

CRANFIELD UNIVERSITY

NASR ABOSATA

CYBERSECURITY OF THE INTERNET OF THINGS: A
DEVELOPMENT OF SECURITY APPROACHES FOR
INDUSTRIAL APPLICATION-BASED DISTRIBUTED
SENSORS

School of Aerospace, Transport and Manufacturing

SATM

PhD in Cybersecurity

This thesis is submitted in partial fulfilment of the requirements for the
degree of Doctor of Philosophy

First Supervisor: Dr. Saba Al-Rubaye

Second Supervisor: Prof. Gokhan Inalhan

(Sep 2022)

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

PhD in Cybersecurity

NASR ABOSATA

CYBERSECURITY OF THE INTERNET OF THINGS: A
DEVELOPMENT OF SECURITY METHODS BASED-
DISTRIBUTED SENSORS FOR INDUSTRIAL
APPLICATIONS

First Supervisor: Dr. Saba Al-Rubaye

Second Supervisor: Prof. Gokhan Inalhan

(Sep 2022)

This thesis is submitted in partial fulfilment of the requirements for the
degree of Doctor of Philosophy

© Cranfield University 2022. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

The nature of communicating in the heterogeneous Internet of Things (IoT) environment and the disparity in resource capabilities between IoT nodes make it difficult to establish the necessary security in connections. In contrast with conventional networks, most IoT components have limited energy and computational capacity and cannot support complicated security schemes. IoT has been utilised extensively in sensing, controlling, and computing for countless applications in numerous domains, including smart homes, cities, transportation, communication, healthcare, and smart grids. Integrating IoT technology for industrial applications requires security techniques for constructing progressively secure, efficient, and adaptable new systems. Even though there are numerous established methods for securing IoT applications, IoT networks face several obstacles in adopting effective security solutions due to their resource-constrained and limiting characteristics, such as limited memory, low cost, fewer computational capabilities, and low power consumption. The IoT necessitates effective authentication and detection mechanisms that protect security objectives and match its capabilities and features. This study addresses the new IoT security challenges for distributed wireless sensor networks (DWSNs). WSNs initiated this change by introducing unattended wireless topologies comprising resource-constrained nodes, in which the radio spectrum ceased to be the primary resource worthy of optimisation. Resource-constrained IoT nodes are exposed to many unanticipated security threats not faced by traditional network systems. Therefore, IoT cybersecurity is constrained by limited resources rather than an absence of advanced security solutions and developing solutions compatible with IoT device capabilities is imperative.

This thesis attempted to fill the gap by addressing the theoretical and practical aspects of IoT/Industrial IoT (IIoT) cybersecurity. A comprehensive evaluation of the literature in the second chapter shows that several authentications, encryption, and detection schemes have been developed to provide security and system integrity without efficiently optimising security complexity to meet the constraints of resource devices.

A new mutual authentication scheme is proposed: a payload encryption-based optimisation scheme (PEOS) for lightweight authentication on the Constrained Application Protocol (CoAP). This incorporates the design of a scenario on the Contiki

OS that is included in the Cooja simulator to provide a realistic hardware environment to analyse IoT applications. The test scenario was run several times under different data sizes to investigate the impact of authentication schemes on the performance of resource-constrained nodes. The results of the computational calculation are used to compare the performance of resource-constrained nodes under our proposed scheme, a basic DTLS, and schemes in previous work. The results show that our PEOS improves the DTLS handshaking and retransmission processes, significantly reducing overhead without impacting CoAP performance over distributed sensors. Furthermore, PEOS implemented the parallel execution of S-Boxes in SubBytes, delayed MixColumns in the Advanced Encryption Standard protocol, and successfully reduced the need for additional storage registers. From the results, with a large payload size, the throughput of the proposed scheme was improved by 8.7% compared to the existing PbMA by consuming 1550 microjoules under a 50-node random topology.

Finally, a novel detection approach, called customised intrusion detection (CID), was proposed for heterogeneous IIoT networks using federated transfer learning (FTL). The proposed FT-CID model uses an improved grey wolf optimiser (IGWO) technique to reduce noise and irrelevant features from a simulated IIoT dataset. This incorporates the design of a scenario on the Contiki OS to provide a realistic hardware environment in which to analyse IoT applications. Advanced metering infrastructure datasets were collected from the Contiki OS experiment under different data sizes and attack conditions. Machine learning was used to validate the CID strategy computationally. Regarding distinguishing abnormal from normal traffic, the results confirm exceptionally high prediction accuracy compared with different ML algorithms. The proposed FT-CID model yields 10.63% higher intrusion detection accuracy than the centralised CNN model on the heterogeneous IIoT network or the AMI applications.

Keywords: Internet of Things (IoT), Security, Cryptography, Industrial IoT (IIoT), Application, AMI, Sensors, Cyberattacks, Countermeasures, Distributed sensors, Intrusion detection, Machine learning, Algorithms, SHA, AES, DTLS, Detection, Simulation, Operating systems.

ACKNOWLEDGEMENTS

The completion of my PhD thesis could not have been achievable without the help and encouragement of several exceptional individuals. Consequently, I would use this opportunity to express my appreciation to those who have helped me in a variety of ways.

I would first express my deep gratitude and appreciation to my beautiful family, who accompanied me on my PhD journey. I could not have accomplished this feat without the assistance of my family members. In addition to being academically rigorous, these four years of my life were emotionally taxing because I was away from my parents and siblings. My father, my eternal cheerleader, who would have been ecstatic with my success, passed away recently and left behind a legacy of great sadness and sorrow. I am grateful to my mother, the wonderful woman who has showered me with love and kindness since my birth. I am also grateful to my wife, for her incredible support and for standing tall by my side whenever I was down and feeling weak. I will never forget to express gratitude to my wonderful children, who have always brought me joy and happiness in all the more difficult times of the PhD.

I am pleased with my academic accomplishment, but it would not have been possible without the lovely and knowledgeable individuals who didn't think twice about offering all the support that was required. I want to thank my amazing and knowledgeable supervisors, Dr Saba Al-Rubaye, Professor Gokhan Inalhan, and Dr Christos Emmanouilidis. Thank you for your support, encouragement, endless kindness, and the confidence you put in me.

Finally, I learned that the area of scientific research is large and even limitless, and that each researcher can examine a situation from a unique perspective. Considering this, I aspire to have made a scientific contribution to the knowledge of cybersecurity that will be useful and a road map for researchers in this field.

Table of Contents

ABSTRACT	i
ACKNOWLEDGEMENTS	iv
Table of Contents	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
1 Chapter 1: Introduction.....	1
1.1 Internet of Things (IoT)	1
1.2 IoT Security (Motivation)	1
1.3 Research Aim and Objectives.....	3
1.4 Research Methodology	4
1.5 Research Contributions.....	10
1.6 Thesis Structure	11
1.7 List of publications.....	13
1.8 References:	14
2 Chapter 2: Cybersecurity for the Internet of Things and System Integrity:	
Challenges and Requirements.....	16
2.1 Abstract	16
2.2 Introduction.....	16
2.3 Classification of Security Attacks	19
2.3.1 Perception Layer with Security Attacks.....	19
2.3.2 Network Layer with Security Attacks	21
2.3.3 Application Layer with Security Attacks	22
2.4 Connectivity and System Integrity Approaches	23
2.4.1 Communication and Security	25
2.4.2 Network Security.....	31
2.4.3 Application Security	31
2.5 Network Simulations and Operating Systems.....	32
2.5.1 Simulators.....	32
2.5.2 Operating Systems.....	34
2.6 Security Solutions for IoT and Industrial Systems	36

2.6.1 Communication Security Countermeasures	36
2.6.2 1.6.2 Network Routing Security Countermeasures.....	37
2.6.3 1.6.3 Transport and Application Security Countermeasures.....	47
2.7 Industrial IoT System Integrity	55
2.7.1 Case Study on IIoT Applications: Smart Grids.....	56
2.8 Knowledge Gap	58
2.9 Chapter Summary	59
2.10 References.....	60
3 CHAPTER 3: Efficient encryption scheme for IoT based on distributed sensor networks	79
3.1 Abstract	79
3.2 Introduction.....	79
3.3 The Constrained Application Protocol (CoAP).....	81
3.4 Cryptography Algorithms	82
3.5 Architecture of the Proposed Work vs. Basic Architecture.....	84
3.6 System Model and Network design	84
3.6.1 Mechanism	86
3.6.2 Procedure.....	87
3.6.3 Performance metrics	88
3.7 Performance and Experimental Results	90
3.8 Results and Discussion	91
3.8.1 Communication duration	91
3.8.2 Energy Consumption.....	92
3.9 Chapter Summary	95
3.10 References.....	96
4 Chapter 4: Efficient Payload Encryption-based Authentication Scheme for IoT based Sensor Networks	98
4.1 Abstract	98
4.2 Introduction.....	99
4.3 Case Study: Authentication Schemes for a Smart Grid Application	102
4.4 Preliminaries	104
4.4.1 Problem Statement.....	105

4.4.2 Problem Formulation	105
4.5 Materials and Methods	Error! Bookmark not defined.
4.5.1 System Model and Network Model.....	106
4.5.2 Overview of the Proposed Scheme	108
4.5.3 Lightweight DF-CoAP for Distributed Sensor Networks Using Payload Encryption-based Four-Way Handshaking	110
4.6 Attack and Cost Analysis.....	116
4.7 Experimental Evaluation.....	118
4.7.1 Time, Computation, and Storage Complexity.....	119
4.7.2 Experiment Results for a 30-Node Topology	122
4.7.3 Simulation Results for a 50-Node Topology	127
4.8 Chapter Summary	131
4.9 References	133
5 Chapter 5: Customised Intrusion Detection for IoT based on Machine learning algorithms called FTL-CID	136
5.1 Abstract	136
5.2 Introduction.....	137
5.2.1 Problem Statement.....	139
5.3 Preliminaries	139
5.3.1 Network Architecture.....	139
5.4 System Model.....	140
5.5 Proposed System Design Overview.....	142
5.5.1 Dataset Construction.....	146
5.5.2 Pre-processing and Feature Selection	147
5.5.3 Optimised FTL assisted Edge IDS Learning	148
5.6 Experimental Evaluation.....	155
5.6.1 Experimental Setup.....	155
5.6.2 Dataset Description.....	156
5.6.3 Performance Metrics	156
5.7 Experimental Results.....	157
5.7.1 Training Data Size Vs. Precision	158
5.7.2 Training Data Size Vs. Recall	159

5.7.3 Number of Communication Rounds Vs. F-Measure.....	161
5.7.4 Intrusion Detection Accuracy	163
5.7.5 Training Data Size Vs. False Positive Rate	164
5.8 Chapter Summary	165
5.9 References	166
6 CHAPTER 6: CONCLUSIONS AND FUTURE WORK	168
Conclusions	168
6.2 Future Work	171
7 APPENDICES	172

LIST OF FIGURES

Figure 1-1: Methodological approach.....	9
Figure 2-1: High-level IoT/IIoT architecture	18
Figure 2-2: Classification of IoT layers, threats, and security approaches	20
Figure 2-3: 6LoWPAN stack.....	26
Figure 2-4: RPL routing mechanism	29
Figure 2-5: DTLS Handshake.....	30
Figure 2-6: Simulations and operating systems for IoT and WSN networks.....	36
Figure 3-1: AES algorithm process	83
Figure 3-2: CoAP with our proposed architecture vs. Basic CoAP-DTLS stack.....	84
Figure 3-3: IoT network distributed sensors based	85
Figure 3-4: AES-SHA-256 encryption/decryption mechanism	86
Figure 3-5: Flowchart of the proposed AES-based authentication.....	90
Figure 3.6: Communication duration for different payload sizes.....	91
Figure 3.7: Energy consumption for different plaintext sizes.....	92
Figure 3.8: Energest type CPU	93
Figure 4-1: Block diagram architecture of smart grid applications	99
Figure 4-2: System model architecture of the proposed scenario	105
Figure 4-3: Block diagram of the proposed work.....	110
Figure 4-4: Two-way authentication scheme using payload encryption.....	112
Figure 4-5: Communication between server and smart sensors.....	116
Figure 4-6: Experiment scenarios based on the Contiki OS. (a) Network design diagram for 30 nodes; (b) Network design diagram for 50 nodes.....	119

Figure 4-7: Payload size vs. throughput for a 30-node topology	123
Figure 4-8: Payload size vs. average energy consumption for a 30-node topology...	124
Figure 4-9: Payload size vs. communication duration for a 30-node topology	125
Figure 4-10: Payload size vs. message size overhead for a 30-node topology.....	127
Figure 4-11: Payload size vs. throughput for a 50-node topology	128
Figure 4-12: Payload size vs. average energy consumption for a 50-node topology .	129
Figure 4-13: Payload size vs. communication duration for a 50-node topology	130
Figure 4-14: Payload size vs. message size overhead for a 50-node topology.....	131
Figure 5-1: Architecture of heterogeneous IIoT network	140
Figure 5-2: Overview design of the proposed FT-CID model.....	144
Figure 5-3: Process Flow of Proposed FT-CID Model.....	145
Figure 5-4: Federated Transfer learning in FT-CID.....	151
Figure 5-5: Performance of Precision for FT-CID, CNN, and MLP	158
Figure 5-6: Performance of Recall for FT-CID, MLP, and LR	159
Figure 5-7: Performance of F-Measure	161
Figure 5-8: Performance of Intrusion Detection Accuracy	163
Figure 5-9: Performance of False Positive Rate for FT-CID, CNN, and MLP	164

