacks on MQTT communication: DoS, Spam, Poison, and MitM attacks. DoS attacks attempt to waste network resources by unnecessarily flooding control messages. Botnet groups can also infect clients and launch attacks on MQTT communication, such as DoS, Poison, and Spam. MitM attackers can modify MQTT packets between servers and clients, reducing efficiency. The chosen ciphertext attack also attempts to identify the plaintext by selecting specific ciphertexts.

Security against SlowDoS Attack: A robust cryptographic mechanism with server support can prevent attackers from accessing the blind token and modifying messages between smart sensors and the server. However, a SlowDoS attacker can copy a valid message and slowly resend it to the server, extending the KeepAlive parameter value and potentially gaining network access. To take advantage of all available connections to the broker, the SlowDoS attacker establishes many connections with minimal resource consumption. If the attacker establishes all available MQTT connections, it leads to a DoS attack, which can negatively impact MQTT performance, even when the KeepAlive parameter is used. Implementing the ICP-ABE scheme in MQTT prevents this attack since the server determines the standard KeepAlive parameter, and all clients revoke the key independently, preventing the attacker from sending packets again for an extended period. ALK values are generated in a predetermined manner and vary during communication. If a node resends a packet using the hash value of the encrypted message with the old ALK, the server and the client can identify attack packets based on mismatched hash values.

Security against Spam and Poison Attack: The proposed ICP-ABE scheme protects against Spam and Poison attacks in MQTT communication. The spam attacker sends unsolicited commercial messages to many clients, which can significantly impact a group of clients who have published similar content. However, due to the differences in published topics between clients in MQTT, it is not always possible to broadly implement the spam attack. On the other hand, poisoning attacks aim to inject erroneous data into the IoT network. Such an attack is possible when the attacker compromises the publishers

and obtains secret information. To prevent these attacks, the proposed ICP-ABE scheme uses different secret keys, such as the secret key, the blind key, and ALK, in various steps to ensure MQTT communication security.

Security against MitM Attacks: MitM attacks can break security schemes, making them a significant threat to MQTT communication. Once an attacker has compromised nodes, they can execute various MitM attacks. Although key revocation is often effective in preventing MitM attacks, using it may not always be possible. Furthermore, the PRESENT algorithm used in the proposed scheme has the disadvantage that it is susceptible to biased input in the keyspace, which attackers can exploit. If an attacker obtains the secret key, they can apply a brute-force method to trace the blind and ALK keys, compromising the strength of the proposed scheme.

7.4.2 Computational Complexity Analysis

To estimate the computational cost of the proposed ICP-ABE scheme, we consider the operations involved in secret key generation, blind key generation, and Pub/Sub messages using the PRESENT algorithm.

Attribute-based Signature Scheme and Self-Key Revocation: Both schemes incur complexity in computations, which also impacts the entire computational complexity of ICP-ABE. Consider that N numbers of attributes with t attribute policy size can impact U numbers of users in ICP-ABE. Table 7.3 defines the steps and notation for estimating the complexity of the attributes enabled by the PRESENT-based self-key revocation algorithm.

The ICP-ABE algorithm exploits an attribute-based PRESENT algorithm for secret key sharing and self-key revocation.

Secret Key Generation and Blind Key Sharing: The PRESENT algorithm executes the XOR operation for 31 rounds and generates the round key for blind key sharing. Additionally, it performs a shift and a permutation in the S-box layer and the p-layer, respectively. The cost of performing an XOR operation between two bytes and a one-byte rotation is equal. The cost of applying XOR between two bytes is denoted X ,

Table 7.3: List of Notations for Complexity.

Steps	Notations for complexity
$O(N)$	Key Generation or Sharing
$O(N.t)$	Signing/Verification
$O(U.N)$	Key Revocation
$O(1)$	Encryption/Decryption
$O(1)$	Key Scheduling
H	Key Size
X	XOR operations
L	S-box lookups

and the rotation of one byte is denoted $O(1)$. Similarly, a table lookup in the S-box is denoted as $1L$ for one-byte, and the cost of two-byte multiplication is denoted as $1M$. For shifting operations, it experiences $O(31)$ for 31 rounds; for permutation, it experiences $O(6)$. In addition, it performs 12 XOR operations ($O(12) \times 4$) for a 4×4 matrix, resulting in $O(48)$. The key schedule executes 16 transformations and generates an additional $64M$ confusion. Table 7.4 shows the notation used for the PRESENT operations performed in ICP-ABE.

Table 7.4: List of Notations for PRESENT in ICP-ABE.

Notation	Description
$O(1)$	XOR operation cost
$1R$	One-bit rotation operation cost
$1L$	S-box table lookup cost in one-byte
$1M$	Two-byte multiplication cost
$O(31)$	Total shifting operation cost of PRESENT with 31 rounds
$O(6)$	Total permutation operation cost
$O(12) \times 4$	Total cost for 12 XOR operations performed on a 4×4 matrix, resulting in $O(48)$.
16	Number of transformations executed during the key scheduling process
$64M$	Extensive cost produced by confusion owing to two-byte multiplications of key schedule.

Furthermore, communication complexity is determined by the length of the ciphertext, that is, the blind encrypted with a secret key. The initial complexity of the PRESENT

algorithm is as follows.

$$O(\text{PRESENT}) = 248X + 496L + O(85) + 64M \quad (7.9)$$

The terms XOR operations (X) and S-box lookups (L) are negligible, as they are constant factors, and their contributions are insignificant in a real-time healthcare IoT environment. Apart from this neglect, a critical factor, the length of the key size, is that H bits impact blind key sharing and are included in cost estimation. Therefore, the simplified computational cost of PRESENT is estimated as follows:

$$O(\text{PRESENT}) = O(85) + 64M + H \quad (7.10)$$

Secure Pub/Sub Messages: In most existing schemes, the secret key size is determined based on the number of attributes used in the scheme. However, in the proposed scheme, the secret key is generated through an XOR operation between the blind key, the blind token, and the topic previously accessed. In the case of the generation of ciphertext involving attributes N_{AR} , the complexity can be determined as follows.

$$O(\text{Ciphertext}) = H + H + N_{AR} * 2H = 2H + 2NH \quad (7.11)$$

Notably, if the ICP-ABE has many attributes, the complexity of $O(\text{Ciphertext})$ is also high. However, it is crucial to note that ICP-ABE limits the attributes in the key generation and revocation operations of ICP-ABE, resulting in minimum complexity. For clear analysis, consider that the size of the input messages is equal to that of the output ciphertext. This means that the computational cost of performing certain operations is directly related to H , the size of the data being processed. However, the proposed scheme reduces computational complexity using a self-key revocation scheme, shrinking the XOR complexity to a single operation. The complexity of the XOR operation is added as $O(1)$, and Equation 7.11 is simplified, as shown in Equation 7.12.

$$O(\text{Ciphertext}) = H + H + O(1) = 2H + O(1) \quad (7.12)$$

Thus, the computational complexity of the proposed ICP-ABE scheme is estimated by adding Equations 7.10 and 7.12, which are represented as follows.

$$O(\text{ICP-ABE}) = O(86) + 64M + 3H \quad (7.13)$$

Unlike conventional CP-ABE schemes, which scale with the number of attributes, ICP-ABE focuses more on fixed operations O and M instead of being directly influenced by large N ; therefore, the complexity appears minimal. In addition, the inclusion of $3H$ indicates that while there is some dependency on data size, the overall complexity remains manageable compared to conventional CP-ABE schemes, where both n and t can dramatically inflate complexity.

7.4.3 Time Complexity Analysis

MQTT request-response and data communication metrics are used to estimate the time complexity of ICP-ABE. The duration of the request/response is defined as the time it takes for a client to send a request for a topic and receive a response. The publisher acknowledges the client's request through a set of attributes, while the server recognises the request through a blind key. The server sends a response after verifying the client node. The time complexity, $Time_{Comp}$, is calculated at the end of the client using the following equation.

$$Time_{Comp} = R_q R_s + TCommunication \quad (7.14)$$

$R_q R_s$ can be estimated as the sum of the round-trip time taken by topic requests R_q and the response from the server R_s . Furthermore, the time taken to deliver the data is added to $R_q R_s$ to calculate the overall time complexity of the proposed work. However, in the existing scheme [267], $R_q R_s$ depends on the number of attributes and the length of the secret key, leading to a significant increase in time complexity. In contrast, the proposed scheme generates the blind key differently, rather than based on attributes, resulting in a considerable reduction in time complexity compared to existing works.

7.5 Evaluation and Validation

7.5.1 Provable Security

In the proposed ICP-ABE scheme, the PRESENT algorithm secures MQTT communication, and blind tokens are used to generate blind keys for secure data transfer. The security strength of the scheme is evaluated based on the key size of the PRESENT algorithm and the attribute-based encryption scheme. The success and failure probability of key tracing in a cryptographic algorithm with a key size of Ks are represented by $PS(t)$ and $PF(t)$, respectively. The number of keys that an attacker tries to process before time t is denoted by $K_x(t)$, and the total number of possible keys is 2^n . The term n represents the number of bits in the key.

$$PF(t) = 1 - PS(t) \quad (7.15)$$

$$PS(t) = K_x(t) / 2^n \quad (7.16)$$

By combining (7.15) and (7.16), we get:

$$PF(t) = 1 - (K_x(t) / 2^n) \quad (7.17)$$

The PRESENT algorithm provides better security against brute force and MitM attacks than the ECC algorithm used in [395]. However, according to [280], the RSA-based solution offers a better security strength.

The security strength of ICP-ABE involves several critical factors, such as key size, efficiency of a random function, relationship factor between bits, and attribute-ciphertext relationship factor. PRESENT generates subkeys randomly, and the proportion of zeros and ones determines the efficiency of subkeys produced by the PRESENT algorithm used in ICP-ABE. The purpose of measuring the efficiency of random functions is to determine whether the number of ones and zeros is equal to or greater than 50% during the generation of subkeys. If the generated subkeys do not meet these conditions, PRESENT does not

meet the essential characteristics of randomness. Therefore, the generation of subkeys in the PRESENT algorithm weakens the security of ICP-ABE. In Equation 7.18, we define the Efficiency of a Random Function (ERF), where a_0 and a_1 represent zeros and ones, respectively, in a sequence of bits n while generating subkeys in the present algorithm.

$$ERF = \frac{(a_0 - a_1)^2}{n} \quad (7.18)$$

The existing work in [395] and [280] investigates a hybrid of RSA and the Elliptic Curve Digital Signature Algorithm (ECDSA), and ECC algorithms, respectively. However, using large secret keys by RSA leads to inadequate ERF and compromised security due to the multiplication of two prime numbers.

The second criterion for security strength estimation is the bit-relationship test of ICP-ABE, which verifies its bit confusion and diffusion properties. In this test, the n is the input bits are mapped onto the m output bits. The bit-relation function is denoted as follows.

$$RF(\text{Plaintext})_n = RF(\text{Ciphertext})_m \quad (7.19)$$

Note that m is not necessarily equal to n bits. However, if $m \ll n$, it reduces the efficiency of RF , while increasing the security strength of ICP-ABE against cryptanalysis attacks such as MitM. In [280], RSA $RF(\text{Plaintext})_n$ is used, where the length of the prime numbers $p * q$ is not equal to the security strength of the bit length n . A poor bit relationship function can allow attackers to breach the security scheme.

The attribute-ciphertext relationship factors indicate the impact of the number of attributes on the ciphertext length. The ciphertext length affects the network performance in terms of delay and throughput, while a short length facilitates the key tracing by attackers. Here, N_{AR} represents the number of attributes, k denotes the average size of an attribute, and l represents the number of rows in the access structure.

$$\text{Ciphertext Length} = \frac{n}{m} + (N_{AR} * k) + l \quad (7.20)$$

In ICP-ABE, the impact of $tn * k$ on ciphertext length is negligible because ALK-based hashing always has the same length for any input data length, i.e. encrypted message. The proposed scheme only considers the length of the attribute in the generation of a key, but it does not affect the creation of a ciphertext. Therefore, the second term, $N_{AR} * k$ in Equation 7.20 is null for the proposed scheme. The negative impact of RSA [395] and ECC [280] on the security strength of the attribute-based encryption scheme is higher than that of ICP-ABE.

The following table 7.5 compares the security strength of the proposed ICP-ABE with four existing methods: simple PRESENT, KSA-PRESENT, RSA-ECC, S-MQTT and SSL/TLS.

Table 7.5: Attack Prevention Ability of Various Algorithms Against Different Attacks.

Security Breaches	Attack Prevention Ability of Algorithms					
	ICP-ABE	Simple-PRESENT	KSA-PRESENT	RSA-ECC	S-MQTT	SSL/TLS
DoS	High	High	High	High	High	Low
Slow DoS	High	No	No	Medium	No	Low
Spam	High	Medium	High	Low	High	Low
Poison	High	Medium	High	High	Low	Medium
MitM	High	Medium	High	Medium	Medium	Very High

7.5.2 Simulation based Evaluation

As discussed earlier, the MQTT protocol has been extended to include PRESENT, ALK, and self-key revocation, and validation has been performed using the Cooja simulation environment. The Cooja simulation is a Contiki OS extension highly suitable for the environment comprising resource-limited devices, especially in MQTT-enabled IoT applications such as Healthcare 4.0. Therefore, the Cooja is considered an appropriate simulation platform for the ICP-ABE. The main reason behind selecting Cooja-based simulations is that the Cooja can provide a realistic simulation environment by accounting for real-time characteristics such as wireless communication, resource consumption,

and network topology, and performing simulations of MQTT in Cooja permits one to observe how ICP-ABE-enabled MQTT behave in a controlled realistic environment. The Cooja simulation has supported the energy consumption model of resource-limited IoT devices.

The proposed ICP-ABE model runs on Ubuntu 18.04 LTS with the Intel i3 2.5 GHz CPU and 4 GB of memory. Our results, shown in the following, indicate that ICP-ABE performs well with and without attacks. The proposed work can be considered a strong and lightweight security scheme for MQTT v5 in an IoT Healthcare 4.0 environment. To evaluate the performance of ICP-ABE and the existing CP-ABE [346], KSA-PRESENT [268], MQTT-PRESENT [267], RSA-ECC [280], SMQTT [395], and SSL/TLS [30] schemes, and simulations on IoT nodes were performed using Ubuntu 14.04 LTS 64-bit and Contiki-3.0. Table 7.6 presents the simulation parameters.

Table 7.6: Simulation Model.

Parameters	Values
Application Protocol	MQTT v5
Total Number of Nodes	31 (Router-1, Publisher-10 and Subscriber-20)
Number of Attacker Nodes	4
Dos Attack Nodes	2
MitM Attack Nodes	2
Simulation Area	$100m \times 100m$
Transmission Range	$50m$
Simulation Time	5 Minutes
MQTT-Broker	Mosquitto-rsmb broker-1.3.0.2
Algorithms	ICP-ABE, CP-ABE, Simple PRESENT, KSA-PRESENT, RSA-ECC, S-MQTT and SSL/TLS

7.6 Simulation Results and Discussion

Table 7.7, Table 7.8, and Table 7.9 show the simulation of the results of the existing KSA-PRESENT, MQTT-PRESENT, RSA-ECC, SMQTT, SSL/TLS, and ICP-ABE works under normal and different attack scenarios. The attacker reduces performance efficiency

by delaying, flooding, or losing MQTT traffic transferred between clients and servers. The results are obtained for different attacker scenarios to demonstrate the impact of the attacker on network performance.

7.6.1 Simulation Results

Due to one table being too long, we split three tables to show the simulation results, which are Table 7.7, 7.8 and 7.9.

Table 7.7: Results of Existing Works under Normal and Different Attack Scenarios - 1.

Performances	Attack Scenarios	ICP-ABE	CP-ABE	KSA-PRE	MQTT-PRE	RSA-ECC	S-MQTT	SSL/TLS
Throughput	Without	356.8	356.8	174	132.8	66.8	185.28	366.8
	MitM	45.28	38.12	22.8266	15.1466	48.26	39.4666	47.68
	DoS	457.44	350.32	451.226	355.626	139.73	449.72	476.8
	With	452.2663	349.36	354.6721	349.3662	160.85333	452.638	477.2
PDR	Without	95.204	94.25	93.75	91.966	97.462	90.752	95.62
	MitM	96.6976	89.78	19.741	14.0316	65.476	29.838	97.62
	DoS	96.870	87.65	32.766	27.740	60.091	33.486	97.624
	With	96.3566	90.32	44.5859	27.0134	63.414	36.217	97.62
Delay	Without	29.936	34.10	29.953	29.958	29.98	29.99	29.96
	MitM	29.96	36.21	29.99	29.92	29.98	29.90	29.99
	DoS	29.92	39.82	29.95	29.90	29.99	29.94	29.99
	With	29.925	35.55	29.9845	29.944	29.981	29.911	29.99

7.6.2 Discussion

Figure 7.8 illustrates the results of the performance comparison of seven different MQTT security algorithms: ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT, and SSL/TLS. The results are obtained for four scenarios: without attack, MitM attack, DoS attack, and with attack. The proposed ICP-ABE achieved

Table 7.8: Results of Existing Works under Normal and Different Attack Scenarios - 2.

Performances	Attack Scenarios	ICP-ABE	CP-ABE	KSA-PRE	MQTT-PRE	RSA-ECC	S-MQTT	SSL/TLS
Energy Consumption	Without	0.001577	0.001822	0.001555	0.00160	0.00158	0.001577	0.00165
	MitM	0.00201	0.00259	0.001914	0.0020	0.00127	0.00160	0.00311
	DoS	0.00164	0.00214	0.00163	0.00167	0.00141	0.00147	0.00216
	With	0.001497	0.00201	0.001725	0.001490	0.001378	0.0015377	0.00171
CPU Energy Consumption	Without	0.0000355	0.0000458	0.0000382	0.0000392	0.0000491	0.0000490	0.0000378
	MitM	0.00003125	0.0000396	0.00003125	0.00003125	0.0000311	0.0000234	0.0000312
	DoS	0.0000271	0.0000345	0.00002767	0.00003125	0.0000292	0.00003125	0.0000312
	With	0.0000468	0.0000591	0.0000449	0.00004692	0.00003191	0.000046890	0.0000512
Execution Time	Without	0.26	0.34	0.30	0.40	2.1	3.14	9.417
	MitM	0.0502	0.0645	0.0922	0.0378	8.347	0.329	9.4111
	DoS	0.428	0.545	0.425	0.257	4.704	0.502	9.4277
	With	0.473	0.589	0.55	0.349	5.33	4.14	9.431

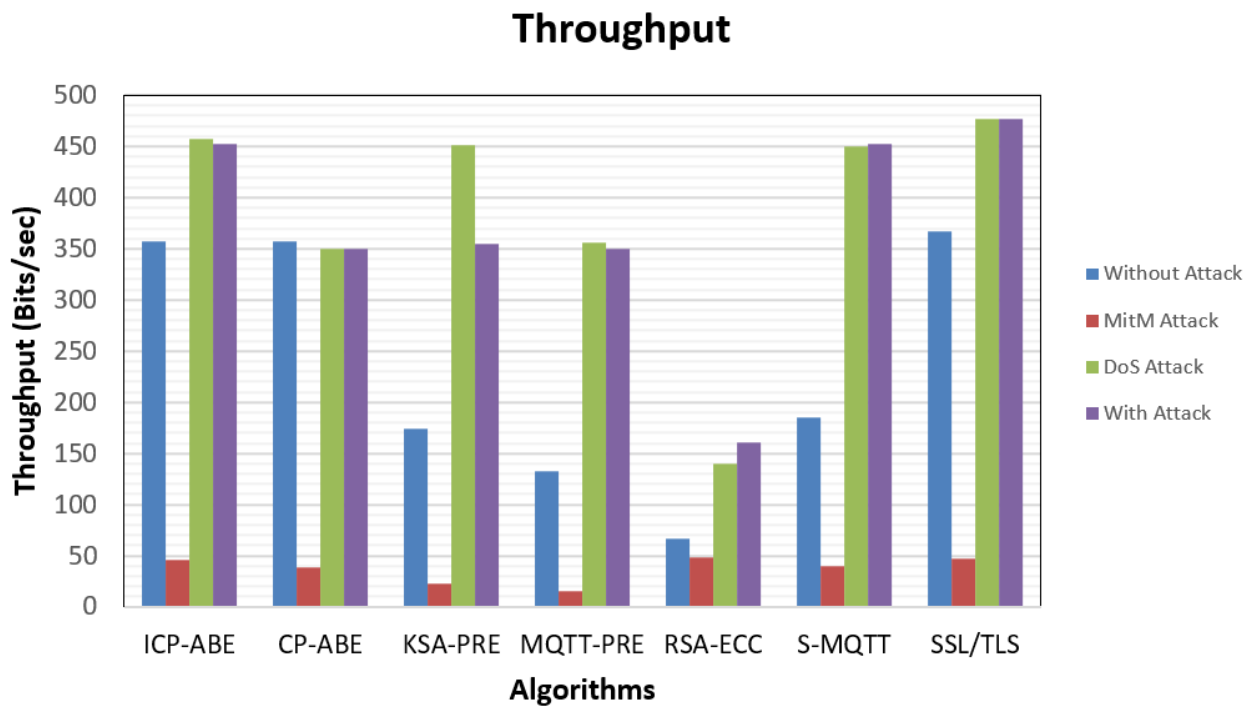


Figure 7.8: Algorithms Vs Throughput

Table 7.9: Results of Existing Works under Normal and Different Attack Scenarios - 3.

Performances	Attack Scenarios	ICP-ABE	CP-ABE	KSA-PRE	MQTT-PRE	RSA-ECC	S-MQTT	SSL/TLS
Communication Overhead	Without	64	64	80	80	128	128	128
	MitM	128	128	80	80	128	128	128
	DoS	128	80	128	128	128	128	128
	With	64	80	80	80	128	128	128
Strength Evaluation Criteria	Without	1.2	1	0.399	0.399	1	1	1.2
	MitM	1	1	0.399	0.399	1	1	1.2
	DoS	1	1	1	1	1	1	1.2
	With	1.2	1	0.399	0.399	1	1	1.2
Computation Overhead	Without	34.22	38.5	40.38	47.214	51.68	37.19	53.2
	MitM	45.22	42.7	42.38	48.34	55.89	39.39	60.2
	DoS	50.32	44.2	43.18	46.45	57.29	36.26	60.2
	With	37.22	41.5	43.38	50.214	54.68	40.19	55.86

356.8, 45.28, 457.44, and 452.27 bits/second throughput results for without attack, MitM attack, DoS attack, and with attack scenarios, respectively. Existing CP-ABE and proposed ICP-ABE accomplish nearly similar PDR under without-attack scenarios, as both exploit attribute-based encryption strategies for fine-grained access control while guaranteeing efficient packet delivery. Figure 7.8 shows that S-MQTT and SSL/TLS perform nearly the same or higher performance than the proposed ICP-ABE. The main reason is that SSL/TLS often has highly complex, heavy-weight mathematical operations with large keys. This complexity can ensure a higher security level and improve throughput compared to the ICP-ABE algorithm. For example, ICP-ABE and SSL/TLS attain 356.8 and 366.8 bits/second of throughput values, respectively, in the without-attack scenario. However, it varies by 452.27 and 477.2 bits/second, respectively, with the attack scenario.

Figure 7.9 demonstrates the results of the PDR comparison of seven different MQTT security works performed for various attacker scenarios. Although ICP-ABE, RSA-ECC and SSL/TLS attain almost 95% of PDR in an attack scenario, ICP-ABE outperforms

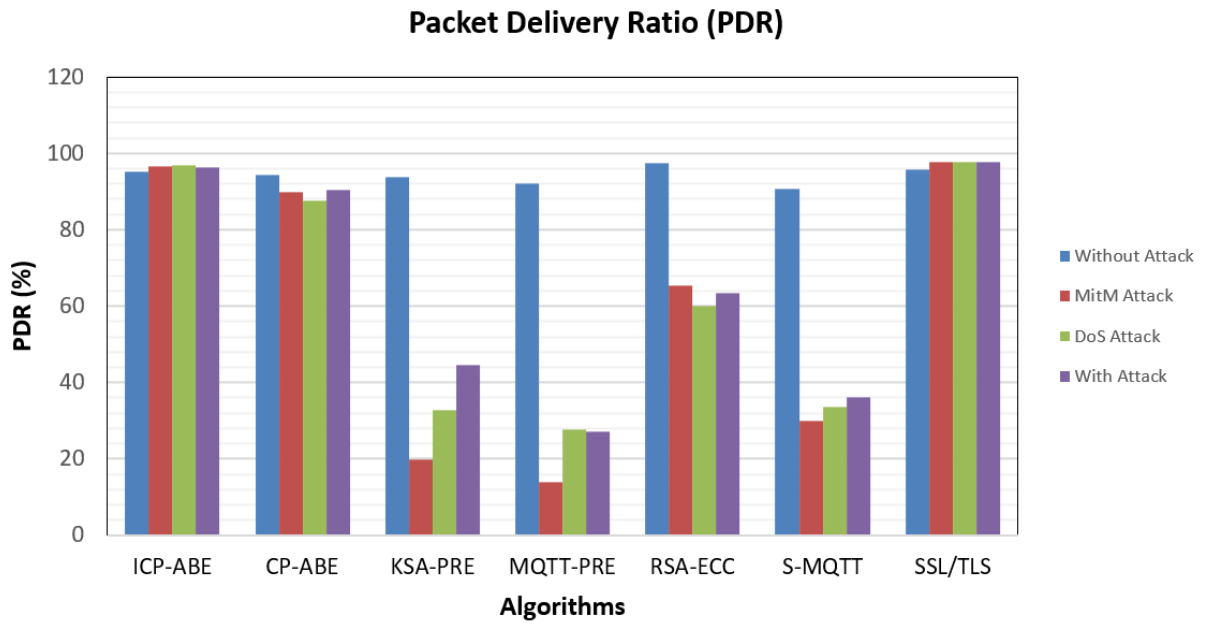


Figure 7.9: Algorithms Vs Packet Delivery Ratio

in the attacker presence scenario. From the results of Figure 7.9, it is observed that the proposed ICP-ABE outperforms the existing work, except SSL/TLS, in the scenario of an attacker's presence. Using the ICP-ABE, the length of the ciphertext and secret keys is independent of the number of attributes, reducing the proposed algorithm's complexity. As a result, it is more lightweight than other algorithms. Compared to SMQTT, RSA-ECC explores hybrid techniques and adds complexity to the communication process. When MitM and DoS attacks occur, attribute-based keys and ciphertext lengths degrade the performance of CP-AEB and SMQTT. Hence, Therefore, existing CP-ABE attains decreased PDR results under DoS and MitM with attack scenarios. This issue is addressed by ICP-ABE, which is more resistant to DoS and MitM attacks than CP-ABE and SMQTT. As a result, MQTT communication efficiency is significantly increased. The performance of ICP-ABE is enhanced even under malicious scenarios by segregating the functions of attribute auditing and key extraction. This scenario ensures MQTT security and helps achieve the highest performance, even in malicious scenarios. For example, ICP-ABE, RSA-ECC, and SSL/TLS obtain 96.36%, 63.41, and 97.62% of PDR values for the attacker scenario.

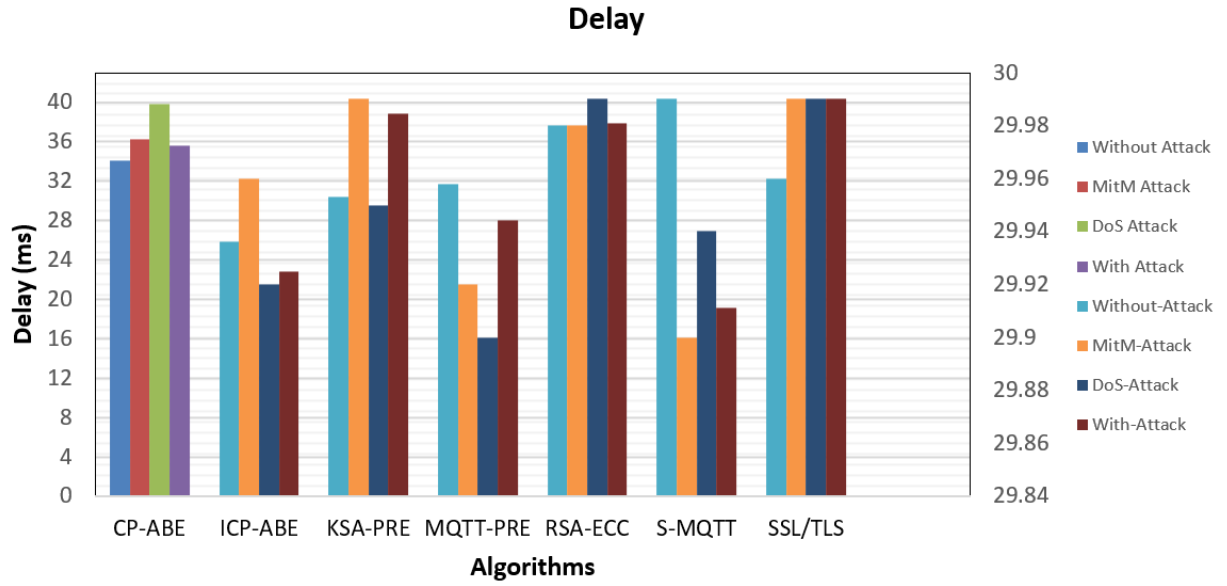


Figure 7.10: Algorithms Vs Delay

Figure 7.10 shows the comparison results of CP-ABE, ICP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT and SSL/TLS delay. To analyse the performance efficiency of such algorithms under different scenarios, results are obtained without attack, MitM attack, DoS attack, and with attack. The proposed ICP-ABE outperforms all existing schemes when there is no attacker in the network. The lightweight attributes-based key revocation strategy in the proposed model minimises the delay. For instance, the ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT, and TLS/SSL are 29.936 ms, 34.1 ms, 29.953 ms, 29.958 ms, 29.98 ms, 29.99 ms, and 29.96 ms, respectively, under the without attack scenario. Each algorithm incurs some delay to protect against the attacker's presence. However, the proposed model achieves the accepted delay level of 29.96 ms, 29.92 ms, and 29.93 ms delay values for the scenarios of MitM attack, DoS attack, and with attack, respectively.

In Figure 7.11, the results of the execution time of Seven different MQTT security strategies are observed. Performance is compared in four scenarios. The results show that the proposed ICP-ABE, CP-ABE, KSA-PRESENT, and MQTT-PRESENT accomplish less execution time results. A lightweight attribute-based self-key revocation scheme and the KeepAlive parameter value included in the ICP-ABE can prevent attackers from

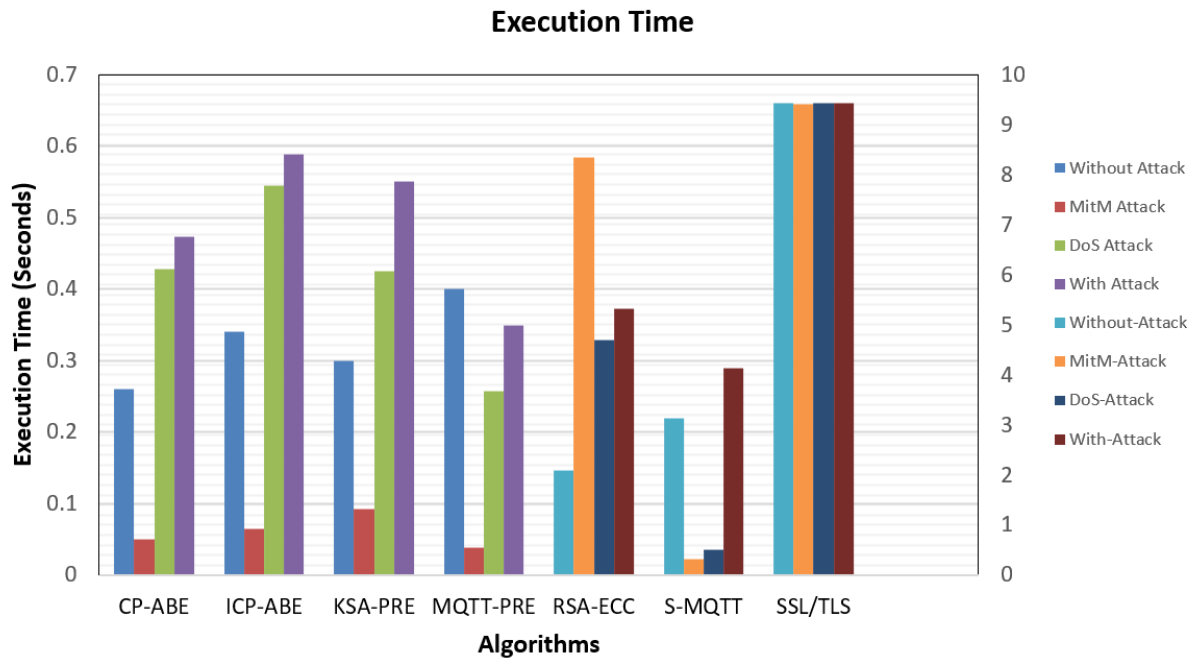


Figure 7.11: Algorithms Vs Execution Time

accessing the credential data after authentication. Therefore, it incurs a minimum delay even when malicious nodes are in the network. For example, ICP-ABE obtains 0.473 seconds of execution time in an attack scenario. In contrast, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT and SSL/TLS achieve 0.589, 0.55, 0.349, 5.33, 4.14, and 9.431 seconds of delay, respectively, in the same scenario. The delay of RSA-ECC and SSL/TLS is higher than that of the proposed scheme in all scenarios. The reason is that RSA-ECC and SSL/TLS include complex heavy-weight key structures to ensure high security against DoS and MitM attacks, resulting in high execution time.