LIST OF TABLES

Table 1-1: Research objectives, deliverables, and contributions.....	4
Table 2-1: IoT stack based on standardised security solutions	25
Table 2-2: Classification of intrusion detection systems for the network layer of the IoT	42
Table 2-3: Classification of IoT security solutions for transport and applications	50
Table 3-1: CoAP message format.....	82
Table 3.2: Average communication duration for data transmission during the request and response messages, the average energy consumption, and the Energex type CPU for different payload sizes from 2 to 16 bytes	94
Table 4-1: Algorithm 1. Secure Authentication and Communication.....	114
Table 4-2: Experimental parameters	119
Table 4-3: Storage cost of the proposed PEOS and existing payload-based encryption schemes	121
Table 5-1: FTL assisted RPL IDS Learning	154
Table 5-2: Comparative Experimental Results of Precision.....	159
Table 5-3: Comparative Experimental Results of Recall.....	160
Table 5-4: Comparative Experimental Results of F-Measure	162
Table 5-5: Comparative Experimental Results of Intrusion Detection Accuracy.....	164
Table 5-6: Comparative Experimental Results of False Positive Rate.....	165

LIST OF ABBREVIATIONS

IoT	Internet of Things
IIoT	Industrial IoT
SoC	System-on-Chip
WSNs	Wireless sensor networks
DoS	Denial of Service
DDoS	Distributed DoS (DDoS)
IDS	Intrusion Detection system
OS	Operating System
SHA	Secure Hash Algorithm
AES	Advanced Encryption Standard
DTLS	Datagram Transport Layer Security
OP-AES	Optimised Advanced Encryption Standard
FT-CID	Federated Transfer Customised Intrusion detection
AMI	Advanced Meter Infrastructure
SG	Smart Grid
SM	Smart Manufacturing
CNN	Convolutional Neural Network
MLP	Multilayer Perceptron
LR	Logistic Learning
ML	Machine learning
PEOS	Payload Encryption-based Optimisation Scheme
CIA	Confidentiality, integrity, and availability
PBMA	Payload-Based Mutual Authentication
IGWO	Improved Grey Wolf Optimiser
RPL	Routing Protocol for Low-Power and Lossy Networks Protocol
SGD	Stochastic Gradient Descent
ICSs	Industrial Control Systems

TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
MITM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
IPsec	Internet Protocol Security
MAC	Medium Access Control
6LoWPAN	Low-Power Wireless Personal Area Networks
AES-CCM	Advanced Encryption Standard-Counter
AES-CTR	Advanced Encryption Standard-Counter Mode
CBC-MAC	Cipher Block Chaining Message Authentication Code
ACK	Acknowledgement
NACK	Non-Acknowledgement
LoRaWan	Long-Range Wide Area Network
TCP	Transmission Control Protocol
HTTP	Hypertext Transfer Protocol
VPN	Virtual Private Network
AH	Authentication Header
ESP	Encapsulating Security Payload
SSL	Secure Socket Layer
SNT	Scalable Network Technologies
RBAC	Role-based access control
ECC	Elliptic Curve Cryptography
LLN	Low-Power and Lossy Network
SCADA	Supervisory control and data acquisition
DTMC	Discrete-time Markova Chain
TL	Transfer Learning
ECG	Electrocardiography
RSA	Rivest–Shamir–Adleman
TPM	Trusted Platform Module
IBE	Identity-Based Encryption

SHA224	Secure Hash Algorithm
DES	Data Encryption Standard
TEA	Tiny Encryption Algorithm
NTSA	National Transport and Safety Authority
ICSs	Industrial Control Systems
HANs	Home Area Networks
M2M	Machine-to-Machine
DF-CoAP	Default CoAP
PALK	Password-based Anonymous Lightweight Key
FTL	Federated Transfer Learning
FedAvg	Federated Averaging
TP	True positive
TN	True negative
FP	False positive
FN	False negative
OTA	Over-the-Air

Chapter 1: Introduction

1.1 Internet of Things (IoT)

Kevin Ashton popularised the term “Internet of Things” in 1990 [1] to refer to “taggable” products that could employ radio-frequency identification (RFID) chips to enable devices to become electronically recognisable and hence accessible for Internet interactions. With the proliferation of low-cost processors and System-on-Chip (SoC)-based devices, the definition has grown to encompass wireless and Internet-connected sensors and actuators, such as home automation systems, as well as smart meters, transportation, grids and manufacturing [2].

The IoT is defined in various ways. In this thesis, the IoT will be defined as follows. IoT devices are systems that include sensors, actuators, or both, which connect to the Internet either directly or through an intermediate service provider. A sensor is a sub-component of an IoT device, which measures and transmits data to the Internet and to cloud platforms. A sensor might be used in the same way as a weather sensor (e.g., thermometers, anemometers and barometers), accelerometer, GPS unit, or air quality or light sensor. People counts are also frequently used, as are medical sensors (to measure heart rate, blood sugar, etc.) and industrial sensors (smart meters, production line monitoring, etc.) [3]. Thus, the IoT refers to a network of devices and Internet technologies designed to interact and communicate with these devices, such as cloud servers, websites, and gateways. In an IoT network, the number of sensors might differ from the tens, in healthcare and home automation, to the thousands, in industrial automation and agriculture, relying on (typically software) applications.

1.2 IoT Security (Motivation)

The number of IoT-connected devices was predicted to reach 75 billion by 2025 [4]. Although these devices enhance people’s lives and improve business efficiency, they also increase the likelihood of vulnerability to attacks from hackers and cybercriminals. With the emergence of the IoT and the expansion of the Internet to encompass the entire globe, security has become a significant concern. Wireless sensor networks (WSNs) and resource-constrained embedded devices characterise the lowest layer of the IoT architecture: the perception layer. These devices have limited memory, processing,

power, and energy capacities. Thus, this makes them susceptible to a multitude of attacks. As IoT devices automate essential operations, such as traffic control, information security is vital [5]. According to the authors of [6], approximately 70% of the most frequently used IoT devices are vulnerable to several types of attack. These attacks include eavesdropping, replay attacks, denial of service (DoS)/distributed DoS (DDoS), and Sybil and blackhole attacks. For example, in 2016, a Domain Name System (DNS) provider that supported Internet services and platforms, including PayPal, VISA and Twitter, was attacked by a DDoS through the vulnerabilities of IoT devices such as Internet Protocol (IP) cameras, printers and residential gateways that were infected by Miria malware [7], [8]. Although the IoT has provided the Internet with the tools and ability to act, this makes the security issue even more complex. With the advent of the IoT era, the scope of the network has expanded beyond all previous restrictions [2]. Essential security considerations, such as confidentiality, integrity, authentication, availability and non-repudiation, should be appropriate to guarantee protection of the IoT.

The IoT devices are considering the vast expansion and distribution of IoT devices across numerous application domains, such as vehicles, household appliances, smart grids and aerospace/defence applications. Several studies have highlighted industrial IoT privacy, security issues and challenges. Today, embedded devices are commonly used in various domains. This emphasises the critical necessity to not only protect these systems against cyberattacks, but also to consider the impact of the security solutions on performance, as this could have catastrophic effects on both humans and the economy. Nevertheless, compared with conventional multitasking systems, IoT devices face many challenges in adopting effective security solutions due to their resource-constrained and limiting characteristic features, such as small memory capacity, low cost, relatively few computation capabilities, and low power consumption. Effective security solutions in authenticated and detection schemes that preserve security objectives and match the capabilities and features of the IoT are required.

Therefore, there is a need to design a security mechanism to secure IoT applications using optimised cryptographic algorithms for heterogeneous IoT networks. More investigation is needed to address the impact of security mechanisms on performance. There needs to be more research on the design, execution and evaluation of the effect of intrusion

detection systems using machine learning or deep learning algorithms for heterogeneous IoT networks. Most implementations used traditional detection and did not detect several types of attacks. Thus, a deep learning approach will make the IDs robust to various attacks.

This research seeks to fill the gap identified above by developing a comprehensive security strategy to address IoT security challenges and their corresponding mitigation in the IoT/IIoT (Industrial IoT) environment [7]. As long as improved countermeasures need to use additional resources in order to be practicable, this thesis presents [9], [10], [11] a new security technique that is efficient for resources that are constrained and that interfere with the properties of IoT devices. This strategy offers a new approach that could have a substantial and positive influence on performance in an IoT-enabled industrial environment by increasing confidence in data integrity and processes.

1.3 Research Aim and Objectives

This research aims to Design, develop and investigate the industrial IoT (IIoT) network integrity system, including cybersecurity consideration, by applying cryptography and detection schemes to achieve a resilient system. Five related objectives were identified to accomplish this aim and are detailed below. Further to these objectives, table 1.1 links the objective(s) to be achieved with the relevant chapter. The research objectives are as follows:

1. To conduct a comprehension and critical analysis of current academic literature related to IoT/IIoT security issues to define the basis and requirements of this research.
2. To implement a new lightweight authentication scheme using optimised datagram transport layer security (DTLS) with advanced encryption standard (AES) algorithms.
3. To develop a novel detection system using a federated learning approach.
4. To evaluate the impact of the applied authentication and detection schemes by building simulations and implementing the schemes in terms of system performance.

5. To validate the proposed system and methods and disseminate the outcome in conference and journal papers.

Table 1-1: Research objectives, deliverables, and contributions

Objective	Chapter	Deliverables/Contribution	Journal	Status
1	2	Cybersecurity for the Internet of Things and System Integrity: Attacks and Countermeasures, Identifying research gaps (A Review Paper).	MDPI Sensor	Published
2	3	Developing a new security architecture for the CoAP-IoT using an integrated security mechanism based on IoT distributed sensors.	IEEE IoTSMMS	Published
2 and 4	4	Proposing a lightweight authentication scheme for CoAP-IoT-based IoT distributed sensors	MDPI Sensor	Published
3 and 4	5	Developing a novel detection approach with federated transfer learning for RPL-IoT-based IoT distributed sensors	MDPI Sensor	Published
5	2,3,4, and 5	The proposed work, methods, and the outcome were published in conference and journal papers	-	-

1.4 Research Questions

This thesis studies the applied authentication and detection schemes associated with using heterogeneous industrial IoT networks. Therefore, this study will answer the following research questions.

- I. How effective and efficient can the optimised integrated cryptography algorithms perform in a heterogeneous and limited resource IIoT network?
- II. How ML can be an effective mechanism as an intrusion detection approach in a heterogeneous and limited resource IoT?

1.5 Research Methodology

The research approach employed in this thesis is mainly based on experimental research, with analytical research also used at the start of the study. Experiment is a data gathering approach that allows the researcher to explore cause-and-effect correlations between the different research variables [12]. The research design chosen is based on the research goal objectives and aim to solve the research problems for this study's research scope. This study is exploratory since it seeks fresh insights into ways to develop unique security protocols for systematically integrating and constructing safe IIoT data exchange. The methodologies for clarifying the needed knowledge to fulfil the scope of this study will include literature review, experiments, computer simulations, and document analysis.

There are two main methods to research design: qualitative and quantitative. Often, qualitative research is more qualitative than quantitative research, and vice versa [13] [14]. Consequently, a mixed-method approach [15] was developed, defined as a blend of qualitative and quantitative methodologies. In further depth, qualitative research tries to investigate and comprehend the meaning that people or groups attribute to a human or societal situation. Using numerical representations to evaluate objective theories, a quantitative technique is employed to examine the correlations between variables [15]. These factors are quantifiable, allowing statistical analysis of the data. The mixed-method approach is a study that combines qualitative and quantitative methods while also including certain philosophical assumptions. If a mix of qualitative and quantitative methodologies is conceivable, all study forms are viable for both approaches within specific boundaries [13]. For this study, both qualitative and quantitative methodologies will be used. Consequently, the mixed-method approach enables researchers to conduct well-managed and well-documented research by presenting qualitative data using quantitative analysis methods [15].

Two types of data sources were used in this research: (1) Primary data sources; a simulation based on the Cooja-Contiki open-source operating system (OS) was used in this research to design, test setup, investigate gather information on the impact of the proposed security solutions schemes on heterogeneous IoT networks. Then, the gathered information will be evaluated and validated by comparing it with existing work and other techniques in terms of impact on performance, such as throughput, communication

duration, energy consumption network overhead, and detection accuracy. (2) Secondary data sources, the published journal articles, conference papers and technical reports.

A literature review was conducted to ascertain the security challenges, risks, and countermeasure mechanisms associated with the existing architectures of IoT and IIoT layers, security risks and vulnerabilities, and the most prevalent threat methodologies and countermeasures for addressing some of these security risks. The literature review established a thorough understanding of the existing research and methods for integrating security IoT technology within the various layers, as well as their strengths and weaknesses. The review also identified a knowledge gap in the study of designing an appropriate security scheme for IoT-embedded devices for heterogeneous IoT networks.

From the comprehensive investigation of the simulation and tools used in chapter 2, a simulation based on the Cooja-Contiki open-source operating system (OS) was used in this research to design, test setup, and gather information on the impact of the proposed security solution schemes on heterogeneous IoT networks. The data collected from the simulation were then evaluated and validated by comparing them with existing work and other techniques in terms of impact on performance, such as throughput, communication duration, energy consumption, network overhead, and detection accuracy. More details regarding the system overview, model architecture, data collection, and so on are included within the implementation scenarios. To this end, five research phases were conducted to achieve the research aim and objectives and are included below. In addition, figure 1.1 shows a high-level summary of the research methods used in this research.

Phase-1: The first phase was crucial for contextualising and understanding the research context of “Internet of Things applications security” through the literature review. The related work discusses several previous relevant results of this study to assess existing limitations and identify areas and existing gaps that can be studied and improved in IoT and IIoT security. The key outcomes and gaps in IoT and IIoT security were analysed critically. Most of the literature review had been conducted and this phase accomplished objective 1.