Figure 7.12 illustrates the CPU results and the average energy consumption for ICP-ABE, CP-ABE, Simple-PRESENT, KSA-PRESENT, RSA-ECC, SMQTT, and SSL/TLS, respectively. The results are achieved for different scenarios, including without attack, MitM attack, DoS attack, and with attack. The proposed scheme employs blind key generation, reducing the secret key length and the ciphertext. As a result of the reduced key size, ICP-ABE reduces CPU load and consumes minimal energy to identify and authenticate IoT devices. Figure 7.9 shows that the CPU energy consumption of ICP-ABE is much lower than other existing methods. For example, the proposed ICP-ABE consumes

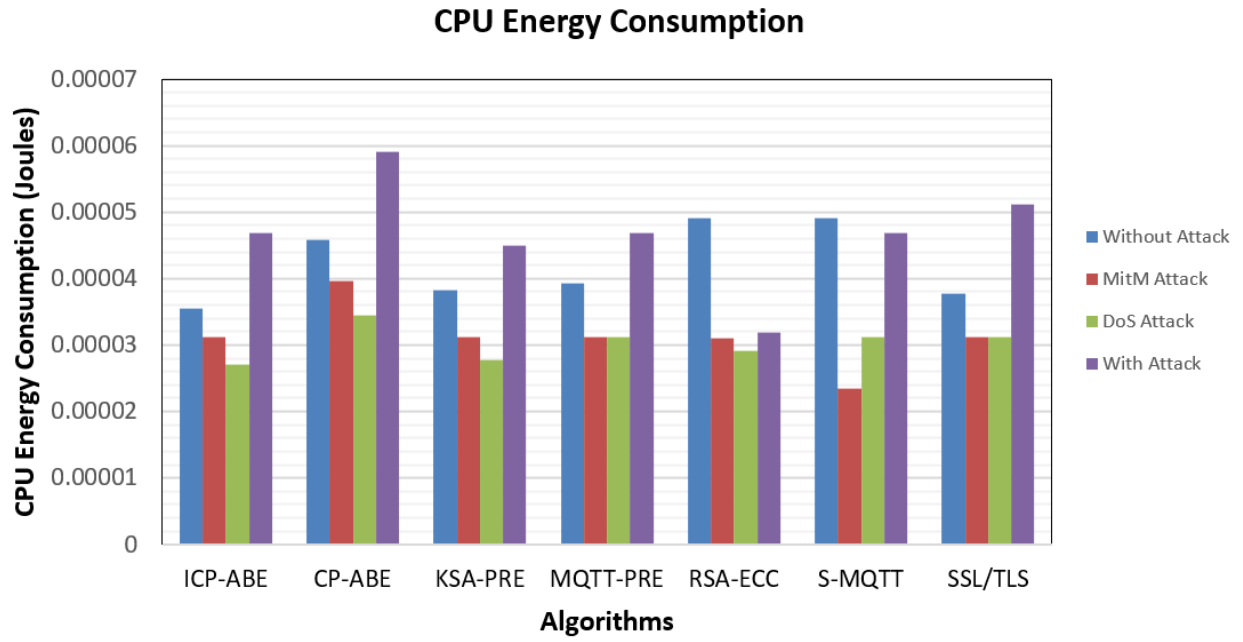


Figure 7.12: Algorithms Vs CPU Energy Consumption

0.00003125 and 0.00271 joules of CPU energy for the MitM and DoS attack scenarios, respectively, as shown in Figure 7.9. The figure clearly shows that the energy consumption of SSL/TLS is higher than that of ICP-ABE. This is caused by the heavy-weight key structure and complex algorithm of SSL/TLS. For example, ICP-ABE and SSL/TLS attain 0.0000468 and 0.0000512 CPU energy joules, respectively, in attack scenarios.

The average energy consumption results of ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT and SSL/TLS are compared under without attack, MitM attack, DoS attack, and with attack scenarios as shown in Figure 7.13. Unlike ICP-ABE, the simple PRESENT and KSA PRESENT algorithms do not consider the secret key length for key revocation. Thus, it significantly increases resource consumption at the nodes. Also, the existing CP-ABE accomplishes higher average energy consumption than the ICP-ABE owing to its complex decryption operations and higher computational overhead in attribute-based access control. Compared to the proposed scheme, the RSA-ECC and SSL/TLS schemes have the additional disadvantage of the key store used in the existing work, which degrades the performance of the protocol. Unlike that, the proposed ICP-ABE sets the value of the KeepAlive parameter too high, and attackers may be able

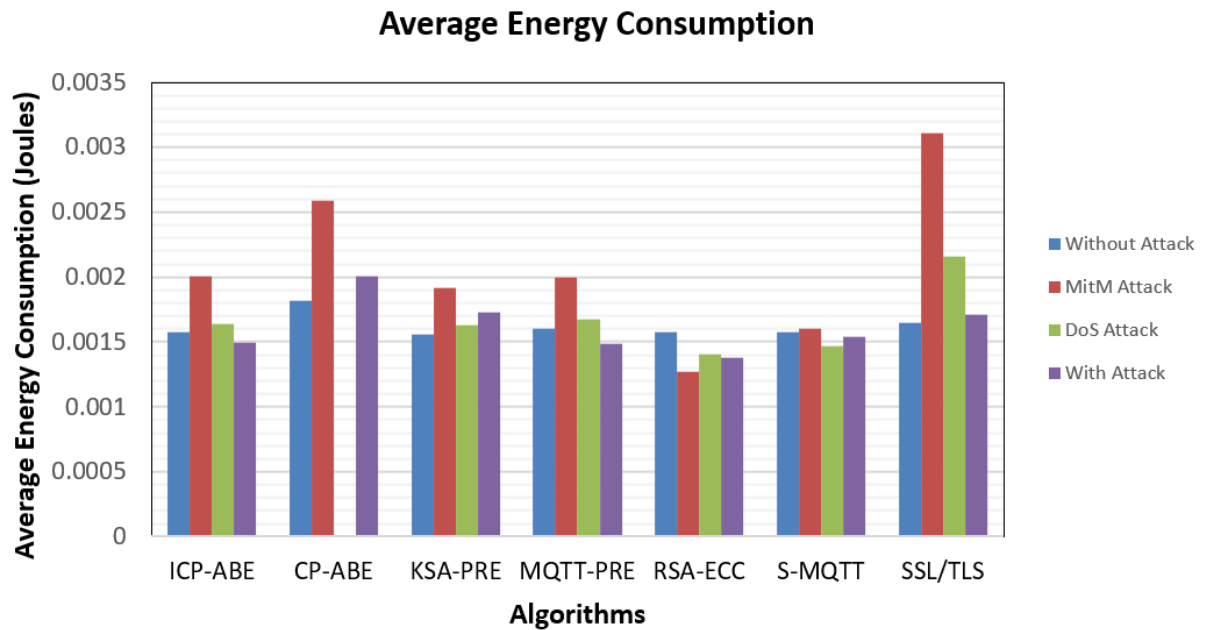


Figure 7.13: Algorithms Vs Average Energy Consumption

to send unnecessary control messages and request messages to the server. When used in the context of IoT, SMQTT tends to result in unnecessary energy consumption due to the length of the secret key, which increases with the number of attributes. Moreover, revocation of the self-key reduces overall energy consumption by sharing the secret key separately. For example, the energy consumption of ICP-ABE and SSL/TLS is 0.001497 and 0.00171 joules, respectively, when the combination of MitM and DoS attackers is present in the network.

Figure 7.14 illustrates the computation overhead of the ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT, and SSL/TLS algorithms. The results are obtained for scenarios without attack, MitM attack, and DoS attack. The figures show that the computation overhead of the proposed ICP-ABE is reduced compared to other existing protocols. This is because the proposed model considers the advantages of the PRESENT and ABE models in blind token generation and self-key revocation. Thus, it simplifies the algorithm process of ICE-ABE and optimises the lightweight design without compromising the security level and performance efficiency of MQTT. For example, ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT, and

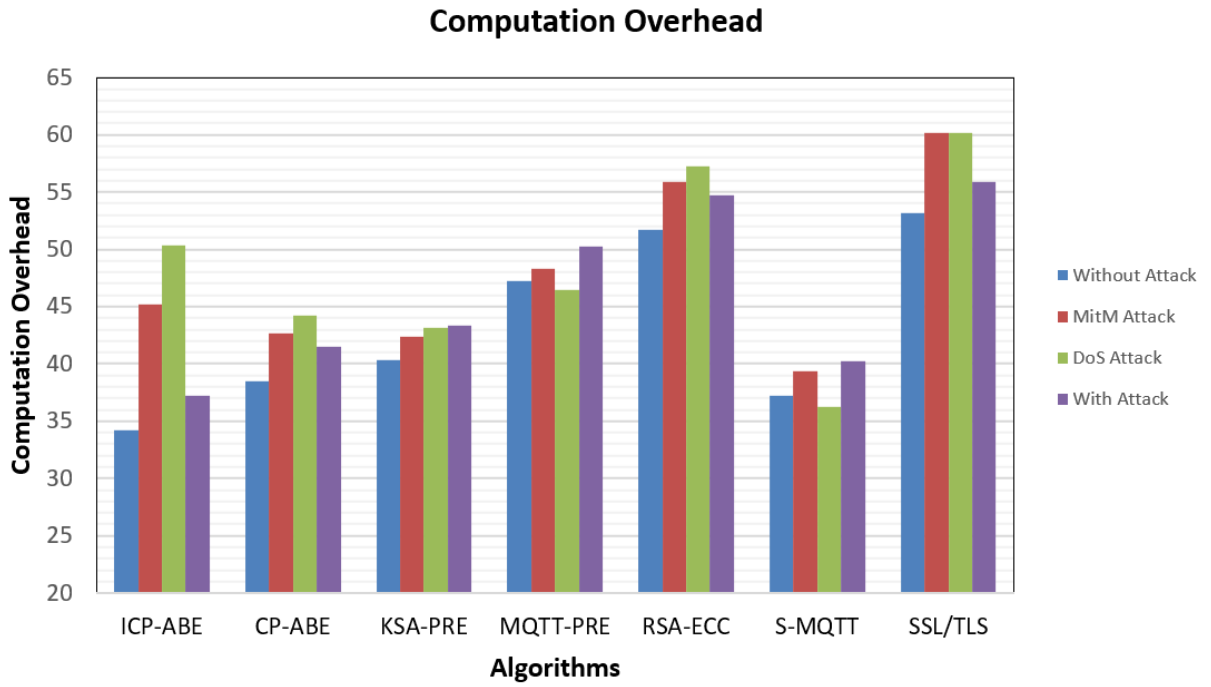


Figure 7.14: Algorithms Vs Computation Overhead

SSL/TLS obtain 37.22, 41.50, 43.38, 50.214, 54.68, 40.19, 55.86 of computation overhead, respectively, under the attack scenario.

Figure 7.15 shows the results of the comparison of strength evaluation criteria for the ICP-ABE, CP-ABE, KSA-PRESENT, MQTT-PRESENT, RSA-ECC, S-MQTT and SSL/TLS algorithms obtained in normal and vulnerable environments, including DoS and MitM attacks. The length of the ciphertext is a crucial factor in determining the suitability of a security scheme for IoT devices. However, unlike the proposed scheme, both existing schemes, CP-ABE and S-MQTT, increase the length of the ciphertext as the number of attributes increases, resulting in a degraded security strength. For example, in MitM and DoS attack scenarios, the security strength of ICP-ABE is 1. Although the existing work satisfies one of the security strength criteria for all scenarios due to its key scheduling and attribute-based ciphertext creation method, its communication overhead increases due to the longer ciphertext length.

Figure 7.16 illustrates the results of the communication overhead comparison of seven different MQTT security algorithms. The results are obtained for four scenarios: without attack, MitM attack, DoS attack, and with attack. The results demonstrate that the

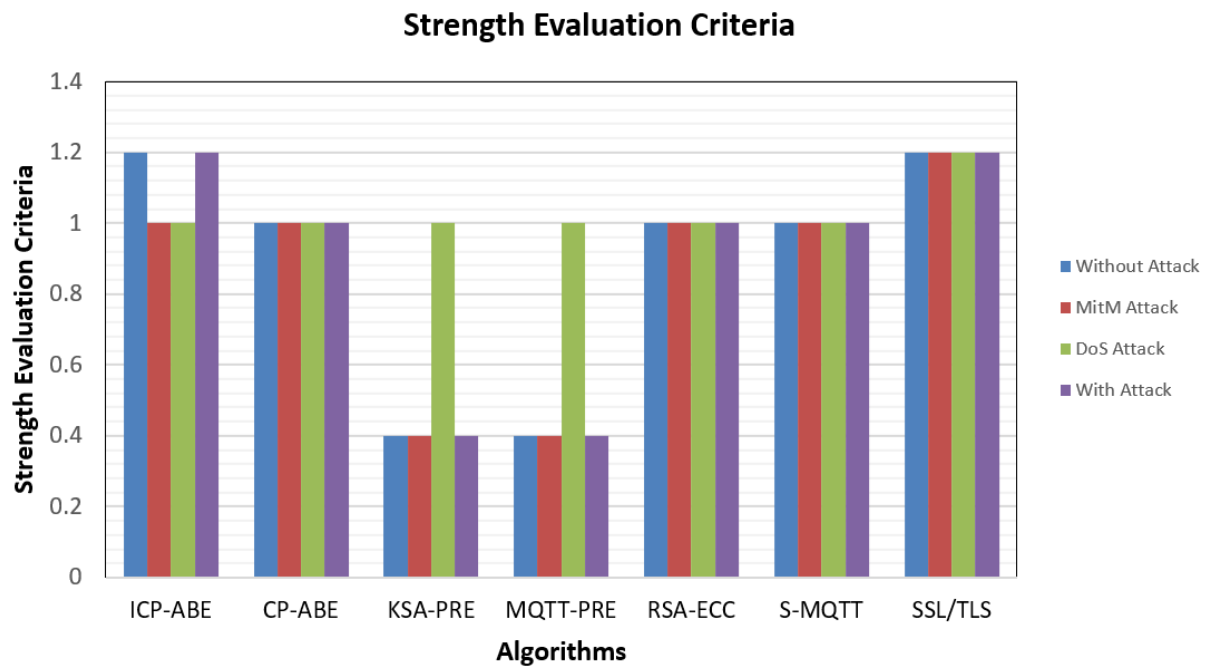


Figure 7.15: Algorithms Vs Strength Evaluation Criteria

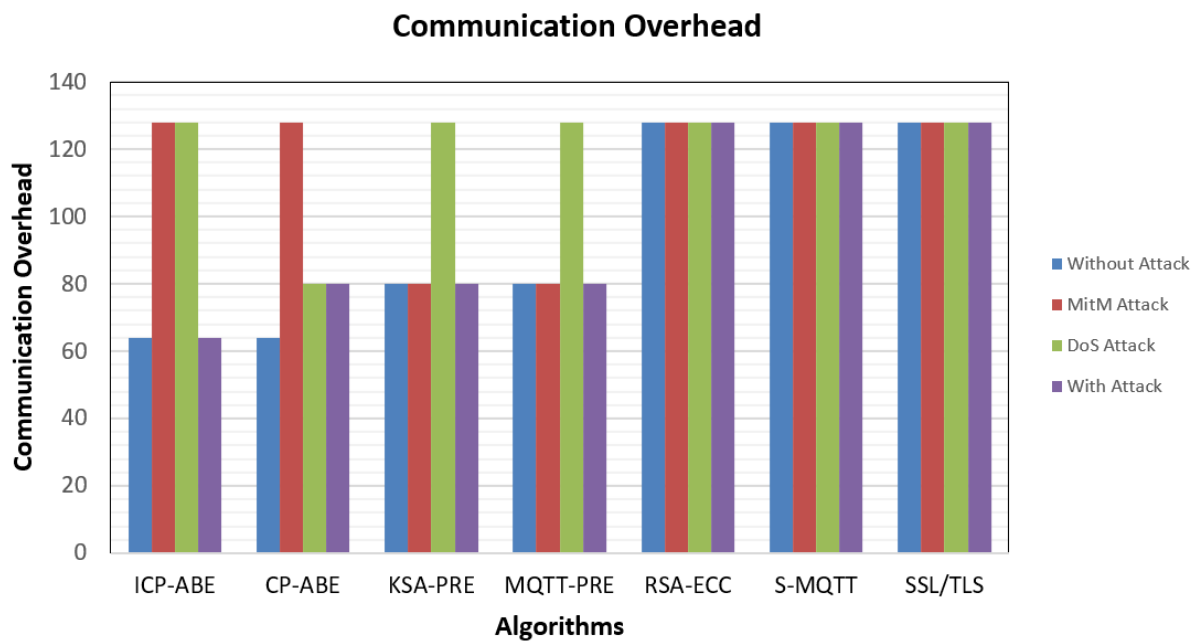


Figure 7.16: Algorithms Vs Communication Overhead

proposed ICP-ABE communication overhead is minimal in both the without- and with-attack scenarios compared to other protocols. Furthermore, the communication overhead of CP-ABE, Simple-PRESENT, KSA-PRESENT, RSA-ECC, and S-MQTT is high in all scenarios compared to the proposed scheme. This is because the length of the secret key increases with the number of attributes, resulting in unnecessary communication overhead and poor strength when used in an IoT environment. For example, the communication overhead of ICP-ABE is only 64 bits in both with and without scenarios. The CP-ABE, KSA-PRESENT and MQTT-PRESENT accomplish 80 bits of communication, and the RSA-ECC, S-MQTT, and SSL/TLS achieve 128 bits of communication overhead under an attack scenario, owing to the different key scheduling and attribute-based ciphertext creation process.

7.6.3 Summary

The performance analysis of the proposed ICP-ABE demonstrates its superiority in terms of different metrics under various scenarios to balance security and efficiency in MQTT-enabled healthcare 4.0 environments. Unlike existing CP-ABE, which includes key generation and attribute auditing mechanisms, leading to high computational complexity, especially in resource-restricted environments, the proposed ICP-ABE decouples these roles to shrink encryption and decryption latencies. This improved structure, consolidated with the exploitation of the lightweight PRESENT algorithm for securely sharing keys, achieved a 10.31% reduction in computational overhead in attack scenarios and made it highly suitable for resource-constrained healthcare devices. In addition, the integration of topic-based encryption in ICP-ABE enhances processing efficiency by providing only security to relevant MQTT topics, improving throughput by 29.47% and shrinking energy consumption by 25.52% compared to the existing CP-ABE in attack scenarios. Using a blind key revocation strategy allows the MQTT broker to verify the identities of publishers without accessing their attributes, safeguarding privacy without overburdening the broker. These enhancements make ICP-ABE contribute to high security, better scalability, and acceptable latency under varying healthcare IoT network conditions and

maximum throughput. Thus, ICP-ABE is a practical, efficient, and secure solution to transmit real-time medical data in healthcare 4.0 IoT systems.

7.7 Chapter Summary

This work proposes a novel ICP-ABE scheme for MQTT to improve its security. The ABE scheme has been made more efficient by separating attribute auditing from secret key generation. In particular, a combination of MQTT blind keys associated with attributes and the ICP-ABE algorithm has reduced the computational burden of existing attribute-based schemes without compromising network security. The scheme also prevents client nodes from executing SlowDoS attacks on MQTT servers. This scheme also prevents client nodes from performing SlowDoS attacks on the MQTT server by disabling the KeepAlive parameter on the server. In addition, the scheme supports the implementation of the proposed algorithm on devices with limited resources. This system uses the PRESENT algorithm to safely exchange anonymous tokens between different devices. Furthermore, we evaluated and compared the proposed scheme with four existing schemes, Simple-PRESENT, KSA-PRESENT, RSA-ECC SMQTT, and SSL/TLS, using provable security and formal analysis methods. The simulation results indicate that ICP-ABE significantly improves the performance and security of MQTT while exhibiting lightweight and secure characteristics in both the standard and attack scenarios.

However, despite our efforts, there are still some issues with the ICP-ABE scheme that we aim to address in future work, such as the revocation of malicious users and tracking malicious users with similar attribute sets. We focus on improving the security of the ICP-ABE solution and testing its usefulness in other IoT application layer protocols. Furthermore, we plan to evaluate the results of the improved ICP-ABE solution against existing lightweight security solutions and implement it on a natural healthcare IoT platform. We are also improving secure communication for IoT devices using the Pub/Sub architecture. This improvement will be a significant focus of our future work.

OCP-ABE: An MQTT-Based Lightweight Authentication Scheme for the Internet of Medical Things

This work proposes a novel authentication scheme using Optimised CP-ABE (OCP-ABE). ICP-ABE, introduced in Chapter 5, effectively reduces computational overhead by separating key extraction from attribute auditing and exploiting lightweight PRESENT-based cryptography. However, it remains inadequate for critical healthcare 4.0-specific application domains. In the practical healthcare IoT domain, ICP-ABE does not provide satisfactory protection against attribute correlation attacks, which may expose sensitive patient information due to overlapped attribute sets. It also exploits a static key revocation strategy, unsuitable for dynamic healthcare networks where healthcare devices frequently join and leave, such as during emergency patient handovers or mobile monitoring. Furthermore, ICP-ABE does not incorporate secure MQTT session handling, making the communication layer susceptible to session hijacking and replay attacks, which pose severe threats in time-sensitive and mission-critical applications such as remote surgery operations and ICU telemetry systems.

To effectively meet these shortcomings with ICP-ABE, the proposed OCP-ABE introduces several key improvements over ICP-ABE for securing MQTT-based data transmission in healthcare 4.0 IoT environments. Although ICP-ABE improves performance efficiency by separating attribute auditing from secret key generation, shrinking computational overhead, and mitigating slowDoS attacks, OCP-ABE is proposed to optimise access control and authentication further to ensure better confidentiality. So, OCP-ABE

introduces an indirect revocation scheme and self-key update mechanism, enhancing traceability and preventing unauthorised access. The indirect self-key revocation mechanism in OCP-ABE guarantees that compromised or outdated keys can be dynamically invalidated without requiring continuous broker-side intervention, shrinking latency and accomplishing scalability in environments with varying numbers of devices. OCP-ABE also implements confusion and diffusion-improved properties of ciphertexts for strengthening privacy, shrinks the risk of attribute-based inference, and protects against statistical attacks, even when vulnerability is minimised with correlated attribute sets. In addition, OCP-ABE includes the flagging of the secure MQTT session directly into the TCP-level communication to diminish connection overhead, neglecting the need for frequent reestablishment of the MQTT session and avoiding the impact of connection-based attacks such as session hijacking and SlowDoS. Cryptographically, OCP-ABE employs the Fast-PRESENT algorithm, which enables parallel execution of S-boxes and decremental bit-shift operations for faster encryption, surpassing the exploitation of ICP-ABE of the PRESENT algorithm. Although ICP-ABE successfully reduces the computational burden and improves network security, it lacks the enhanced cryptographic optimisation and access control refinements provided by OCP-ABE. These advancements make OCP-ABE more suitable for the high-efficiency, low-latency, and robust security demands of Healthcare 4.0 IoT scenarios.

The primary objective of OCP-ABE is to ensure optimised access control, authentication, and confidentiality for MQTT-based data transmission in IoMT. The proposed scheme is validated through simulations and demonstrates its strong security measures using Avalanche Effect (AE), Correlation Coefficient (CC), and the semi-equivalent key test using Hamming Weight (HW). The security strength is proved through the high AE and HW values and the low CC values of the proposed scheme. The simulation assessed the performance of Base-MQTT, OCP-ABE, Pre-AugPAKE [269], SPECK [264], and SIMON [263]. The results demonstrate the superiority of the proposed scheme in terms of performance.

8.1 Introduction for OCP-ABE

The Internet of Medical Things (IoMT) offers digital healthcare systems that allow people to experience quality health treatment in the comfort of their homes [471,472]. Ensuring the security, efficiency, and lightweight nature of these systems for the timely prediction and treatment of critical diseases can save millions of lives while reducing the need for hospital visits. COVID-19 has disrupted various technologies and created significant opportunities for developing IoMT. In IoMT, patients and organisations often need to share and store sensitive data [472]. Intelligent healthcare systems must select appropriate data communication protocols that provide reliability and security. The MQTT protocol is one of the standardised protocols used in IoT for effective communication [473]. It defines a subscriber-publisher interaction facilitated by an MQTT broker. However, communication security and data integrity remain significant concerns. IoMT is vulnerable to different security attacks, such as eavesdropping, hijacking, DoS, MitM, message tampering, device cloning, and denial of power attacks. Ensuring security in this environment is tedious due to the specific resource limitations of IoMT: low battery capacity, less memory, and minimum processing power. However, securing IoMT is crucial to ensure timely care for patients.

Currently, several researchers are working to improve the protocol to ensure the confidentiality of data transmission. A promising technique for securing MQTT communication is Ciphertext-Policy Attribute-Based Encryption (CP-ABE). For example, in the context of IoMT data sharing, a patient may want to share medical data with nurses and physicians in the field of ophthalmology without knowing their specific identities. Communication security is crucial in ensuring the confidentiality and privacy of data exchanged in IoMT. A potential solution to achieve data security is to encrypt sensitive data with a well-defined access policy. Several security schemes have been developed using CP-ABE, although some security issues persist [474]. In traditional CP-ABE, a node's secret key is associated with its corresponding attributes, and the ciphertext is associated with an access policy. The receiving nodes can decrypt the ciphertext only when the access policy matches the defined one. In the IoMT environment, patients can encrypt their health

data using an access policy such as “Doctor and Ophthalmologist” OR “Nurse and Ophthalmologist” and upload the resulting ciphertext to the MQTT broker. This approach allows only nurses and doctors in the field of ophthalmology to access patient data.

Despite advances in research on MQTT security, several security challenges still require further investigation. These challenges include the length of the attribute-based key and its ciphertext [475, 476], the presence of correlated keys due to similar attribute sets [254, 477], the absence of a lightweight revocation scheme [280, 354, 358, 466, 478], and the limited resources available to handle MQTT security features [479, 480]. Additionally, while lightweight, the PRESENT algorithm [481] has limitations due to the high complexity of its S-box and key scheduling algorithms compared to other lightweight block cypher schemes [482]. The MQTT broker generally serves as an intermediary between publishers and subscribers to facilitate communication. Establishing a secure and efficient security method is crucial to ensure safe communication between all parties involved.

To address these concerns, the Optimised CP-ABE (OCP-ABE) and Fast-PRESENT schemes are proposed to enhance security. Although existing methods often showcase their lightweight properties through simulations, focusing on execution time and energy consumption, they usually lack a thorough analysis of their security strength. In this proposed work, the Cooja simulator [483] demonstrates the security strength of OCP-ABE while running the MQTT protocol. In particular, the potential validation of the security strength reveals the superiority of the proposed method using the AE, CC, and semi-equivalent key tests using HW.

8.2 Problem Formulation and Proposed OCP-ABE Scheme

This section defines the OCP-ABE application scenarios and explains the system’s entity, attributes, and access structures. It also describes the system and the attacker models. A remote heart rate monitoring system is considered a use case to explain the proposed OCP-ABE design. In the IoMT environment, the Data User (*DU*) explores the medical data through various medical devices, such as heartbeat rate monitoring. For example,

when a heart rate monitor measures and records a heart rate on the server for later usage, users such as doctors and nurses with specific attributes can access the data from the server. Using the fundamental CP-ABE in IoMT, a patient can encrypt medical data with the access policy such as “Doctor” AND “Ophthalmologist” OR “Nurse” AND “Ophthalmologist” and upload the ciphertext to the MQTT Broker (*MB*); then, only nurses and doctors in ophthalmology can access the medical data through subscribing the topic “heartbeat rate”.

The proposed scheme takes a specific IoMT application scenario, such as heartbeat rate monitoring, as an example to deploy and demonstrate the efficiency of the OCP-ABE scheme, as shown in Figure 8.1. In this scenario, heartbeat rate monitoring sensors are attached to the patient side and are referred to as MQTT publishers. These sensors or wearables are connected to the *MB* and frequently update the heartbeat rate of patients in critical stages. Two cases occurred, emergency and non-emergency. The heartbeat rate is abnormal during emergencies, and immediate care is needed to prevent patients from experiencing life-threatening events. Patients are continuously monitored during non-emergency situations, and their heartbeat rate is normal. Healthcare providers can obtain specific heartbeat information through topic subscriptions.

Data owners (*DO*), such as patients, doctors and medical institutions, apply the security scheme to their data and upload it to the *MB*. The *MB* shares the data only when the entities meet the access structure. Physicians who operate PCR and scanning devices want to obtain medical data with specific attributes. If the attributes of a doctor meet the access structure requirements, the Doctor can receive encrypted data about a particular patient.

The intelligent IoMT App enables data users to enter their username and password to log into the system, as shown in Figure 8.1. When *DO* has authenticated a user, the login page is redirected to the system dashboard page. The login credentials are the blind key (B_k) and the attribute A . The dashboard page shows the current status of all intelligent IoMT devices obtained from *MB*. *MB* establishes data communication between smart *DU* and *DO* by publishing/subscribing request messages. If *DU* subscribes to one of the *MB* topics, the *MB* will verify the user credentials.

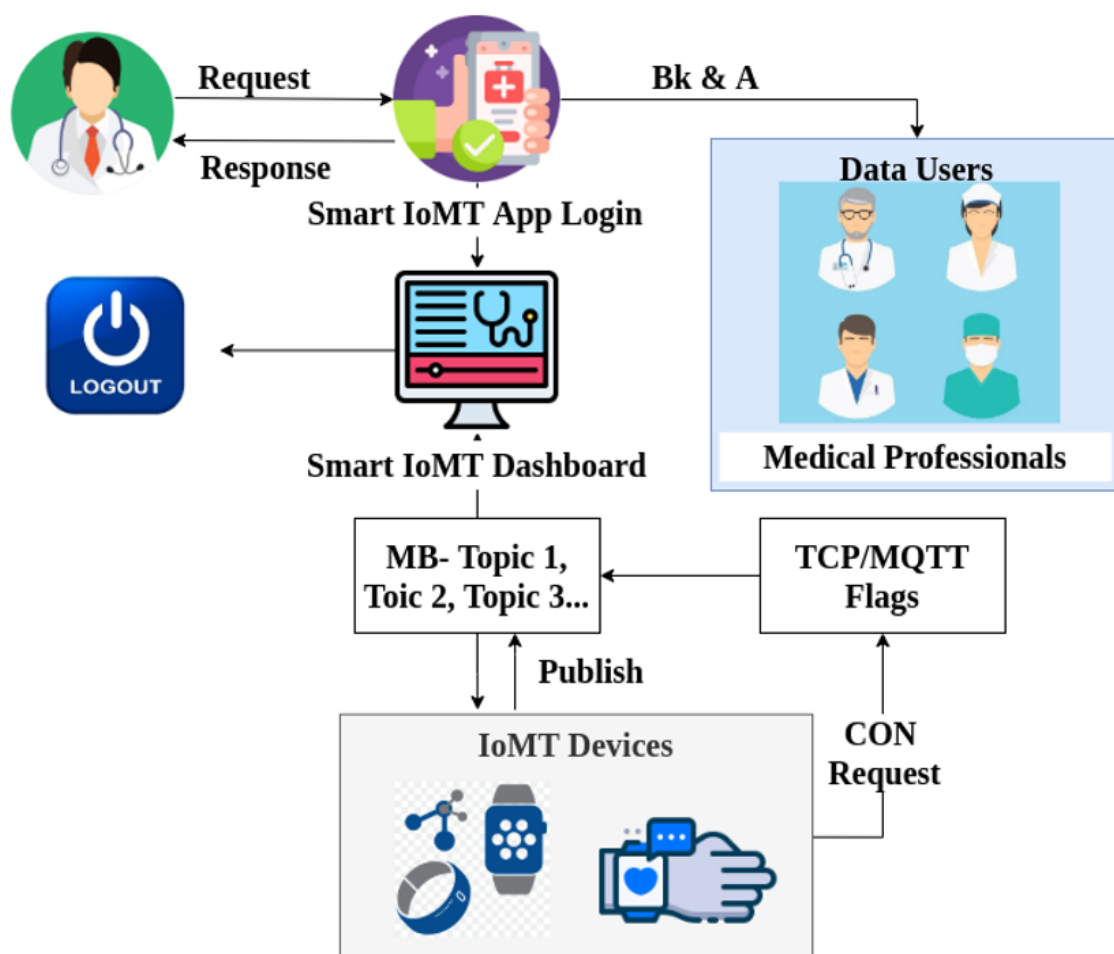


Figure 8.1: Smart IoMT Application Scenario

There are two database-authentication situations:

- 1) If the user is authenticated, she/he can access the system and log in successfully.
- 2) If the device credentials provided by DU have not been validated, the subscription is not allowed. Furthermore, the proposed system allows DU to connect with the application by specifying the device ID , B_k and its attributes in the TCP certificate. When a DU sends a subscription or publishes a request to an MB , the MB is responsible for validating the device by authenticating the TCP certificate. Access is granted successfully if the device is authenticated without needing MQTT connection establishment messages. If it is not authenticated, access is denied. Reduce MQTT connection messages for invalid user access.

8.2.1 System Entities

The proposed work comprises the network, security, and revocation model of N IoMT nodes. In the proposed MQTT protocol using OCP-ABE, there are three entities involved in IoMT: the MQTT Broker, the Data User, and the Data Owner. The DU are doctors, nurses, researchers, and others. Patient details are stored in the MQTT broker. Table 8.1 lists the notes used in the work.

- **MQTT Broker** (MB): We consider the MQTT broker to be a completely trusted entity because the MQTT broker is fully visible to the encrypted and decrypted data as a valid carrier. The main functionalities of the MQTT broker are to generate and store the Secret Key (S_k), Blind Key (B_k), Time Factor (T_F), and system parameters of the IoMT nodes. Moreover, the MQTT broker participates in user revocation using the time factor. Using such a factor, the MQTT broker is responsible for disabling the revoked IoMT nodes from accessing the data by notifying the IoMT nodes to perform self-key revocation, except for the revoked IoMT nodes.
- **Data User** (DU): DU has a series of attributes (A) and only accesses exciting data in the system, and the number of DU belongs to N . Generally, the IoMT nodes N are divided into the revoked IoMT nodes R_N and the UnRevoked IoMT nodes (UR_N). If the DU is not on the list of UR_N or $DU \notin UR_N$ and attributes of the DU

Table 8.1: List of used Notations.

Notation	Explanation
DU	Data user
DO	Data owner
mDU	Malicious DU
At	Attacker
MB	MQTT broker
Top	Topic
N	Number of nodes
$\mathbb{A} = \{A_1, A_2, \dots, A_n\}$	Set of attributes
R_N	Revoked IoMT nodes
UR_N	Unrevoked IoMT nodes
AP	Access policy
P	n-byte string policy
S	n-byte string
S_k	Secret key
B_t	Blind token
B_k	Blind key
T_F	Time factor
Ct	Ciphertext
Pt	Plaintext
Ms	Message
MF	MQTT flag
K	Key register
$Non-RL$	Non-Revoked IoMT node list
ALK	Attribute Length using blind Key
Bt_{Si}	Blind tokens created for specific attributes of devices
$(A_{pi} \text{ and } A_{pj})$	Correlation attribute set
$ Ct_i - Pt_i $	Flipped ciphertext bits due to the changes in plaintext
Ct_i	i^{th} value from the first set
Pt_i	i^{th} value from the second set

(A_{DU}) meet the Access policy (AP) in the ciphertext; the DU is eligible to decrypt the ciphertext. The DU is also named a subscriber.

- **Data Owner (DO):** DO , the owner of the data transmitted in the system or the data publisher and the number of DO belongs to N . DO is responsible for deciding the access policy. For instance, the Doctor can access the patient's health details, not their data. To ensure the security of the data, DO transmits and stores encrypted data on the MQTT broker. The ciphertext includes the access structure decided by DO .

8.2.2 Access Structure and Attributes

The proposed work defines the access control policy using attributes and a set of attributes $\mathbb{A} = \{A_1, A_2, \dots, A_n\}$. For example, patient details include name, ID, disease, treatment, address, phone number, and age. An attribute set for a particular IoMT node i is represented as $S_i \in \mathbb{A}$. Moreover, S denotes the n-byte string, denoted as $S_1, S_2, S_3, \dots, S_n$. The definitions for those strings are given below.

$$S = \begin{cases} S_i = 0, & A_i \notin S \\ S_i = 1, & A_i \in S \end{cases} \quad (8.1)$$

Considering a value of n equals 5, which means that the number of bits is 5, and assume IoMT node attribute string is denoted as $S = S_1 S_2 S_3 S_4 S_5 = 11011$. It denotes that the first and last two attributes (A_1, A_2, A_4, A_5) represent a string S . Access policy is denoted as AP , an n-byte string of access policy ($p_1 p_2 p_3 \dots, p_n$). The definition is given below.

$$AP = \begin{cases} p_i = 0, & A_i \notin P \\ p_i = 1, & A_i \in P \end{cases} \quad (8.2)$$

Like the attribute, considering that a value of n is equal to 5, assume that its specified access policy is denoted as $AP = p_1 p_2 p_3 p_4 p_5 = 10111$. The first and last three attributes (A_1, A_3, A_4, A_5) are included in the access control policy AP . Moreover, $|S|$ and $|AP|$

represent the number of attributes in S and AP , respectively, if $AP \in S$ means that the set of attributes S satisfies the access control policy AP .

8.2.3 System and Attacker Model

The proposed model improves the OCP-ABE strategy to support a high communication rate, an efficient revocation mechanism, and robust traceability. This approach is accomplished by indirectly reducing the size of the ciphertext, offering effective key revocation, and ensuring reliable tracking of malicious activities within the resource-limited IoMT environment.

Definition 1 ((Setup ID and \mathbb{A}) and Key Generation): The setup algorithm installs IoT devices with Identity, ID and the list of attributes \mathbb{A} or S byte strings. Moreover, MB generates the secret key (S_k) using the Fast-PRESENT encryption scheme and the blind token (B_t) for each attribute $A_n \in \mathbb{A}$. Hence, blind tokens are also created for specific attributes of devices Bt_{Si} according to a particular Data User DU access control policy. The DU shares the Bt_{Si} with the DO in an encrypted format using a secret key (S_k). This process is executed once, and if any changes occur in the attribute set of a particular DO , it requests to send the \mathbb{A} for accessing the data from DO . Using ID and Bt_{Si} , a blind key B_k is generated on both sides DO and DU by performing the XOR operation.

Function 1 (Key Generation and Management): Initialises IoT devices with identity and attributes to generate a secret key using Fast-PRESENT. Create unique blind keys for each attribute instead of a single key for a set of attributes, improving security against attribute-related attacks.

$$B_K = Bt_{S1} \oplus Bt_{S2} \cdots \oplus Bt_{Si} \oplus ID \quad (8.3)$$

Definition 2 (Encrypt (Ms ; \mathbb{A} ; B_k)): This algorithm considers the message (Ms), and an access tree structure over the attributes \mathbb{A} . It encrypts the Ms using B_K and results in ciphertext Ct , and DO takes hashing for encrypted data using B_k and attribute length. This process is called Attribute Length Key (ALK). This process is also executed

in the DU while subscribing to the particular topic from the MQTT server. ALK helps maintain the consistency and integrity of encrypted data.

Function 2 (Encryption Function): The data owner secures the data before uploading it to the MQTT broker for access-controlled sharing.

Definition 3 (Revoke ($ID, Non-RL \cup DU, T_F$)): The revocation algorithm is implemented by MB . By updating the list of non-revoked IoMT nodes, non-RL with Time Factor (T_F), the non-revoked IoMT nodes update their secret key by performing an XOR operation with T_F . Due to the usage of the updated blind key, the revoked IoMT nodes cannot decrypt the ciphertext received from DO after revocation. In particular, the decryption rights of nodes in Non-RL are not affected.

Function 3 (Revocation Function): Manages access revocation using indirect key updates. Legitimate IoMT nodes update their keys by XOR with the time factor, preventing revoked users from accessing updated data without resending new keys.

Definition 4 (Decrypt($Ct; B_k$)): The decryption algorithm in DU takes Ct as input, the blind key new B_k of a client. It outputs the requested content Ms only if the request satisfies the new B_k . Otherwise, the output is \perp .

Function 4 (Decryption Function): Ensures that only authorised users with matching attributes can access the decrypted content.

Traceability and Masquerade Attacker Model

Essential MQTT communications have several potential vulnerabilities. Using a robust cryptographic mechanism, an attacker cannot read encrypted messages using the publish-subscribe model. However, traceability and masquerade attacks analyse the messages sent to MB and try to identify secret keys to know the details of the patients over IoMT. It is demonstrated in Figure 8.2. From the revoked DU , the masquerade attack tries to get the blind key directly. From the normal DUs , the traceability attackers try to obtain the secret key, since, to receive the blind key, the private key is essential.

- A traceability attacker collects several requests from data users to identify the secret key.

- The masquerade malicious IoMT nodes send blind keys for a particular access structure after it is left from the group and degrade the system security.

The proposed OCP-ABE-based MQTT scheme is designed to avoid such attacks.

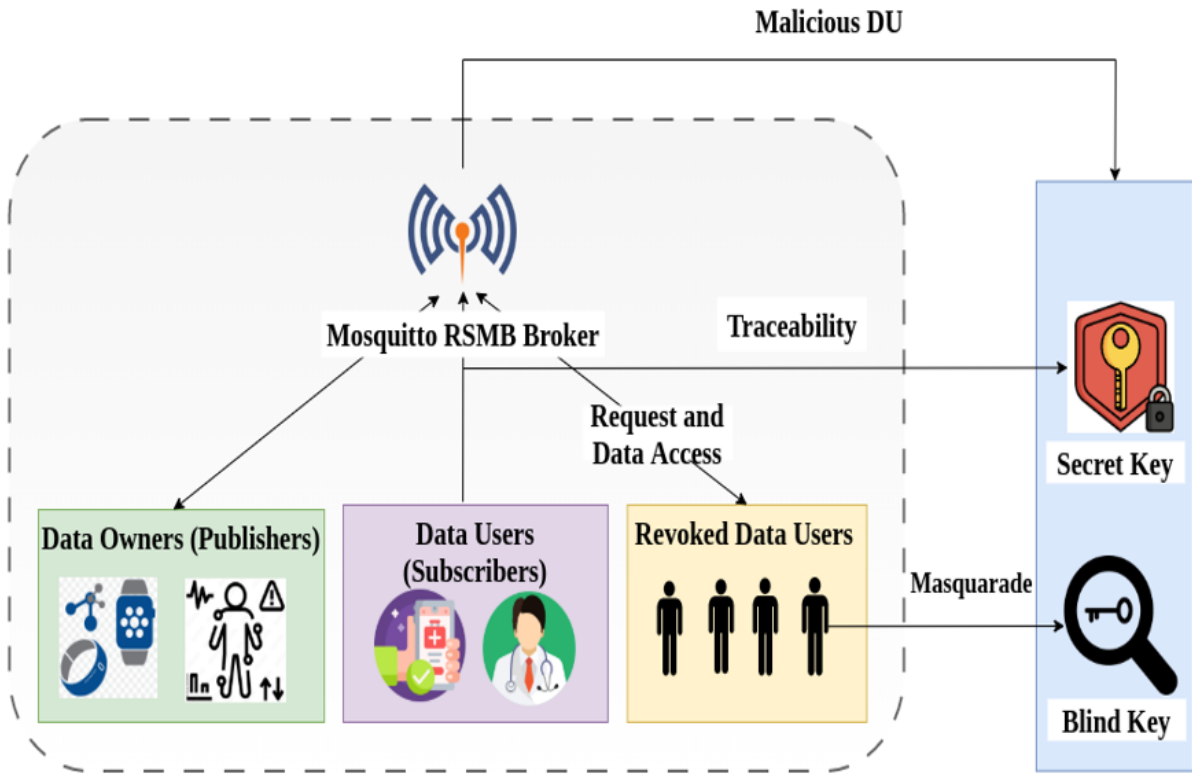


Figure 8.2: MQTT Attacks

Initialisation: At the beginning of the attack scenario, an attacker first declares the access policy AP of an IoMT node and is left in the group in the Masquerade attack. After that, it sends its previous secret information to other malicious DU (mDU). The traceability attack attempts to identify the private key by determining the similarity between the ciphertexts.

Setup: After generating the security parameter Bt_{Si} , mDU executes algorithm Setup to obtain system parameters. Then, mDU shares public parameters with At .

Attack: At this stage, At sends a query request to MB , and it attaches its ID as subscriber or DU for DO . The MB sends the topic requested by the At , which degrades the privacy of the system.

The MQTT security scheme is insecure if an attacker At wins the above game. To

overcome such issues, the proposed work designs an optimised CP-ABE scheme without increasing the computational complexity.

8.2.4 Overview of the OCP-ABE

The main components of the proposed OCP-ABE scheme are Fast-PRESENT-enabled encryption, a topic and time factor-based indirect key revocation, and secure data publishing/subscribing, as shown in Figure 8.3. In the initial mechanism, Fast-PRESENT efficiently verifies identity and generates blind keys based on the attributes for secure encryption and decryption functionalities. Hence, parallel execution of S-boxes increases the PRESENT speed without affecting performance. Secondly, the indirect revocation of self-keys using topic and time factors effectively manages resource scarcity and strengthens security. Finally, secure publishing and subscription-based MQTT communication are accomplished in IoMT.

Fast-PRESENT Algorithm: It is an optimised version of the conventional PRESENT algorithm that enhances efficiency by running the S-boxes in parallel. Thus, it is well-suited for resource-limited environments like IoMT devices. The Fast-PRESENT algorithm modifies the conventional PRESENT by executing all 16 S-box operations simultaneously. It assigns each 4-bit block of the 64-bit state to a separate processing unit, allowing the S-box executions to occur in parallel. This approach notably reduces the time needed for the substitution layer in every round of encryption. Moreover, this key innovation of parallel S-Box execution significantly shrinks the time required for encryption while maintaining greater security. This enhancement makes it particularly well-suited for IoMT applications with low computational costs and strong data security.

Topic and Time Factor-based Indirect Key Revocation: The indirect key revocation mechanism consolidates topic-based access control and time factor constraints to enhance security and manages the length of the key in a Fast-PRESENT-integrated OCP-ABE scheme. This strategy ensures effective self-key revocation while considering and maintaining the lightweight nature of IoMT environments. To overcome the high computational cost of direct revocation, the authorities of indirect revocation only allow

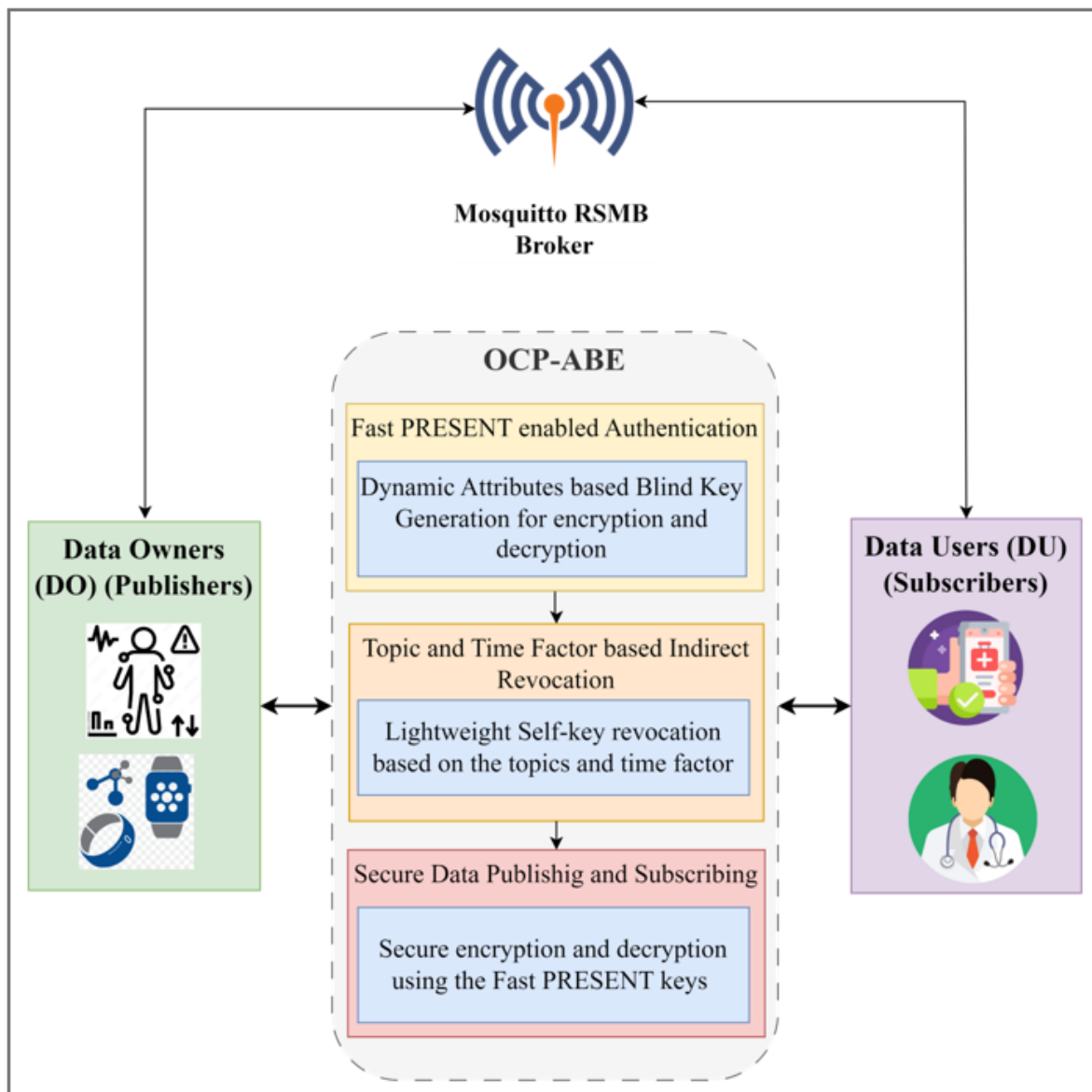


Figure 8.3: Block Diagram of Proposed Methodology

legitimate IoMT nodes to update their decryption keys periodically. Increase system overhead while detecting all malicious IoMT nodes in the network. To further enhance security with minimum cost, the proposed OCP-ABE strategy incorporates a time factor into the self-key revocation process of the previously proposed ICP-ABE. Upon receiving a server notification, only legitimate IoMT clients autonomously perform self-key updates based on the shared time factor, ensuring timely and cost-reducing key management against attacks. Thus, it only allows the secret key to be generated by legitimate users and improves the scheme's security without increasing the computational complexity.

Secure Data Publishing/Subscribing: A significant problem with exploiting a single-blind key for an attribute set is that it increases the risk of key tracing by a malicious IoMT client possessing a correlated attribute set. To effectively address this, the proposed OCP-ABE scheme individually assigns a unique blind key to each attribute rather than exploiting a single blind key for the entire set of attributes. In addition, enhanced confusion and diffusion properties assist in obscuring the relationship between plaintext and ciphertext, making it more complex for attackers to infer the original information or deduce the keys for encryption, even if they obtained the attribute set. Thus, it effectively mitigates the impact of vulnerability. In addition, stronger encryption and decryption properties ensure that only authorised users with correct decryption keys can access the information, indirectly preserving confidentiality in IoMT communication. Moreover, using identity and blind keys during secret key generation enhances security by identifying and revoking malicious users in the publishing/subscribing model. When a group of subscribers makes minimal changes in their attribute set, those IoMT nodes update their keys according to the modified attribute set and notify the server without reinitialising the process from scratch. Improves the security and performance of the existing ICP-ABE scheme. To reduce the control overhead, the OCP-ABE introduces the MQTT flag in the TCP messages and uses the TCP messages as MQTT connection establishment messages without losing device authentication.

The following Figure 8.4 shows the key sharing and revocation operations of the OCP-ABE scheme. Figure 8.5 shows the encryption and decryption communication of the proposed scheme.

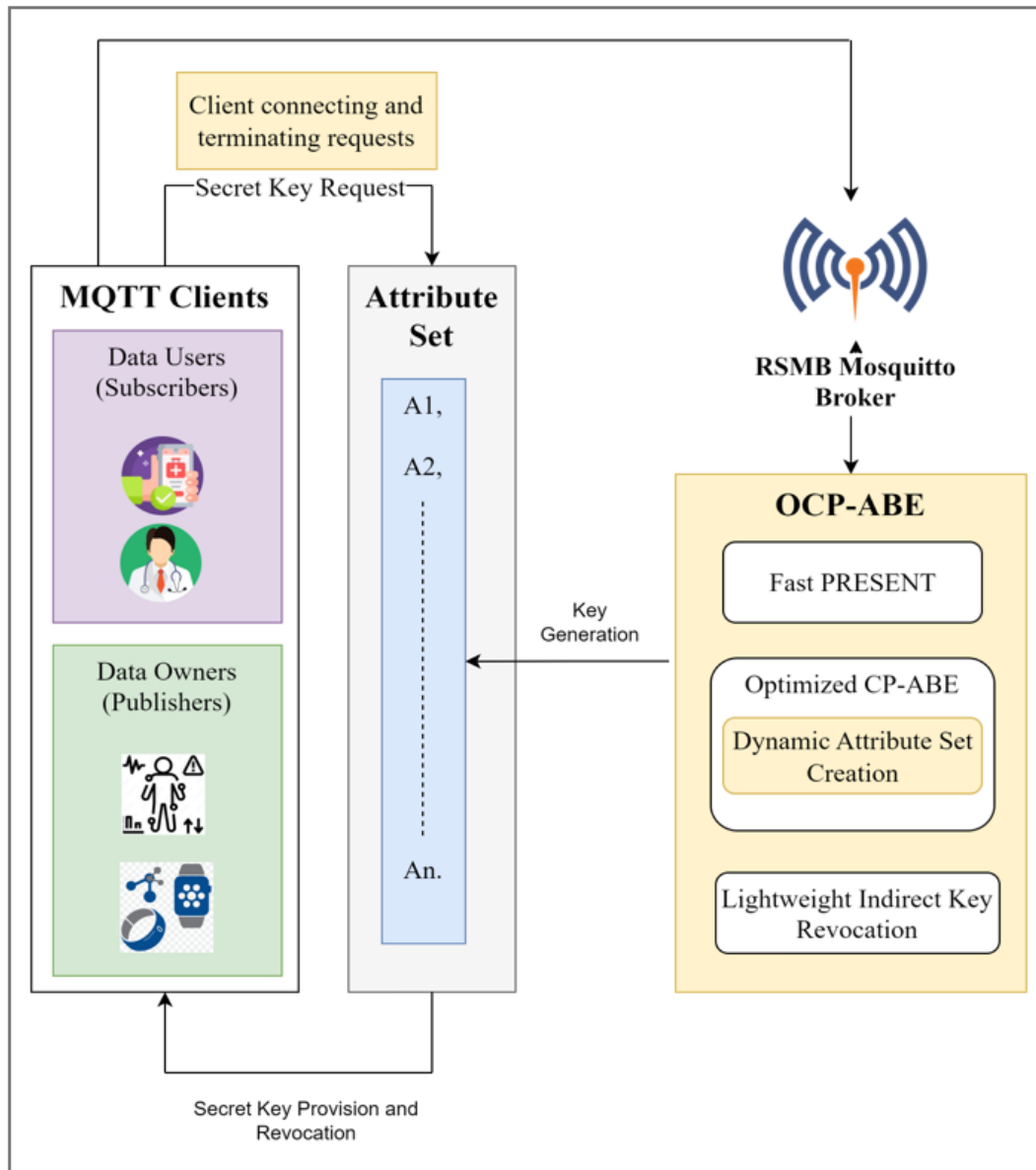


Figure 8.4: Key Sharing and Revocation Operations of OCP-ABE Scheme

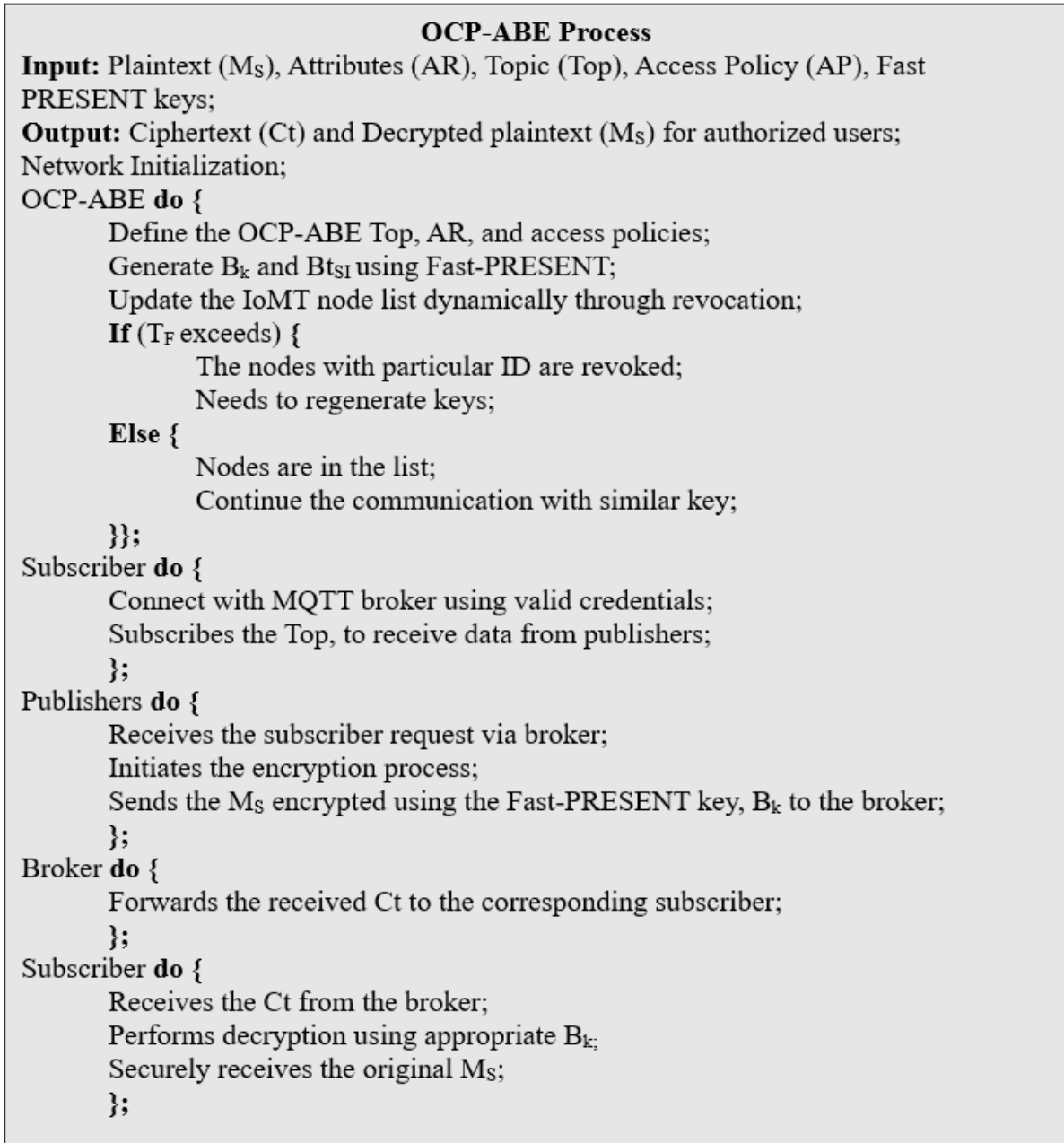


Figure 8.5: Process of OCP-ABE

8.3 Optimised CP-ABE Scheme using Attribute Specific Blind Key Generation and MQTT Flag

Most ABE schemes allow the MQTT server to generate secret keys with knowledge of each user's attribute information, increasing the ciphertext length. Malicious IoMT nodes can easily trace it due to the common key for a set of attributes. With the correlated attribute set (A_{P_i} and A_{P_j}), the secret keys are also correlated with each other. To avoid such a problem, the proposed scheme provides a separate B_t to each attribute instead of providing the same B_t for an attribute set. The MB chooses several random B_t for each attribute $A_t \in \mathbb{A}$. The algorithm randomly generates $B_{t_{S_1}}, B_{t_{S_2}}, \dots, B_{t_{S_n}}$ for each attribute. Moreover, the proposed work exploits XOR functions between $B_{t_{S_i}}$ and ID for access trees and ciphertext generation. The blind key B_k is published as encrypted $B_k = \{AP, ID\}$.

$$E.B_K = Enc_{S_k}(B_{t_{S_i}} \oplus ID) \quad (8.4)$$

Like ICP-ABE, OCP-ABE also separates the two functions of attribute auditing and key extraction functions along with multiple B_t values. Both the DO and DU are installed with the knowledge of the blind token for each attribute A in its corresponding attribute set, \mathbb{A} . The MQTT broker is only allowed to know the blind token, but it does not know the corresponding attributes of those tokens. Moreover, using small-size keys limits the length of the ciphertext and reduces the computation and communication costs. The MB verifies the blind key according to the attributes, and its identity returns encrypted data with a signature to the DU . Even in the same group \mathbb{A} , every IoMT node has a different B_k due to the incorporation of ID. Data encryption is performed using a blind key and is estimated using the formula 8.4.

The main aim of the proposed scheme is to develop a lightweight authentication solution to the MQTT security problem. The proposed scheme introduces a new flag (MF) in TCP packets, and MF contains the identity and encrypted B_k and the corresponding attribute set using a secret key. The MQTT uses TCP transmission sessions. The idea is

to sandwich the MQTT connection messages within TCP flags. MF includes everything related to monitoring communication security, including node identity, B_k , and AP . If the credentials are satisfied, the MB responds to the DU without initialising separate MQTT connection messages. Otherwise, the TCP message is discarded. When the user credentials are satisfied, the TCP and MQTT connection messages work together to log into the IoMT application. Table 8.2 shows the MF in the TCP message.

Table 8.2: MF Flag in TCP Message.

Application Layer (OCP-ABE MQTT)	
TCP	TCP Flags
	TMF
Network and Lower Layer	

8.3.1 Fast-PRESENT Algorithm-Based Encryption

The blind token is evidence that the IoMT nodes have the exact attributes. Moreover, the blind token never reveals attributes, IoMT node identity, and data content information to others. In particular, the blind token is initially shared using the secret key generated with the help of the Fast-PRESENT algorithm. The specific processes of the proposed scheme are explained in the following to show the role of the PRESENT algorithm.

1. The DU and DO store their attributes and the relevant blind key to the attribute set. The DU appends MQTT Flag, MF in TCP message and uses the same transport layer message as a request to DO to receive the content of interest.
2. The DO is responsible for auditing the blind key and its corresponding attributes. Suppose $AP \in S$ means that the attribute set S satisfies the access control policy AP . If it meets the attributes policy, DO shares the blind key with the MB . PRESENT encryption is used to send the blind key securely. Generate the secret key by running 31 rounds of XOR operation with a key size of 80 bits, an improved S-Box, and a new KSA. Figures 8.6 and 8.7 illustrate the operation and process of

the Fast-PRESENT algorithm. The Fast-PRESENT has the following functions.

- (a) AddRoundKey: It appends a round key to each state.
 - (b) PS-Box layer: Replaces each byte in an array with a sub-byte. The type of PS-Box is 4-bit to 4-bit, and it is executed in parallel instead of executing them individually.
 - (c) Improved KSA: It helps provide random and independent round keys irrespective of the secret key in every round.
 - (d) pLayer: It permutes each state into the predefined position.
3. After sending the interesting content, *DO* and *DU* change the blind key separately by executing the XOR operation between the previous blind key, the blind token, and the topic previously accessed.

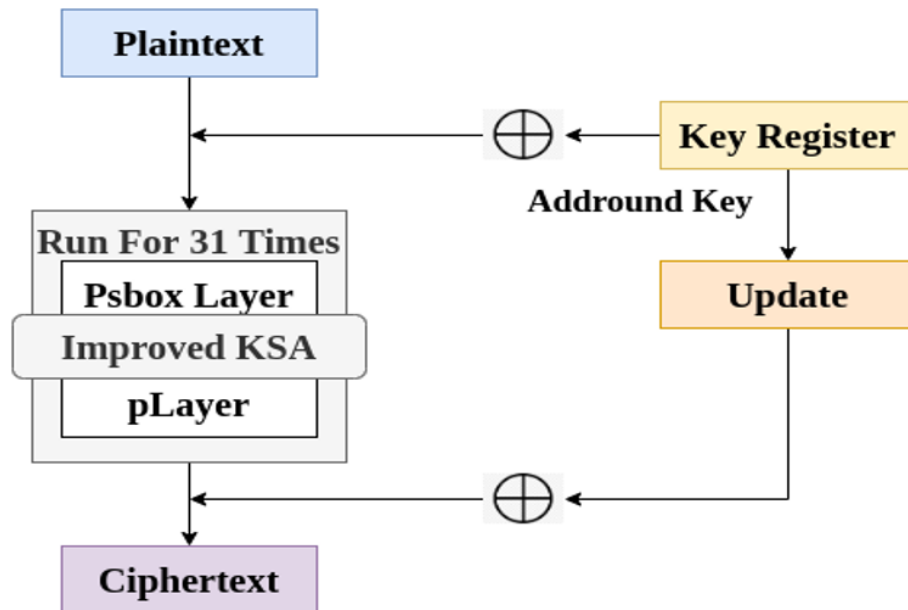


Figure 8.6: Operations in Fast-PRESENT Algorithm

The SubBytes uses a lookup table called the S-box with 16×16 matrix byte values. This 16×16 matrix is designed using the Rijndael S-box. However, the PRESENT algorithm explores a single S-box, which may increase the possibility of conflict in accessing a given table. If the table is currently accessed for a byte, it shares a busy message with

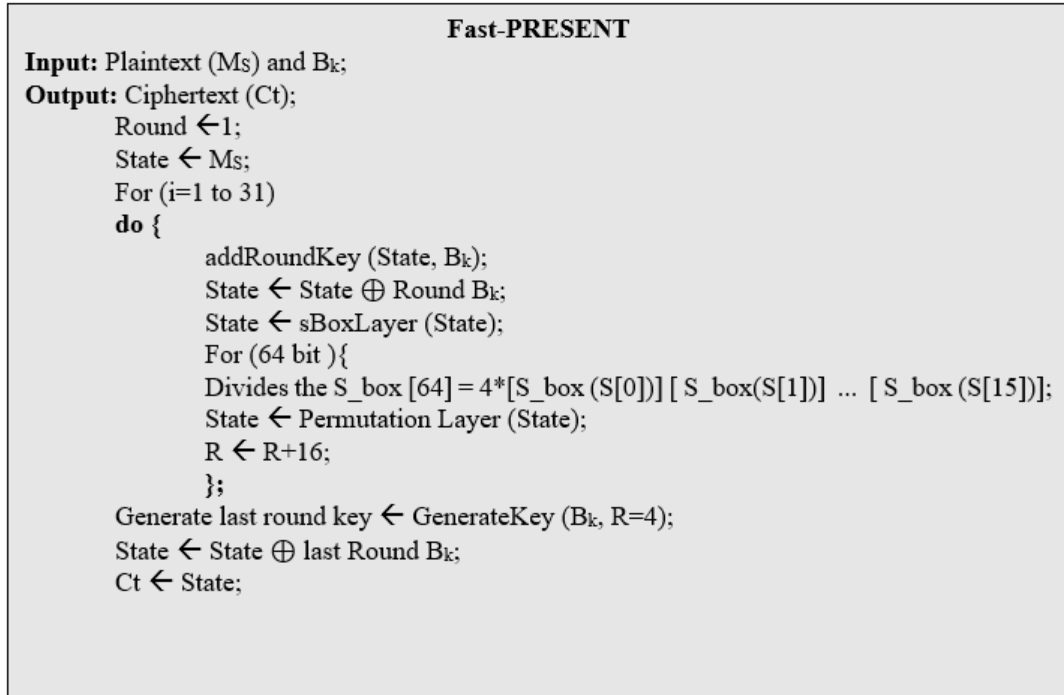


Figure 8.7: Process in Fast-PRESENT Algorithm

others for each table, so it cannot be accessed in the current cycle. Increase time complexity and reduce the performance of the PRESENT algorithm. Thus, the proposed scheme introduces a parallel operation instead of a serial operation.