Phase-2: From phase-1, the research gaps were identified in order to develop the rationale for the need for designing virtual experiments. An empirical study was performed to

investigate the impact of security integrated with an IoT Constrained Application Protocol (CoAP)-based secure hash algorithm (SHA) and our Advanced Encryption Standard (AES) algorithm on performance, energy consumption and timely execution. This phase contained two stages in order to design, execute and evaluate the experiment. The first stage involved the operational architectures, including the software experiments required to develop the SHA and AES algorithms. The second stage included implementing and assessing the proposed algorithms by running the Contiki OS software in Cooja simulator to acquire results. This phase was intended to achieve objectives 2 and 5.

Phase-3: This phase extends phase-2 to propose a lightweight authentication scheme for distributed sensors. The authentication scheme integrates and optimises important features of the Datagram Transport Layer Security (DTLS) protocol in the CoAP architecture instead of implementing the DTLS in a separate channel. The proposed work involved designing a payload encryption scheme and an Optimised Advanced Encryption Standard (OP-AES). Similar to phase 2, the same software was used to collect the data (i.e., the Contiki OS) to execute and evaluate the results. The results were evaluated and compared with an existing solution in terms of various performance metrics, including communication duration, throughput, energy consumption and network overhead. This phase achieved objectives 2 3, and 5.

Phase-4: In this phase, an intrusion detection mechanism based on a federated ML-based approach was implemented to detect intrusions, including rank, version, hello flood, DoS, blackhole, Sybil, selective forwarding, sinkhole, theft, and wormhole attacks. The design process of federated transfer customised intrusion detection FT-CID contained three steps: dataset collection, a federated transfer learning (FTL)-assisted edge learning approach to an IDS and intrusion detection. The datasets were collected from a heterogeneous IoT network executed using Contiki OS software, an advanced meter infrastructure (AMI). This phase achieved objectives 2 and 5.

Phase-5: Upon successfully establishing dataset collection in phase-4, the datasets were trained, tested, and validated using an FTL approach. The performance of the FT-CID was then evaluated and compared with convolutional neural network (CNN), multilayer perceptron (MLP), and Logistic learning (LR) learning techniques using various metrics,

Chapter 1: Introduction

including precision, recall, F-measures, false positives, and intrusion detection accuracy. This phase achieved objectives 4 and 5.

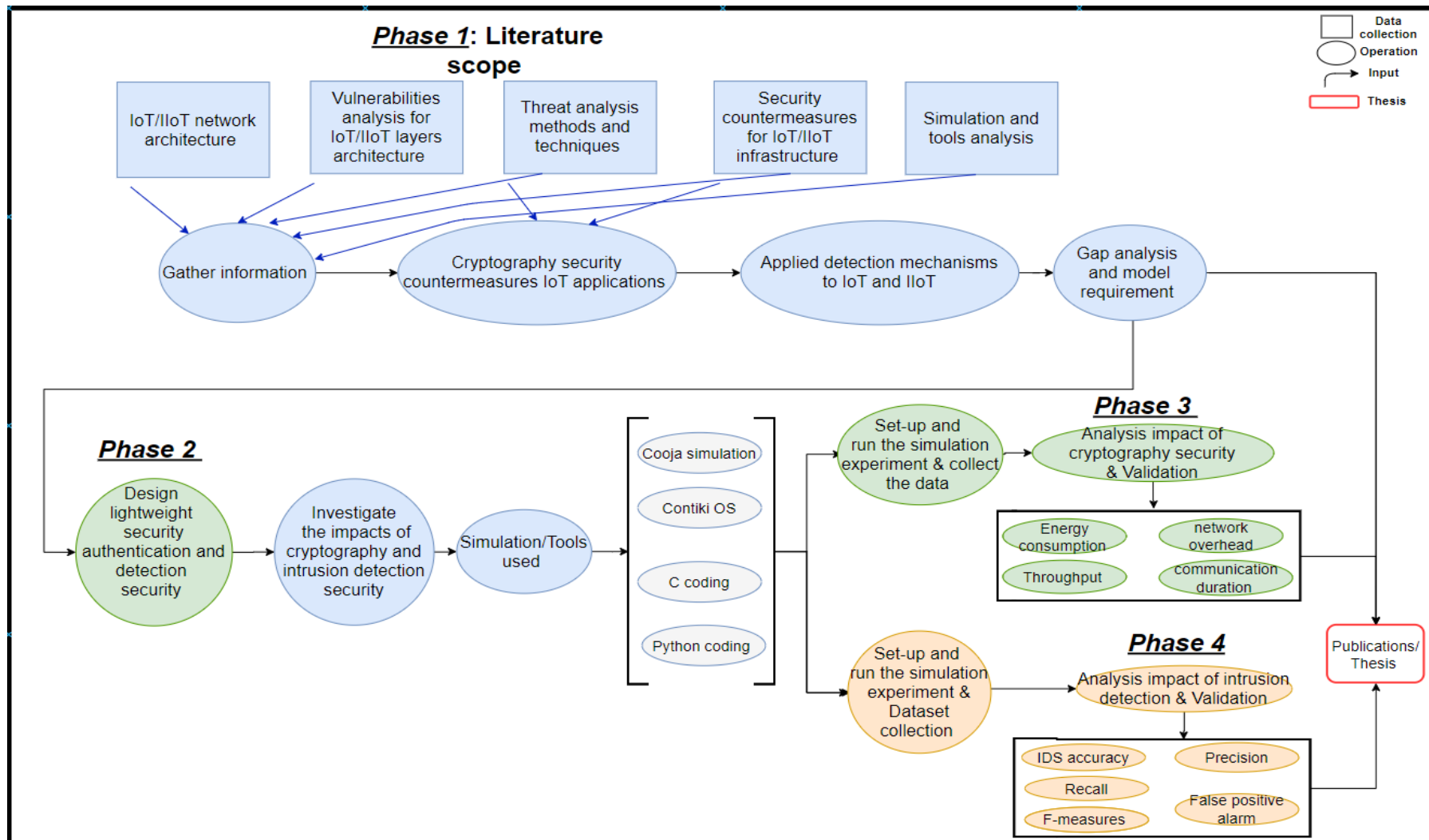


Figure 1-1: Methodological approach

1.6 Research Contributions

This research aims to study the impact of cryptography and detection mechanisms on several performance aspects of IoT networks in accomplishing resilient security systems. Generally, this thesis focuses on designing and applying integrated security protocols for heterogeneous IoT networks. To meet these needs, this thesis highlights the following four primary areas of focus as a contribution to developing integrated, efficient security for the IoT-enabled smart applications.

1. *A comprehensive review of the taxonomy of IoT layers, in which attacks and security countermeasures are presented followed by a critical analysis of their practical implementation in previous IoT security solutions focusing specifically on IoT layers.*
2. *A new security architecture has been developed for an IoT network based on the CoAP protocol using an integrated security protocol mechanism based on cryptography. We compare the performance of our proposed AES algorithm with an SHA in terms of communication duration and energy consumption. The integrated security scheme was executed and evaluated using Cooja simulation on the Contiki OS. The results show that the communication duration of the AES is six times faster than the SHA and was stable and energy efficient when the data size increased.*
3. *An efficient and secure authentication scheme for distributed sensor networks is proposed. Our Payload Encryption-based Optimisation Scheme (PEOS) is a mutual authentication scheme for the CoAP. Our proposed PEOS integrates and optimises DTLS in CoAP architecture to fix this and avoids a second DTLS channel. PEOS DTLS handshaking and retransmission are improved by payload encryption and NACK (i.e., negative acknowledgement) messages. The overhead is minimised without affecting the CoAP over distributed sensors by deleting duplicate protocol versions and sequence number features. The PEOS scheme uses an OP-AES. PEOS introduces parallel S-Boxes in SubBytes and delays MixColumns in the AES, reducing the need for additional storage registers. PEOS implements dynamic key generation to prevent CoAP key traceability and guessing attacks on dispersed sensors. Using tokens and dynamic key creation, the PEOS also avoids DDoS and confidentiality attacks. The efficiency of PEOS over distributed sensors in terms of lightweight security and authentication is evaluated on the Contiki OS using the Cooja simulator. Moreover, the proposed work is compared with an existing payload-based scheme*

and basic DTLS in the sensor network scenario. Based on the results with a large payload size, the throughput of the proposed method is improved by 8.7% compared to an existing payload-based mutual authentication (PbMA) scheme, consuming 1550 microjoules in a 50-node random topology.

4. We present a novel CID for heterogeneous IIoT network using FTL. The proposed FT-CID model uses an improved grey wolf optimiser (IGWO) technique to reduce noise and irrelevant features from a simulated RPL-IIoT dataset (RPL being the Routing Protocol for Low-Power and Lossy Networks Protocol). In the FT-CID, the deep FTL algorithm largely relies on a CNN algorithm to generate local models on two IIoT networks: global model updating and local model optimisation. In the FT-CID, transfer learning involves parameter transfer and relational knowledge transfer during global model updating and local model optimisation with stochastic gradient descent (SGD). The experimental results illustrate that the proposed FT-CID model yields 10.63% higher intrusion detection accuracy than the centralised CNN model on a heterogeneous IIoT network or applications such as AMI.

1.7 Thesis Structure

This thesis is structured in accordance with university requirements as a “paper-format thesis”. This thesis consists of a collection of chapters based on previously published works. Chapters 2, 4 and 5 have been published in the MDPI *Sensors* journal, and chapter 3 has been published in IEEE Xplore. The chosen methodological techniques for achieving the stated purposes and goals are offered in accordance with the corresponding chapters. All published articles were authored by the initial author and examined and approved for publication by two primary supervisors and one associate supervisor, as well as by MDPI *Sensors* and IEEE Xplore reviewers.

Chapter 2: This chapter presents an examination of the literature on the integrity of IIoT systems, highlighted along with a case study discussing the security solutions for an essential IIoT application: the smart grid. A taxonomy of IoT layers and attack and security countermeasures is also presented, followed by a critical analysis of their practical implementation in previous IoT security solutions focusing specifically on IoT layers. Simulation tools and OSs used to investigate security solutions are also presented.

Chapter 1: Introduction

Finally, research limitations and recommendations highlighted for future work are discussed.

Chapter 3: This chapter develops a new security architecture to secure the communication of a distributed sensor network based on the CoAP using symmetric cryptography rather than a costly computation protocol, such as the DTLS. An experiment is implemented using the Contiki OS under various operating conditions to measure the performance of the proposed security scheme. This chapter contains a model with two parameters for analysing the statistical shape of the impact of security on performance using the script editor provided on the Contiki OS platform, which is illustrated further in chapter 4. To assess the success of this strategy in delivering the impact of security on performance, a statistical classification method is created using the script editor platform integrated with Contiki and used for the gathered data.

Chapter 4: This chapter extends the work in chapter 3 and proposes a lightweight authentication scheme for heterogeneous distributed IoT sensors. This chapter presents a new PEOS for lightweight authentication that modifies the DTLS handshaking and retransmission processes using payload encryption and NACK messages. It also removes duplicate features of the protocol version and sequence numbers without having a negative impact on the performance of the CoAP. Moreover, the PEOS attempts to improve the CoAP over distributed sensors in optimised AES operations, such as the parallel execution of S-Boxes in SubBytes and delayed MixColumns. We develop the work presented in chapter 3 to provide a more extensive statistical framework for a wide range of low-embedded sensors. We also address the limitations by using the NACK mechanism and developing a detection algorithm. Several tests were carried out under various operating conditions to assess the performance and resource consumption of the lightweight authentication proposed. We describe four performance metrics for analysing the statistical shape of the impact of security on performance and compare it with basic DTLS and existing solutions. Moreover, the efficiency of the PEOS is evaluated on the Contiki OS using the Cooja simulator.

Chapter 5: This chapter proposes a novel detection using ML model to customise a distributed IDS to detect routing intrusions in heterogeneous IIoT networks. The design process of the proposed work contains three steps: dataset collection, an ML-assisted edge

learning approach to an IDS, and intrusion detection. The experimental model uses the Contiki OS to generate two RPL-IIoT datasets with normal and abnormal traffic for two IIoT networks. The final part of this chapter includes several performance metrics and the experimental setup, as well as an evaluation of the proposed model in relation to heterogeneous IIoT networks.

Chapter 6 concludes the research and discusses the potential directions for future research.

1.8 List of publications

- 1- Abosata, N., Al-Rubaye, S., & Inalhan, G. (2022). Lightweight Payload Encryption-Based Authentication Scheme for Advanced Metering Infrastructure Sensor Networks. *Sensors*, 22(2), 534. <https://doi.org/10.3390/s22020534>
- 2- Abosata, N.; Al-Rubaye, S.; Inalhan, G.; Emmanouilidis, C. Internet of Things for System Integrity: A Comprehensive Survey on Security, Attacks and Countermeasures for Industrial Applications. *Sensors* 2021, 21, 3654. <https://doi.org/10.3390/s21113654>
- 3- Abosata, Nasr Rajb A., Andrew H. Kemp, and Mohsen Razavi. "Secure Smart-Home Application Based on IoTCaP protocol." 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, 2019. <https://doi.org/10.1109/IOTSMS48152.2019.8939199>
- 4- Abosata, N., Al-Rubaye, S., & Inalhan, G. (2023). Customised Intrusion Detection for an Industrial IoT Heterogeneous Network Based on Machine Learning Algorithms Called FTL-CID. *Sensors*, 23(1), 321. <https://doi.org/10.3390/s21113654>

1.9 References:

1. Ashton K. RELATED CONTENT RFID-Powered Handhelds Guide Visitors at Shanghai Expo Despite Sluggish Growth, Taiwan's RFID Industry Remains Committed Mobile RTLS Tracks Health-care Efficiency RFID Journal LIVE! 2010 Report, Part 2 That 'Internet of Things' Thing. That 'Internet of Things' Thing-RFID Journal. 2010. Available at: <http://www.rfidjournal.com/article/print/4986>
2. Vermesan O., Friess P. Internet of things applications: From research and innovation to market deployment. Internet of Things Applications: From Research and Innovation to Market Deployment. 2014. 1–371 p. Available at: <http://riverpublishers.com/river>
3. Maede Zolanvari, Jain R. IoT Security: A Survey. 2015; : 1–15. Available at: http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_sec/index.html
4. Girard M. Standards for Cybersecure IoT Devices: A Way Forward. 2020. Available at: www.postscapes.com/
5. Russell B., Van Duren D. Practical Internet Of Things Security. Practical Internet of Things Security. 2016. 281–302 p. Available at: www.packtpub.com
6. Angrishi K. Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV): IoT Botnets. 13 February 2017; Available at: <http://arxiv.org/abs/1702.03681>
7. Abosata N., Al-Rubaye S., Inalhan G., Emmanouilidis C. Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. Sensors. MDPI AG; 2021. Available at: DOI:10.3390/s21113654
8. Flashpoint—Mirai Botnet Linked to Dyn DNS DDoS Attacks. Available at: <https://www.flashpoint-intel.com/blog/cybercrime/mirai-botnet-linked-dyn->

Chapter 1: Introduction

dns-ddos-attacks/ (Accessed: 16 March 2021)

9. Nasr Abosata., Andrew H. Kemp., Mohsen Razavi. Secure Smart-Home Application Based on IoTCoAP protocol, Sixth International Conference on Internet of Things: Systems, Management and Security (IoTSMMS): Granada, Spain, October 22-25, 2019, IEEE.
10. Abosata N., Al-Rubaye S., Inalhan G. Lightweight Payload Encryption-Based Authentication Scheme for Advanced Metering Infrastructure Sensor Networks. *Sensors*. MDPI; 1 January 2022; 22(2). Available at: DOI:10.3390/s22020534
11. Abosata N., Al-rubaye S., Inalhan G. Customised Intrusion Detection for an Industrial IoT Algorithms Called FTL-CID. 2023;
12. King GRD. Creswell's appreciation of arabian architecture. *Muqarnas*. 1991; 8: 94–102. Available at: DOI:10.1163/22118993-90000268
13. Kumar R. Research Methodology a step-by-step guide for beginners. Fifth. Aly Owen (ed.) London: SAGE; 2018. 441 p.
14. LEAVY P. RESEARCH DESIGN. Leavy P (ed.) New York London: The Guilford Press; 2017. 1–320 p. Available at: <https://www.ptonline.com/articles/how-to-get-better-mfi-results>
15. Creswell WJ., Creswell JD. Research Design: Qualitative, Quantitative and Mixed Methods Approaches. Fifth edit. *Journal of Chemical Information and Modeling*. Los Angeles: SAGE; 2018. 1–388 p. Available at: <https://lccn.loc.gov/2017044644>

Chapter 2: Cybersecurity for the Internet of Things and System Integrity: Challenges and Requirements

2.1 Abstract

The growth of the Internet of Things (IoT) offers numerous opportunities for developing industrial applications such as smart grids, smart cities and smart manufacturers. By utilising these opportunities, businesses engage in creating the Industrial IoT (IIoT). The IoT is vulnerable to hacks and, therefore, requires various techniques to achieve the level of security required. Furthermore, the wider implementation of the IIoT causes an even greater security risk in relation to its benefits. A comprehensive literature review and critical analysis of current academic literature were conducted for this study in order to identify the essential aspects of developing the cybersecurity landscape of IoT/IIoT heterogeneous networks.

2.2 Introduction

There are increasing concerns among experts and policy-makers regarding protecting information and information technology (IT) infrastructure from attacks. People, process and technology enterprise constituents are the prime targets of cyber-sabotage and the security of industrial systems has become of paramount concern to organisations across various industries [1]. Most industrial control systems (ICSs) are upgraded legacy systems with connectivity issues that are susceptible to potential attacks. These ICSs were not designed for such connectivity, so there is a need to upgrade their security designs. This is being made possible by the increasingly common use of the IoT, which connects every piece of equipment in a network to the Internet to facilitate the communication and management of the various pieces of hardware. The result is that more ICSs are interconnected both with each other and to the Internet, drastically increasing the amount and scale of cyberattack vectors. The key to the successful application of the IoT in industry is to enable real-time monitoring of the network infrastructure and its associated service operations to support the automation of data delivery to achieve secure and high-quality services [2]. However, criminals, terrorists, and hacktivists can launch a wide variety of cyberattacks against IoT-enabled infrastructure [3]. One paper, authored in collaboration with the cybersecurity research firm MedSec (Miami, Florida), outlined two

potential forms of cybersecurity breaches in numerous St. Jude Medical pacemaker models: (1) a “crash” attack, which could lead to device communication disabling or high-rate pacing; and (2) a “battery drain” assault [4]. This type of incident serves as a practical illustration of the harm that can result when hackers gain access to crucial facilities and steal sensitive data.

The architecture of the IoT and the IIoT consists of four components: devices and equipment, networks, the cloud, and applications. Figure 2.1 demonstrates the high-level design of the four layers of the IoT: the perception layer, the network layer, the processing layer, and the application layer [5]. The perception layer comprises numerous kinds of sensors, security cameras, robots, etc. This inventory of machines can be found in an industrial environment, where they could be followed by equipment such as industrial robots and automated guided vehicles (AGVs). These devices gather sensory data, track environmental factors and transport raw materials [5], [6]. The network layer may have different kinds of connectivity networks, including WiFi/IEEE 802.15.4, Bluetooth, LoRa, 6LoWPAN and NarrowBand-IoT, and is responsible for relaying information to the processing systems of the following layer. The basis of this layer is industrial Ethernet, which transmits data either to the cloud or to other computers [5], [6]. The processing layer consists of databases and servers and carries out many operations, such as decision-making, refining computation algorithms and storing large volumes of data. The application layer manages and ensures the meeting of the application-specific needs of the end-user. Some significant IoT/IIoT applications include smart homes, smart robots, smart healthcare, and vehicle ad-hoc networks (VANETs), and smart grid, smart automation and smart factory warehousing are known as IIoT applications [5], [6].

IoT and IIoT applications are crucial to the support of services and sensitive data infrastructures. The amount of data generated will increase with the rise in applications for healthcare, household and industrial use [7]. Basic security concepts, such as confidentiality, integrity, authentication, availability, and non-repudiation, should be appropriate to guarantee IoT protection. However, IoT devices are constrained by their power and memory. Therefore, security difficulties have become a concern when contemplating employing an appropriate security strategy that considers all those limitations.

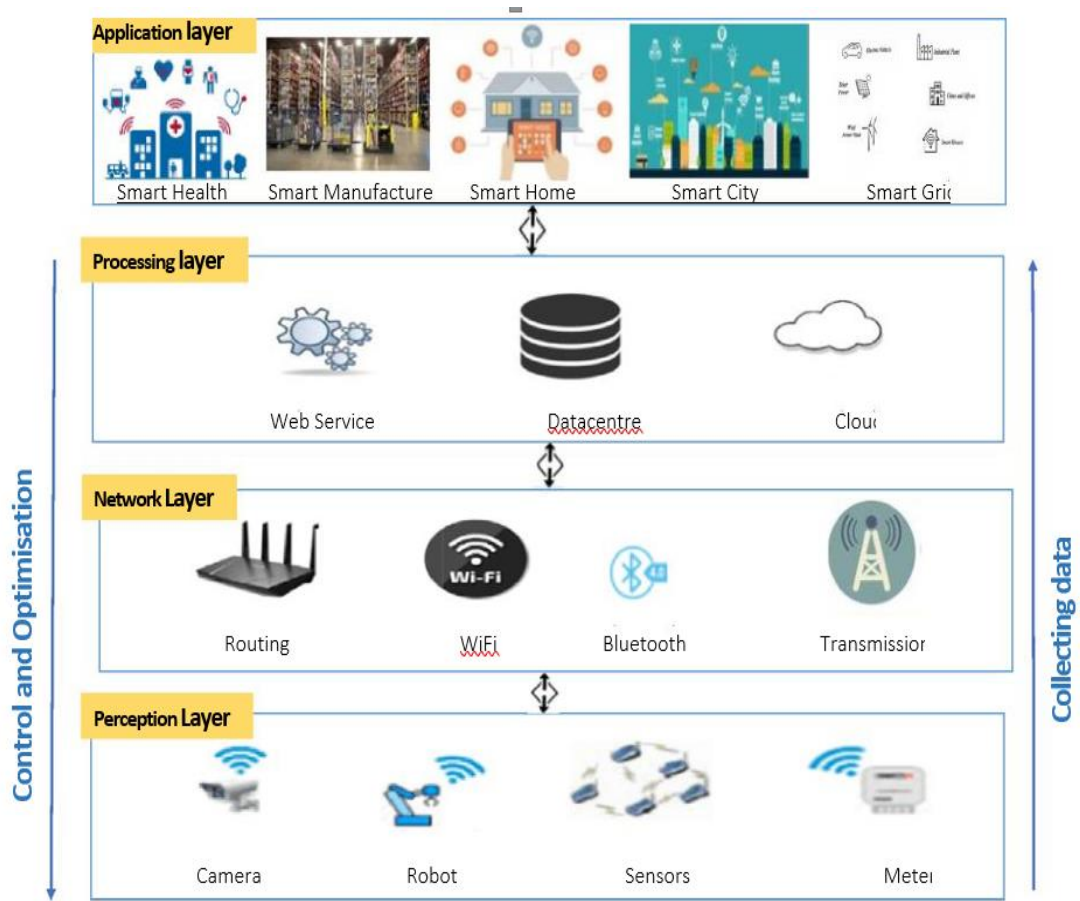


Figure 2-1: High-level IoT/IIoT architecture

This chapter aims to present a comprehensive literature review to evaluate studies on cyberattacks, countermeasures, simulations and tools, and analyses the security solutions commonly implemented. The major contributions of the chapter include an assessment of the integrity of IIoT systems, which has been highlighted along with a case study discussing the existing security solutions for an important IIoT application: the smart grid. A taxonomy of IoT layers, attacks and security countermeasures is also presented, followed by a critical analysis of their practical implementation in previous IoT security solutions, focusing specifically on IoT layers. Some of the simulation tools and operating systems (Oss) used to investigate security solutions are then presented. Finally, some of the limitations are highlighted and recommendations for future work are discussed.

The remainder of the chapter is organised as follows. Section 2.3 presents the layers of the IoT and their corresponding vulnerability to security attacks and potential solutions. Section 2.4 presents the security services required and discusses standardised security

solutions. Section 2.5 highlights the testbed tools that are used in IoT research. Section 2.6. discusses solutions presented in existing studies and provides a critical analysis of this research. Section 2.7 highlights the issue of the integrity of the IIoT system, as well as some of the existing studies on the industrial application of smart grids. Section 2.8 presents open research directions and section 2.9 concludes the chapter.

2.3 Classification of Security Attacks

The IoT encompasses a wide range of devices and equipment, ranging from small, embedded devices to large, advanced servers. There is, therefore, a need to highlight the security issues at the different IoT layers. Figure 2.2 shows a classification of IoT security issues along with potential security solutions for each of the IoT layers. The most common type of IoT architecture is divided into three layers: perception, network, and application layers. These three layers contain a large range of information with various enabling features and technologies. The following sections present the IoT/IIoT layers with the potential each layer has for receiving security attacks.

2.3.1 Perception Layer with Security Attacks

The perception, or device, layer includes objects with attached sensors, smart meters, robots, cameras, etc. The perception layer identifies and collects the target sensor data, such as those related to movements, vibrations, chemicals in the atmosphere, heat, orientation, humidity or acceleration. These data are sent to the network layer and then to an information processing system [8].

Node Capture Attacks: These attacks involve attackers capturing/replacing a node or modifying hardware, exposing sensitive data for the management of digital rights, such as access/cryptography keys. Replacement nodes may then act maliciously, making the whole IoT network insecure [9].