The proposed scheme creates 2×2 tables and organises them into groups to enable parallel operation. Each group includes 16 small S-boxes. Each group contains 16 small s-boxes. The proposed scheme implements parallel operation in SubBytes, which is equal to the group size. If the group size is 16, 16 bytes can be processed simultaneously. The advantages of parallel SubByte operation are simplifying table indexing and reducing the computational complexity of the PRESENT algorithm, as shown below. Box 1 is represented in Figure 8.8.

- (1) A and A1: The first two bits in the left-most position denote a group.
- (2) B and B1: The next left-most bits denote a row within a group.
- (3) C and C1: The next left-most bits denote a column within a group.
- (4) D and D1: The two right-most bits denote the index for getting a substitution value.

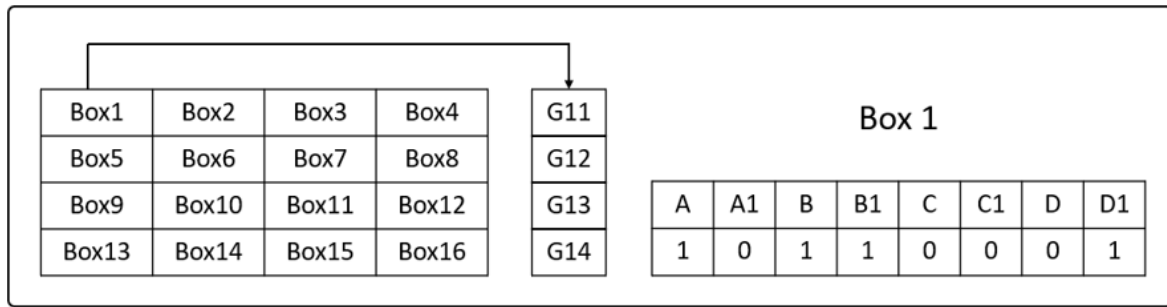


Figure 8.8: Parallel SubByte Operation in Fast n PRESENT

New Key Scheduling Algorithm in Fast-PRESENT Scheme

Another problem associated with the PRESENT algorithm is the simple KSA algorithm. Apply left shift, 8-bit substitution, and XOR operation with the least significant bit of the round counter. Per the S-box and XOR in PRESENT, the keys can be modified only with a few bits in each round. Increases the possibility of estimating statistical dependence between round keys. To solve such issues, Fast-PRESENT offers a new key scheduling algorithm.

The 128-bit secret key is stored in the key register, K , and the left 64-bit key in K is considered the round key K_1 . After that, the following steps are repeated to generate 31 round keys.

Step 1. Store the secret key to a $K = K_{127}, K_{126}, \dots, K_0$.

Step 2. Split the key K into four blocks and each K should contain 32 bits:

- (a) $x1 = [K_{127}, \dots, K_{96}] \Rightarrow J1 = PS - box[x1]$.
- (b) $x2 = [K_{95}, \dots, K_{64}] \Rightarrow J2 = PS - box[x2]$.
- (c) $x3 = [K_{63}, \dots, K_{32}] \Rightarrow J3 = reverse(x3 \oplus J2)$.
- (d) $x4 = [K_{31}, \dots, K_0] \Rightarrow J4 = reverse(x4 \oplus J1)$.

Step 3. Concatenate $K = J1 J2 J3 J4$.

Step 4. Apply decremental bits shift, such as 64, 32, and 16 bits to the left of

- (a) $K = [K_{64} K_{63} \dots K_{127} K_{126} \dots K_{67} K_{66} K_{65}]$.
- (b) $K = [K_{33} K_{32} \dots K_{64} K_{63} \dots K_{31} K_{32}]$.

$$(c) \ K = [K_{17} \ K_{18} \ \cdots \ K_{33} \ K_{32} \ \cdots \ K_{14} \ K_{15} \ K_{16}].$$

Step 5. XOR 5 bits of K with the least significant bit of round counter I , where $K[K_{17} \ K_{18} \ K_{19} \ K_{20} \ K_{21}] = [K_{17} \ K_{18} \ K_{19} \ K_{20} \ K_{21}] \oplus i$. The left-most 64-bit round key, K_i from K , is used as the key in the AddRoundKey function. Steps 1 to 5 are repeated for 31 round keys, effectively improving the security of the Fast-PRESENT algorithm without increasing the computational cost.

Dynamic Attribute-Based Signature Scheme

After encrypting the data, DO takes hashing for Attribute Length using blind key, ALK . The ALK is used to provision hashing for encrypted data. The DU executes the same process while subscribing to a particular topic on the MQTT server. The next component is the self-key agreement at the DO and DU . The MB stores the encrypted published data with the index of the encrypted topic. While accessing a particular topic, the DU IoMT nodes send a request message and the encrypted topic using a blind key. The MB provides a message to the corresponding DU by matching the encrypted topic in the index. The DU explores the generated ALK to decrypt the received message from the MB .

Finally, the ALK is updated using the previously accessed topic and the current blind key. It avoids secure key sharing among devices and ensures the security of the system. However, it is suitable for static attribute sets. However, IoMT nodes are dynamic in some environments, and their decryption rights may be removed at any time. Therefore, traditional fixed ABE schemes are not suitable for a dynamic environment. It is essential to design a dynamic ABE scheme to fulfil this requirement. When a DO wants to change its attribute set partially, it updates its S values to the corresponding DUs by encrypting S using S_k . They estimate the blind key using the new attribute set without reinitialising the process from scratch. Improves both the security and the performance of existing CP-ABE schemes.

8.3.2 Lightweight Indirect Revocation Scheme

In the OCP-ABE system, malicious IoMT nodes that share their decryption keys with others must be removed from the system. Hence, the user revocation mechanism should be designed for the OCP-ABE system. Based on existing techniques, the IoMT node revocation system is divided into direct and indirect revocation schemes.

Direct Revocation: The *MB* shares the revocation list with all the IoMT nodes in the network. The Data owner encrypts the data with a specified revocation list, and only the non-revoked IoMT nodes can encrypt the corresponding ciphertext. However, direct revocation results in high complexity since each *DO* needs to keep a revocation IoMT node identity list revocation and breaches the anonymity of the IoMT node in the ABE system.

Indirect Revocation: The *MB* informs only the unrevoked users to update their decryption keys periodically. Revocation users do not receive update requests and cannot decrypt the new ciphertexts.

The existing work provides a self-key revocation scheme. OCP-ABE introduces the Time Factor (TF), TF-based key update, by combining the advantages of the self-key revocation scheme with the indirect revocation scheme.

Setup : The *DO* executes the setup algorithm and passes the current *TF* value to the IoMT nodes in the non-RL list.

Key Updation: The IoMT nodes in non-RL output a new B_k using *TF* and previously used B_k .

$$new B_k = B_k \oplus TF \quad (8.5)$$

The IoMT nodes in the non-RL are unaware of the key update and are disabled from accessing the topic illegally using the proposed OCP-ABE scheme. In addition, according to the proposed scheme, *DUs* that fail to access the subscribed topic within a specific period will be removed from the group.

The following advantages are achieved using the proposed OCP-ABE scheme compared to the existing scheme.

1. Malicious users can be traced through the Pub/Sub model.
2. Misbehaving IoMT nodes can be revoked in time with minimal computational cost.
3. Reduced ciphertext size and key improve transmission efficiency and reduce the time for the decryption operation.
4. Appending a blind token for each attribute increases the complexity for malicious IoMT nodes to break the security scheme.

8.3.3 Secure Data Publishing/Subscribing

The OCP-ABE provides a fine-grained, secure data access control and encryption/decryption strategy. In an IoMT, where healthcare devices often deal with sensitive information, OCP-ABE ensures that only authorised users or *DU* can decrypt and access particular information based on their attributes. Thus, the secure data publishing and subscribing scheme improves the confidentiality, authentication, and privacy of data transmitted between *DO* and *DU*, leveraging OCP-ABE for access control and Fast-PRESENT encryption. In OCP-ABE, the topic- and time-factor-based key revocation strategy ensures that subscriber access can be dynamically revoked, ensuring up-to-date access control against malicious activities. This property compromises the key of one subscriber or spoofs it and does not affect the other subscribers or the entire security of IoMT communication. Moreover, this secure data publishing and subscribing scheme based on OCP-ABE offers robust security while ensuring that IoMT devices can efficiently transmit data within resource constraints, making it a highly suited solution for a healthcare environment with more sensitive data.

8.4 Attack Analysis and Security Strength Evaluation

This section provides insight into the attack analysis and security strength evaluation, designed to validate the OCP-ABE and Fast PRESENT. The attack analysis explains the attack prevention capability of OCP-ABE-MQTT in replay, traceability, and masquerade attacks. The security strength is evaluated from three perspectives: secret keys, plaintext,

and ciphertext. Three tests are conducted to evaluate the security strength of the proposed scheme, such as the avalanche effect, correlation coefficient, and semi-equivalent key test using the Cooja simulator. It estimates the ciphertext changes when it makes minor changes in the secret key and plaintext. The high correlation in ciphertexts tends to poor security strength while varying the secret keys or plain texts at a minimum level.

8.4.1 Attack Analysis

The proposed OCP-ABE scheme ensures secure communication while implementing MQTT over IoMT. Provides security against various attacks. The possibility of security provisioning against different attackers in IoMT applications is discussed below.

Security Against Traceability Attack: Generally, the MQTT, which has a strong cryptographic mechanism, makes it difficult for an attacker to change the messages between the *DU* and *MB*. However, an attacker can copy a valid request message from the *DU* and resend it to the server. It disrupts the network access for legitimate users and tends to waste resources. Another primary type of network attack is the blind key traceability attack. Among a group of *DU*, any IoMT node can be compromised by a malicious node and share its secret key with malicious *DU* to access the topic illegally. In addition, using the identity and blind key during the generation of the secret key helps to correctly identify malicious users from a group of *DU* in the publish/subscribe model by executing the following equation 8.6.

$$ID_{DU} = (Bt_{Si} \oplus E.B_k) \quad (8.6)$$

The following definition, if trace $(ID_{DU}) \in$ nodes with the same \mathbb{A} , then $ID_{DU} \cup RL$ denotes the possibility of the presence of an attack in IoMT. This statement determines that the attacker and the data user have similar attribute sets, and there is a chance to trace secret keys easily. This definition states that if the data user and the attacker have a similar attribute set and the attacker also traces the secret key, then they are appended in the revocation list, *RL*. In other words, the ID of the extracted *DU* and requested *DU* are not matched in such attacker cases. Moreover, the corresponding IoMT node to

the ID_{DU} is considered malicious and added to the RL . Finally, OCP-ABE cancels the request from the malicious DU and improves the security of MQTT communication while identifying the exact identity of the malicious DU without extensive computational cost.

Security against Masquerade Attack: The masquerade attacks only impact when the revoked user list is not updated for a long time. The malicious DU shares its credentials with an attacker and attempts to damage the confidentiality of the IoMT network data. However, the proposed work explores the lightweight indirect revocation scheme using TF and ensures fresh, nonrevoked list maintenance in a group. Moreover, the proposed scheme provides better data confidentiality, even in the worst case of secret key tracing, because it uses a frequent self-key update scheme. Thus, the proposed scheme secures MQTT communication and considerably explores the resources of medical sensors.

Security against DoS Attack: DoS attack aims at making an MB and DO unavailable to its intended subscribers or DO by temporarily disrupting communication services. It is accomplished after the success of a masquerade or traceability attack. The copied request messages flood towards the MB and overload it to prevent all legitimate requests from being transmitted to the server. Due to the usage of a specific ID and TF in the proposed scheme, the attack messages are successfully identified by the MB . Thus, the frequent arrival of messages with the id of mDU is successfully rejected by MB .

Security against Guessing Attack: Another attack is the blind key guessing attack. The legitimate intelligent sensors in a group may be compromised to obtain the blind key of legitimate medical devices. Blind key guessing attacks are performed by brute force or dictionary attacks. The brute-force attacks analyse every overheard message, possible code, and secret key until they identify the private information. However, it takes considerable time to determine the secret key. A dictionary attack explores a dictionary of standard secret keys and attempts to trace the exact blind key of legitimate devices. However, dynamically generated blind keys using a self-key revocation scheme cannot be identified by both brute-force and dictionary attacks. As a result, it ensures security against guessing attacks on IoMT applications.

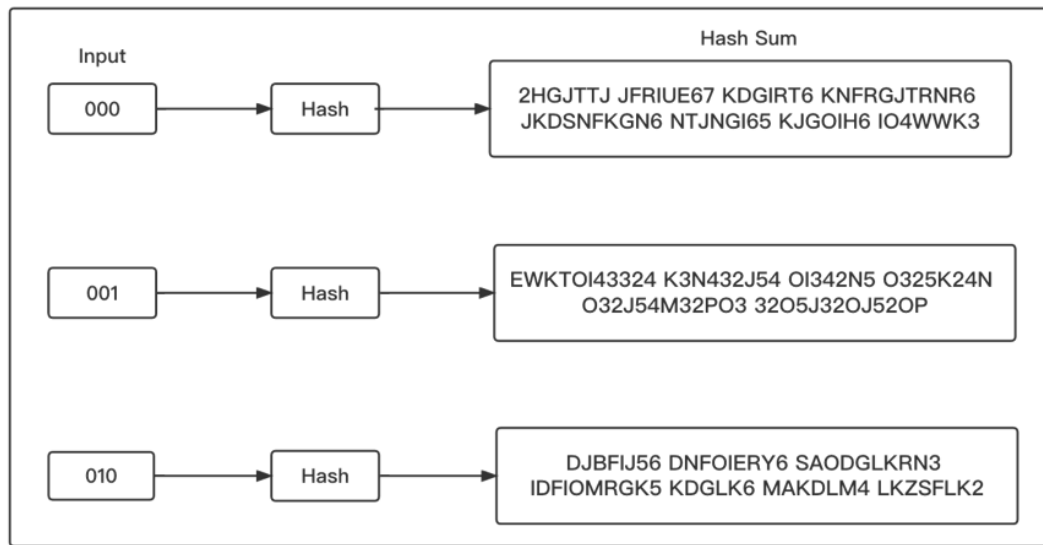


Figure 8.9: Hash Function to Show Avalanche Effect

8.4.2 Avalanche Effect

The avalanche effect is the desired property to evaluate cryptographic algorithms, such as block cyphers and hash functions. A small change in the key or plaintext in block cyphers significantly changes the ciphertext. When a cryptographic algorithm fails to exhibit the avalanche effect to a significant degree, it includes poor randomisation, and thus it eases the data prediction on such a cryptographic algorithm. Hence, the avalanche effect is considered a desirable security condition. The following figure demonstrates the hash function to show the avalanche effect. The SHA-1 hash function in Figure 8.9 shows a good avalanche effect. When a single bit is changed on input, the hash sum changes considerably.

$$\text{Avalanche Effect (AE) in percentage} = \frac{1}{x} \sum_{i=1}^x |Ct_i - Pt_i| \times 100 \quad (8.7)$$

The formula 8.7 is used to calculate the avalanche effect, adopted from [484]. x denotes the length of plaintext/ciphertext, and Ct_i and Pt_i are the ciphertext and plaintext bits, respectively. The term $|Ct_i - Pt_i|$ denotes the bits of ciphertext flipped due to the changes in plaintext. The calculated value is converted into a percentage by multiplying by 100. The proposed work keeps the plaintext constant for security evaluation, while the secret

key is modified bitwise. The proposed scheme takes ten secret keys for the avalanche test and analyses them by flipping 1 or 2 bits. The AE and its ten secret key percentage values are listed below. An avalanche effect denotes the total number of bits in the ciphertext affected by bit flipping in the plaintext. The value of a good avalanche effect should be 50%

Table 8.3: Avalanche Effect.

S.no	Changes (Bits)		AE-Value		AE-Value(%)		Avg AE	Avg AE(%)
	1-bit	2-bit	1-bit	2-bit	1-bit	2-bit		
1	101001	010011	0.16	0.83	16	83	0.575	57.5
2	101011	010111	0.33	1	33	100	0.83	83
3	100100	110000	0.33	0.33	33	33	0.495	49.5
4	100110	110100	0.5	0.5	50	50	0.75	75
5	100101	110001001	0.5	0.5	50	50	0.75	75
6	100111	110110	0.66	0.66	66	66	0.99	99
7	101100	110001	0.16	0.5	16	50	0.41	41
8	101110	110101	0.33	0.66	33	66	0.66	66
9	101101	110011	0.33	0.66	33	66	0.66	66
10	101010	010101	0.16	0.83	16	83	0.575	57.5

8.4.3 Correlation Coefficient

The correlation coefficient evaluates the relationship between plaintext and its corresponding ciphertext. If the ciphertext is entirely independent of the plaintext, the Correlation Coefficient (CC) results in 0%. Various ciphertexts are identified for different plaintexts by varying the secret key, and the correlation among them is listed. The CC is calculated using equation 8.8 with the support of AE.

$$R = \frac{\sum_{i=1}^s (Pt_i - AE) (Ct_i - AE)}{\left(\sqrt{\sum_{i=1}^s (Pt_i - AE)^2} \sqrt{\sum_{i=1}^s (Ct_i - AE)^2} \right)} \quad (8.8)$$

Table 8.4 provides a format for storing the CC test. This test is applied with ten different plain texts, shown in Table 8.4. This test shows the relationship between the

Table 8.4: Correlation Coefficient Test.

Plain-text	R-Value		Max R
	1-bit	2-bit	
1	-0.1972	-0.498	-0.1972
2	-0.292	0	0
3	-0.292	-0.292	-0.292
4	-0.337	-0.337	-0.337
5	-0.337	-0.337	-0.337
6	-0.523	-0.523	-0.523
7	-0.1972	-0.337	-0.1972
8	-0.292	-0.523	-0.292
9	-0.292	-0.523	-0.292
10	-0.1972	-0.498	-0.1972

plaintext and its corresponding ciphertext generated using the proposed scheme. This test is executed from the attacker's perspective, who has access to the ciphertext and is guessing the secret key. The CC denotes the relationship between the plaintext and its corresponding ciphertext. The value of R ranges from 0, +1 to -1 . ± 1 denotes the strongest possible agreement, and 0 is the strongest possible disagreement. The table shows that most CC values are close to zero, indicating that the proposed scheme is in great agreement with the minimum correlation coefficient, high variation of the ciphertext and better security strength.

8.4.4 Semi Equivalent Key Test

This test keeps the plaintext constant and observes the change in the ciphertext, with a bit difference in the secret key. The new Fast PRESENT key scheduling algorithm ensures no equivalent keys in rounds. Because of plaintext encryption, they use equivalent keys, resulting in the same ciphertext or small differences. Those keys are subjected to the Hamming weight test using the following equation to validate the security strength of the Fast PRESENT algorithm in the proposed scheme. Table 8.5 lists the results of the semi-equivalent key test. The high HW value denotes the deviation between ciphertexts

while changing plaintexts.

$$\text{Hamming Weight} = \frac{\text{Number of non zero bits}}{\text{Total Bits}} \quad (8.9)$$

It denotes the nonzero bits in the ciphertexts while changing the plaintexts. A high number of zero bits in the ciphertext can act as a helping hand for differential cryptanalysis attacks. The upper 50% of the HW denotes the high-security strength of the proposed scheme, as shown in Table 8.5.

Table 8.5: Semi Equivalent Key Test.

Plain-text	Hamming Weight		Avg HW
	1-bit	2-bit	
1	0.5	0.5	0.5
2	0.67	0.67	0.67
3	0.33	0.33	0.33
4	0.5	0.5	0.5
5	0.5	0.44	0.47
6	0.66	0.66	0.66
7	0.5	0.5	0.5
8	0.66	0.66	0.66
9	0.66	0.66	0.66
10	0.5	0.5	0.5

8.5 Performance Evaluation of OCP-ABE

To evaluate the performance of OCP-ABE MQTT and existing MQTT with SIMON and Speck encryption algorithms [262] and Pre-AugPAKE [269], the simulation on IoT nodes is conducted using Ubuntu 18.04 LTS 64-bit, Instant Contiki-3.0, and VMware Player 16.1.2. In the IoT environment, all existing and proposed schemes are evaluated to show the efficiency and complexity of the proposed method. To validate the efficiency of OCP-ABE more clearly, it is compared with the ICP-ABE scheme proposed in the previous chapter. The evaluation results are obtained for the MQTT v5 settings, and Table 8.6 lists the simulation parameters.

Table 8.6: Simulation Model.

Simulator	Cooja
Number of nodes	15, 30 and 60
Broker	Mosquitto RSMB
Transmission Range	50m
Area	100m \times 100m
Application Layer Protocol	MQTT v5
MAC Layer Protocol	802.15.4
Security Algorithms	OCP-ABE, ICP-ABE, SIMON, SPECK, Pre-Aug-PAKE, Base-MQTT
Simulation Time	300 seconds

8.5.1 Performance Evaluation of OCP-ABE MQTT without Attacks

Different performance metrics, such as throughput, PDR, delay, execution time, energy, and CPU energy consumption, are evaluated without implementing application-layer attacks.

Throughput and Packet Delivery Ratio (PDR): Figure 8.10 illustrates the performance of MQTT in terms of throughput with SIMON, SPECK, Pre-AugPAKE, and OCP-ABE symmetric encryption algorithms under 15 and 30 node topologies. Moreover, it is simulated without any attackers. The proposed scheme, OCP-ABE, attains better throughput and PDR, followed by the MQTT-based scheme. The Base MQTT works better than all other algorithms due to the exclusion of cryptographic functions. The optimised CP-ABE for MQTT data encryption works well regarding throughput and PDR. For instance, the Base MQTT and OCP-ABE MQTT deliver throughput of 317.89 and 271.8 bits per second, respectively, and Pre-AugPAKE delivers 80.33 bits under a 15-node scenario.

Moreover, the OCP-ABE implements a Fast PRESENT algorithm with support from a novel key scheduling algorithm. Without any security scheme in place by default, the Base MQTT continuously delivers a large number of packets. Moreover, SIMON does not attain noticeable results in performance measures. Although both SIMON and SPECK integrate

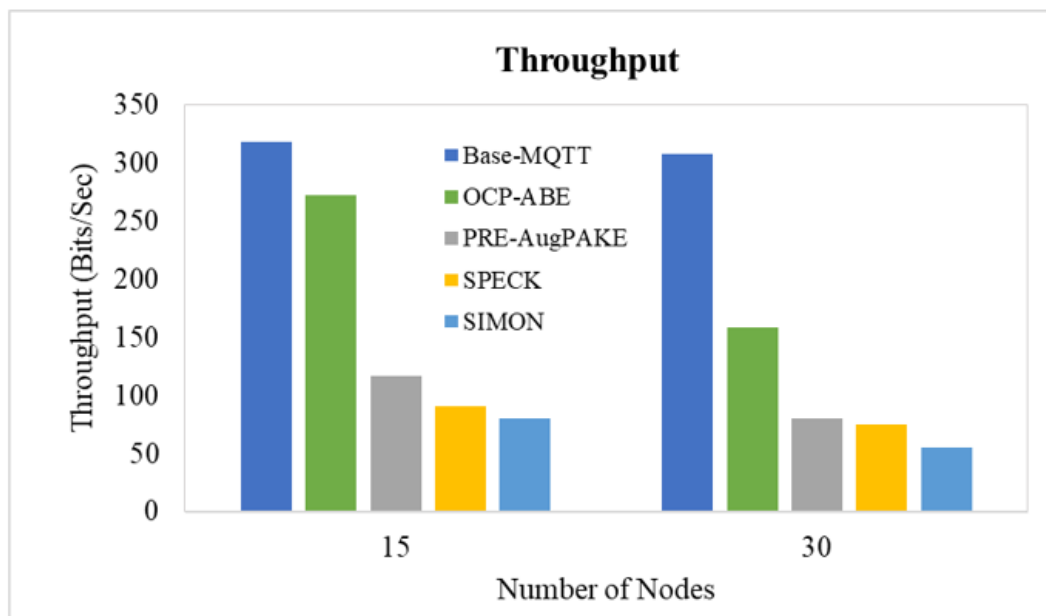


Figure 8.10: Number of Nodes Vs. Throughput

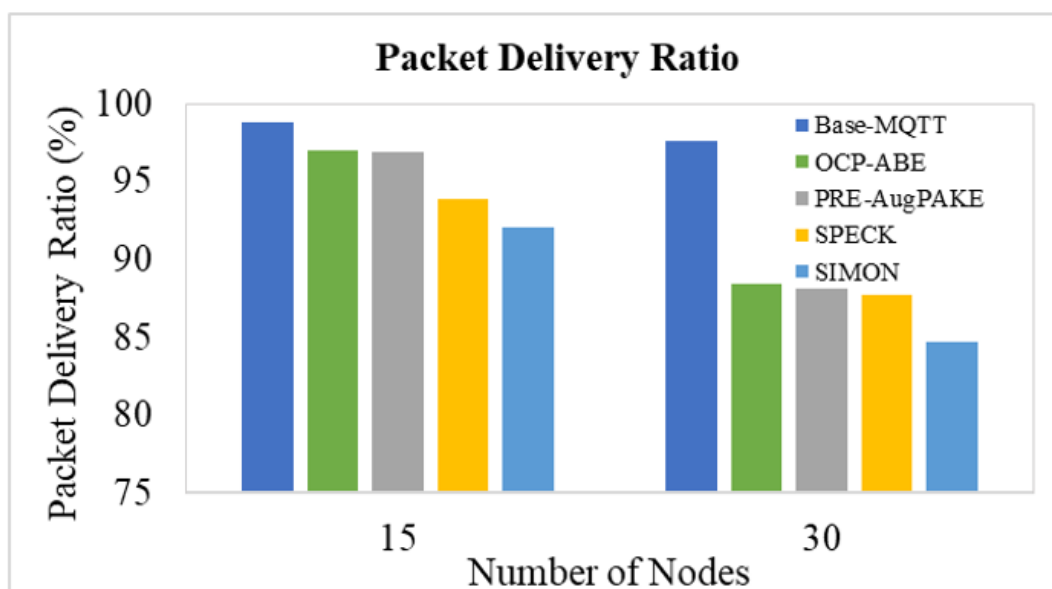


Figure 8.11: Number of Nodes Vs. PDR

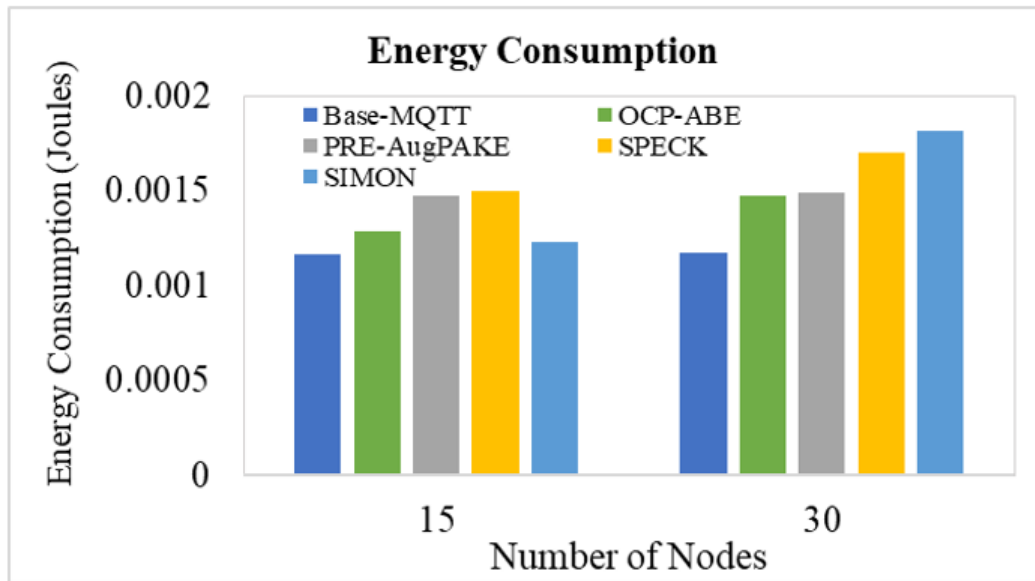


Figure 8.12: Number of Nodes Vs. Energy Consumption

lightweight cryptography, they eliminate the support of S-boxes. The cryptography size and the RAM requirements are large in SIMON and SPECK. Less complex algorithms in SIMON help to achieve better packet delivery, but tend to have a high delay and poor throughput, especially in the IoT environment. For example, SIMON and SPECK achieve 93.87% and 92.02% of PDR, respectively, for 15-node scenarios in Figure 8.11. In addition, the throughput of these algorithms is 91.13 and 80.22 bits per second, respectively, in Figure 8.10. Moreover, the impact of scalability on the selected algorithms is evaluated. Compared to the 30-node topology, the 15-node topology performs better on throughput and PDR. For example, the proposed OCP-ABE improves 8.65%

Figures 8.12 and 8.13 demonstrate the energy consumption and CPU energy consumption for the base MQTT, the proposed OCP-ABE, and existing security schemes while implementing MQTT v5. In the network with 15 IoMT devices and without attack, all algorithms result in low energy consumption, whereas Pre-AugPAKE, SPECK, and SIMON correspond to high energy consumption. Among those algorithms, the main reason behind the difference in energy consumption is that different cryptography operations are not performed unnecessarily during encryption. Due to the absence of attackers, Base-MQTT spends fewer energy resources. For instance, the Base MQTT spends only 0.00116 and 0.00117 joules of energy consumption under 15 and 30-node scenarios, respectively.

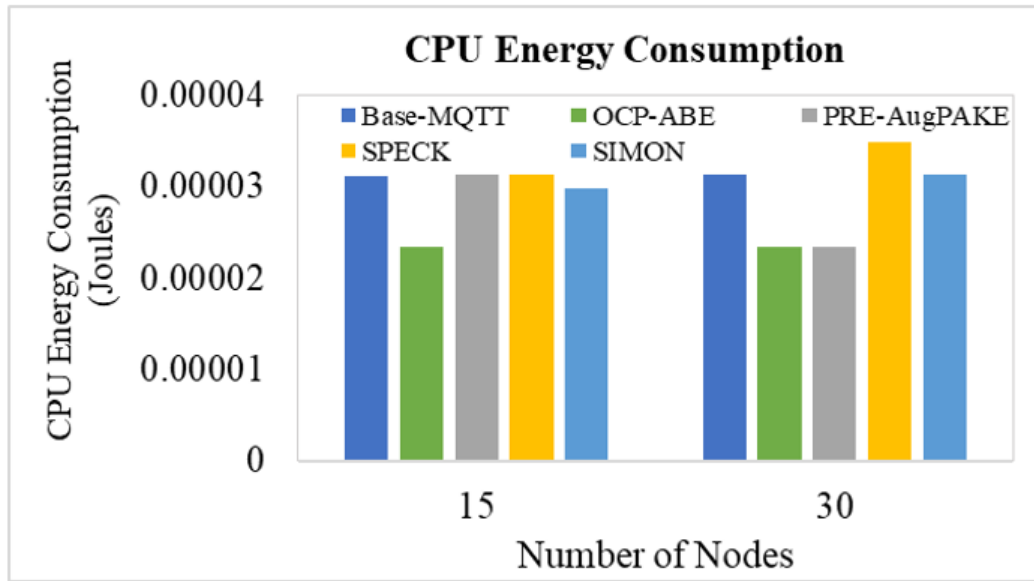


Figure 8.13: Number of Nodes Vs. CPU Energy Consumption

The Pre-Aug-PAKE and SPECK attain noticeable energy consumption. Pre-AugPAKE incorporates multiple schemes, such as the PRESENT and AugPAKE algorithms, and tends to have high energy consumption. For instance, the Pre-AugPAKE algorithm consumes 0.001476 joules in a 15-node topology, and in the same scenario, the OCP-ABE consumes 0.00129 joules. The proposed scheme reduces the complexity of the PRESENT and CP-ABE algorithms and effectively uses the energy resources of nodes. Compared to 15-node topologies, OCP-ABE increases energy consumption by 0.000186 Joules under 30-node topologies. Many DO and DU are involved in communication under 30-node topologies, which increases communication delay and the energy consumption to complete communication.

Delay is the average difference between the received time and the sent time of the packets. With complex cryptography algorithms, the time required to complete the encryption and decryption process is high. Figures 8.14 and 8.15 illustrate the performance of the proposed, base, and all existing algorithms in terms of delay and execution time, respectively, obtained for 15 and 30-node scenarios. In Figure 8.14, it is clear that the delay of Base MQTT and OCP-ABE is better than all other algorithms over 15 and 30-node topologies. The Base-MQTT incurs a nearly 1.938 ms delay in delivering the data packets under 15-node topologies. According to Figure 8.14, the SPECK provides the data pack-

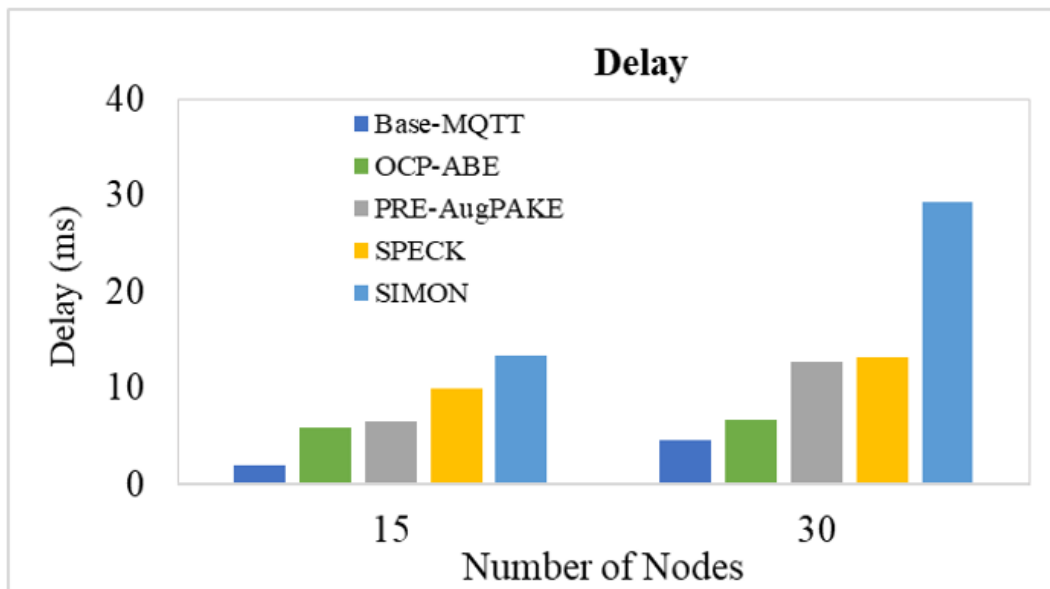


Figure 8.14: Number of Nodes Vs. Delay

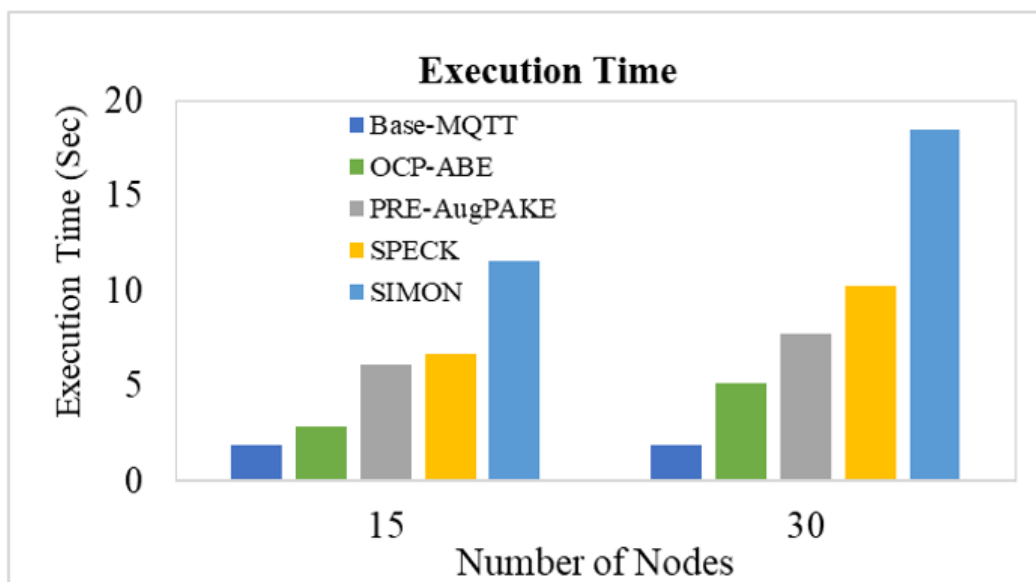


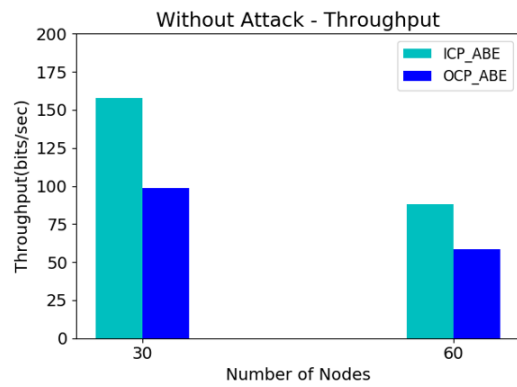
Figure 8.15: Number of Nodes Vs. Execution Time

ets with a delay of 6.68 ms. The number of rounds and exclusion of S-box usage tend to cause SIMON to have expensive transpose operations on both plaintext and ciphertext, as well as high communication delay. Moreover, compared to other existing systems, the OCP-ABE accomplishes better delay and execution time results. Under 30-node topologies, OCP-ABE achieved 6.7 ms delay and 5.13 seconds of execution time, respectively. Compared to PRE-Aug-PAKE, SPECK, and SIMON, the proposed OCP-ABE decreases the delay by 6.03 ms, 6.55 ms, and 22.55 ms due to optimisations in CP-ABE and fast PRESENT algorithms.

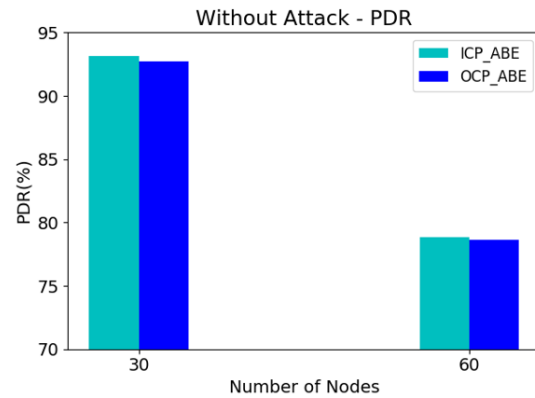
8.5.2 Comparison Results of ICP-ABE and OCP-ABE

This section compares the proposed ICP-ABE and OCP-ABE to analyse the effectiveness of the suitability of healthcare use cases. Although both algorithms utilise lightweight PRESENT and attribute-based encryption schemes, they perform differently when implementing various healthcare use case scenarios.

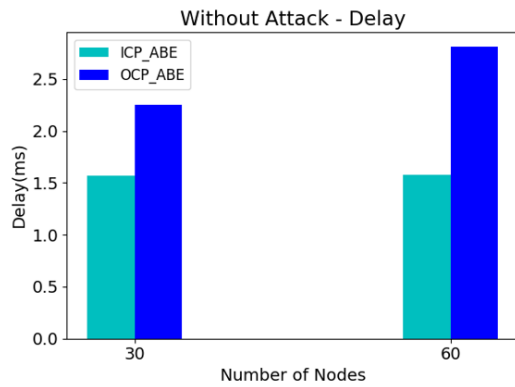
Figure 8.16 compares the performance efficiency of ICP-ABE and OCP-ABE in terms of different performance metrics, such as throughput, PDR, delay, execution time, energy consumption and CPU energy consumption. The results are obtained for 30 and 60-node scenarios without creating attack scenarios like DoS and MitM. Figures of 8.16(a) and 8.16(b) demonstrate that the proposed OCP-ABE decreases throughput and PDR compared to the proposed ICP-ABE scheme. The simpler design of ICP-ABE enables rapid executions, and this speed assists in handling more packets in a particular time compared to OCP-ABE, resulting in improved PDR and throughput. For example, ICP-ABE and OCP-ABE obtain 88.1 and 58.66 bps of throughput and 78.85% and 78.65% of PDR, respectively, for the 60-node scenario. In addition, the OCP-ABE exploits additional processes, such as self-key revocation and fast PRESENT, which escalate the delay and execution time in the network. Figure 8.16(c) shows that the ICP-ABE decreases the delay and execution time by 1.23 ms and 4.1 s compared to the OCP-ABE scheme. The intensive OCP-ABE computations lead to higher energy consumption than the ICP-ABE scheme. Although OCP-ABE improves the security level against multiple attacks,



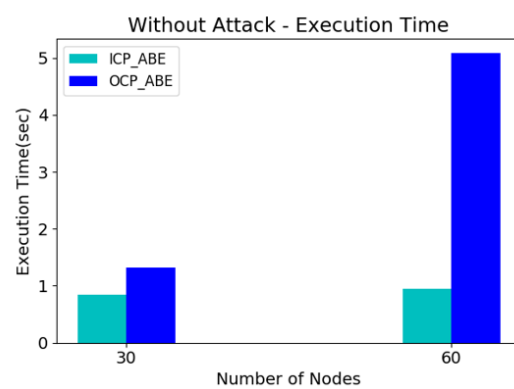
(a) Throughput



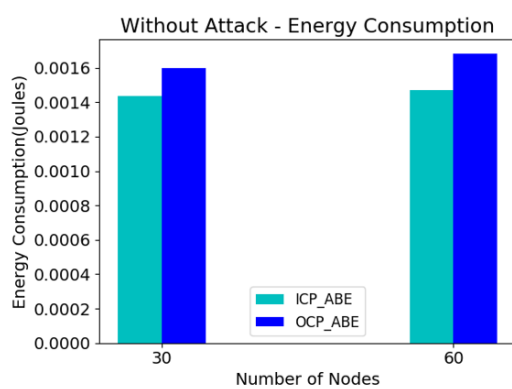
(b) Packet Delivery Ratio



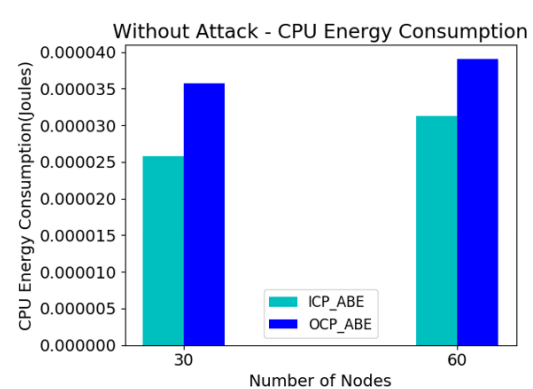
(c) Delay



(d) Execution Time



(e) Energy Consumption



(f) CPU Energy Consumption

Figure 8.16: Performance Results of ICP-ABE and OCP-ABE for Without Attack Scenario

it achieves low performance efficiency under attack scenarios. Moreover, the ICP-ABE performs better than the OCP-ABE without an attack scenario because of its simpler design and fewer computations.

Figure 8.17 illustrates the performance results regarding throughput, PDR, delay, execution time, energy consumption, and CPU energy consumption. To validate the effectiveness of both ICP-ABE and OCP-ABE in different scenarios, the number of nodes varies from 30 to 60 in the presence of both DoS and MitM attackers. The attackers significantly impact performance efficiency by losing, dropping, or manipulating the MQTT packets during transmission. Thus, it significantly reduces the throughput and PDR while increasing the energy consumption and delay in the network. For example, ICP-ABE and OCP-ABE attain 42.02 and 39.89 bps of throughput in an attack scenario when 60 MQTT clients are in the network. In the same scenario, the OCP-ABE incurs a 19.86 ms delay, which is 2.08 ms higher than the ICP-ABE scheme. The reason is that ICP-ABE does not include any key revocation and has more s-box layers in the PRESENT algorithm, such as OCP-ABE, resulting in a minimum delay and execution time. The OCP-ABE achieves a high execution time of 0.18 seconds compared to the ICP-ABE under the 60-node scenario in Figure 8.17(d). In an attacker's presence, it is essential to re-transmit the packets frequently owing to packet losses or manipulations. In addition, implementing security incurs additional energy consumption, resulting in low performance efficiency. For instance, the OCP-ABE and ICP-ABE need 0.001567 joules and 0.001609 joules of energy, respectively, for 60 number of 60-node scenario.

Figure 8.18 plots the performance results of the proposed ICP-ABE and OCP-ABE schemes to analyse their performance comparatively. The results are plotted for 30 and 60-node scenarios in the presence of a DoS attack. Generally, the DoS attacker tries to flood unnecessary traffic into the data transmission to reduce the successful packet delivery. Thus, it incurs a significant level of packet loss in the network. To show the effects of the DoS attacker on performance metrics such as throughput, PDR, delay, execution time, energy consumption, and CPU energy consumption, the node density is varied, which also impacts the attack percentage. Each proposed strategy tries to accomplish better performance and security trade-offs. For instance, the throughput of ICP-ABE and OCP-

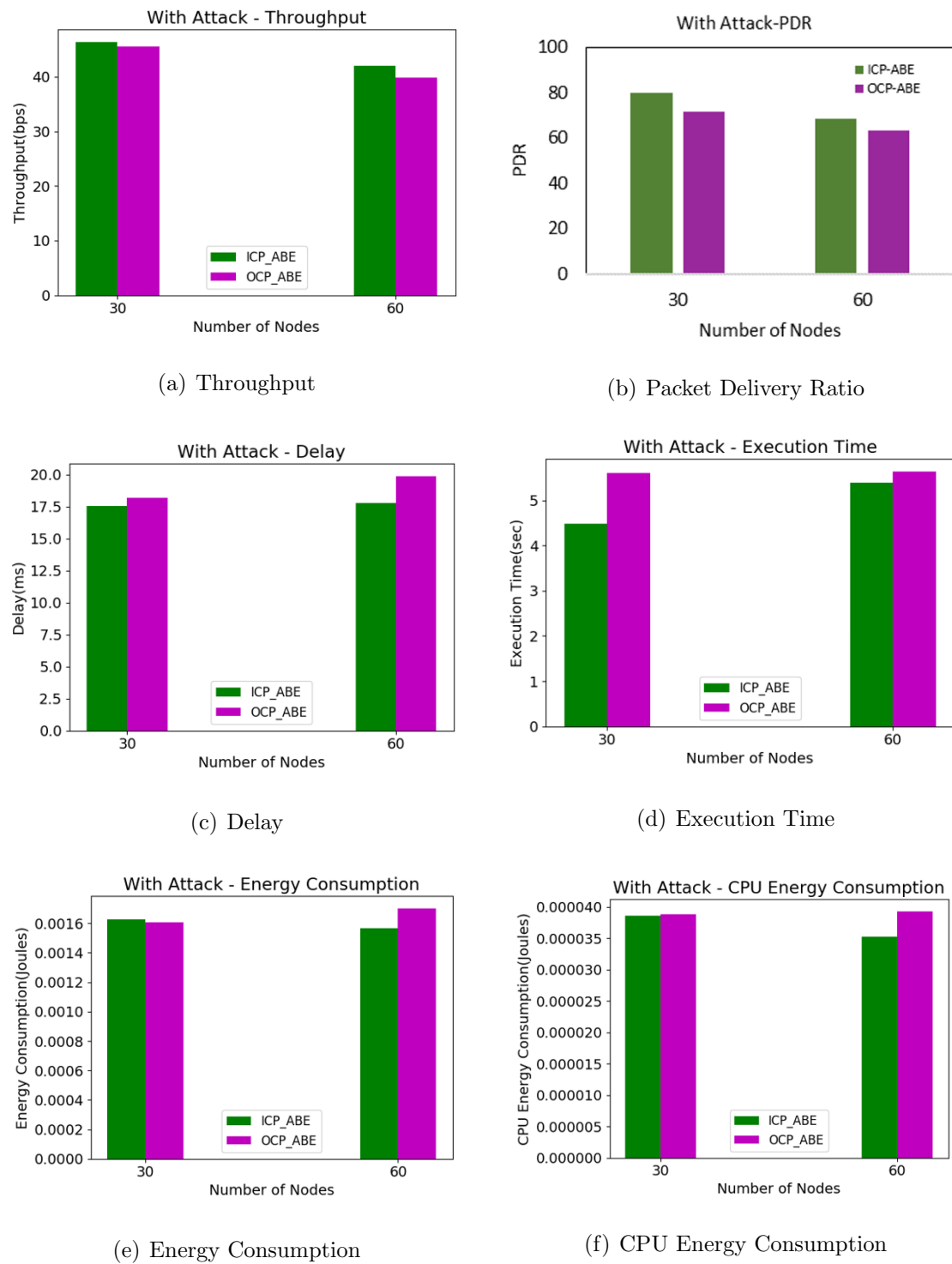


Figure 8.17: Performance Results of ICP-ABE and OCP-ABE for With Attack Scenario

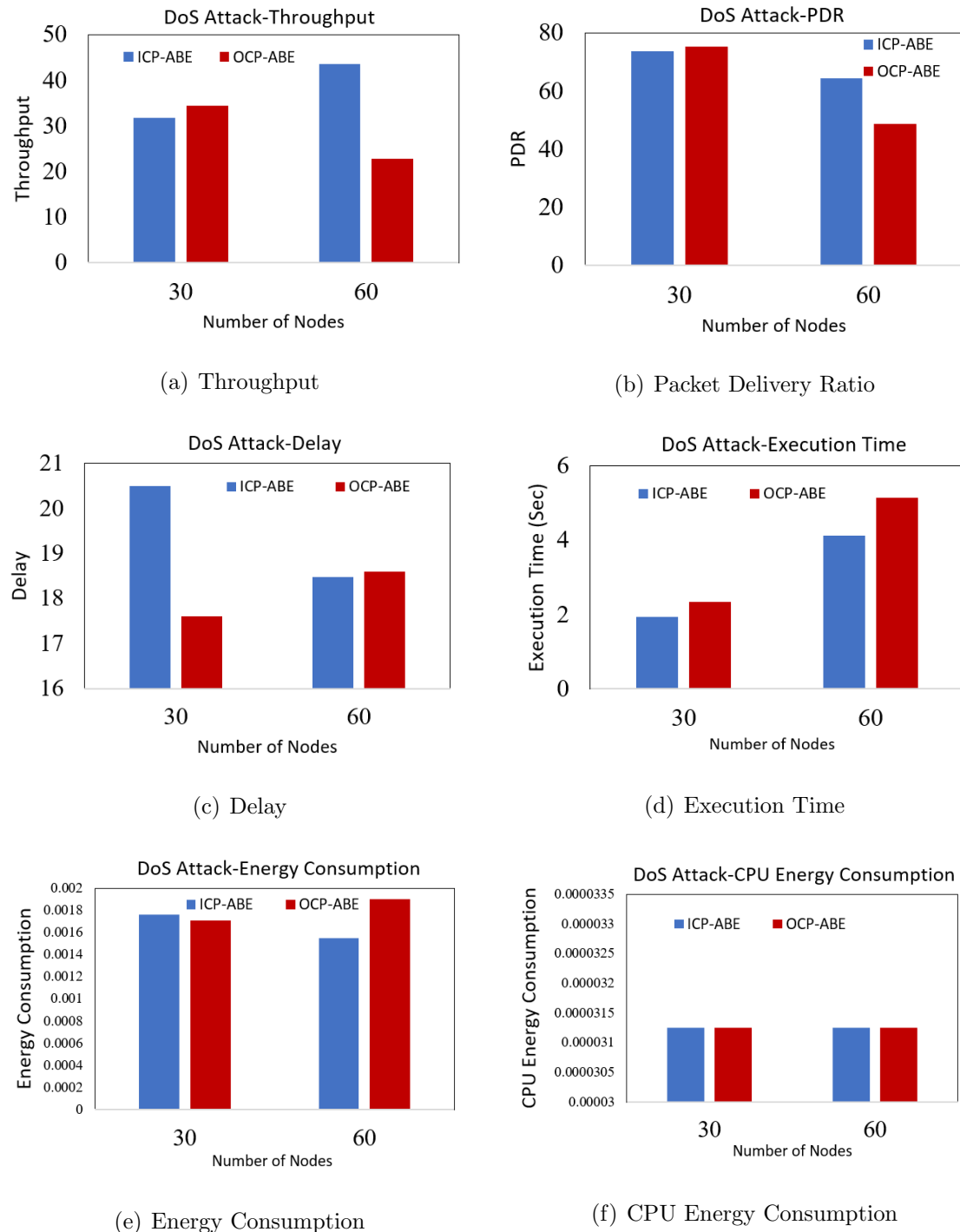


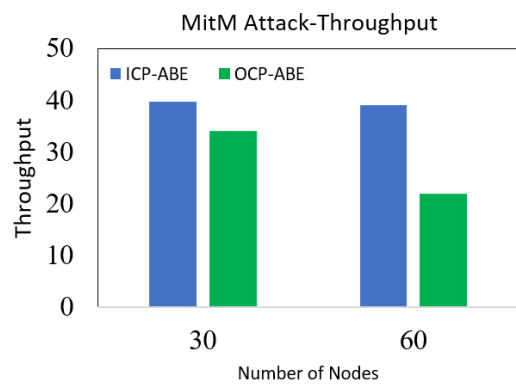
Figure 8.18: Performance Results of ICP-ABE and OCP-ABE for DoS Attack Scenario

ABE is 43.52 bps and 22.82 bps, respectively, for 60-node scenarios. The simpler algorithm structure of ICP-ABE compared to OCP-ABE escalates the throughput and PDR in low and high numbers of node scenarios. To provide strong resilience against multiple attacks, OCP-ABE includes additional processes and limits the number of attributes in the ABE scheme, resulting in increased delay, execution time, and energy consumption. For instance, the OCP-ABE needs 5.132 seconds of execution time for 60 node density, whereas the ICP-ABE requires 4.13 seconds for the same scenario.

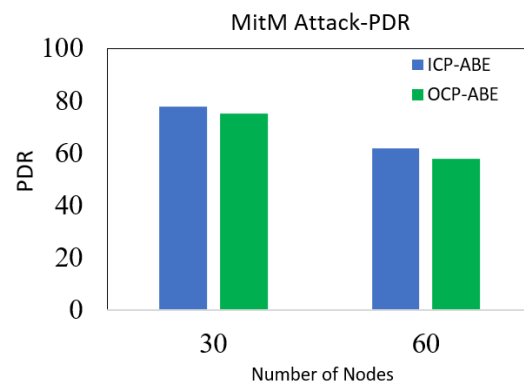
The performance impact of MitM attacks on 30 and 60-node scenarios is obtained in Figure 8.19 for both ICP-ABE and OCP-ABE. MitM attackers lead to significant packet losses by manipulating or intercepting the data transmission process. Especially in integrity-checking scenarios, many data retransmissions are required due to the impact of MitM attacks on performance efficiency. Both ICP-ABE and OCP-ABE algorithms enable strong defence against MitM attacks. However, the implementation of lightweight algorithms also leads to low performance. To analyse these impacts, Figure 8.19 varies the number of attacks by adjusting the density of the node from 30 to 60. PDR of ICP-ABE and OCP-ABE is 61.82% and 57.92%, respectively, for the 30-node scenario. In particular, the PDR of OCP-ABE is less than that of ICP-ABE; the reason is that OCP-ABE includes a key revocation and a fast PRESENT algorithm in its design, which escalates the execution time and reduces the PDR over a specific time interval. For example, it is observed that the execution time of OCP-ABE is 5.86 seconds and 3.54 seconds, respectively, for 60-node density scenarios. The optimisation performed in OCP-ABE also escalates the network's energy consumption and CPU energy consumption.

8.5.3 Results Summary

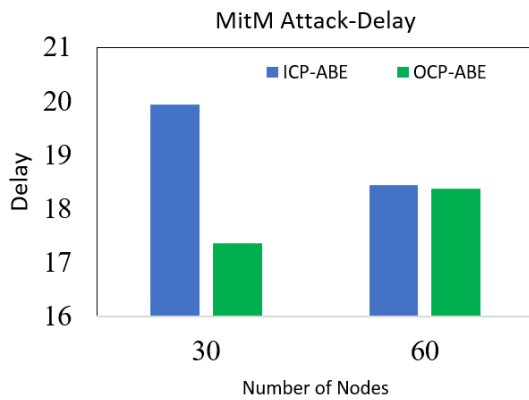
The performance analysis of OCP-ABE by comparing it with ICP-ABE under adversarial conditions underscores the key trade-offs between computational efficiency and strong security. Although OCP-ABE achieves slightly higher delay and execution time owing to its inclusion of key revocation strategies and additional PRESENT S-box layers, it compensates by significantly improving protection against session hijacking and addressing



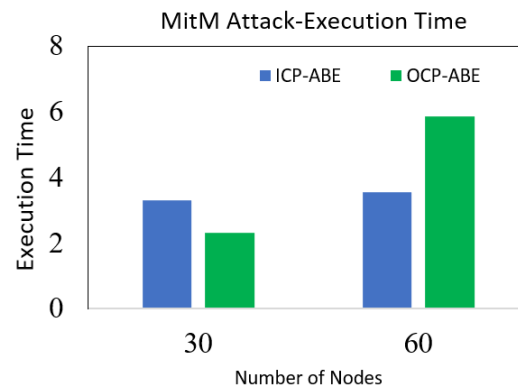
(a) Throughput



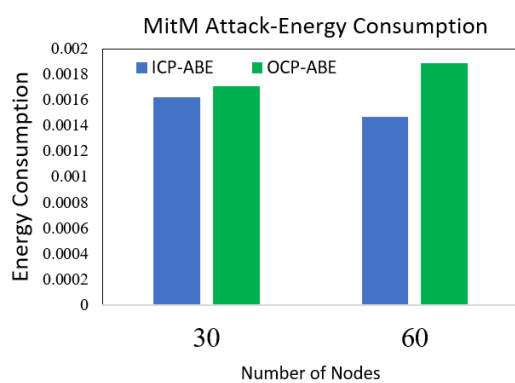
(b) Packet Delivery Ratio



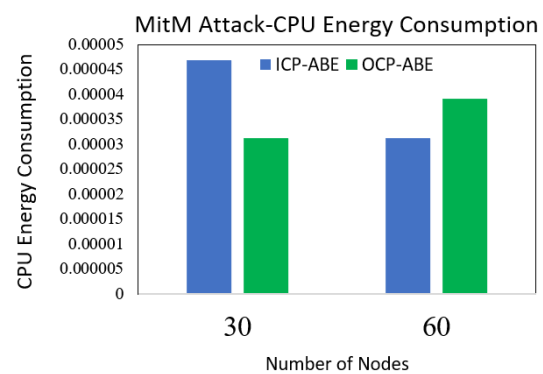
(c) Delay



(d) Execution Time



(e) Energy Consumption



(f) CPU Energy Consumption

Figure 8.19: Performance Results of ICP-ABE and OCP-ABE for MitM Attack Scenario

challenges related to unauthorised access, which ICP-ABE does not fully address. For example, in a 60-node attack scenario, OCP-ABE improves energy efficiency by approximately 2.6%, consuming 0.001567 J compared to 0.001609 J for ICP-ABE. Furthermore, OCP-ABE maintains a competitive throughput of 39.89 bps despite the security overhead, demonstrating its efficiency in practical healthcare 4.0 scenarios. These enhancements are highly beneficial for healthcare IoT applications, where it is crucial to ensure continuous authentication, secure session control, and lightweight cryptographic security. Moreover, OCP-ABE is a more flexible and robust security solution that effectively meets the architectural and operational gaps observed and resolved in ICP-ABE.

8.6 Chapter Summary

This work presents a lightweight OCP-ABE scheme based on the MQTT protocol to implement and improve security in resource-constrained IoMT environments. The proposed scheme performs fast data encryption using a Fast-PRESENT scheme combined with a new key scheduling algorithm, and it also uses a dynamic attribute-based signature scheme for data confidentiality and security. In addition, MQTT blind keys are associated with attributes, and an indirect revocation scheme provides fast and light encryption on IoMT devices. Furthermore, parallel execution of S-boxes in PRESENT SubBytes and the decrease in bit shift operation in the key scheduling algorithm can prevent *DU* from dealing with a heavy computational burden. The dynamic attribute-based signature scheme proves the verifiability of the data user group and identifies the compromised user. In an IoMT scenario, the proposed method has enabled the MQTT flag in TCP control messages and minimised the number of control messages. Finally, the proposed scheme has been compared with existing techniques such as Base MQTT, MQTT with SIMON, SPECK, and Pre-AugPAKE algorithms under the IoT environment. The OCP-ABE MQTT achieves better throughput with minimal delay and energy consumption, followed by the base MQTT.

Conclusions and Future Work

This thesis proposed three research contributions to improve the security and privacy of MQTT-enabled communication over a Healthcare 4.0 environment. This section provides the conclusions and future directions of these three objectives.

9.1 Contributions

Technological advances in digital information and communication have profoundly impacted the IoT, motivating several new breeds of opportunities in practical application scenarios, notably Healthcare 4.0, which deeply influences human lives. This research improves the reliability and security features of the MQTT protocol by resolving the security challenges of data-sensitive Healthcare 4.0. To accomplish the objectives, this research significantly offers three contributions by analysing the existing MQTT security strategies. The proposed research designed novel security solutions for MQTT-enabled Healthcare 4.0 by addressing security vulnerabilities without compromising performance in a resource-constrained environment.

9.1.1 Contributions

The initial contribution evaluated the performance of five lightweight symmetric key cryptography algorithms under MitM and DoS attack scenarios over MQTT-enabled Healthcare 4.0 and ensured data confidentiality. In-depth analysis of cryptography algorithms using formal security analysis, tool-based security analysis, and simulation-based analysis reveals potential strengths and weaknesses. The comprehensive validation is performed under diverse network conditions for varied metrics and scenarios.

The results indicate that the FBC is highly adaptable for resource-constrained medical

devices owing to its robust and effective cryptographic operations. For instance, under 64 bits/second data rate with an attack scenario, the FBC accomplishes 470 bits/second throughput, whereas AES, DES, LED, and PRESENT obtain 450.13, 208.24, 321.84, and 150.28 bits/second throughput values, respectively. Even when compared with FBC, the PRESENT and AES are complex, enabling high security while achieving acceptable performance nearly equal to FBC in non-ideal environments. For instance, the FBC accomplishes 86.57% of the PDR value, whereas the PRESENT attains 79.12% when the data rate is 64 bits/second in the presence of DoS and MitM attackers. Finally, this study concludes that PRESENT is more effective in securing communication over MQTT-enabled Healthcare 4.0 scenarios.

The second contribution, proposed a novel ICP-ABE method to improve the lightweight authentication mechanism that exploits the advantages of lightweight cryptography and improved encryption. By consolidating the MQTT protocol with blind keys and attributes, the ICP-ABE has significantly minimised the computational burden without sacrificing security and performance. ICP-ABE also prevents SlowDoS attacks on MQTT servers from being achieved by the client devices. Additionally, ICP-ABE is improved by making it suitable for deployment on resource-limited devices, exploiting the PRESENT algorithm to share anonymous tokens among MQTT clients in medical environments securely. Finally, the ICP-ABE scheme was evaluated against four existing strategies, simple-PRESENT, KSA-PRESENT, RSA-ECC SMQTT and SSL/TLS, utilising proven security and formal analysis methods. The effectiveness of ICP-ABE is demonstrated through simulations. The results show that ICP-ABE and RSA-ECC accomplish 93.75% and 97.46% of PDR without an attack scenario, whereas ICP-ABE improves PDR by 32.95% than the RSA-ECC algorithm under an attack scenario. The results show that ICP-ABE significantly improves performance and security in MQTT-enabled healthcare communications while maintaining lightweight and robust characteristics in standard and attack scenarios.