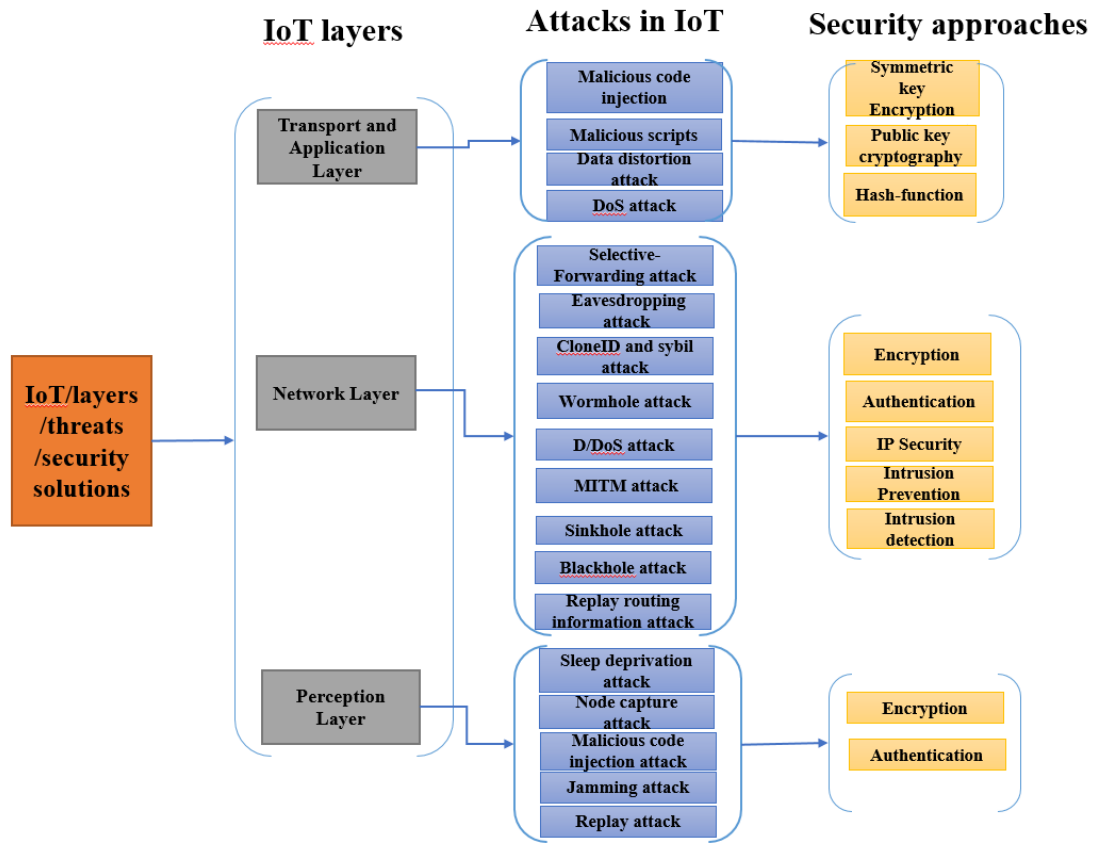


Figure 2-2: Classification of IoT layers, threats, and security approaches

Malicious Code Injection Attacks: Malicious codes are injected within the node's memory via its debug modules. These codes can carry out undesirable activities and may even allow an attacker to access the entire network. Attacks generally occur when upgrading software/firmware through an over-the-air (OTA). When devices are in operating mode (e.g., during a scheduled firmware update), an intruder could insert trojans into the machine (requires device reboot). The security challenge, in this case, is divided into two categories: proper authentication, and identification of an edge device in the network and ensuring that drivers or malware are not installed on peripheral devices under the guise of updates or upgrades [9].

Sleep Deprivation Attacks: Sleep deprivation attacks resemble denial-of-service (DoS) attacks (see the next subsection for more details about DoS attacks) as they drain the edge device's batteries, and these devices are usually intended for low-power operation. Consumption is increased by making hardware modifications or injecting infinitely-looping codes into the memory [9].

Jamming Attacks: These attacks interfere with the tag reader air interface, disrupting communications or altering communications. Jamming is done through a long-distance, powerful transmitter or passively, by, for example, shielding, which can succeed due to the sensitivity of the interface. Radio-frequency identification (RFID) systems can be jammed through radio noise that matches the system's frequency [10].

Replay Attacks: This type of attack involves repeating an authentication code used by an authorised individual, either by cloning the authorised tag or sending a signal again after eavesdropping on signals sent by a device with the correct antenna and card. Replay attacks require the specific data that a tag sends in communications [10].

2.3.2 Network Layer with Security Attacks

The network layer is responsible for transmitting data to and from various devices or applications through interfaces or gateways between heterogeneous networks and via a variety of communication technologies and protocols. The network layer receives the processed data from the perception layer and determines the routes for transmitting it to the IoT devices, hub and gateway through integrated networks [11].

Selective-Forwarding Attacks: These are DoS attacks. Only selected packets are forwarded by malicious nodes with the aim of disrupting the route of the path, although any protocol could be targeted. Thus, attackers might forward every RPL (i.e., Routing Protocol for Low-Power and Lossy Networks) control message while dropping all other packets. When combined with a sinkhole or other attack, selective forwarding can be highly damaging [11].

Eavesdropping Attacks: This type of attack relies on the signal sent out by RFID tags when required by RFID readers. The attack works by eavesdropping on the signal being sent to an RFID reader with authorisation to identify the frequency and tag group used. Clear text is used by the majority of RFIDs based on cost and memory limitations, and this allows the eavesdropping to succeed [10].

Clone ID and Sybil Attacks: Clone ID attacks involve copying a legitimate node's identity to a second node for reasons such as accessing a greater proportion of a network or counteracting vote schemes. Sybil attacks utilise a number of logical entities for one

physical node to control an extensive network area with no need for more nodes to be used [11].

Wormhole Attacks: These attacks generally target traffic flows and network typologies. Wormhole attacks are performed by producing a tunnel that links two attackers for the selective transmission of traffic via this route [12].

Denial of Service (DoS) Attacks: In DoS attacks, a targeted network/computation source is disrupted, potentially reducing capacity on the network. In the IoT, DoS attacks can be a simple DoS attack or a distributed denial-of-service (DDoS) attack. A simple attack needs a tool for sending packets to crash or restart a system/network, whereas a DDoS may use one attacker with less force than a proxy. These attacks can disrupt and prevent access to networks [13].

Man-in-the-Middle (MITM) Attacks: These attacks involve the interception and alteration of node-to-node communications and use a range of strategies. After a node–node link is interrupted and the data altered in real time, the link can then be monitored by the attackers [14].

Sinkhole Attacks: Sinkhole attacks compromise nodes within a network and use them to transmit false routing data to adjacent nodes, claiming to have the shortest route to the base and then dropping or modifying packets routed through them [15].

Blackhole Attacks: These attacks involve a malicious node placed within a network that quietly drops every packet routed through it, with nothing being passed on [12].

Spoofed, Altered, Replayed Routing Information: Mutual direct attacks involve spoofing, altering or replaying routing information, targeting routing data in node-to-node exchanges. Spoofing attacks utilise issues due to the ability to detect an IoT device within a system by, for example, producing a fake error message or creating a routing loop [13].

2.3.3 Application Layer with Security Attacks

The final layer in this novel architecture design is the application layer. This layer formats and presents data and delivers a range of applications to diverse types of users by defining different smart applications for use by the IoT, including smart health, homes, cities, industries and transportation. This layer provides the user with a particular application

based on object sensor data [11]. Security is a central challenge in this layer, with frequent issues arising. These security issues include the following.

Malicious Code Injections: These attacks exploit coding within the software, which damages systems or leads to other unwanted impacts and can avoid detection by antivirus applications. The code may be self-activating or activated when the user takes a specific action [14].

Malicious Scripts: These involve networks or IoT devices that are connected to the Internet. The attack is carried out by running malicious codes or x-Scripts, which look like legitimate scripts and which the user must access, and can involve data theft and systems failure [16].

Data distortion attack: These attacks use code within the software to damage systems or lead to other unwanted impacts and avoid detection by antivirus applications. The code may be self-activating or activated when the user takes a specific action [16].

2.4 Connectivity and System Integrity Approaches

The IoT allows devices to share and transfer data among users and devices to accomplish specific goals. Therefore, security is intrinsic to the deployment of the IoT due to the sensitivity and connectivity of applications, such as in relation to the military and defence and to smart homes, healthcare, and railway systems [17], [18]. In order to put in place effective countermeasures, we must explicitly define the security goals we are attempting to achieve. Confidentiality, integrity, and availability (the CIA triad) are the general security goals for all systems and services. In addition, IoT networks should be constructed with authentication, lightweight security, non-repudiation, and response protection in mind. The following are the most important security services in the IoT in relation to industrial applications.

Confidentiality: This is crucial to ensure that the messages sent are secure and accessible to only authorised objects because an intruder could catch the data flow between sender and receiver and confidential data could be exposed. Moreover, these data should be unknown to intermediate users. In the IoT, the objects could be machines, devices, sensors, and internal or external objects. For example, it is required in an IoT system that

the end-to-end message is secret. Furthermore, the data stored in IoT device should be unobservable by unauthorised users. Data confidentiality services are supported through mechanisms such as encryption and decryption [19], [20].

Integrity: This is important to ensure that data exchange between multiple IoT devices is accurate. This means that the data come from the correct source and ensures the data are not modified during transmission [21]. For instance, the stored data of a medical patient should not be modified. The most useful forms of protection for providing an integrity service are message integrity codes (MICs) and the hash function mechanism. Furthermore, maintaining end-to-end security communication in an IoT system is crucial for the integrity feature [22], [23].

Authentication: This ensures that data are sent from the right device and not modified during transmission. In communication, each object should be able to identify as well as authenticate each other. Nevertheless, this process is very challenging due to the nature of the IoT, as numerous objects, including people, devices, processing units, and service providers, may be required to communicate with each other. Thus, an authentication mechanism is necessary for every piece of communication within the IoT [24].

Availability: The vision behind the IoT is to link as many devices as possible. Availability ensures that all the data are available to the objects that are connected whenever the data are needed. Nevertheless, the data component is used in the IoT, and devices and services should be available and reachable whenever needed [25]. Firewalls and intrusion detection systems (IDSs) are the most effective security mechanisms for detecting malicious activity and intrusions and for ensuring the availability of services [26].

Lightweight Solutions: Lightweight security is considered a unique feature due to the power capabilities and computational speed of IoT devices. A lightweight solution should be considered when designing and executing protocols in the encryption and authentication of IoT devices. These types of algorithms can run with limited capabilities on IoT devices [27], [28].

Replay Protection: A data packet stored in an intermediate node can be compromised and replayed later. The replayed data can comprise a sensor reading, such as a

temperature, or a blockchain transaction. It is crucial to provide a mechanism for detecting replayed or duplicate messages. This can be accomplished through a nonce mechanism, integrity-protected sequence number, or timestamp [29].

2.4.1 Communication and Security

Communication in the IoT must be preserved by a designated security mechanism, as discussed in section 1.4. Security can be provided in the different layers by combining Internet security solutions. Generally, the protection of IoT communication can be delivered through end-to-end security or intermediate devices [30]. An IoT stack with a standardised security solution at the IoT layers is shown in Table 2.1.

Table 2-1: IoT stack based on standardised security solutions

IoT layer	Protocols	Security solution	IoT layer
Application layer	CoAP, MQTT	CoAP, MQTT	Application layer
Transport layer	UDP, TCP	DTLS, TLS	Transport layer
Network layer	IPv6, RPL	IDS, Secure RPL, IPsec	Network layer
Perception layer	IEEE 802.14.5	IEEE 802.24.5 security	Perception layer

IEEE 802.15.4 Security: Link Layer

The IEEE 802.15.4 protocol is widely used for short-range communication in the IoT environment. It is specifically responsible for information transmission at the physical and medium access control (MAC) layers. The IEEE 802.14.5 protocol is used as link layer security in communication networks, such as the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN). The link layer provides secure communication through hop-by-hop aggregation to all the nodes in the network. The 802.15.4 specification outlines different security sets that can be included in encryption and authentication (AES-CCM), encryption only (AES-CTR), authentication only (AES-CBC-MAC), and no security. All the communication is protected through a pre-shared key. The shared key must be secure because if an attacker gains that key, the whole network is compromised [31]. Although the IEEE 802.15.4 protocol provides critical

security services, it does have some limitations. More precisely, it is incapable of safeguarding the privacy and confidentiality of acknowledgement (ACK) messages.

6LoWPAN

6LoWPAN provides limited and low-power devices with native IPv6 connectivity at a minimal overhead cost. This IPv6 support enables connectivity with other hosts and routers on the Internet. Many of 6LoWPAN's capabilities have been adopted by physical link layer technologies, such as BLE. 6LoWPAN was initially developed to provide full IPv6 compatibility under IEEE 802.15.4. Typically, the terminology 6LoWPAN network refers to an LLN which employs 6LoWPAN principles [32], [33]. Figure 2.3 illustrates the 6LoWPAN stack. Note that 6LoWPAN adaptation layer is located between the IPv6 and IEEE 802.15.4 link layers in the 6LoWPAN-based stack. In many implementations, the UDP and IPv6 layers are directly integrated into 6LoWPAN via network and transport APIs. As a result, 6LoWPAN adaptation is commonly depicted as one of the IPv6 layers. The IPSec/Internet Key Exchange (IKE) combination is a possibility for supporting 6LoWPAN security [33]. Well-known network security protocols are utilised to construct a secure connection between two authenticated nodes. IKE is used to establish up security associations (SAs) across entities, whereas IPSec offers encryption and authentication.

application	CoAP	
transport	UDP	ICMP
network	IPv6	
	6LoWPAN	
link	IEEE 802.15.4	
physical	IEEE 802.15.4	

Figure 2-3: 6LoWPAN stack [33]

LoRaWan Security

The long-range wide area network (LoRaWan) was designed to enhance the functionality of low-power wide area networks (LPWANs) in terms of power consumption, storage, long-range communication and transmission cost. The four critical components of the LoRaWAN architecture are as follows: end nodes, gateways, networks, and application servers. End nodes are typically IoT devices that gather data from their physical environment and transmit them to gateways through the LoRa physical layer. The

gateways then transmit these data to a network server. This can be carried out through IEEE 802.11 (Wi-Fi), satellite, IEEE 802.3 (Ethernet) or other systems. The network server is responsible for data control by conducting the necessary security operations and scanning for duplicate packets. It then sends the information to application servers, which function as the foundation for software applications [34]. Two layers of encryption are used in LoRaWan technology. The first security layer is responsible for authenticating the data on the end nodes. This process is performed between the end nodes and the network server using an AES-CTR 128 secret key, called the network session key. The second layer ensures that the privacy of the end nodes is protected by using an AES-CTR 128 secret key, called the application key, between the end nodes and the application servers. As a consequence, the LoRaWaN technology protection of these keys is important. If a secret key is compromised, a potential attacker would have access to and could alter data. In addition, when communicating between end nodes and gateways, it is worth remembering that the length of the payload remains constant before and after encryption. An attacker can exploit this by decrypting the encrypted messages and recovering the network session key [35].