The contribution proposed an optimised CP-ABE-based lightweight cryptography scheme is using attribute-specific blind-key generation and the Fast-PRESENT algorithm. This scheme potentially improved the security and performance efficiency of MQTT-

enabled resource-limited IoMT environments. The OCP-ABE ensures data confidentiality and security by performing fast data encryptions through a Fast-PRESENT strategy consolidated with a novel key scheduling algorithm and utilising a dynamic attribute-based signature strategy. Additionally, parallel execution of S-boxes in Fast-PRESENT SubBytes and shrinking the bit shift operation in the key scheduling in OCP-ABE can prevent DU from managing a heavy computational load. The dynamic attribute-based signature model enables verifiability for DU groups and effectively determines compromised users within them. By activating the MQTT flag and reducing the number of control message exchanges, the OCP-ABE optimises TCP control messages in the IoMT environment. Finally, the OCP-ABE scheme has been rigorously evaluated using simulations and compared with conventional methods, including Base MQTT, Pre-AugPAKE, SPECK, and SIMON algorithms, and ICP-ABE within IoMT environments. The results show that the proposed OCP-ABE improves the PDR by 8.65% under 15 node topologies, compared to the 30-node topologies scenario. Although the extensive self-key revocation and Fast-PRESENT make OCP-ABE highly suitable to dynamic IoMT, it incurs acceptable PDR and throughput values when compared with ICP-ABE. For instance, the ICP-ABE and OCP-ABE obtain 88.1 bps and 58.66 bps of throughput and 78.85% and 78.65% of PDR for 60 nodes scenario. This result indicates that the OCP-ABE reduces throughput by 29.44 bits/second and shrinks the PDR by 0.2% due to its complex structure. However, the OCP-ABE MQTT achieves performance results, outperforming other existing strategies Pre-AugPAKE, SPECK, and SIMON under different scenarios, with the base MQTT configuration being the closest in performance.

The potential outcome of this research ensures improved MQTT by addressing the reliability and efficiency of Healthcare 4.0 services. This research facilitates smooth and secure data communication by ensuring regulatory compliance, making Healthcare 4.0 a transformative technology for an efficient and patient-centric healthcare ecosystem.

9.2 Future Directions

The three strategies focus on improving MQTT security, and several aspects still need improvement in the future.

Although the first objective evaluates the performance of five different symmetric key cryptography algorithms comprehensively, there are still some aspects to focus on in future that are,

I Integration with Real-Time Healthcare 4.0: Future work should focus on implementing the evaluated symmetric key cryptography strategies into realistic Healthcare 4.0 environments to assess their performance under real-world constraints, such as dynamic network conditions, delay strict lines, and resource consumption. This future work enables a more comprehensive evaluation of the sustainability of each algorithm for practical use cases of emergent healthcare environments.

II Development of Hybrid Strategy: Another future direction is the development of hybrid cryptographic algorithms that consolidate the strengths of the analysed symmetric algorithms and bring different benefits in the medical environment. The integration of lightweight and robust features from diverse algorithms can provide a way to design a more secure and effective encryption model tailored to Healthcare 4.0 use cases.

However, despite the efforts, there is still some space for improving the ICP-ABE scheme as follows,

I Attacker Revocation and Extension with Other Protocols: Future work of ICP-ABE plans to extend it to revoke malicious users and track malicious users with similar attribute sets in real-time. Also, it should focus on enhancing the ICP-ABE solution's security and testing its usefulness in other IoT application layer protocols.

II Real-time Dynamic Healthcare Attribute Revocation: Another notable direction for future research of ICP-ABE is to introduce a real-time dynamic attribute revocation strategy. By incorporating real-time updates and efficient attribute revocation without incurring extensive computational overhead, the ICP-ABE will be more optimised for maintaining high security and flexibility, and this is especially crucial for rapidly changing realistic Healthcare 4.0 scenarios where attributes may frequently vary.

Here are the two notable future works for the proposed OCP-ABE scheme.

I Integration with AI Models for Adaptive Security: Future work should explore incorporating the OCP-ABE strategy with various artificial intelligence algorithms to build an adaptive security solution. This approach can offer dynamic encryption parameter adjustments and key revocation policies according to real-time data patterns and threat detection, improving the resiliency and efficiency of OCP-ABE in various MQTT-enabled resource-restricted IoMT use cases.

II Energy Conscious Design and Seamless Interoperability across Heterogeneous Platforms: Another future direction is to optimise the OCP-ABE further to make it suitable for ultra-low power devices that are wearable and implantable medical devices used in IoMT applications. Further, the seamless interoperability across heterogeneous IoMT platforms is another critical future direction.

References

- [1] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of Things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, 2020.
- [2] A. Rejeb, K. Rejeb, H. Treiblmaier, A. Appolloni, S. Alghamdi, Y. Alhasawi, and M. Iranmanesh, "The internet of things (IoT) in healthcare: Taking stock and moving forward," *Internet of Things*, vol. 22, p. 100721, 2023.
- [3] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, "A survey on intelligent internet of things: applications, security, privacy, and future directions," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.
- [4] J. Li and P. Carayon, "Health care 4.0: A vision for smart and connected health care," *IJSE Trans. Healthc. Syst. Eng.*, vol. 11, no. 3, pp. 171–180, 2021.
- [5] S. Krishnamoorthy, A. Dua, and S. Gupta, "Role of emerging technologies in future IoT-driven healthcare 4.0 technologies: A survey, current challenges and future directions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 361–407, 2023.
- [6] K. H. Almotairi, "Application of Internet of Things in healthcare domain," *Journal of Umm Al-Qura University for Engineering and Architecture*, vol. 14, no. 1, pp. 1–12, 2023.
- [7] H. H. Alshammari, "The Internet of Things healthcare monitoring system based on MQTT protocol," *Alexandria Engineering Journal*, vol. 69, pp. 275–287, 2023.
- [8] K. Wei, L. Zhang, Y. Guo, and X. Jiang, "Health monitoring based on internet of medical things: Architecture, enabling technologies, and applications," *IEEE Access*, vol. 8, pp. 27 468–27 478, 2020.
- [9] A. Kumar, R. Krishnamurthi, A. Nayyar, K. Sharma, V. Grover, and E. Hossain, "A novel smart healthcare design, simulation, and implementation using healthcare 4.0 processes," *IEEE Access*, vol. 8, pp. 118 433–118 471, 2020.
- [10] D. Kant, A. Johannsen, and R. Creutzburg, "Analysis of IoT security risks based on the exposure of the MQTT protocol," *Electronic Imaging*, vol. 33, pp. 1–8, 2021.

- [11] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of Things security: Requirements, challenges, and solutions," *Internet of Things*, vol. 14, p. 100129, 2021.
- [12] A. E. Omolara, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, H. Arshad *et al.*, "The Internet of Things security: A survey encompassing unexplored areas and new insights," *Computers & Security*, vol. 112, p. 102494, 2022.
- [13] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: A survey," *IEEE Access*, vol. 8, pp. 201 071–201 086, 2020.
- [14] S. Quincozes, T. Emilio, and J. Kazienko, "Mqtt protocol: fundamentals, tools and future directions," *IEEE Latin America Transactions*, vol. 17, no. 09, pp. 1439–1448, 2019.
- [15] H. C. Hwang, J. Park, and J. G. Shon, "Design and implementation of a reliable message transmission system based on MQTT protocol in IoT," *Wireless Personal Communications*, vol. 91, pp. 1765–1777, 2016.
- [16] S. Xuan and D. Kim, "Development of cloud of things based on proxy using OCF IoTivity and MQTT for P2P internetworking," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 729–741, 2020.
- [17] L. M. Pham, N.-T.-T. Le, and X.-T. Nguyen, "Multi-level just-enough elasticity for MQTT brokers of Internet of Things applications," *Cluster Computing*, vol. 25, no. 6, pp. 3961–3976, 2022.
- [18] S. Akhbarifar, H. H. S. Javadi, A. M. Rahmani, and M. Hosseinzadeh, "A secure remote health monitoring model for early disease diagnosis in cloud-based IoT environment," *Personal and Ubiquitous Computing*, vol. 27, no. 3, pp. 697–713, 2023.
- [19] S. Lakshminarayana, A. Praseed, and P. S. Thilagam, "Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.
- [20] R. Banno, J. Sun, M. Fujita, S. Takeuchi, and K. Shudo, "Dissemination of edge-heavy data on heterogeneous MQTT brokers," in *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*. IEEE, 2017, pp. 1–7.
- [21] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in *2017 international conference on engineering & MIS (ICEMIS)*. Ieee, 2017, pp. 1–6.
- [22] H. Koziolk, S. Grüner, and J. Rückert, "A comparison of MQTT brokers for distributed IoT edge computing," in *Software Architecture: 14th European Conference, ECSA 2020, L'Aquila, Italy, September 14–18, 2020, Proceedings 14*. Springer, 2020, pp. 352–368.
- [23] R. A. Abdelouahid, M. Oqaidi, and A. Marzak, "Towards to a new iot interoperability architecture," in *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*. IEEE, 2018, pp. 148–154.

- [24] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "Mqtt vulnerabilities, attack vectors and solutions in the internet of things (IoT)," *IETE Journal of Research*, vol. 69, no. 6, pp. 3368–3397, 2023.
- [25] A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," *Cyberspace: Jurnal Pendidikan Teknologi Informatasi*, vol. 2, no. 2, pp. 109–134, 2019.
- [26] I. Vaccari, M. Aiello, and E. Cambiaso, "Slowite, a novel denial of service attack affecting MQTT," *Sensors*, vol. 20, no. 10, p. 2932, 2020.
- [27] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A review on the study on MQTT security challenge," in *2020 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2020, pp. 128–133.
- [28] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *2017 4th International conference on electrical engineering, computer science and informatics (EECSI)*. IEEE, 2017, pp. 1–6.
- [29] S. Tian and V. G. Vassilakis, "On the efficiency of a lightweight authentication and privacy preservation scheme for MQTT," *Electronics*, vol. 12, no. 14, p. 3085, 2023.
- [30] I. L. B. M. Paris, M. H. Habaebi, and A. M. Zyoud, "Implementation of SSL/TLS security with MQTT protocol in IoT environment," *Wireless Personal Communications*, vol. 132, no. 1, pp. 163–182, 2023.
- [31] P. S. Bangare and K. P. Patil, "Enhancing MQTT security for internet of things: Lightweight two-way authorization and authentication with advanced security measures," *Measurement: Sensors*, vol. 33, p. 101212, 2024.
- [32] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, "A review and state of art of internet of things (IoT)," *Archives of Computational Methods in Engineering*, pp. 1–19, 2021.
- [33] A. J. Hintaw, S. Manickam, S. Karuppayah, M. A. Aladaileh, M. F. Aboalmaaly, and S. U. A. Laghari, "A robust security scheme based on enhanced symmetric algorithm for MQTT in the Internet of Things," *IEEE Access*, vol. 11, pp. 43 019–43 040, 2023.
- [34] Z. Liu, T. Liang, J. Lyu, and D. Lang, "A security-enhanced scheme for MQTT protocol based on domestic cryptographic algorithm," *Computer Communications*, vol. 221, pp. 1–9, 2024.
- [35] S. Rajalakshmi, P. Duraisamy *et al.*, "A review on lightweight cryptographic algorithms in Internet of Things," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2023, pp. 1448–1451.
- [36] F. Thabit, O. Can, A. O. Aljahdali, G. H. Al-Gaphari, and H. A. Alkhzaimi, "Cryptography algorithms for enhancing IoT security," *Internet of Things*, vol. 22, p. 100759, 2023.

- [37] M. Chui, M. Collins, and M. Patel, “The internet of things: Catching up to an accelerating opportunity,” *McKinsey*, 2021.
- [38] F. Dahlgvist, M. Patel, A. Rajko, and J. Shulman, “Growing opportunities in the Internet of Things,” *McKinsey & Company*, vol. 22, 2019.
- [39] T. Ahmad and D. Zhang, “Using the internet of things in smart energy systems and networks,” *Sustainable Cities and Society*, vol. 68, p. 102783, 2021.
- [40] T. I. I. F. Database, “Global IoT connections forecast to reach 40 billion in 2033,” <https://transformainsights.com/news/iot-connections-40-billion-2033>, may 2024, accessed: 2024-7-12.
- [41] M. Husnain, K. Hayat, E. Cambiaso, U. U. Fayyaz, M. Mongelli, H. Akram, S. Ghaz-anfar Abbas, and G. A. Shah, “Preventing MQTT vulnerabilities using IoT-enabled intrusion detection system,” *Sensors*, vol. 22, no. 2, p. 567, 2022.
- [42] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, “Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2024.
- [43] E. Elemam, A. M. Bahaa-Eldin, N. H. Shaker, and M. Sobh, “Formal verification for a PMQTT protocol,” *Egyptian Informatics Journal*, vol. 21, no. 3, pp. 169–182, 2020.
- [44] A. Velinov and A. Mileva, *Running and testing applications for Contiki OS using Cooja simulator*. Information Technology and Education Development Conference, 2016.
- [45] M. Bansal and Priya, “Performance comparison of MQTT and CoAP protocols in different simulation environments,” *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*, pp. 549–560, 2021.
- [46] N. Muraleedharan and B. Janet, “A deep learning based HTTP slow DoS classification approach using flow data,” *ICT Express*, vol. 7, no. 2, pp. 210–214, 2021.
- [47] S. Villamil, C. Hernández, and G. Tarazona, “An overview of Internet of Things,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2320–2327, 2020.
- [48] A. H. M. Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y.-J. Park, “A survey on trend and classification of internet of things reviews,” *Ieee Access*, vol. 8, pp. 111 763–111 782, 2020.
- [49] V. Parihar, A. Malik, Bhawna, B. Bhushan, and R. Chaganti, “From smart devices to smarter systems: The evolution of artificial intelligence of things (AIoT) with characteristics, architecture, use cases and challenges,” in *Engineering Cyber-Physical Systems and Critical Infrastructures*. Cham: Springer International Publishing, 2023, pp. 1–28.

- [50] “Types of wireless communication protocols in IOT,” <https://iotdesignpro.com/articles/different-types-of-wireless-communication-protocols-for-iot>, accessed: 2024-1-6.
- [51] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, “Iot elements, layered architectures and security issues: A comprehensive survey,” *sensors*, vol. 18, no. 9, p. 2796, 2018.
- [52] M. Ahmid, O. Kazar, and E. Barka, “Internet of Things overview: Architecture, technologies, application, and challenges,” in *Advances in Information Security*. Cham: Springer International Publishing, 2024, pp. 1–19.
- [53] S. A. Ansar, S. Arya, S. Aggrawal, S. Saxena, A. Kushwaha, and P. C. Pathak, “Security in IoT layers: Emerging challenges with countermeasures,” in *Algorithms for Intelligent Systems*. Singapore: Springer Nature Singapore, 2023, pp. 551–563.
- [54] F. Lamonaca, C. Scuro, D. Grimaldi, R. S. Olivito, P. F. Sciammarella, and D. L. Carnì, “A layered IoT-based architecture for a distributed structural health monitoring system system,” *Acta Imeko*, vol. 8, no. 2, pp. 45–52, 2019.
- [55] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, “A survey of IoT security based on a layered architecture of sensing and data analysis,” *Sensors*, vol. 20, no. 13, p. 3625, 2020.
- [56] S. Hamdan, M. Ayyash, and S. Almajali, “Edge-computing architectures for Internet of Things applications: A survey,” *Sensors*, vol. 20, no. 22, p. 6441, 2020.
- [57] S. A. Chelloug and M. A. El-Zawawy, “Middleware for Internet of Things: Survey and challenges,” *Intelligent Automation & Soft Computing*, pp. 1–9, 2017.
- [58] S. Pratap Singh, V. Kumar, A. Kumar Singh, and S. Singh, “A survey on Internet of Things (IoT): layer specific vs. domain specific architecture,” in *Second International Conference on Computer Networks and Communication Technologies: ICCNCT 2019*. Springer, 2020, pp. 333–341.
- [59] H. G. Hamid and Z. T. Alisa, “Survey on IoT application layer protocols,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1663–1672, 2021.
- [60] I. Ahmad, M. S. Niazy, R. A. Ziar, and S. Khan, “Survey on IoT: security threats and applications,” *Journal of Robotics and Control (JRC)*, vol. 2, no. 1, pp. 42–46, 2021.
- [61] W. Kassab and K. A. Darabkh, “A–z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations,” *Journal of Network and Computer Applications*, vol. 163, p. 102663, 2020.
- [62] K. Priya Dharshini, D. Gopalakrishnan, C. Shankar, and R. Ramya, “A survey on IoT applications in smart cities,” *Immersive Technology in Smart Cities: Augmented and Virtual Reality in IoT*, pp. 179–204, 2022.

- [63] S. Ketu and P. K. Mishra, "A contemporary survey on iot based smart cities: architecture, applications, and open issues," *Wireless Personal Communications*, vol. 125, no. 3, pp. 2319–2367, 2022.
- [64] M. Rupp, M. Schneckenburger, M. Merkel, R. Börret, and D. K. Harrison, "Industry 4.0: A technological-oriented definition based on bibliometric analysis and literature review," *J. Open Innov.*, vol. 7, no. 1, p. 68, 2021.
- [65] M. S. Farooq, M. Abdullah, S. Riaz, A. Alvi, F. Rustam, M. A. L. Flores, J. C. Galán, M. A. Samad, and I. Ashraf, "A survey on the role of industrial IoT in manufacturing for implementation of smart industry," *Sensors*, vol. 23, no. 21, p. 8958, 2023.
- [66] M. Noor-A-Rahim, J. John, F. Firyaguna, H. H. R. Sherazi, S. Kushch, A. Vijayan, E. O'Connell, D. Pesch, B. O'Flynn, W. O'Brien *et al.*, "Wireless communications for smart manufacturing and industrial IoT: Existing technologies, 5g and beyond," *Sensors*, vol. 23, no. 1, p. 73, 2022.
- [67] M. A. Tunc, E. Gures, and I. Shayea, "A survey on iot smart healthcare: Emerging technologies, applications, challenges, and future trends," *arXiv preprint arXiv:2109.02042*, 2021.
- [68] A. Ahad, M. Tahir, M. Aman Sheikh, K. I. Ahmed, A. Mughees, and A. Numani, "Technologies trend towards 5g network for smart health-care using IoT: A review," *Sensors*, vol. 20, no. 14, p. 4047, 2020.
- [69] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications," *Automation in Construction*, vol. 101, pp. 111–126, 2019.
- [70] V. A. Orfanos, S. D. Kaminaris, P. Papageorgas, D. Piromalis, and D. Kandris, "A comprehensive review of IoT networking technologies for smart home automation applications," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, p. 30, 2023.
- [71] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, and I. Traore, "A survey on IoT-enabled smart grids: emerging, applications, challenges, and outlook," *Energies*, vol. 15, no. 19, p. 6984, 2022.
- [72] S. Kirmani, A. Mazid, I. A. Khan, and M. Abid, "A survey on IoT-enabled smart grids: technologies, architectures, applications, and challenges," *Sustainability*, vol. 15, no. 1, p. 717, 2022.
- [73] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for Health Care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
- [74] J. Li and R. Malekian, "The Internet of Things: Impact and Implications for Health Care," *Journal of Medical Internet Research*, vol. 22, no. 11, 2020.
- [75] M. Kumar and S. C. Sharma, "Recent advancements in emerging technologies for healthcare applications," *Journal of Healthcare Engineering*, 2021.

- [76] M. S. Hossain and G. Muhammad, “Cloud-assisted industrial internet of things (IIoT) – enabled framework for health monitoring,” *Comput. Netw.*, vol. 101, pp. 192–202, 2016.
- [77] S. Sakr and A. Elgammal, “Towards a comprehensive data analytics framework for smart healthcare services,” *Big Data Res.*, vol. 4, pp. 44–58, 2016.
- [78] M. Chen, Y. Ma, J. Song, C.-F. Lai, and B. Hu, “Smart clothing: Connecting human with clouds and big data for sustainable health monitoring,” *Mob. Netw. Appl.*, vol. 21, no. 5, pp. 825–845, 2016.
- [79] V. S. Naresh, S. S. Pericherla, P. S. R. Murty, and S. Reddi, “Internet of things in healthcare: Architecture, applications, challenges, and solutions,” *Computer Systems Science & Engineering*, no. 6, 2020.
- [80] S. M. Karunarathne, N. Saxena, and M. K. Khan, “Security and privacy in IoT smart healthcare,” *IEEE Internet Comput.*, vol. 25, no. 4, pp. 37–48, 2021.
- [81] M. Kumar, A. Kumar, S. Verma, P. Bhattacharya, D. Ghimire, S.-H. Kim, and A. S. M. S. Hosen, “Healthcare internet of things (H-IoT): Current trends, future prospects, applications, challenges, and security issues,” *Electronics (Basel)*, vol. 12, no. 9, p. 2050, 2023.
- [82] S. Anmulwar, A. K. Gupta, and M. Derawi, “Challenges of IoT in healthcare. IoT and ICT for healthcare applications,” pp. 11–20, 2020.
- [83] S. U. A. Laghari, W. Li, S. Manickam, P. Nanda, A. K. Al-Ani, and S. Karuppayah, “Securing MQTT ecosystem: Exploring vulnerabilities, mitigations, and future trajectories,” *IEEE Access*, vol. 12, pp. 139 273–139 289, 2024.
- [84] P. Bajpayi, S. Sharma, and M. S. Gaur, “AI driven IoT healthcare devices security vulnerability management,” in *2024 2nd International Conference on Disruptive Technologies (ICDT)*. IEEE, 2024, pp. 366–373.
- [85] E. Longo and A. E. Redondi, “Design and implementation of an advanced MQTT broker for distributed pub/sub scenarios,” *Computer Networks*, vol. 224, p. 109601, 2023.
- [86] G. Muhammad, F. Alshehri, F. Karray, A. El Saddik, M. Alsulaiman, and T. H. Falk, “A comprehensive survey on multimodal medical signals fusion for smart healthcare systems,” *Information Fusion*, vol. 76, pp. 355–375, 2021.
- [87] A. PS, S. Dilip Kumar, and V. KR, “Mqtt implementations, open issues, and challenges: A detailed comparison and survey,” *International Journal of Sensors Wireless Communications and Control*, vol. 12, no. 8, pp. 553–576, 2022.
- [88] M. O. Al Enany, H. M. Harb, and G. Attiya, “A comparative analysis of MQTT and IoT application protocols,” in *2021 International Conference on Electronic Engineering (ICEEM)*. IEEE, 2021, pp. 1–6.

- [89] D. Shanmugapriya, A. Patel, G. Srivastava, and J. C.-W. Lin, "Mqtt protocol use cases in the Internet of Things," in *Big Data Analytics: 9th International Conference, BDA 2021, Virtual Event, December 15-18, 2021, Proceedings 9*. Springer, 2021, pp. 146–162.
- [90] D. B. Ansari, A.-U. Rehman, and R. Ali, "Internet of things (Iot) protocols: a brief exploration of MQTT and CoAP," *International Journal of Computer Applications*, vol. 179, no. 27, pp. 9–14, 2018.
- [91] "MQTT - the standard for IoT messaging," <https://mqtt.org/>, accessed: 2024-1-6.
- [92] R. Giambona, A. E. C. Redondi, and M. Cesana, "MQTT+: Enhanced syntax and broker functionalities for data filtering, processing and aggregation," *14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet'18)*, 2018.
- [93] O. Standard, "Mqtt version 5.0," *Retrieved June*, vol. 22, no. 2020, p. 1435, 2019.
- [94] M. A. Tariq, M. Khan, M. T. Raza Khan, and D. Kim, "Enhancements and challenges in CoAP—a survey," *Sensors*, vol. 20, no. 21, p. 6391, 2020.
- [95] O. Standard, "Mqtt version 3.1. 1," *URL <http://docs.oasis-open.org/mqtt/mqtt/v3>*, vol. 1, p. 29, 2014.
- [96] Z. Kang, R. Canady, A. Dubey, A. Gokhale, S. Shekhar, and M. Sedlacek, "A study of publish/subscribe middleware under different IoT traffic conditions," in *Proceedings of the International Workshop on Middleware and Applications for the Internet of Things*, 2020, pp. 7–12.
- [97] J. Nam, Y. Jun, and M. Choi, "High performance IoT cloud computing framework using pub/sub techniques," *Applied Sciences*, vol. 12, no. 21, p. 11009, 2022.
- [98] N. Ferraz Junior, A. A. Silva, A. E. Guelfi, and S. T. Kofuji, "Performance evaluation of publish-subscribe systems in IoT using energy-efficient and context-aware secure messages," *Journal of Cloud Computing*, vol. 11, no. 1, p. 6, 2022.
- [99] S. Gruener, H. Koziolk, and J. Rückert, "Towards resilient iot messaging: an experience report analyzing MQTT brokers," in *2021 IEEE 18th International Conference on Software Architecture (ICSA)*. IEEE, 2021, pp. 69–79.
- [100] J. Kotak, A. Shah, A. Shah, and P. Rajdev, "A comparative analysis on security of MQTT brokers," in *2nd Smart Cities Symposium (SCS 2019)*. Institution of Engineering and Technology, 2019, pp. 7 (5 pp.)–7 (5 pp.).
- [101] "Eclipse mosquitto," <https://mosquitto.org/>, Jan. 2018, accessed: 2024-9-11.
- [102] "HiveMQ community edition," <https://github.com/hivemq/hivemq-community-edition/wiki>, accessed: 2024-1-23.
- [103] "EMQX overview," <https://docs.emqx.com/en/emqx/latest/>, accessed: 2024-1-6.
- [104] "Welcome," <https://docs.vernemq.com/>, accessed: 2024-1-6.

- [105] “What is AWS IoT?” <https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>, accessed: 2024-1-23.
- [106] “MQTT topics, wildcards, & best practices – MQTT essentials: Part 5,” <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>, Aug. 2019.
- [107] J. Moen, “Topic alias - MQTT 5.0 new features,” <https://www.emqx.com/en/blog/mqtt5-topic-alias>, accessed: 2024-1-6.
- [108] G. C. Hillar, *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd, 2017.
- [109] V. Thirupathi and K. Sagar, “A survey on MQTT bridges, challenges and its solutions,” in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*. IEEE, 2022, pp. 58–62.
- [110] A. Detti, L. Funari, and N. Blefari-Melazzi, “Sub-linear scalability of MQTT clusters in topic-based publish-subscribe applications,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1954–1968, 2020.
- [111] M. Houimli, L. Kahloul, and S. Benaoun, “Formal specification, verification and evaluation of the mqtt protocol in the Internet of Things,” in *2017 International conference on mathematics and information technology (ICMIT)*. IEEE, 2017, pp. 214–221.
- [112] E. Shahri, P. Pedreiras, and L. Almeida, “Enhancing MQTT with real-time and reliable communication services,” in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*. IEEE, 2021, pp. 1–6.
- [113] M. Jazzar and M. Hamad, “An analysis study of IoT and DoS attack perspective,” in *Algorithms for Intelligent Systems*. Singapore: Springer Nature Singapore, 2022, pp. 127–142.
- [114] B. Bendele and D. Akopian, “A study of IoT MQTT control packet behavior and its effect on communication delays,” *Electronic Imaging*, vol. 29, pp. 120–129, 2017.
- [115] R. Wadullah Tareq and T. Ahmed Khaleel, “Implementation of MQTT protocol in health care based on IoT systems: a study,” *International journal of electrical and computer engineering systems*, vol. 12, no. 4, pp. 215–223, 2021.
- [116] S. S. Kumar, D. P. Rajan, and Y. M. Iggalore, “Mqtt: As default, secured protocol for IoT communication and its practical implementation,” in *Internet of Things*. CRC Press, 2020, pp. 189–208.
- [117] N. I. Jaya and M. F. Hossain, “A prototype air flow control system for home automation using MQTT over websocket in aws IoT core,” in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2018, pp. 111–1116.

- [118] S. Nazir and M. Kaleem, “Reliable image notifications for smart home security with MQTT,” in *2019 International Conference on Information Science and Communication Technology (ICISCT)*. IEEE, 2019, pp. 1–5.
- [119] Y. Kyung, T. K. Kim, and Y. Kim, “Retained message delivery scheme utilizing reinforcement learning in MQTT-based IoT networks,” *Journal of Internet of Things and Convergence*, vol. 10, no. 2, pp. 131–135, 2024.
- [120] N. H. Motlagh, M. A. Zaidan, R. Morabito, P. Nurmi, and S. Tarkoma, “Towards large-scale IoT deployments in smart cities: Requirements and challenges,” in *Learning Techniques for the Internet of Things*. Cham: Springer Nature Switzerland, 2024, pp. 105–129.
- [121] S. Tarkoma, *Publish/subscribe systems: design and principles*. John Wiley & Sons, 2012.
- [122] Y. Teranishi, R. Banno, and T. Akiyama, “Scalable and locality-aware distributed topic-based pub/sub messaging for IoT,” in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015.
- [123] D. Happ and A. Wolisz, “Limitations of the Pub/Sub pattern for cloud based IoT and their implications,” in *2016 Cloudification of the Internet of Things (CIoT)*. IEEE, 2016.
- [124] S. Coroller, S. Chabridon, M. Laurent, D. Conan, and J. Leneutre, “Position paper: Towards end-to-end privacy for publish/subscribe architectures in the Internet of Things,” in *Proceedings of the 5th Workshop on Middleware and Applications for the Internet of Things*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 35–40.
- [125] M. Noura, M. Atiquzzaman, and M. Gaedke, “Interoperability in Internet of Things: Taxonomies and open challenges,” *Mob. Netw. Appl.*, vol. 24, no. 3, pp. 796–809, 2019.
- [126] M. Diyan, B. Nathali Silva, J. Han, Z. Cao, and K. Han, “Intelligent Internet of Things gateway supporting heterogeneous energy data management and processing,” *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 2, 2022.
- [127] O. Logvinov, B. Kraemer, C. Adams, J. Heiles, G. Stuebing, M. Nielsen, and B. Mancuso, “Standard for an architectural framework for the Internet of Things (IoT) ieee p2413,” *IEEE-P2413 Working Group. Technical Report*, 2016.
- [128] I. Ishaq, D. Carels, G. Teklemariam, J. Hoebeke, F. Abeele, E. Poorter, I. Moerman, and P. Demeester, “IETF standardization in the field of the Internet of Things (IoT): A survey,” *J. Sens. Actuator Netw.*, vol. 2, no. 2, pp. 235–287, 2013.
- [129] V. Gazis, “A survey of standards for machine-to-machine and the Internet of Things,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 482–511, 2017.
- [130] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, “A review of security standards and frameworks for IoT-based smart environments,” *IEEE Access*, vol. 9, pp. 121 975–121 995, 2021.

- [131] P. Bellavista, A. Corradi, and A. Reale, “Quality of service in wide scale publish—subscribe systems,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 3, pp. 1591–1616, 2014.
- [132] P. Colombo, E. Ferrari, and E. D. Tümer, “Regulating data sharing across MQTT environments,” *J. Netw. Comput. Appl.*, vol. 174, no. 102907, p. 102907, 2021.
- [133] D. Happ, N. Karowski, T. Menzel, V. Handziski, and A. Wolisz, “Meeting IoT platform requirements with open pub/sub solutions,” *Ann. Telecommun.*, vol. 72, no. 1-2, pp. 41–52, 2017.
- [134] B. S. Sahay and J. Ranjan, “Real time business intelligence in supply chain analytics,” *Inf. Manage. Comput. Secur.*, vol. 16, no. 1, pp. 28–48, 2008.
- [135] E. Atilgan, I. Ozcelik, and E. N. Yolacan, “MQTT security at a glance,” in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*. IEEE, 2021.
- [136] S. Yuvaraj, M. Manigandan, V. Dhandapani, R. Saajid, and S. Nikhilesh, “Internet of things integrated with multi-level authentication for secured IoT data stream through TLS/SSL layer,” in *Big-Data-Analytics in Astronomy, Science, and Engineering*. Cham: Springer International Publishing, 2022, pp. 245–258.
- [137] A. Hue, G. Sharma, and J.-M. Dricot, “Privacy-enhanced MQTT protocol for massive IoT,” *Electronics (Basel)*, vol. 11, no. 1, p. 70, 2021.
- [138] S. K. Matharu, “Exploiting SSL/TLS vulnerabilities in modern technologies,” University of Alberta, Tech. Rep., jan 2021.
- [139] C. Butpheng, K.-H. Yeh, and H. Xiong, “Security and privacy in IoT-cloud-based e-health systems—a comprehensive review,” *Symmetry (Basel)*, vol. 12, no. 7, p. 1191, 2020.
- [140] A. Corallo, M. Lazoi, M. Lezzi, and A. Luperto, “Cybersecurity awareness in the context of the industrial Internet of Things: A systematic literature review,” *Comput. Ind.*, vol. 137, no. 103614, p. 103614, 2022.
- [141] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, “IoT security: Ongoing challenges and research opportunities,” in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*. IEEE, 2014.
- [142] N. Miloslavskaya, A. Nikiforov, K. Plaksiy, and A. Tolstoy, “Standardization issues for the Internet of Things,” in *New Knowledge in Information Systems and Technologies: Volume 2*. Springer, 2019, pp. 328–338.
- [143] N. C. Mandal, G. Shahariar, and M. T. R. Shawon, “Effectiveness of transformer models on IoT security detection in stackoverflow discussions,” in *Proceedings of International Conference on Information and Communication Technology for Development: ICICTD 2022*. Springer, 2023, pp. 125–137.

- [144] K. Ueda, C. Sasaki, and A. Tagami, "Pub/sub meets MLS: End-to-end encrypted group data sharing over publish-subscribe," in *2024 IFIP Networking Conference (IFIP Networking)*, vol. 12. IEEE, 2024, pp. 1–6.
- [145] H. Kurdi and V. Thayananthan, "Authentication mechanisms for IoT system based on distributed MQTT brokers: review and challenges," *Procedia Comput. Sci.*, vol. 194, pp. 132–139, 2021.
- [146] P. Kalpana Devi, M. Manasa, S. N. C. Prakash, and B. V. Teja, "An OAuth-based authentication system for IoT networks using LabVIEW," in *Sustainable Communication Networks and Application*. Singapore: Springer Nature Singapore, 2022, pp. 621–628.
- [147] B. Daddala, H. Wang, and A. Y. Javaid, "Design and implementation of a customized encryption algorithm for authentication and secure communication between devices," in *2017 IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2017.
- [148] S. Karthikeyan, R. Patan, and B. Balamurugan, "Enhancement of security in the internet of things (IoT) by using x.509 authentication mechanism," in *Lecture Notes in Electrical Engineering*. Singapore: Springer Singapore, 2019, pp. 217–225.
- [149] A. Bhawiyuga, M. Data, and A. Warda, "Architectural design of token based authentication of MQTT protocol in constrained IoT device," in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*. IEEE, 2017.
- [150] C. Newman, A. Menon-Sen, A. Melnikov, and N. Williams, "Salted challenge response authentication mechanism (scram) SASL and GSS-API mechanisms," Internet Engineering Task Force (IETF), Tech. Rep., 2010.
- [151] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Security of Internet of Things based on cryptographic algorithms: a survey," *Wirel. Netw.*, vol. 27, no. 2, pp. 1515–1555, 2021.
- [152] "MQTT security fundamentals," <https://www.hivemq.com/mqtt/mqtt-security-fundamentals/>.
- [153] S. Suhail, R. Hussain, A. Khan, and C. S. Hong, "On the role of hash-based signatures in quantum-safe Internet of Things: Current solutions and future directions," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 1–17, 2021.
- [154] "MQTT message data integrity - MQTT security fundamentals," <https://www.hivemq.com/blog/mqtt-security-fundamentals-mqtt-message-data-integrity/>, Jun. 2015.
- [155] M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai, "Analysis of vulnerabilities in MQTT security using shodan API and implementation of its countermeasures via authentication and ACLs," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018.

- [156] S. Bhatt, F. Patwa, and R. Sandhu, "Access control model for AWS Internet of Things," in *Network and System Security*. Cham: Springer International Publishing, 2017, pp. 721–736.
- [157] S. Nakamura, L. Ogiela, T. Enokido, and M. Takizawa, "An information flow control model in a topic-based publish/subscribe system," *J. High Speed Netw.*, vol. 24, no. 3, pp. 243–257, 2018.
- [158] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38 431–38 441, 2019.
- [159] S. Pal, M. Hitchens, V. Varadharajan, and T. Rabehaja, "Policy-based access control for constrained healthcare resources in the context of the Internet of Things," *J. Netw. Comput. Appl.*, vol. 139, pp. 57–74, 2019.
- [160] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT privacy and security: Challenges and solutions," *Appl. Sci. (Basel)*, vol. 10, no. 12, p. 4102, 2020.
- [161] W.-L. Chin, W. Li, and H.-H. Chen, "Energy big data security threats in IoT-based smart grid communications," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 70–75, 2017.
- [162] Y. Meng, W. Zhang, H. Zhu, and X. S. Shen, "Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures," *IEEE Wirel. Commun.*, vol. 25, no. 6, pp. 53–59, 2018.
- [163] A. N. Doss, D. Shah, G. F. Smaisim, M. Olha, and S. Jaiswal, "A comprehensive analysis of internet of things (IOT) in enhancing data security for better system integrity - a critical analysis on the security attacks and relevant countermeasures," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2022.
- [164] S. Siby, R. R. Maiti, and N. O. Tippenhauer, "Iotscanner: Detecting privacy threats in iot neighborhoods," in *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*. New York, NY, USA: ACM, 2017.
- [165] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing IoTs in distributed blockchain: Analysis, requirements and open issues," *Future Gener. Comput. Syst.*, vol. 100, pp. 325–343, 2019.
- [166] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-based authentication scheme for the Internet of Things," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016.
- [167] S. Tu, M. Waqas, S. U. Rehman, M. Aamir, O. U. Rehman, Z. Jianbiao, and C.-C. Chang, "Security in fog computing: A novel technique to tackle an impersonation attack," *IEEE Access*, vol. 6, pp. 74 993–75 001, 2018.
- [168] S. Nakhodchi, A. Dehghantanha, and H. Karimipour, "Privacy and security in smart and precision farming: A bibliometric analysis," in *Handbook of Big Data Privacy*. Cham: Springer International Publishing, 2020, pp. 305–318.

- [169] M. Torky and A. E. Hassanein, “Integrating blockchain and the Internet of Things in precision agriculture: Analysis, opportunities, and challenges,” *Comput. Electron. Agric.*, vol. 178, no. 105476, p. 105476, 2020.
- [170] A. Hussain, T. Ali, F. Althobiani, U. Draz, M. Irfan, S. Yasin, S. Shafiq, Z. Safdar, A. Glowacz, G. Nowakowski, M. S. Khan, and S. Alqhtani, “Security framework for IoT based real-time health applications,” *Electronics (Basel)*, vol. 10, no. 6, p. 719, 2021.
- [171] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (coap)-no. rfc7252.” Internet Engineering Task Force (IETF), Tech. Rep., 2014.
- [172] “SSL/TLS strong encryption: An introduction,” https://httpd.apache.org/docs/current/ssl/ssl_intro.html.
- [173] K. Hughes-Lartey, M. Li, F. E. Botchey, and Z. Qin, “Human factor, a critical weak point in the information security of an organization’s Internet of Things,” *Heliyon*, vol. 7, no. 3, p. e06522, 2021.
- [174] N. Abughazaleh, R. Bin, M. Btish, and Hemalatha, “DoS attacks in IoT systems and proposed solutions,” *Int. J. Comput. Appl.*, vol. 176, no. 33, pp. 16–19, 2020.
- [175] N. Koroniotis, N. Moustafa, and E. Sitnikova, “A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework,” *Future Gener. Comput. Syst.*, vol. 110, pp. 91–106, 2020.
- [176] M. H. Ali, M. M. Jaber, S. K. Abd, A. Rehman, M. J. Awan, R. Damaševičius, and S. A. Bahaj, “Threat analysis and distributed denial of service (DDoS) attack recognition in the Internet of Things (IoT),” *Electronics (Basel)*, vol. 11, no. 3, p. 494, 2022.
- [177] N. Ravi and S. M. Shalinie, “Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, 2020.
- [178] R. Vishwakarma and A. K. Jain, “A survey of DDoS attacking techniques and defence mechanisms in the IoT network,” *Telecommun. Syst.*, vol. 73, no. 1, pp. 3–25, 2020.
- [179] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network intrusion detection for IoT security based on learning techniques,” *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, 2019.
- [180] G. De La Torre Parra, P. Rad, K.-K. R. Choo, and N. Beebe, “Detecting Internet of Things attacks using distributed deep learning,” *J. Netw. Comput. Appl.*, vol. 163, no. 102662, p. 102662, 2020.
- [181] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, “Towards sflow and adaptive polling sampling for deep learning based DDoS detection in SDN,” *Future Gener. Comput. Syst.*, vol. 111, pp. 763–779, 2020.

- [182] O. Salem, K. Alsubhi, A. Shaafi, M. Gheryani, A. Mehaoua, and R. Boutaba, "Man-in-the-middle attack mitigation in Internet of Medical Things," *IEEE Trans. Industr. Inform.*, vol. 18, no. 3, pp. 2053–2062, 2022.
- [183] S. Thavamani, D. Mahesh, U. Sinthuja, and G. Manoharan, "Crucial attacks in internet of things via artificial intelligence techniques: The security survey," in *INTERNATIONAL CONFERENCE ON RESEARCH IN SCIENCES, ENGINEERING & TECHNOLOGY*. AIP Publishing, 2022.
- [184] X. de Carné de Carnavalet and P. C. van Oorschot, "A survey and analysis of TLS interception mechanisms and motivations: Exploring how end-to-end TLS is made "end-to-me" for web traffic," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–40, 2023.
- [185] D. Diaz-Sanchez, A. Marin-Lopez, F. A. Mendoza, P. A. Cabarcos, and R. S. Sherratt, "TLS/PKI challenges and certificate pinning techniques for IoT and M2M secure communications," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3502–3531, 2019.
- [186] A. Karale, "The challenges of IoT addressing security, ethics, privacy, and laws," *Internet of Things*, vol. 15, no. 100420, p. 100420, 2021.
- [187] M. Azrour, J. Mabrouki, A. Guezzaz, and A. Kanwal, "Internet of Things security: Challenges and key issues," *Secur. Commun. Netw.*, vol. 2021, pp. 1–11, 2021.
- [188] R. F. Ali, A. Muneer, P. D. D. Dominic, S. M. Taib, and E. A. A. Ghaleb, "Internet of Things (IoT) security challenges and solutions: A systematic literature review," in *Communications in Computer and Information Science*. Singapore: Springer Singapore, 2021, pp. 128–154.
- [189] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W.-C. Hong, "Internet of Things: Evolution, concerns and security challenges," *Sensors (Basel)*, vol. 21, no. 5, p. 1809, 2021.
- [190] M. I. Adawy, M. Tahboush, O. Aloqaily, and W. Abdulraheem, "Man-in-the-Middle attack detection scheme on data aggregation in wireless sensor networks." *International Journal of Advances in Soft Computing & Its Applications*, vol. 15, no. 2, 2023.
- [191] M. R. F. Eslava, J. C. H. Lozada, M. H. Bolaños, and J. S. Gutiérrez, "Firewall system for the Internet of Things," in *Communications in Computer and Information Science*. Cham: Springer Nature Switzerland, 2023, pp. 73–85.
- [192] S. Saif, P. Das, and S. Biswas, "Lsea-iomt: On the Implementation of Lightweight Symmetric Encryption Algorithm for Internet of Medical Things (IoMT)," in *Frontiers of ICT in Healthcare: Proceedings of EAIT 2022*. Springer, 2023, pp. 565–575.
- [193] P. J. Denning, *Computers under attack: intruders, worms, and viruses*. New York: ACM Press, 1990.
- [194] M. Aqeel, F. Ali, M. W. Iqbal, T. A. Rana, M. Arif, and M. R. Auwal, "A review of security and privacy concerns in the Internet of Things (IoT)," *J. Sens.*, vol. 2022, pp. 1–20, 2022.

- [195] A. Gazet, “Comparative analysis of various ransomware virii,” *J. Comput. Virol.*, vol. 6, no. 1, pp. 77–90, 2010.
- [196] Maneesha, Savitha, Jeevika, Nithiskumar, and Sangeetha, “Deep learning approach for intelligent intrusion detection system,” *International Research Journal on Advanced Science Hub*, vol. 3, no. SpecialICARD 3S, pp. 45–48, 2021.
- [197] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and A. Ribagorda, “Evolution, detection and analysis of malware for smart devices,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 2, pp. 961–987, 2014.
- [198] M. J. Farooq and Q. Zhu, “Modeling, analysis, and mitigation of dynamic botnet formation in wireless IoT networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 9, pp. 2412–2426, 2019.
- [199] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, “Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations,” *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2702–2733, 2019.
- [200] L. Arnaboldi, R. M. Czekster, C. Morisset, and R. Metere, “Modelling load-changing attacks in cyber-physical systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 353, pp. 39–60, 2020.
- [201] D. Acarali, M. Rajarajan, N. Komninos, and B. B. Zarpelão, “Modelling the spread of botnet malware in IoT-based wireless sensor networks,” *Secur. Commun. Netw.*, vol. 2019, pp. 1–13, 2019.
- [202] E. Rattanalerdnusorn, M. Pattaranantakul, P. Thaenkaew, and C. Vorakulpipat, “IoTDePT: Detecting security threats and pinpointing anomalies in an IoT environment,” in *Proceedings of the 2020 9th International Conference on Software and Computer Applications*. New York, NY, USA: ACM, 2020.
- [203] X. Liu, X. Du, X. Zhang, Q. Zhu, H. Wang, and M. Guizani, “Adversarial samples on android malware detection systems for IoT systems,” *Sensors (Basel)*, vol. 19, no. 4, p. 974, 2019.
- [204] A. Rodriguez-Mota, P. J. Escamilla-Ambrosio, J. Happa, and J. R. C. Nurse, “Towards IoT cybersecurity modeling: From malware analysis data to IoT system representation,” in *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2016.
- [205] A. Mosenia and N. K. Jha, “A comprehensive study of security of Internet-of-Things,” *IEEE Trans. Emerg. Top. Comput.*, vol. 5, no. 4, pp. 586–602, 2017.
- [206] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, “Securing the internet of things (IoT): A security taxonomy for IoT,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018.

- [207] D. Dinculeană and X. Cheng, “Vulnerabilities and limitations of MQTT protocol used between IoT devices,” *Appl. Sci. (Basel)*, vol. 9, no. 5, p. 848, 2019.
- [208] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, “SpiderCast: A scalable interest-aware overlay for topic-based pub/sub communication,” in *Proceedings of the 2007 inaugural international conference on Distributed event-based systems*. New York, NY, USA: ACM, 2007.
- [209] J. Simla., R. Chakravarthy, and M. Leo., “An experimental study of IoT-Based topologies on MQTT protocol for agriculture intrusion detection,” *Measur. Sens.*, vol. 24, no. 100470, p. 100470, 2022.
- [210] E. Onica, P. Felber, H. Mercier, and E. Rivière, “Confidentiality-preserving publish/subscribe: A survey,” *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–43, 2017.
- [211] B. Mishra, “Performance evaluation of MQTT broker servers,” in *Computational Science and Its Applications – ICCSA 2018*. Cham: Springer International Publishing, 2018, pp. 599–609.
- [212] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, “A survey on access control in the age of Internet of Things,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, 2020.
- [213] Y. Sharaf-Dabbagh and W. Saad, “On the authentication of devices in the Internet of Things,” in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2016.
- [214] I. Yaqoob, E. Ahmed, M. H. u. Rehman, A. I. A. Ahmed, M. A. Al-garadi, M. Imran, and M. Guizani, “The rise of ransomware and emerging security challenges in the Internet of Things,” *Comput. Netw.*, vol. 129, pp. 444–458, 2017.
- [215] Y. Pan, J. White, D. Schmidt, A. Elhabashy, L. Sturm, J. Camelio, and C. Williams, “Taxonomies for reasoning about cyber-physical attacks in IoT-based manufacturing systems,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 3, p. 45, 2017.
- [216] I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, “Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices,” *Sensors (Basel)*, vol. 24, no. 12, p. 4008, 2024.
- [217] P. Gope and B. Sikdar, “Lightweight and privacy-preserving two-factor authentication scheme for IoT devices,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, 2019.
- [218] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, “Certificate-based anonymous device access control scheme for IoT environment,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, 2019.
- [219] C. L. Lin, H. M. Sun, and T. Hwang, “Three-party encrypted key exchange: attacks and a solution,” *ACM SIGOPS Operating Systems Review*, vol. 34, no. 4, pp. 12–20, 2000.

- [220] G. Padmavathi, “DSSS with ISAKMP key management protocol to secure physical layer for mobile adhoc network,” *Int. J. Netw. Secur. Appl.*, vol. 4, no. 1, pp. 69–76, 2012.
- [221] H. M. Kamali, K. Z. Azar, S. Roshanisefat, A. Vakil, H. Homayoun, and A. Sasan, “ExTru: A lightweight, fast, and secure expirable trust for the Internet of Things,” in *2020 IEEE 14th Dallas Circuits and Systems Conference (DCAS)*. IEEE, 2020.
- [222] P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, “Internet of Things: Security and solutions survey,” *Sensors (Basel)*, vol. 22, no. 19, p. 7433, 2022.
- [223] O. I. Abiodun, E. O. Abiodun, M. Alawida, R. S. Alkhawaldeh, and H. Arshad, “A review on the security of the Internet of Things: Challenges and solutions,” *Wirel. Pers. Commun.*, vol. 119, no. 3, pp. 2603–2637, 2021.
- [224] M. Castañeda, Á. Luis, J. Antonio Aveleira, and H. Mata, “Characterization of threats in IoT from an MQTT protocol-oriented dataset,” *Complex & Intelligent Systems*, vol. 9, no. 5, pp. 5281–5296, 2023.
- [225] E. Džaferović, A. Sokol, A. A. Almisreb, and S. Mohd Norzeli, “DoS and DDoS vulnerability of IoT: A review,” *Sustainable Engineering and Innovation*, vol. 1, no. 1, pp. 43–48, 2019.
- [226] Z. Zhou, A. Gaurav, B. B. Gupta, H. Hamdi, and N. Nadjah, “A statistical approach to secure health care services from DDoS attacks during COVID-19 pandemic,” *Neural Comput. Appl.*, pp. 1–14, 2021.
- [227] L. R. Baratta, D. Harford, C. A. Sinsky, T. Kannampallil, and S. S. Lou, “Characterizing the patterns of electronic health record-integrated secure messaging use: Cross-sectional study,” *J. Med. Internet Res.*, vol. 25, p. e48583, 2023.
- [228] T. Prantl, L. Iffländer, S. Herrnleben, S. Engel, S. Kounev, and C. Krupitzer, “Performance impact analysis of securing MQTT using TLS,” in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*. New York, NY, USA: ACM, 2021.
- [229] C.-Y. Chen, G.-L. Hung, and H.-Y. Hsieh, “A study on a new type of DDoS attack against 5G ultra-reliable and low-latency communications,” in *2020 European Conference on Networks and Communications (EuCNC)*. IEEE, 2020.
- [230] R. Buenrostro-Mariscal, P. C. Santana-Mancilla, O. A. Montesinos-López, M. Vazquez-Briseno, and J. I. Nieto-Hipolito, “Prioritization-driven congestion control in networks for the Internet of Medical Things: A cross-layer proposal,” *Sensors (Basel)*, vol. 23, no. 2, p. 923, 2023.
- [231] N. J. Palatty, “How many cyber attacks per day: The latest stats and impacts in 2024,” <https://www.getastra.com/blog/security-audit/how-many-cyber-attacks-per-day/>, Dec. 2023, accessed: 2024-2-11.
- [232] Q. Wang, S. Ji, Y. Tian, X. Zhang, B. Zhao, Y. Kan, Z. Lin, C. Lin, S. Deng, A. X. Liu *et al.*, “MPInspector: A systematic and automatic approach for evaluating

- the security of IoT messaging protocols,” in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX, 2021, pp. 4205–4222.
- [233] Y. Shah and S. Sengupta, “A survey on classification of cyber-attacks on IoT and IIoT devices,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2020.
- [234] S. S. Ambarkar and N. Shekokar, “Critical and comparative analysis of DoS and version number attack in healthcare IoT system,” in *Proceeding of First Doctoral Symposium on Natural Computing Research*. Singapore: Springer Singapore, 2021, pp. 301–312.
- [235] Z. Alwaisi, S. Soderi, and R. De Nicola, “Detection of energy consumption cyber attacks on smart devices,” *Springer*, pp. 160–176, 2024.
- [236] Z. Li, T. Sen, H. Shen, and M. C. Chuah, “A study on the impact of memory DoS attacks on cloud applications and exploring real-time detection schemes,” *IEEE ACM Trans. Netw.*, vol. 30, no. 4, pp. 1644–1658, 2022.
- [237] A. R. Alkhafajee, A. M. A. Al-Muqarm, A. H. Alwan, and Z. R. Mohammed, “Security and performance analysis of MQTT protocol with TLS in IoT networks,” in *2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA)*. IEEE, 2021, pp. 206–211.
- [238] Y. Al-Hadhrami and F. K. Hussain, “DDoS attacks in IoT networks: a comprehensive systematic literature review,” *World Wide Web*, vol. 24, no. 3, pp. 971–1001, 2021.
- [239] M. Goworko and J. Wytrebowicz, “A secure communication system for constrained IoT devices-experiences and recommendations,” *Sensors (Basel)*, vol. 21, no. 20, p. 6906, 2021.
- [240] K. Sowjanya and M. Dasgupta, “Survey of symmetric and asymmetric key management schemes in the context of IoT based healthcare system,” in *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*. IEEE, 2020.
- [241] D. Swessi and H. Idoudi, “A survey on Internet-of-Things security: Threats and emerging countermeasures,” *Wirel. Pers. Commun.*, vol. 124, no. 2, pp. 1557–1592, 2022.
- [242] A. Ghasempour, “Internet of Things in smart grid: Architecture, applications, services, key technologies, and challenges,” *Inventions*, vol. 4, no. 1, p. 22, 2019.
- [243] J. Deogirikar and A. Vidhate, “Security attacks in IoT: A survey,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, 2017.
- [244] R. Sharma and R. Arya, “Security threats and measures in the Internet of Things for smart city infrastructure: A state of art,” *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 11, 2023.

- [245] A. Tewari and B. B. Gupta, "Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework," *Future Gener. Comput. Syst.*, vol. 108, pp. 909–920, 2020.
- [246] L. S. Abdulla, M. K. Mahmood, A. F. Salih, and S. M. Karim, "Analysis and evaluation of symmetrical key ciphers for Internet of Things smart home," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, p. 1191, 2021.
- [247] V. O. Nyangaresi, M. Ahmad, A. Alkhayyat, and W. Feng, "Artificial neural network and symmetric key cryptography based verification protocol for 5G enabled Internet of Things," *Expert Syst.*, vol. 39, no. 10, 2022.
- [248] Z. Ashraf, A. Sohail, and M. Yousaf, "Robust and lightweight symmetric key exchange algorithm for next-generation IoE," *Internet of Things*, vol. 22, no. 100703, p. 100703, 2023.
- [249] A. Sarkar, "A symmetric neural cryptographic key generation scheme for IoT security," *Appl. Intell.*, 2022.
- [250] G. C. C. F. Pereira, R. C. A. Alves, F. L. d. Silva, R. M. Azevedo, B. C. Albertini, and C. B. Margi, "Performance evaluation of cryptographic algorithms over IoT platforms and operating systems," *Secur. Commun. Netw.*, vol. 2017, pp. 1–16, 2017.
- [251] P. Kumar and B. Dezfouli, "Implementation and analysis of QUIC for MQTT," *Comput. Netw.*, vol. 150, pp. 28–45, 2019.
- [252] A. Varghese, N. Narendra, M. Singh, S. Vl, R. Ma, M. G. Chandra, and P. Balamuralidhar, "SMART: secure mobile augmented reality for tele-assistance," in *2015 Asia Pacific Conference on Multimedia and Broadcasting*. IEEE, 2015.
- [253] A. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, no. 1, 2017.
- [254] L. Bisne and M. Parmar, "Composite secure MQTT for Internet of Things using ABE and dynamic s-box AES," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE, 2017.
- [255] O. Yerlikaya and G. Dalkiltc, "Authentication and authorization mechanism on Message Queue Telemetry Transport protocol," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2018.
- [256] A. Rhbech, H. Lotfi, A. Bajit, A. Barodi, S. El Aidi, and A. Tamtaoui, "An optimized and intelligent security-based message queuing protocol S-MQTT applied to medical IOT COVID-19 DATA monitoring platforms," in *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*. IEEE, 2020.
- [257] X. Zhou and X. Tang, "Research and implementation of RSA algorithm for encryption and decryption," in *Proceedings of 2011 6th International Forum on Strategic Technology*. IEEE, 2011.

- [258] A. Varma and S. UniKrishnan, "Effect of payload security in MQTT protocol over transport and application layer," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1166, no. 1, p. 012019, 2021.
- [259] Shilpa, Vidya, and S. Pattar, "MQTT based secure transport layer communication for mutual authentication in IoT network," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 60–66, 2022.
- [260] S. Amanlou and K. A. A. Bakar, "Lightweight security mechanism over MQTT protocol for IoT devices," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020.
- [261] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, "Report on the development of the advanced encryption standard (AES)," *J. Res. Natl. Inst. Stand. Technol.*, vol. 106, no. 3, pp. 511–577, 2001.
- [262] S. Iyer, G. V. Bansod, P. Naidu, and S. Garg, "Implementation and evaluation of lightweight ciphers in MQTT environment," in *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*. IEEE, 2018.
- [263] N. Alassaf, A. Gutub, S. A. Parah, and M. Al Ghamdi, "Enhancing speed of SIMON: A light-weight-cryptographic algorithm for IoT applications," *Multimed. Tools Appl.*, vol. 78, no. 23, pp. 32 633–32 657, 2019.
- [264] R. A. F. Lusto, A. M. Sison, and R. P. Medina, "Performance analysis of enhanced SPECK algorithm," in *Proceedings of the 4th International Conference on Industrial and Business Engineering*. New York, NY, USA: ACM, 2018.
- [265] O. Sadio, I. Ngom, and C. Lishou, "Lightweight security scheme for MQTT/MQTT-SN protocol," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE, 2019.
- [266] O. Sabri, B. Al-Shargabi, A. Abuarqoub, and T. A. Hakami, "A lightweight encryption method for IoT-based healthcare applications: A review and future prospects," *Comput. Syst.*, vol. 157, pp. 288–302, 2024.
- [267] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Viskelson, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.
- [268] M. Imdad, S. N. Ramli, and H. Mahdin, "An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys," *Symmetry (Basel)*, vol. 14, no. 3, p. 604, 2022.
- [269] I. Sahmi, A. Abdellaoui, T. Mazri, and N. Hmina, "MQTT-PRESENT: Approach to secure Internet of Things applications using MQTT protocol," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 11, no. 5, p. 4577, 2021.

- [270] K. K. Coelho, M. Nogueira, M. C. Marim, E. F. Silva, A. B. Vieira, and J. A. M. Nacif, "LORENA: Low memORy symmetric-key geNerAtion method for based on group cryptography protocol applied to the Internet of healthcare Things," *IEEE Access*, vol. 10, pp. 12 564–12 579, 2022.
- [271] N. M. Hamed and A. A. Yassin, "Secure patient authentication scheme in the health-care system using symmetric encryption," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 18, no. 1, pp. 71–81, 2022.
- [272] A. Bisht, A. K. Das, D. Niyato, and Y. Park, "Efficient personal-health-records sharing in Internet of Medical Things using searchable symmetric encryption, blockchain, and IPFS," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 2225–2244, 2023.
- [273] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–495, 2018.
- [274] Q. Feng, D. He, H. Wang, L. Zhou, and K.-K. R. Choo, "Lightweight collaborative authentication with key protection for smart electronic health record system," *IEEE Sensors Journal*, vol. 20, no. 4, pp. 2181–2196, 2019.
- [275] T. Hasija, K. R. Ramkumar, B. Singh, A. Kaur, and S. K. Mittal, "Symmetric key cryptography: Review, algorithmic insights, and challenges in the era of quantum computers," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2023, pp. 1–6.
- [276] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: A solution to secure IoT," *Wirel. Pers. Commun.*, vol. 112, no. 3, pp. 1947–1980, 2020.
- [277] V. Thirunavukkarasu, A. S. Kumar, P. Prakasam, and G. Suresh, "Elliptic curve cryptography based key management and flexible authentication scheme for 5G wireless networks," *Multimedia Tools and Applications*, vol. 82, pp. 21 131–21 145, 2023.
- [278] Z. Laaroussi and O. Novo, "A performance analysis of the security communication in CoAP and MQTT," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021.
- [279] A. A. Diro, N. Chilamkurti, and P. Veeraraghavan, "Elliptic curve based cyber-security schemes for publish-subscribe Internet of Things," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Cham: Springer International Publishing, 2017, pp. 258–268.
- [280] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over MQTT protocol a comparaisn between RSA and elliptic curve," in *2016 Third International Conference on Systems of Collaboration (SysCo)*. IEEE, 2016.
- [281] D. Kumar and M. Kumar, "A comparative analysis of elliptic curve-based cryptographic techniques for Internet of Things," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 43, no. 4, pp. 238–251, 2023.

- [282] A. E. Adeniyi, R. G. Jimoh, and J. B. Awotunde, "A systematic review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," *Comput. Electr. Eng.*, vol. 118, no. 109330, p. 109330, 2024.
- [283] M. Kumar, M. Sethi, S. Rani, D. K. Sah, S. A. AlQahtani, and M. S. Al-Rakhami, "Secure data aggregation based on end-to-end homomorphic encryption in IoT-based wireless sensor networks," *Sensors (Basel)*, vol. 23, no. 13, 2023.
- [284] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Des. Test*, vol. 34, no. 4, pp. 7–17, 2017.
- [285] A. A. Diro, N. Chilamkurti, and N. Kumar, "Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing," *Mob. Netw. Appl.*, vol. 22, no. 5, pp. 848–858, 2017.
- [286] A. Lohachab and Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," *J. Inf. Secur. Appl.*, vol. 46, pp. 1–12, 2019.
- [287] W. Elgenaidi and T. Newe, "Trust security mechanism for marine wireless sensor networks," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*. IEEE, 2015.
- [288] E. L. Sanaa, A. Bajit, and A. Barodi, "An optimized security vehicular internet of Things-IoT-application layer protocols MQTT and COAP," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. IEEE, vol. 2020, pp. 1–6.
- [289] Z. Y. M. Yusoff, M. K. Ishak, L. A. B. Rahim, and O. Ali, "Elliptic curve cryptography based security on MQTT system for smart home application," in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2022.
- [290] M. I. Ahmed and G. Kannan, "Secure end to end communications and data analytics in IoT integrated application using IBM watson IoT platform," *Wirel. Pers. Commun.*, vol. 120, no. 1, pp. 153–168, 2021.
- [291] X. Li, J. Peng, S. Kumari, F. Wu, M. Karuppiah, and K.-K. Raymond Choo, "An enhanced 1-round authentication protocol for wireless body area networks with user anonymity," *Comput. Electr. Eng.*, vol. 61, pp. 238–249, 2017.
- [292] K. Sowjanya, M. Dasgupta, and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 129–146, 2020.
- [293] Z. Zhao, C. Hsu, L. Harn, Q. Yang, and L. Ke, "Lightweight privacy-preserving data sharing scheme for internet of medical things," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 8402138, 2021.

- [294] M. Masud, G. S. Gaba, S. Alqahtani, G. Muhammad, B. B. Gupta, P. Kumar, and A. Ghoneim, "A lightweight and robust secure key establishment protocol for Internet of Medical Things in COVID-19 patients care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 694–15 703, 2020.
- [295] R. Banoth and R. Regar, *Classical and Modern Cryptography for Beginners*. Springer, 2023.
- [296] H. Aroraa, S. P. Singhb, and S. Prasadc, *Analysis of Asymmetric Cryptographic Algorithms: A Review*. Pune, India: ASM, 2023.
- [297] V. Jayaprakash and A. K. Tyagi, "Security optimization of resource-constrained internet of healthcare things (IoHT) devices using asymmetric cryptography for blockchain network," in *Proceedings of International Conference on Network Security and Blockchain Technology*. Singapore: Springer Nature Singapore, 2022, pp. 225–236.
- [298] M. E. H. Kahla, M. Beggas, A. Laouid, M. AlShaikh, and M. Hammoudeh, "An IoMT image crypto-system based on spatial watermarking and asymmetric encryption," *Multimed. Tools Appl.*, 2024.
- [299] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, pp. 28 177–28 193, 2021.
- [300] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in IoT networks: A survey," *Future Gener. Comput. Syst.*, vol. 129, pp. 77–89, 2022.
- [301] N. A. Gunathilake, A. Al-Dubai, and W. J. Buchana, "Recent advances and trends in lightweight cryptography for IoT security," in *2020 16th International Conference on Network and Service Management (CNSM)*, vol. 18. IEEE, 2020, pp. 1–5.
- [302] V. Rao and K. V. Prema, "A review on lightweight cryptography for Internet-of-Things based applications," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 9, pp. 8835–8857, 2021.
- [303] L. M. Shamala, D. G. Zayaraz, D. K. Vivekanandan, and D. V. Vijayalakshmi, "Lightweight cryptography algorithms for Internet of Things enabled networks: An overview," *J. Phys. Conf. Ser.*, vol. 1717, p. 012072, 2021.
- [304] Z. Dewamuni, B. Shanmugam, S. Azam, and S. Thennadil, "Bibliometric analysis of iot lightweight cryptography," *Information*, vol. 14, no. 12, p. 635, 2023.
- [305] A. Fotovvat, G. M. E. Rahman, S. S. Vedaiei, and K. A. Wahid, "Comparative performance analysis of lightweight cryptography algorithms for IoT sensor nodes," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8279–8290, 2021.
- [306] A. Beg, T. Al-Kharobi, and A. Al-Nasser, "Performance evaluation and review of lightweight cryptography in an Internet-of-Things environment," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019.

- [307] P. Panahi, C. Bayılmış, U. Çavuşoğlu, and S. Kaçar, "Performance evaluation of lightweight encryption algorithms for IoT-based applications," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 4015–4037, 2021.
- [308] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The day after mirai: A survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices," in *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security*. SCITEPRESS - Science and Technology Publications, 2017.
- [309] S. P. Kannoja and J. Kurmi, "Performance analysis of SSL/TLS crypto libraries: Based on operating platform," *Journal of Scientific Research*, vol. 66, no. 02, pp. 91–100, 2022.
- [310] P. Li, J. Su, and X. Wang, "ITLS: Lightweight transport-layer security protocol for IoT with minimal latency and perfect forward secrecy," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6828–6841, 2020.
- [311] —, "Itls/idthls: Lightweight end-to-end security protocol for Iot through minimal latency," in *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*. New York, NY, USA: Association for Computing Machinery, 2019, p. 166–168.
- [312] D. D. Kumar, J. D. Mukharzee, C. V. D. Reddy, and S. M. Rajagopal, "Safe and secure communication using SSL/TLS," in *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2024, pp. 1–6.
- [313] L. Perugini and A. Vesco, "On the integration of Self-Sovereign identity with TLS 1.3 handshake to build trust in IoT systems," *Internet of Things*, vol. 25, no. 101103, p. 101103, 2024.
- [314] A. Kumar, S. Garg, and R. Singh, "Secure communication in IoT-based healthcare systems using TLS/SSL: Challenges and solutions," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4567–4578, 2020.
- [315] M. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "A survey on TLS/SSL encryption for IoT security in healthcare applications," *Journal of Medical Systems*, vol. 43, no. 11, 2019.
- [316] P. Sharma, R. Jain, and K. Gupta, "Enhancing security in healthcare IoT devices using TLS 1.3: A performance analysis," *International Journal of Network Security*, vol. 23, no. 4, pp. 678–689, 2021.
- [317] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Rupprechter, and G. Pregartner, "Securing smart maintenance services: Hardware-security and TLS for MQTT," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. IEEE, 2015.
- [318] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *2017 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2017.