IPsec: Network Level

Internet Protocol Security (IPsec) provides end-to-end security for the network layer and contains a number of protocols, such as the Encapsulating Security Payload (ESP) protocol, which ensures confidentiality and integrity, and the Authentication Header (AH) protocol, which ensures authentication and integrity [36]. IPsec can be used with IP-based protocols and the transport layer, such as the User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Constrained Application Protocol (CoAP) and Hypertext Transfer Protocol (HTTP). IPsec is considered the most appropriate in terms of end-to-end security for the IoT because of the default security policies that run on constrained devices [37]. Virtual private network (VPN) technology is also considered the first line of protection for IoT networks, as it enables a private and secure tunnel to be established between communicating parties to safeguard the data exchanged against tampering and passive and active intruders. The tunnel construction process begins with establishing a preparatory encrypted and stable tunnel, and then the encryption keys and parameters are negotiated from inside that channel [38]. In addition to IPsec and the VPN,

RPL provides security mechanisms to control messages. Although this is optional, the integrity of the authentication process and the confidentiality of the control message are guaranteed [39].

RPL

WSN routing protocols provide routing elements relevant to the Internet of Things. Similar to IoT networks, WSNs are limited in terms of power consumption, resulting in resource, computational, and communication constraints that significantly impact routing. The Routing for Low Power (RPL) protocol is a hierarchy IPv6 address-centric routing protocol. RPL was created with three objectives: energy efficiency, reliability, and scalability [33]. RPL constructs IPv6 networks containing thousands of embedded devices utilising multiple low-power wireless hops that combine dependable transmissions with a battery life of several years. RPL is based on the premise that every device of a WPAN can function as a router, facilitating the deployment of the network. This indirectly restricts the features of the routing mechanism to ensure support for devices with low complexity [40]. RPL is a protocol designed to operate at the IPv6 network level and, as such, differs from mechanisms that operate at lower layers, such as 6LoWPAN layer 2 mesh forwarding. The procedure begins with establishing a DODAG (Destination Oriented Directed Acyclic Graph), which allows devices to monitor the routing topology. A DODAG is constructed using an objective function that utilises a combination of constraints and metrics to find the optimal path. DODAGs may be generated by objective functions that seek the optimal path that (1) minimises ETXs (metric) across encrypted links (constraint) or (2) low latency (metric) across unencrypted links (constraint), or (3) reduces power consumption (metric) or (4) utilises wireless links (constraint). The IoT gateway, the LBR (Low Power PAN Border Router), often called as root, initiates the creation of the DODAG [33].

Multiple roots can serve several devices that overlap. The graph creation is enabled by the transmission of three control routing messages over ICMPv6: (1) DODAG Advertisement Object (DAO) (DAO), (2) DODAG Information Solicitation (DIS), and (3) DODAG Information Object (DIO). The LBR begins by periodically sending DIO messages to its neighbours that contain information about the DODAG. Those devices which receive the DIO message and elect to join the graph are categorised as LBR

children. For each child device, an LBR assumes the role of parent and provides itself with the rank that indicates its rank in the network hierarchy related to a root and serves as the DV path cost. Generally, the closer a device is to the root, the lower its rank. The rank's conceptual purpose is to allow the routing algorithm to identify the presence of loops. If the device has been configured with routing capabilities, it can broadcast the route to the LBR by transmitting its own DIO to its neighbours [33]. The route allows all devices in the network to submit datagrams to the LBR via the routing method shown in Figure 2.4.

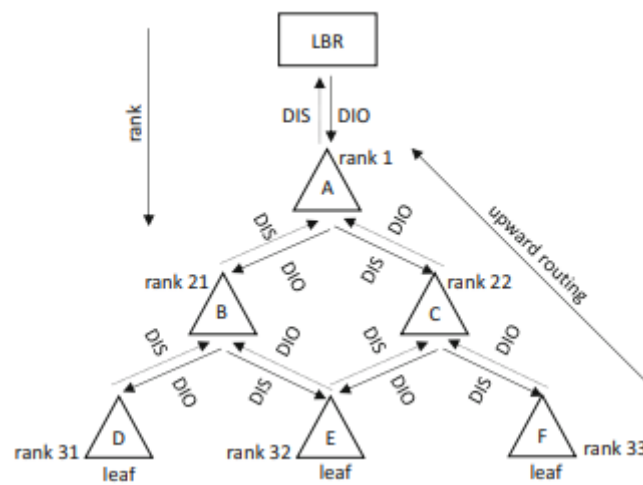


Figure 2-4: RPL routing mechanism [33]

CoAP: Transport Layer

In addition to the IPsec datagram mentioned above, the Secure Socket Layer (SSL) and Transport Layer Security (TLS) are the most commonly used security protocols in web protocols that run over the TCP. However, they are not efficient for use in communication with smart objects in low-power wireless networks [39]. Another version of TLS is Datagram Transport Layer Security (DTLS), which runs over the UDP. DTLS provides integrity, authentication and confidentiality solutions, as well as end-to-end security between applications and transport layers. The CoAP is an IoT web protocol, and DTLS is used as a security solution for the CoAP [41], [42]. DTLS was developed to be as resilient as TLS and to offer comparable levels of security. Unlike TLS, DTLS does not need reliable transport to function. Specifically, DTLS utilises UDP as its transport protocol, which makes it more suitable to be used in 6LoWPAN as well as IoT scenarios

[32]. DTLS addresses the critical issue of reliability in the security context by providing (1) message size management, (2) loss handling, and (3) datagram reordering. Note that stream-based TCP transport handles these features under TLS. DTLS consists of two layers: (1) the low layer, known as DTLS Record Protocol, which ensures the privacy of connections through symmetric encryption and message authentication codes, and (2) the high layer, which facilitates the DTLS Handshake Protocol, the DTLS Change Cipher Spec Protocol, the DTLS Alert Protocol, and the DTLS Application Data Protocol. The DTLS Handshake Protocol allows security parameters to be negotiated. A DTLS Change Cipher Spec Protocol is utilised to indicate changes in ciphering strategies and to specify which records are secured by the negotiated parameters. The DTLS Alert Protocol is utilised to indicate an error condition and terminate a session. Lastly, the DTLS Application Data Protocol to ensure that the DTLS Record Protocol fragments and transmits messages [33]. Figure 2.5 describes a DTLS handshake. A flight is any communication from client to server or from server to client. A client initiates a session, while a server acknowledges the session establishment request.

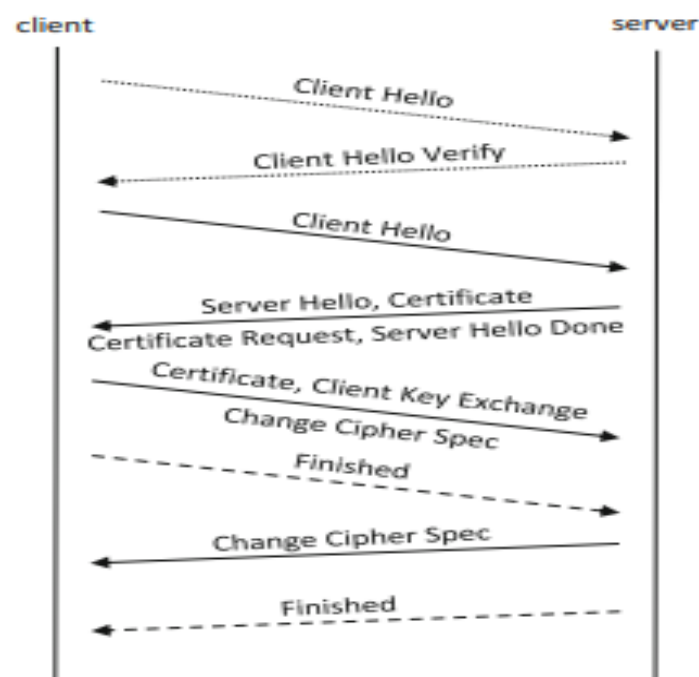


Figure 2-5: DTLS Handshake [33]

2.4.2 Network Security

The main task of the network layer is to address and route data packets for the seamless integration of a large number of devices into a single collaborative network. The most common network layer protocols are the IPv4/IPv6, RPL and 6LoWPAN. Although integrity and confidentiality services secure the messages within the communication, several types of attack are likely to accrue on networks, significantly disrupting the availability of the security services [43]. For example, eavesdropping attacks can analyse traffic transmitted over a network and affect the privacy of the data. Traffic is then vulnerable to various attacks, such as DoS, MITM, and illegal access attacks. DoS attacks can be launched to disrupt networks and the network layer of the IoT is highly vulnerable to MITM attacks. The security of the communication will be exposed if the attacker gains access to the keying of the devices [44]. However, protecting the network in the IoT is crucial in order to secure services such as integrity, availability and confidentiality when transferring information at the network layer. Firewalls, IDSs and key management can be employed to secure a network against routing attacks (Sybil, rankhole, blackhole, clone ID, etc.) [45].

Furthermore, 6LoWPAN is susceptible to attacks from both inside and outside the network. As discussed above, some security solutions can protect 6LoWPAN, such as IPsec, 802.14.5 and DTLS. RPL networks are also vulnerable to routing attacks aimed at disrupting a network. The RPL specification defines various security modes: “unsecured” RPL control messages; “pre-installed” RPL, in which nodes have been configured with a symmetric key to generate a secured RPL message; and “authenticated”, which are used to operate the device as a router. Furthermore, when a device joins a network using both a pre-installed mode and a preconfigured key, every RPL message has a protected variant, as well as AES/CCM procedures, which are utilised to support integrity and confidentiality [31], [46].

2.4.3 Application Security

There are numerous challenges to ensuring the security of applications because of the absence of standards that control their communication and development processes. For instance, identity authentication and access permissions can be a reason for concern. There are many different IoT applications, such as smart homes, smart healthcare and

smart cities. For example, a smart home application can provide air conditioning and temperature measurement to the sensor requesting such information. For applications with a different purpose for customers with widely varying needs, it is challenging to accomplish authentication and secure access permissions [6], [43]. Data security is a major factor in the application layer. In order to guarantee access privileges and data usage, data encryption and distortion technologies are mostly used in protecting the privacy of data. Furthermore, recovery mechanisms and data backup must be implemented appropriately and techniques to protect data privacy should be properly selected, such as DNS, TLS and SSL [47], [48].

2.5 Network Simulations and Operating Systems

Designing, developing and evaluating new IoT products and protocols prior to being deployed in a specific environment demands testing and assessment using various tools. For instance, prototyping may not be widespread when using a large number of device nodes during the original exploratory design and evaluation phases, and this is because of the economic and operational restrictions. This is especially the case when the reliability and utility of the protocol under consideration have not yet been demonstrated. In addition, creating reliable, repeatable experiences that include real hardware can be complicated and often requires specific expertise and field knowledge. Thus, some simulators and OSs offer a better choice for setting up reliable scenarios and experiments. However, as mentioned earlier, testing, analysing and evaluating a real testbed are costly and challenging. Therefore, simulators offer a high level of accuracy for scenarios involving heterogeneous elements, energy efficiency, scalability and low cost [49]. This section examines several simulators and OSs that are used in IoT research.

2.5.1 Simulators

The IoT paradigm predates network protocol research and many previously available tools for wireless sensor networks (WSNs) or basic networking research have been modified to include IoT-specific elements. A comprehensive survey conducted in [50] highlighted approximately 21 open-source simulators that have been used in IoT/IIoT research. This section mainly focuses on the widespread use of the following open-source simulators by academic and industrial user groups: Cooja, OMNeT++, NS-3, QualNet3, and SCADASim.

Cooja is a simulator configured for emulating a network with sensor nodes and supports several of the sensor motes included in Wismote, Sky, Z1, etc. Cooja enables synchronised simulations in three levels: machine code instruction, the OS, and the application level. Therefore, the majority of protocols and standards are implemented through Contiki [51].

OMNeT++ is a network simulator for modelling communication networks, distributed and other parallel systems, or multiprocessors. OMNeT++ is open source, discrete event, and C++ based and can be used by education, academic and commercial institutions for simulating distributed systems and computer networks [52].

NS-3 is a new open-source simulator that was developed to replace the old NS-2. It is a powerful tool for supporting network optimisation and modelling, including the Internet-stack implementation module TCP/UDP/IPv4/IPv6 stack, and NetDevice operations, such as IEEE802.15.4, CSMA and WiFi. It also supports the 6LoWPAN stack [53].

QualNet is a commercial simulation tool introduced by Scalable Network Technologies (SNT) and the software is designed for large, distributed applications and heterogeneous networks. It has an additional extension sensor network library that can be used for IoT-specific simulation and supports IEEE 802.15.4 networks [49], [54].

SCADASim is open-source software that aims to provide a framework for rapidly developing flexible SCADA system simulations. SCADASim is a discrete-event simulation engine built on top of OMNET++. The SCADASim architecture comprises three primary components: SSScheduler, a real-time scheduler; SSGate, a communication port that implements protocols for communication with the external environment; and SSProxy, a simulation object that simulates an external component within the simulation environment. SCADASim also offers several tools for developing network typologies (the NED language editor) and a plug-in extension architecture. Plug-ins allow for customisation of the actions of the default simulation engine. For instance, a simulation can be enhanced with a different message scheduler, thus altering the default behaviour of the message scheduling [55].

2.5.2 Operating Systems

The IoT OS is well suited to IoT devices with low-to-moderate resource constraints. Small IoT devices are highly specialised and place a strict restriction on IoT OSs to be extremely hardware-specific with minimal capabilities. Medium-sized IoT devices allow the inclusion of a full Internet Protocol (IP) suite and various applications to run on top of the network stack. The devices also have additional capabilities and can act as servers, hosts or Internet routers. Many aspects of OSs were designed for the IoT environment. This section highlights some of these OSs: TinyOS, Contiki, RIOT, Raspbian, Mbed, and Zephyr. However, based on the literature analysis, the most widely used OSs are Contiki and TinyOS.