- [319] Ö. Şeker, G. Dalkılıç, and U. C. Çabuk, “MARAS: Mutual authentication and role-based authorization scheme for lightweight Internet of Things applications,” *Sensors (Basel)*, vol. 23, no. 12, p. 5674, 2023.
- [320] S.-J. Horng, X.-Z. Hu, B. Li, and N. Xiong, “Personal identification via heartbeat signal,” in *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*. IEEE, 2018.
- [321] D. Alduwaile and M. S. Islam, “Single heartbeat ECG biometric recognition using convolutional neural network,” in *2020 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE, 2020.
- [322] P. Verma and D. S. Gupta, “A pairing-free data authentication and aggregation mechanism for Intelligent Healthcare System,” *Comput. Commun.*, vol. 198, pp. 282–296, 2023.
- [323] W. Yuan, “DCAGS-IoT: Dynamic cross-domain authentication scheme using group signature in IoT,” *Applied Sciences*, vol. 13, no. 10, 2023.
- [324] M. A. Mobarhan and M. Salamah, “REPS-AKA5: A robust group-based authentication protocol for IoT applications in LTE system,” *Internet of Things*, vol. 22, no. 100700, p. 100700, 2023.
- [325] G. Singh, “GBEAKA: Group-based efficient authentication and key agreement protocol for LPiMT using 5G,” *Internet of Things*, vol. 22, no. 100688, p. 100688, 2023.
- [326] Z. Yang, Z. Wang, F. Qiu, and F. Li, “A group key agreement protocol based on ECDH and short signature,” *J. Inf. Secur. Appl.*, vol. 72, no. 103388, p. 103388, 2023.
- [327] W. Ali and A. A. Ahmed, “An authenticated group shared key mechanism based on a combiner for hash functions over the industrial Internet of Things,” *Processes*, vol. 11, 2023.
- [328] C. Trivedi and U. P. Rao, “Secrecy aware key management scheme for Internet of Healthcare Things,” *The Journal of Supercomputing*, vol. 79, pp. 12 492–12 522, 2023.
- [329] M. Alshahrani and I. Traore, “Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain,” *J. Inf. Secur. Appl.*, vol. 45, pp. 156–175, 2019.
- [330] H. Ap and K. Kulothungan, “Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for Internet of Things,” *EURASIP Journal on Wireless Communications and Networking*, no. 1, 2019.
- [331] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou, and C. Tsatsoulis, “Review of security and privacy for the internet of medical things (iomt),” in *2019 15th international conference on distributed computing in sensor systems (DCOSS)*. IEEE, 2019, pp. 457–464.

- [332] A. Altameem, Prabu, Senthilnathan, R. C. Poonia, and A. K. J. Saudagar, "A hybrid AES with a chaotic map-based biometric authentication framework for IoT and Industry 4.0," *Systems*, vol. 11, no. 1, p. 28, 2023.
- [333] V. O. Nyangaresi, "Privacy preserving three-factor authentication protocol for secure message forwarding in wireless body area networks," *Ad Hoc Netw.*, no. 103117, p. 103117, 2023.
- [334] H. Xu, C. Hsu, L. Harn, J. Cui, Z. Zhao, and Z. Zhang, "Three-factor anonymous authentication and key agreement based on fuzzy biological extraction for industrial Internet of Things," *IEEE Trans. Serv. Comput.*, vol. 16, no. 4, pp. 3000–3013, 2023.
- [335] R. M. A. Haseeb-Ur-Rehman, M. Liaqat, A. H. M. Aman, A. A. Almazroi, M. K. Hasan, Z. Ali, and R. L. Ali, "LR-AKAP: A lightweight and robust security protocol for smart home environments," *Sensors (Basel)*, vol. 22, no. 18, p. 6902, 2022.
- [336] K. Kim, J. Ryu, Y. Lee, and D. Won, "An improved lightweight user authentication scheme for the Internet of Medical Things," *Sensors (Basel)*, vol. 23, no. 3, p. 1122, 2023.
- [337] C.-W. Lien and S. Vhaduri, "Challenges and opportunities of biometric user authentication in the age of IoT: A survey," *ACM Comput. Surv.*, 2023.
- [338] H. Amintoosi, M. Nikooghadam, M. Shojafar, S. Kumari, and M. Alazab, "Slight: A lightweight authentication scheme for smart healthcare services," *Computers and Electrical Engineering*, vol. 99, p. 107803, 2022.
- [339] S. F. Aghili, H. Mala, M. Shojafar, and P. Peris-Lopez, "Laco: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT," *future generation computer systems*, vol. 96, pp. 410–424, 2019.
- [340] P. K. Dhillon and S. Kalra, "Multi-factor user authentication scheme for IoT-based healthcare services," *J. Reliab. Intell. Environ.*, vol. 4, no. 3, pp. 141–160, 2018.
- [341] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52 018–52 027, 2020.
- [342] S. S. Sahoo, S. Mohanty, and B. Majhi, "A secure three factor based authentication scheme for health care systems using IoT enabled devices," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 1, pp. 1419–1434, 2021.
- [343] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2649–2656, 2021.
- [344] S. R. Moosavi, T. N. Gia, A.-M. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, and H. Tenhunen, "Sea: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways," *Procedia Computer Science*, vol. 52, pp. 452–459, 2015.

- [345] P.-S. Cheng-Chi and M.-S. Chung, "A survey on attribute-based encryption schemes of access control in cloud environments," *Int. J. Netw. Secur.*, vol. 15, pp. 231–240, 2013.
- [346] R. Fredy and J. L. Stiv Leon-Aguilar, "CP-ABE encryption over MQTT for an IoT system with raspberry pi," in *56th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2022.
- [347] R. Xiao and X. Liu, "Analysis of the architecture of the mental health education system for college students based on the Internet of Things and privacy security," *IEEE Access*, vol. 9, pp. 81 089–81 096, 2021.
- [348] A. Mughaid, A. Al-Arjan, M. Rasmi, and S. AlZu'bi, "Intelligent security in the era of AI: The key vulnerability of RC4 algorithm," in *2021 International Conference on Information Technology (ICIT)*. IEEE, 2021.
- [349] E. Luo, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, and M. Atiquzzaman, "Privacyprotector: Privacy-protected patient data collection in IoT-based health-care systems," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 163–168, 2018.
- [350] A. Andreas, C. X. Mavromoustakis, G. Mastorakis, D.-T. Do, J. M. Batalla, E. Pallas, and E. K. Markakis, "Towards an optimized security approach to IoT devices with confidential healthcare data exchange," *Multimedia Tools and Applications*, vol. 80, pp. 31 435–31 449, 2021.
- [351] X. C. Yin, Z. G. Liu, B. Ndibanje, L. Nkenyereye, and S. Riazul Islam, "An IoT-based anonymous function for security and privacy in healthcare sensor networks," *Sensors*, vol. 19, no. 14, p. 3146, 2019.
- [352] A. Refaey, A. Sallam, and A. Shami, "On IoT applications: a proposed SDP framework for MQTT," *Electron. Lett.*, vol. 55, no. 22, pp. 1201–1203, 2019.
- [353] N. Asghar, M. Mamoon, and M. Fleury, "Key management protocols for secure wireless multimedia services: a review," *Recent Patents on Telecommunication (Discontinued)*, vol. 1, no. 1, pp. 41–53, 2012.
- [354] M. Hamad, "SEEMQTT: secure end-to-end MQTT-based communication for mobile IoT systems using secret sharing and trust delegation," *IEEE Internet of Things Journal*, vol. 10, pp. 3384–3406, 2022.
- [355] M. Thamizhselvan, R. Raghuraman, S. Gershon Manoj, and P. Victor Paul, "A novel security model for cloud using trusted third party encryption," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2015.
- [356] C.-I. Fan, C.-H. Shie, Y.-F. Tseng, and H.-C. Huang, "An efficient data protection scheme based on hierarchical ID-based encryption for MQTT," *ACM Trans. Sens. Netw.*, vol. 19, no. 3, pp. 1–21, 2023.
- [357] F. Buccafurri and C. Romolo, "A Blockchain-Based OTP-Authentication scheme for constrained IoT devices using MQTT," in *Proceedings of the 2019 3rd International*

- Symposium on Computer Science and Intelligent Control*. New York, NY, USA: ACM, 2019.
- [358] T. Noguchi, M. Nakagawa, M. Yoshida, and A. G. Ramonet, “A secure secret key-sharing system for resource-constrained IoT devices using MQTT,” in *2022 24th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2022.
- [359] S. Das and S. Namasudra, “A novel hybrid encryption method to secure healthcare data in IoT-enabled healthcare infrastructure,” *Computers and Electrical Engineering*, vol. 101, p. 107991, 2022.
- [360] O. Popoola, M. A. Rodrigues, J. Marchang, A. Shenfield, A. Ikpehai, and J. Popoola, “An optimized hybrid encryption framework for smart home healthcare: Ensuring data confidentiality and security,” *Internet of Things*, vol. 27, p. 101314, 2024.
- [361] S. Balaji, K. Nathani, and R. Santhakumar, “Iot technology, applications and challenges: a contemporary survey, wireless personal communications 108,” *Wireless personal communications*, vol. 108, pp. 363–388, 2019.
- [362] R. R. K. Chaudhary and K. Chatterjee, “A lightweight security framework for electronic healthcare system,” *Int. J. Inf. Technol.*, vol. 14, no. 6, pp. 3109–3121, 2022.
- [363] G. Nebbione and M. C. Calzarossa, “Security of IoT application layer protocols: Challenges and findings,” *Future Internet*, vol. 12, no. 3, p. 55, 2020.
- [364] R. B. Marqas, S. M. Almufti, and R. R. Ihsan, “Comparing symmetric and asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms. xi’an,” *Xi’an Jianzhu Keji Daxue Xuebao/Journal of Xi’an University of Architecture & Technology*, vol. 12, no. 3, pp. 3110–3116, 2020.
- [365] I. Vaccari, M. Aiello, and E. Cambiaso, “SlowTT: A slow denial of service against IoT networks,” *Information (Basel)*, vol. 11, no. 9, p. 452, 2020.
- [366] J. Ahamed, M. Zahid, M. Omar, and K. Ahmad, “AES and MQTT based security system in the Internet of Things,” *J. Discrete Math. Sci. Cryptogr.*, vol. 22, no. 8, pp. 1589–1598, 2019.
- [367] M. M. Salim, L. T. Yang, and J. H. Park, “Lightweight authentication scheme for IoT based E-healthcare service communication,” *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 9, pp. 5025–5032, 2024.
- [368] V. A. Thakor, M. A. Razzaque, and M. R. Khandaker, “Lightweight cryptography for IoT: A state-of-the-art,” *arXiv preprint arXiv:2006.13813*, 2020.
- [369] A. Hassan, “Lightweight cryptography for the Internet of Things,” in *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2021, pp. 780–795.
- [370] K. Tsantikidou and N. Sklavos, “Hardware limitations of lightweight cryptographic designs for IoT in healthcare,” *Cryptography*, vol. 6, no. 3, p. 45, 2022.

- [371] S. Kumar, D. Kumar, R. Dangi, G. Choudhary, N. Dragoni, and I. You, "A review of lightweight security and privacy for resource-constrained iot devices," *Computers, Materials and Continua*, vol. 78, no. 1, pp. 31–63, 2024.
- [372] V. Bhagat, S. Kumar, S. K. Gupta, and M. K. Chaube, "Lightweight cryptographic algorithms based on different model architectures: A systematic review and futuristic applications," *Concurr. Comput.*, vol. 35, no. 1, 2023.
- [373] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *Journal of Cyber Security Technology*, vol. 1, no. 3-4, pp. 187–201, 2017.
- [374] M. Adil, M. K. Khan, N. Kumar, M. Attique, A. Farouk, M. Guizani, and Z. Jin, "Healthcare Internet of Things: Security threats, challenges, and future research directions," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19 046–19 069, 2024.
- [375] G. Marques, R. Pitarma, N. M. Garcia, and N. Pombo, "Internet of Things architectures, technologies, applications, challenges, and future directions for enhanced living environments and healthcare systems: A review," *Electronics (Basel)*, vol. 8, no. 10, p. 1081, 2019.
- [376] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A survey on security and privacy issues in modern healthcare systems: Attacks and defenses," *ACM Transactions on Computing for Healthcare*, vol. 2, no. 3, pp. 1–44, 2021.
- [377] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab, "Securing Internet of Medical Things systems: Limitations, issues and recommendations," *Future Gener. Comput. Syst.*, vol. 105, pp. 581–606, 2020.
- [378] M. Katagi and S. Moriai, "Lightweight cryptography for the Internet of Things," *sony corporation*, pp. 7–10, 2008.
- [379] Z. Ding, D. He, Q. Qiao, X. Li, Y. Gao, S. Chan, and K.-K. R. Choo, "A lightweight and secure communication protocol for the IoT environment," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 3, pp. 1050–1067, 2024.
- [380] J. Furtak, "The cryptographic key distribution system for IoT systems in the MQTT environment," *Sensors (Basel)*, vol. 23, no. 11, 2023.
- [381] S. Ameer, J. Benson, and R. Sandhu, "Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 4032–4051, 2023.
- [382] Y. Chen, R. Xu, E. Blasch, and G. Chen, "A federated capability-based access control mechanism for Internet of Things (IoTs)," in *Sensors and Systems for Space Applications XI*, K. D. Pham and G. Chen, Eds. SPIE, 2018.
- [383] S. Ding, C. Li, and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27 336–27 345, 2018.
- [384] T. Wang, Y. Zhou, H. Ma, and R. Zhang, "Enhanced dual-policy attribute-based encryption for secure data sharing in the cloud," *Secur. Commun. Netw.*, vol. 2022, pp. 1–21, 2022.

- [385] Z. Elbanna, *Integration of Attribute-Based Encryption and IoT: An IoT Security Architecture*. Digitala Vetenskapliga Arkivet, 2023.
- [386] J. Ling, J. Chen, J. Chen, and W. Gan, “Multiauthority attribute-based encryption with traceable and dynamic policy updating,” *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, 2021.
- [387] A. Xiang, H. Gao, Y. Tian, L. Wang, and J. Xiong, “Attribute-based key management for patient-centric and trusted data access in blockchain-enabled IoMT,” *Comput. Netw.*, vol. 246, no. 110425, p. 110425, 2024.
- [388] H. Guo, W. Li, M. Nejad, and C.-C. Shen, “A hybrid blockchain-edge architecture for electronic health records management with attribute-based cryptographic mechanisms,” 2023.
- [389] Z. Zhou, N. Wang, J. Liu, J. Fu, and L. Deng, “The blockchain-based privacy-preserving searchable attribute-based encryption scheme for federated learning model in IoMT,” *Concurr. Comput.*, 2024.
- [390] X. Yang, W. Li, and K. Fan, “A revocable attribute-based encryption EHR sharing scheme with multiple authorities in blockchain,” *Peer Peer Netw. Appl.*, vol. 16, no. 1, pp. 107–125, 2023.
- [391] R. Ding, H. Zhong, J. Ma, X. Liu, and J. Ning, “Lightweight privacy-preserving identity-based verifiable IoT-based health storage system,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8393–8405, 2019.
- [392] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, “Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage,” *IEEE Trans. Inf. Forensics Secur.*, pp. 1–1, 2018.
- [393] X. Zhang, Y. Tang, S. Cao, C. Huang, and S. Zheng, “Enabling identity-based authorized encrypted diagnostic data sharing for cloud-assisted e-health information systems,” *J. Inf. Secur. Appl.*, vol. 54, no. 102568, p. 102568, 2020.
- [394] V. Odelu, A. K. Das, M. Khurram Khan, K.-K. R. Choo, and M. Jo, “Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts,” *IEEE Access*, vol. 5, pp. 3273–3283, 2017.
- [395] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, “Secure MQTT for Internet of Things (IoT),” in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, 2015.
- [396] T.-L. Liao, H.-R. Lin, P.-Y. Wan, and J.-J. Yan, “Improved attribute-based encryption using chaos synchronization and its application to MQTT security,” *Appl. Sci. (Basel)*, vol. 9, no. 20, p. 4454, 2019.
- [397] P. Perazzo, F. Righetti, M. La Manna, and C. Vallati, “Performance evaluation of Attribute-Based encryption on constrained IoT devices,” *Comput. Commun.*, vol. 170, pp. 151–163, 2021.

- [398] J. Jebrane and S. Lazaar, "An enhanced and verifiable lightweight authentication protocol for securing the internet of medical things (IoMT) based on CP-ABE encryption," *Int. J. Inf. Secur.*, vol. 23, no. 6, pp. 3691–3710, 2024.
- [399] T. V. Bezerra, G. Callou, and F. Airton, "A Modeling-Based approach for performance and availability assessment of IoMT systems," *Electronics*, vol. 14, 2025.
- [400] M. S. Ahsan and A. S. K. Pathan, "A comprehensive survey on the requirements, applications, and future challenges for access control models in IoT: The state of the art," *IoT*, vol. 6, no. 1, 2025.
- [401] R. Imam, K. Kumar, S. M. Raza, R. Sadaf, F. Anwer, N. Fatima, M. Nadeem, M. Abbas, and O. Rahman, "A systematic literature review of attribute based encryption in health services," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6743–6774, 2022.
- [402] R. Guo, G. Yang, H. Shi, Y. Zhang, and D. Zheng, "O 3-r-cp-abe: An efficient and revocable attribute-based encryption scheme in the cloud-assisted iomt system," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8949–8963, 2021.
- [403] F. Sammy and S. M. C. Vigila, "An efficient blockchain based data access with modified hierarchical attribute access structure with CP-ABE using ECC scheme for patient health record," *Security and communication networks*, vol. 2022, no. 1, p. 8685273, 2022.
- [404] Y.-W. Hwang and I.-Y. Lee, "A study on cp-abe-based medical data sharing system with key abuse prevention and verifiable outsourcing in the iomt environment," *Sensors*, vol. 20, no. 17, p. 4934, 2020.
- [405] A. Winarno and R. F. Sari, "A novel secure End-to-End IoT communication scheme using lightweight cryptography based on block cipher," *Appl. Sci. (Basel)*, vol. 12, no. 17, p. 8817, 2022.
- [406] A. Biswas, A. Majumdar, S. Nath, A. Dutta, and K. L. Baishnab, "LRBC: a lightweight block cipher design for resource constrained IoT devices," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 5, pp. 5773–5787, 2023.
- [407] A. Alahdal and N. K. Deshmukh, "A systematic technical survey of lightweight cryptography on IoT environment," *International Journal of Scientific & Technology Research*, vol. 9, no. 3, 2020.
- [408] S. Srinath, G. S. Nagaraja, and R. Shahabadkar, "A detailed analysis of lightweight cryptographic techniques on Internet-of-Things," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, 2021.
- [409] N. Lata and R. Kumar, "Analysis of lightweight cryptography algorithms for IoT communication," in *Advances in Intelligent Systems and Computing*. Singapore: Springer Singapore, 2021, pp. 397–406.

- [410] H. Cui and X. Yi, "Secure internet of things in cloud computing via puncturable attribute-based encryption with user revocation," *IEEE Internet of Things Journal*, 2023.
- [411] M. Rasori, M. L. Manna, P. Perazzo, and G. Dini, "A survey on attribute-based encryption schemes suitable for the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8269–8290, 2022.
- [412] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*. IEEE, 2007.
- [413] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, 2014.
- [414] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Inf. Sci. (Ny)*, vol. 479, pp. 640–650, 2019.
- [415] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010.
- [416] R. Chatterjee and R. Chakraborty, "A modified lightweight PRESENT cipher for IoT security," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, 2020.
- [417] A. Biryukov and L. Perrin, "State of the art in lightweight symmetric cryptography," *Cryptology ePrint Archive*, 2017.
- [418] S. Goyal, N. Sharma, B. Bhushan, A. Shankar, and M. Sagayam, "IoT enabled technology in secured healthcare: applications, challenges and future directions," *Cognitive Internet of Medical Things for Smart Healthcare: Services and Applications*, pp. 25–48, 2021.
- [419] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4132–4156, 2021.
- [420] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper," in *Computer Aided Verification: 20th International Conference*. Princeton, NJ, USA: Springer, 2008, pp. 414–418.
- [421] B. Esslinger, *Learning and Experiencing Cryptography with CrypTool and SageMath*. Artech House, 2023.
- [422] K. P. Naik and U. R. Joshi, "Performance analysis of constrained application protocol using Cooja simulator in Contiki OS," in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. IEEE, 2017.

- [423] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of drone environment," *IEEE Access*, vol. 8, pp. 155 645–155 659, 2020.
- [424] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of IoT protocols: A review," *Comput. Netw.*, vol. 174, no. 107233, p. 107233, 2020.
- [425] S. Hick, B. Esslinger, and A. Wacker, "Reducing the complexity of understanding cryptology using CrypTool," in *10th International Conference on Education and Information Systems, Technologies and Applications*. Orlando, Florida, USA: EISTA, 2012.
- [426] A. Meça, "Exploring data encryption standard (DES) through CrypTool implementation: A comprehensive examination and historical perspective," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Cham: Springer Nature Switzerland, 2024, pp. 143–160.
- [427] M. Belkheir, M. Rouissat, M. A. Boukhobza, A. Mokaddem, H. S. A. Belkhira, P. Lorenz, M. Bouziani, M. Beneddine, and A. Reguieg, "An in-depth analysis of application protocols performances in various IoT network environments," in *2024 8th International Conference on Image and Signal Processing and their Applications (ISPA)*. IEEE, 2024, pp. 1–6.
- [428] M. A. Khatun, S. F. Memon, C. Eising, and L. L. Dhirani, "Machine learning for healthcare-IoT security: A review and risk mitigation," *IEEE Access*, vol. 11, pp. 145 869–145 896, 2023.
- [429] T. Mehmood, "COOJA network simulator: Exploring the infinite possible ways to compute the performance metrics of IOT based smart devices to understand the working of IOT based compression & routing protocols," *arXiv preprint arXiv:1712.08303*, 2017.
- [430] J. Roldán-Gómez, J. Carrillo-Mondéjar, J. M. Castelo Gómez, and S. Ruiz-Villafranca, "Security analysis of the MQTT-SN protocol for the Internet of Things," *Appl. Sci. (Basel)*, vol. 12, no. 21, p. 10991, 2022.
- [431] S. Siboni, V. Sachidananda, Y. Meidan, M. Bohadana, Y. Mathov, S. Bhairav, A. Shabtai, and Y. Elovici, "Security testbed for Internet-of-Things devices," *IEEE Trans. Reliab.*, vol. 68, no. 1, pp. 23–44, 2019.
- [432] Benoît Thébaudeau, "An introduction to Cooja," <https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja>, accessed: 2024-2-23.
- [433] R. L. Nicholas Humfrey, "Mosquitto really small message broker," <https://github.com/eclipse-mosquitto/mosquitto.rsmb?tab=readme-ov-file>, accessed: 2024-2-23.
- [434] H. da Rocha, T. L. Monteiro, M. E. Pellenz, M. C. Penna, and J. Alves Junior, "An MQTT-SN-based QoS dynamic adaptation method for wireless sensor networks," in *Advanced Information Networking and Applications*. Cham: Springer International Publishing, 2020, pp. 690–701.

- [435] R. Dwivedi, D. Mehrotra, and S. Chandra, "Potential of internet of medical things (IoMT) applications in building a smart healthcare system: A systematic review," *J. Oral Biol. Craniofac. Res.*, vol. 12, no. 2, pp. 302–318, 2022.
- [436] M. Yaghoubi, K. Ahmed, and Y. Miao, "Wireless body area network (WBAN): A survey on architecture, technologies, energy consumption, and security challenges," *J. Sens. Actuator Netw.*, vol. 11, no. 4, p. 67, 2022.
- [437] A. H. Seh, M. Zarour, M. Alenezi, A. K. Sarkar, A. Agrawal, R. Kumar, and R. A. Khan, "Healthcare data breaches: Insights and implications," *Healthcare (Basel)*, vol. 8, no. 2, p. 133, 2020.
- [438] P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: the definitive guide*. " O'Reilly Media, Inc.", 2009.
- [439] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Comput.*, vol. 10, no. 6, pp. 87–89, 2006.
- [440] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. Van Bemten, I. Askoxylakis, I. Pappaefstathiou, and A. Plemenos, "Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017.
- [441] P. Gupta and I. O. Prabha., "A survey of application layer protocols for Internet of Things," in *2021 International Conference on Communication information and Computing Technology (ICCICT)*. IEEE, 2021.
- [442] I. Dacosta, M. Ahamad, and P. Traynor, "Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties," in *Computer Security – ESORICS 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 199–216.
- [443] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2013.
- [444] A. Sachdeva, "A study of encryption algorithms AES, DES and RSA for security," *Global Journal of Computer Science and Technology*, vol. 13, pp. 32–40, 2013.
- [445] S. Rao, "Performance analysis of DES and triple DES," *International Journal of Computer Applications*, vol. 130, pp. 30–54, 2015.
- [446] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *Twofish: A 128-bit block cipher*. Nashville, TN: John Wiley & Sons, 1998, vol. 15, no. 1.
- [447] T. Nie and T. Zhang, "A study of DES and Blowfish encryption algorithm," in *TENCON 2009 - 2009 IEEE Region 10 Conference*. IEEE, 2009.
- [448] A. A. Zakaria, A. H. Azni, F. Ridzuan, N. H. Zakaria, and M. Daud, "Modifications of key schedule algorithm on RECTANGLE block cipher," in *Communications in Computer and Information Science*. Singapore: Springer Singapore, 2021, pp. 194–206.

- [449] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, “The LED block cipher,” in *Cryptographic Hardware and Embedded Systems – CHES 2011*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 326–341.
- [450] T. Pornin, *Cryptographic Hardware and Embedded Systems-CHES 2001: Third International Workshop*. Paris, France: Springer, 2001.
- [451] E. Gamess, T. N. Ford, and M. Trifas, “Performance evaluation of a widely used implementation of the MQTT protocol with large payloads in normal operation and under a DoS attack,” in *Proceedings of the 2021 ACM Southeast Conference*, vol. 19. New York, NY, USA: ACM, 2021, pp. 154–162.
- [452] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, “Lightweight authenticated-encryption scheme for Internet of Things based on publish-subscribe communication,” *IEEE Access*, vol. 8, pp. 60 539–60 551, 2020.
- [453] E. Barker and W. Barker, *Recommendation for key management, part 2: best practices for key management organization (No. NIST Special Publication (SP) 800-57 Part 2 Rev. 1 (Draft))*. NIST Special Publication, 2018.
- [454] D. Boneh, C. Dunworth, and R. Lipton, “Breaking DES using a molecular computer,” in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Providence, Rhode Island: American Mathematical Society, 1996, pp. 37–65.
- [455] H. Huang, X. Miao, Z. Wu, and Q. Wei, “An efficient ECC-based authentication scheme against clock asynchronous for spatial information network,” *Math. Probl. Eng.*, vol. 2021, pp. 1–14, 2021.
- [456] I. N. R. Hendrawan and I. G. N. W. Arsa, “Zolertia Z1 energy usage simulation with Cooja simulator,” in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, 2017.
- [457] R. Ferdousi, M. Helaluddin, A. Akther, and K. M. Alam, “An empirical study of CoAP based service discovery methods for constrained IoT networks using Cooja simulator,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017.
- [458] C.-S. Park and H.-M. Nam, “Security architecture and protocols for secure MQTT-SN,” *IEEE Access*, vol. 8, pp. 226 422–226 436, 2020.
- [459] A. Bashir and A. H. Mir, “Lightweight secure-MQTT for Internet of Things,” in *Lecture Notes in Electrical Engineering*. Singapore: Springer Singapore, 2020, pp. 57–66.
- [460] R. A. Al-Qassab and M. I. Aal-Nouman, “Performance evaluation of CoAP and MQTT-SN protocols,” in *Communications in Computer and Information Science*. Cham: Springer International Publishing, 2022, pp. 223–236.
- [461] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, “A critical cybersecurity analysis and future research directions for the Internet of Things: A Comprehensive Review,” *Sensors*, vol. 23, no. 8, 2023.

- [462] I. Lee and K. Lee, "The internet of things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015.
- [463] E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello, "Slow DoS attacks: definition and categorisation," *Int. J. Trust Manag. Comput. Commun.*, vol. 1, no. 3/4, p. 300, 2013.
- [464] N.-W. Lo and S.-H. Hsu, "A secure IoT firmware update framework based on MQTT protocol," in *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2020, pp. 187–198.
- [465] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," in *2016 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2016.
- [466] R. S. Bali, F. Jaafar, and P. Zavarasky, "Lightweight authentication for MQTT to improve the security of IoT communication," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*. New York, NY, USA: ACM, 2019.
- [467] S. Moffat, M. Hammoudeh, and R. Hegarty, "A survey on ciphertext-policy attribute-based encryption (CP-ABE) approaches to data security on mobile devices and its application to IoT," in *Proceedings of the International Conference on Future Networks and Distributed Systems*. New York, NY, USA: ACM, 2017.
- [468] R. Ruqayah, Q. Shi, and K. Gyu Myoung Lee, "Survey on revocation in ciphertext-policy attribute-based encryption," *Sensors*, vol. 19, no. 7, 2019.
- [469] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, "Access control in the Internet of Things: Big challenges and new opportunities," *Comput. Netw.*, vol. 112, pp. 237–262, 2017.
- [470] S. Wang, H. Wang, J. Li, H. Wang, J. Chaudhry, M. Alazab, and H. Song, "A fast CP-ABE system for cyber-physical security and privacy in mobile healthcare network," *IEEE Trans. Ind. Appl.*, pp. 1–1, 2020.
- [471] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of smart healthcare monitoring system in IoT environment," *SN Comput. Sci.*, vol. 1, no. 3, 2020.
- [472] J. N. S. Rubí and P. R. L. Gondim, "IoMT platform for pervasive healthcare data aggregation, processing, and sharing based on OneM2M and OpenEHR," *Sensors (Basel)*, vol. 19, no. 19, p. 4283, 2019.
- [473] B. A. Mubdir and H. M. A. Bayram, "Adopting MQTT for a multi protocols IoMT system," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 1, p. 834, 2022.
- [474] G. S. Aljumaie, G. H. Alzeer, R. K. Alghamdi, H. Alsuwat, and E. Alsuwat, "Modern study on internet of medical things (IOMT) security," *International Journal of Computer Science & Network Security*, vol. 21, no. 8, pp. 254–266, 2021.

- [475] W.-T. Su, W.-C. Chen, and C.-C. Chen, “An extensible and transparent thing-to-thing security enhancement for MQTT protocol in IoT environment,” in *2019 Global IoT Summit (GIoTS)*. IEEE, 2019.
- [476] A. Rahman, S. Roy, M. S. Kaiser, and M. S. Islam, “A lightweight multi-tier S-MQTT framework to secure communication between low-end IoT nodes,” in *2018 5th International Conference on Networking, Systems and Security (NSysS)*. IEEE, 2018.
- [477] A. J. Hintaw, S. Manickam, and S. Karuppayah, “A robust security scheme based on enhanced symmetric algorithm for MQTT in the Internet of Things,” *IEEE Access*, vol. 11, pp. 43 019–43 040, 2023.
- [478] P. Anantharaman, K. Palani, and S. Smith, “Scalable identity and key management for publish-subscribe protocols in the Internet-of-Things,” in *Proceedings of the 9th International Conference on the Internet of Things*. New York, NY, USA: ACM, 2019.
- [479] M. Calabretta, R. Pecori, and L. Veltri, “A token-based protocol for securing MQTT communications,” in *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2018.
- [480] J. Toldinas, B. Lozinskis, E. Baranauskas, and A. Dobrovolskis, “MQTT quality of service versus energy consumption,” in *2019 23rd International Conference Electronics*. IEEE, 2019.
- [481] T. K. Goyal, V. Sahula, and D. Kumawat, “Energy efficient lightweight cryptography algorithms for IoT devices,” *IETE J. Res.*, pp. 1–14, 2019.
- [482] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The SIMON and SPECK lightweight block ciphers,” in *Proceedings of the 52nd Annual Design Automation Conference*. New York, NY, USA: ACM, 2015.
- [483] C. Thomson, I. Romdhani, A. Al-Dubai, M. Qasem, B. Ghaleb, and I. Wadhaj, *Cooja simulator manual*. Edinburgh Napier University, 2016.
- [484] A. A. Zakaria, A. Azni, F. Ridzuan, N. H. Zakaria, and M. Daud, “Extended rectangle algorithm using 3d bit rotation to propose a new lightweight block cipher for iot,” *IEEE Access*, vol. 8, pp. 198 646–198 658, 2020.