TinyOS, designed for WSNs, was the most popular OS for public research for many years. It has not, however, been used recently by researchers because of inactive development. TinyOS is an open-source OS designed for wireless devices that are embedded and have low power. It has a programming language based on C, called nesC (network embedded systems C). It also provides several hardware platforms and supports the 6LoWPAN protocol through the IPv6 stack [56].

Contiki OS is open source and the most common OS used for programming IoT sensor nodes. Furthermore, it is used for multi-tasking and making a network more efficient in terms of memory, wireless sensor systems and networks [49]. It also supports IP connectivity for both IPv4 and IPv6. Contiki was developed by a group led by Adam Dunkels at the Swedish Institute of Computer Science [57]. Contiki supports many mechanisms and protocols, such as RPL routing, the CoAP and MQTT application protocols, and 6LoWPAN header compression. Contiki also provides a power-profiling mechanism, called Powertrace, which keeps track of and estimates the energy consumed by each sensor node [58].

RIOT OS is a free, open-source OS created by a grassroots group of businesses, academics and enthusiasts and is distributed worldwide. The RIOT OS is compatible with ARM Cortex-M3 and Cortex-M4 processors and ARM7, AVR Atmega and MSP430 devices. This OS was written in C and C++ and is distributed under the LGPL v2.1 licence. Gcc, valgrind and gdb are specific development tools that can be part of an software development kit (SDK) included with the RIOT OS. The SDK architecture is

also compatible with C and C++ application programming. RIOT is compatible with the vast majority of low-power IoT devices and micro-controller architectures. The RIOT OS is compatible with most common communication and networking protocols, including IPv6, TCP, UDP, CoAP, 6LoWPAN and RPL [59].

Raspbian is the OS most commonly used by researchers in IoT and IIoT projects. It is a low-power, highly extendable, cheap and small board developed by the Raspberry Foundation in the UK to be used for teaching and experimental projects. It runs over a variety of Linux distributions and its primary Raspbian Debian-based OS. There are different varieties of Raspberry Pi available, such as the Pi 3 Model B, Pi 2 Model B, Pi 1 Model A+, B+ and Pi Zero. The latest Pi 4 Model B launched in June 2019; it has 1–4 GB RAM, uses a 1.5 GHz Quad-Core ARM Cortex-A72 CPU, integrated Bluetooth 5.0 and 802.11n wireless LAN [60].

Mbed is a 32-bit ARM cortex-m micro-controller OS developed by ARM in collaboration with its technological partners [39]. Mbed is an open-source OS that can be used on a wide range of products, from small Internet-connected devices to smart cities and smart applications. The OS is written in C and C++ and is governed by Apache Licence 2.0. When compared with the offerings of Microsoft or Google, Device and Component Support with real-time software execution and ease of use by any sensor are some of the key features of the Mbed OS, as are end-to-end security and an extensive collection of drivers and support libraries [59].

Zephyr is a real-time OS designed for IoT applications that are backed by the Linux Foundation. It is common among IoT experts due to the ease with which it integrates with various IoT architectures. Zephyr's interconnectivity technology is one of its most distinguishing features. It is a library-based OS with stable memory security and a highly configurable, customised open-source IoT OS that supports device trees. The OS requires 8 KB RAM and 512 KB ROM. It includes a device development kit with comprehensive documentation, a full set of kernel services, non-volatile storage, and virtual file support, among other features [61].

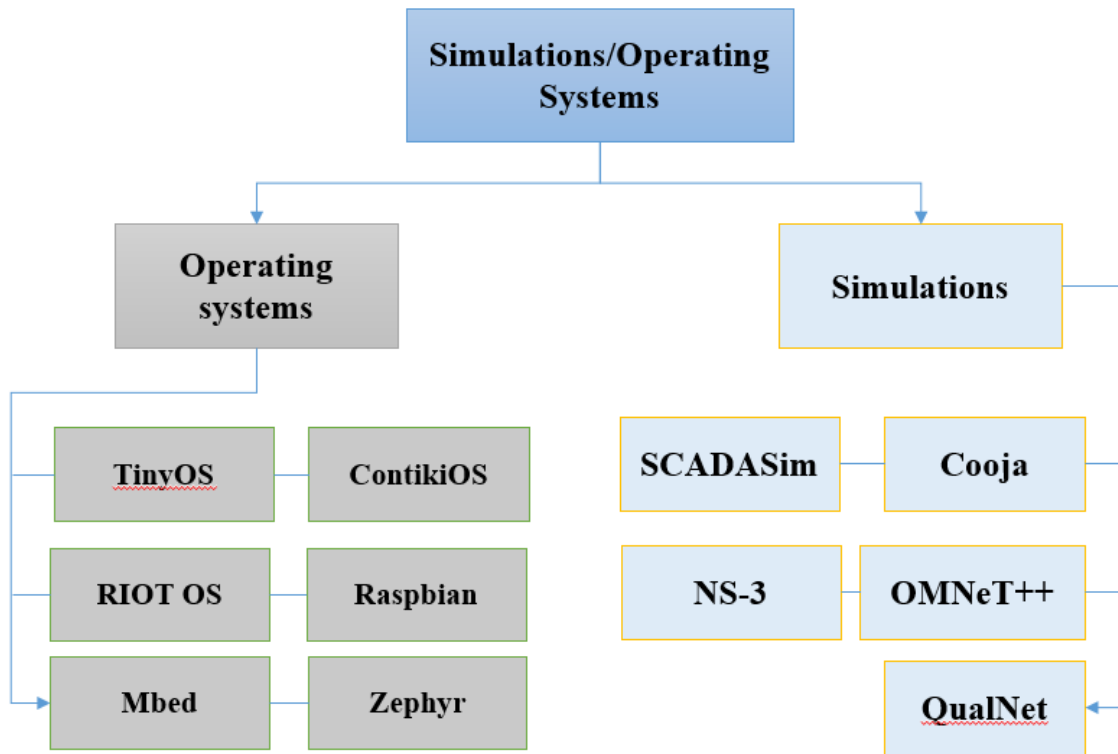


Figure 2-6: Simulations and operating systems for IoT and WSN networks

2.6 Security Solutions for IoT and Industrial Systems

2.6.1 Communication Security Countermeasures

Securing end-to-end communication in an IoT network is crucial. Compressed IP security is proposed in [37] to enable end-to-end security within communications between the traditional Internet and a sensor network. The security approach involves the ESP and AH protocols and shows that a compressed IPsec can check the integrity of messages by encrypting and authenticating them using standardised IPv6 mechanisms. The work is extended in [62], detailing the ESP protocol for IPsec/6LoWPAN and then comparing the solutions by employing link layer security for IEEE802.15.4. An evaluation of the performance of the IPsec/6LoWPAN security and solution testbed is built into IEEE 802.15.4, which re-utilises the crypto device through the actual IEEE 802.15.4 transceivers for the IPsec/6LoWPAN. Although the IPsec is more effective than link layer security for enhancing security and performs better in terms of response time, the IPsec consumes more energy than link layer security.

The authors of [63] propose a method of security analysis, access control improvement and authentication for the IoT and offer a practical protocol for the IoT by breaking down current access control and authentication approaches. The authentication protocol is an efficient, simple and secure key function based on Elliptic Curve Cryptography (ECC) and is utilised to enhance device authentication. Role-based access control (RBAC) is proposed for the access control rule on IoT network applications. However, the practical implementation of the proposed security evaluation was not performed, and the communication overhead for the IoT sensor nodes was high.

2.6.2 1.6.2 Network Routing Security Countermeasures

Although IoT networks enable the encryption and authentication of messages, they are still vulnerable to a range of threats, such as Sybil, blackhole, sinkhole, fragmentation, selective-forwarding and MITM attacks. It is crucial to design a system that can detect these attacks. Many studies have implemented IDSs against attacks in a WSN; however, only some research has evaluated the performance of IDSs against an attack in the IoT. IDSs could, for example, be placed at every node of a low-power and lossy network (LLN). The IDSs deployed in each physical object must be optimised due to the nodes being resource-constrained. The authors of [64] performed lightweight intrusion detection against malicious attacks (DoS) and evaluated the impact of the IDS on the energy consumption of the sensor nodes during an attack. They found that more energy was consumed while an attack was being detected. However, the authors do not show the detection accuracy of their IDS, and they only use one metric and one type of attack.

In response to the limitations of resource-constrained memory size and computing power, the authors of [65] propose a lightweight IDS based on malicious pattern detection. They evaluated detection in terms of energy consumption and communication duration based on two schemes (early detection and auxiliary shifting) to decrease the number of matches required for discovering attacks. Although their approach enhanced the performance and accuracy more quickly than the Wu–Manber algorithm, the overhead for auxiliary shifting was high due to an increase in the pattern. Another distributed IDS architecture was proposed in [66], called INTI (Intrusion detection for sinkhole attacks over 6LoWPAN for Internet of Things), to address sinkhole attacks. The solution combined “watchdog” and “reputation and trust” approaches for discovering and reducing attacks.

INTI contains four roles: first, nodes are categorised as leaders, whereby the leader nodes receive each node's information state; then, each node monitors the routing traffic; the third role is attacker detection; and the last nodes are for isolating the attacker. The authors do not analyse the impact of their solution on the resource-constrained nodes in terms of energy consumption. Although they achieved a lower false positive rate than in [67], it is still high (29%).

A hybrid IDS for use with the IoT, called SVELTE, is proposed in [67] and targets selective-forwarding and sinkhole attacks. SVELTE consists of the main IDS modules, a 6LoWPAN mapper and a mini firewall at the constrained devices and border router. Based on the requests sent by the border router to a sensor node, the sensor node responds with information such as ID, parent ID and rank. SVELTE compares the collected information in order to identify malicious traffic and notifies the border router. The researchers claim that their proposed IDS could be applied to anomaly detection techniques. The ideal IDS result is to get low false positive, low false negative and high true positive rates. SVELTE recorded a 90% true positive rate; however, the true positive rate decreased with an increase in nodes, and there was a high false alarm rate (38%) during the detection of malicious traffic. In a similar study to that involving SVELTE, the authors in [68] propose a network-based IDS that targets wormhole attacks. Their IDS measures the collected, received signal strength to identify suspicious nodes. The results show that the true positive rate was 90% when the network size was small (eight nodes); however, there was a significant decrease in the rate with an increase in network size. Furthermore, the authors evaluated the impact of their IDS using only one metric and one kind of attack.

A specification-based IDS was proposed in [69] for the IoT. The proposed IDS works against repair and rank attacks. The study also designed and implemented a distributed architecture to monitor all the nodes over an RPL finite-state machine. Although the researchers claim that their proposed system could successfully detect routing threats with a sensible overhead, they did not implement the IDS or specify the form of communication among the monitoring nodes, due to all monitoring nodes needing to store the ranks, preferred neighbouring nodes, and IDs. The finite-state machine was implemented on all monitoring nodes with an anomaly state to identify attacks if the

monitoring nodes could not observe whether the node was not an attacker. Further to this, some of the same authors [70] worked with others to implement a specification-based IDS that targets rank, local repair, sinkhole, DoS and neighbour attacks. They employed a group monitoring structure, whereby the network is split into multiple clusters. The IDS is located at the head of each cluster to monitor the cluster based on the configured role to reduce computation and storage. The head cluster sends requests periodically to all cluster members; each member responds with parent information, neighbour lists, and rank information. The results show that IDS can be detected attacks more effectively with only a small amount of overhead.

The authors of [71] propose an efficient, secure route optimisation protocol for the Proxy Mobile IPv6 (PMIPv6). The protocol is intended to enhance the existing routing protocol (PMIPv6) when implementing security, specifically when using mutual authentication, complete forward secrecy, key exchange and privacy to support the protocol. Their approach provided secure transmission and reduced packet loss, latency and delay. The authors of [72] propose a novel trust mechanism which was implemented in a testbed experiment based on the SecTrust-RPL routing protocol. The proposed SecTrust-RPL protocol is intended to enable the detection of Sybil and rank attacks. However, the authors do not evaluate the impact of their approach in terms of performance, energy consumption or performance overhead. A recent study [73] presents a wormhole attack detection mechanism for the IoT RPL routing protocol. Their IDS is placed at the border router as well as at the host sensor nodes. They evaluated the impact of their IDS on the success rate for detecting wormhole attacks. Contiki OS simulation with three random typologies, consisting of 8, 16 and 24 sensor nodes, was run among different run times. The results show that the true positive rate was 96% with 8 sensor nodes but decreased with an increase in the number of sensor nodes to around 87%. The authors do not evaluate the impact of their IDS in terms of performance. They consider only the detection accuracy and aimed to detect only one specific type of attack, rather than examining a range of attacks.

A recent study [74] proposed a hybrid IDS for targeting sinkhole and clone ID attacks and evaluated their ID impact in terms of performance and detection. The study was an extension of SVELTE and focused on improving the detection rate in relation to false

positive alarms. The authors claim that the detection rate was 100%. However, the figures show that a detection rate decrease followed an increase in the number of sensor nodes. Furthermore, the authors identified that the energy and power consumption were higher than for SVELTE. Currently, a study in [75] proposes an IDS against a blackhole attack on the RPL routing protocol. The authors aimed to improve detection efficiency by analysing only the suspected node, rather than exploring the traffic of all the nodes as is the case with watchdog approaches. However, the study limited its evaluation to addressing only the accuracy of the IDS.

Unlike intrusion detection in the IoT, the authors of [76] implemented an intrusion detection framework and architecture for an RPL-based IIoT that used genetic programming. Their detection approach targets two kinds of attack, version number and hello flood attacks, and they simulated a network using the Contiki OS. Their detection system shows that for the flood attack, 96.08% and 99.83% are the worst and the most accurate, respectively. The values were obtained by collecting data in the 500 ms and 5000 consecutive ms periods. The attack version number showed a performance closer to it in flood attack in different periods. The best and worst accuracy obtained with time intervals was 4000 ms (99.42%) and 3000 ms (97.97%), respectively. Although they investigated how genetic programming could achieve high accuracy and low false positive rates in detecting intrusions, they restricted their investigation to detection accuracy only, rather than examining the impact of their intrusion mechanism on performance, such as energy and memory consumption. Table 2.2 presents the classification of IDSs for the network layer of the IoT.

Another study [77] implemented a detection mechanism for an industrial control system, called HAMIDS (a hierarchical monitoring intrusion detection system). They employed a behaviour detection approach to a SCADA water treatment system application. The most important module is the Bro IDS sensor, which works on receiving data from the store in logs containing various network protocols, such as UDP, TCP, the Address Resolution Protocol (ARP) and EtherNet/IP. These logs are collected in the HAMIDS framework and results of the IDS upon detecting threats from the raw records. The results show that HAMIDS was able to detect outsider attacks with zero false positives. However, HAMIDS was still vulnerable to insider attacks, such as DoS and MITM

attacks are most common threats to SCADA in realistic Internet Connection Sharing (ICS) settings, such as rebooting Ethernet and CPU crashes. In IIoT ecosystems, communication models among the devices are crucial for employing security, such as anomaly detection and guarding against possible cyberattacks.

A study in [78] proposes deep packet inspection based on a discrete-time Markova chain (DTMC) model for four types of industrial network datasets: an energy management system, large-scale water treatment, a small-scale water testbed, and an electrical substation. The study evaluates the impact of the DTMC model on the industrial network's datasets among various industrial communication protocols, such as DNP3, EtherNet/IP and Modbus/TCP. While testing the data, the potential of the DTMC model was classified as unknown state, unknown transition or anomalous probability. The results show that the Energy System Monitor (ESM) dataset did not detect any unknown probabilities for two channels out of six; it observed a high number of transitions and states for one of the channels: 248 and 479, respectively. In contrast, the power generation testbed (PGT) dataset did not observe any unknown probabilities or states (for single-state transitions). Afterwards, the maximum numbers of unknown probabilities (5) and unknown transitions (15) were observed. The water plant testbed (WPT) dataset had the highest number of unknown probabilities (111) and 17.3 on average were of unknown transition. It is clear that more investigation is needed due to the high number of unknown transactions and states observed in all three datasets.

It is clear from Table 2.2 that most studies concentrate mainly on insider attacks, whereas outsider attacks from the Internet must also be studied and examined. As can be seen, attacks above the network layer have not been thoroughly studied. This clearly demonstrates that the transport and application layers of the IoT will be vulnerable to attacks, and the IoT is mentioned in the news because of DDoS attacks. For example, in 2016, the DNS provider that supported Internet services and platforms, including PayPal, VISA and Twitter, was attacked by a DDoS through the vulnerabilities of IoT devices, such as IP cameras, printers and residential gateways that were infected by a malware named Mirai [79]. Another issue regarding IoT security research is that the majority of authors keep their implementation source codes private. It would benefit the field of IoT security research if researchers shared their implementation with the public.

Table 2-2: Classification of intrusion detection systems for the network layer of the IoT

Ref	Detection Technique	Security Threats	Validation	Advantages	Disadvantages
[72]	Lightweight IDS Does not evaluate the accuracy.	DoS attack	Simulation (Qualnet)	Evaluates the energy consumption.	Use only one metric and one attack.
[73]	Centralised	Malicious attacks	RaspberryPi	High accuracy rate. Much faster than Wu-Manber.	High overhead.
[74]	Distributed - High accuracy rate.	Snikhole attack	-	High false-positive.	Does not address the impact of IDS on energy consumption.
[75]	Hybrid	Snikhole and Selective forwarding attacks	Cooja-ContikiOS	High accuracy rate and Low overhead.	High false alarm rate 38%
[76]	Centralised	Wormhole attack	Cooja-ContikiOS	High true-positive when the network size small. the true-positive significantly decreases by the increase of network size.	Evaluates only one metric and one attack.
[77]	Distributed	Repair and rank attack	Not available	Effectively detects rank and repair attacks.	Does not validate their approach on a simulation.
[78]	Centralised	Rank local repair, and sinkhole attacks	Cooja-ContikiOS	High detection accuracy Low false-positive.	Slight overhead.
[79]	-	Authentication key exchange and privacy	AVISSPA	Reduces latency and packet delay.	

[80]	Distributed	Sybil and rank attack	Cooja-ContikiOS	Capable of detecting sybil and rank attacks.	Does not evaluate the impact of their approach in terms of performance.
[81]	Hybrid	Wormhole attack	Cooja-ContikiOS	High detection rate sufficient for constrained resources.	Detects only one attack.
[82]	Hybrid	Sinkhole-CloneID attacks	Cooja-ContikiOS	High detection rate.	High energy and power consumption
[83]	Anomaly	Blackhole attack	Cooja-ContikiOS	Improves detection rate.	Packet delivery ratio decreases when the network size increases
[84]	centralised	Version number and hello flood attacks	Cooja-ContikiOS	High accuracy detection. Low false-positive alarms	Evaluates only one metric which detection accuracy. Does not evaluate the impact of the IDs on performance.
[85]	Network behaviour anomaly	Outsider attacks ARP posing and MITM attacks	NS3 and Python script	Detects outsider attacks with zero false-positive.	Not able to detect insider attacks
[86]	Anomaly	Malicious traffic	Discrete Markov Chain models (DTMC)	Evaluates the DTMC on different industrial datasets.	High number of unknown transactions and states.

2.6.2.1 Intrusion detection based on machine learning

Several intrusion detection researchers have employed machine learning (ML) models to protect IoT networks from attacks and threats [80]. The research work in [81] detects automated IIoT attacks using an ML algorithm. The algorithm performs functional shape analysis for extracting and representing features, which is intended to enhance the performance of the ML algorithm during attack detection. By modelling the waveforms using functional representation of the feature extraction, the different types of IIoT attacks are detected through a novel early detection strategy. The work in [82] presents a novel supervised ML-based IDS strategy to identify security attacks in RPL-based cyber security systems. However, conventional ML-based solutions mainly focus on detecting a minimum number of attacks and lack the ability to cover most of the attacks that occur in the RPL-IoT scenario. They are also evaluated with simple IoT datasets, which are ineffective for reflecting the heterogeneity of the IoT. The work in [83] utilises an ML model to identify combined attacks in an IoT environment. It detects attacks against two main RPL objective functions: the Minimum Rank with Hysteresis Objective Function (MRHOF) and Objective Function Zero (OF0). To detect attacks efficiently, the model constructs a novel IoT dataset with optimal IoT features and attacks. The work in [84] presents a DETector of rOutiNg Attacks in RPL (DETONAR) model that combines the advantages of anomaly- and signature-based IDS models to detect malicious behaviour in network traffic. To handle the issue of lacking a dataset in RPL-IoT, DETONAR introduces a Routing Attacks DATaset for RPL (RADAR), in which more than 14 types of RPL attacks are covered. However, most of the above-mentioned solutions exploit a centralised way of detecting attacks, whereas the RPL-IoT is mostly distributed among many regions. Hence, it is crucial to design a novel ML-based IDS solution for handling IoT heterogeneity effectively.

Federated learning (FL) is a distributed learning model in which the machine learns on heterogeneous IoT data with different IoT services enabled, as investigated in [85]. Several network security researchers [86], [87] have adopted FL for intrusion detection. To provide security for IoT networks, the work in [86] presents a Probabilistic Hybrid Ensemble Classification (PHEC) model by adopting FL for local training and model parameter aggregation. Moreover, it develops a noise-tolerant PHEC model for

centralised and federated settings and the classifiers are designed based on weighted convex surrogate loss functions for intrusion detection. The network intrusion detection model in [87] adopts the FL model with joint training by using deep learning to improve intrusion detection accuracy and protect the privacy of network traffic.

Furthermore, several IoT research works have employed an FL model for preserving privacy [88], [89], [90]. The work in [88] proposes a Supportive Twin Delayed DDPG (S-TD3) algorithm to authenticate and preserve the privacy of an IIoT network with the assistance of a federated transfer learning (FTL) model. In addition to ensuring the privacy and security of FTL blockchains, transfer-based blockchains with an authentication mechanism enrich the security and preservation standards of IIoT applications. An IoT anomaly detection model [89] adopts FL with the potential advantage of privacy preservation and the design of a blockchain-based asynchronous and decentralised FL model. Consequently, it avoids single-point failure and ensures data integrity with improved efficiency during anomaly detection. Moreover, it applies generative adversarial networks to design a private FL model to optimise data utility during the training process and to preserve privacy in a decentralised IoT anomaly detection system. In other work, an IoT privacy-preserving framework [90] detects malware without sharing sensitive data through the training and evaluation of both supervised and unsupervised models by the FL model. This supports the privacy preservation of IoT devices, such as those using 5G, Wifi, or B5G networks, during computation offloading and processing.

The work in [91] demonstrates that FL-based IDS models are susceptible to backdoor attacks over the IoT. To discuss and rectify this problem, the research presents a novel data poisoning attack in which a malicious intruder implants a backdoor of aggregated information for the detection model that leads to incorrect classification of malicious and benign information. Thus, the malicious intruder can gradually poison the detection strategy simply by exploiting the compromised IoT devices and injecting an amount of malicious traffic. Moreover, this type of situation has motivated researchers to innovate novel FL IDS models for the IoT to optimise performance. The work in [92] introduces an FL model as a defence against gradient-based reconstruction attacks over resource-limited IoT environments. The model compares analyses of the loss of training and

designs an adaptive communication mechanism by adjusting the frequency and parameter compression, resulting in a high level of security. The framework in [93] advocates a personalised FL model to investigate the heterogeneous issues associated with the IoT environment. It also accomplishes personalisation and maximises intelligent IoT device performance by integrating four different FL methods: FTL, federated meta-learning, federated multi-task learning, and federated distillation. A federated ML-based IDS model in [94] trains IoT device models based on the local data and maintains the security of sensitive IoT information. Further, the FL-based server utilises aggregation with local models to maximise the detection efficiency of the sensors. The work in [95] proposes a Federated Learning-based Attention Gated Recurrent Unit (FedAGRU) model to ensure security over edge wireless networks. The FedAGRU is able to update the universal learning without sharing raw information from the sensors to the server and optimise the learning model. Further, it ignores unimportant parameter transmissions to the server with the help of an attention mechanism and greatly enhances the security performance with minimum communication overhead.

The work in [96] presents an FL-assisted intelligent IDS model that exploits the FedACNN algorithm to detect intrusions in edge-enabled IIoT environments. It solves the issues associated with centralised ML models by transferring the model parameters to the global server with a high level of security, instead of sending raw data. It improves IDS detection accuracy considerably by achieving an accelerated global model using the FL concept. Other work [97] proposes an ensemble multi-view FL-based intrusion detection model, called MV-FLID. It views the various IoT data traces and forms an effective decentralised learning process with the help of FL. Further, the MV-FLID identifies and classifies intrusions and provides strong security against attacks. It also improves learning efficiency by viewing the data of diverse attack classes in multiple aspects. The work in [98] presents a three-layer FL architecture for an IIoT environment. In this model, the gateways are locally trained using a deep neural network (DNN) model and optimise the learning process by globally federating the locally trained models.

Today, transfer learning (TL) models have gained significant attention for their ability to detect intrusions into RPL-based resource-constrained IoT networks by transferring the knowledge from the source domain to the target domain. One study [99] investigates the

role of TL in transferring the knowledge gained from intrusion detection to new devices in order to detect new attacks. The work in [100] uses the advantages of both federated and transfer learning to improve the generalisation training efficiency of the detection system in an IIoT environment. With the integration of edge learning in the IIoT network for resolving the issue of resource-constrained IIoT devices, securing the data transmission with a blockchain becomes crucial, along with a TL model for the FL environment to increase the versatility and efficiency of the training model. A collaborative learning strategy in [101] improves the security performance of the target network domain by optimising the unlabelled data of the target model with the help of a very large volume of labelled source model data through transfer learning. To handle the IoT device training issues with a large dataset, deep transfer models are used in [102] for an IIoT environment.

The work in [103] presents a deep TL-based traffic classification model for the 5G IoT environment. The model fine-tunes the classification model with a few labelled data by extracting abundant labels from multiple domains. To denoise the raw Electrocardiography (ECG) signals of patients in IoT healthcare, the work in [104] utilises an FL-assisted convolutional neural network (CNN)-based autoencoder. Similarly, the work in [105] proposes an FTL-based IDS model for the Internet of medical things environment. The model performs a unique reconstruction of raw input data and provides those unique data to the federation process, resulting in improved healthcare IoT performance. To handle the network resource scarcity issue, the model runs the classification algorithm at the edge devices. However, most of the FL-assisted IDS models can solve the intrusion detection issues of heterogeneous IoT environments, whereas there are no specified FTL-based IDS models for heterogeneous RPL-IoT environments.

2.6.3 1.6.3 Transport and Application Security Countermeasures

Several works have suggested improving the performance and security of DTLS in the CoAP. For example, the authors of [106] applied Rivest–Shamir–Adleman (RSA) cryptography to sensing devices using a particular Trusted Platform Module (TPM). They evaluate their system in terms of latency, energy consumption and memory based on a DTLS cipher suite: TLS-RSA-with-AES-128-CBC-SHA. Further to this, the same

proposal is described [107] and further evaluated using an experiment in WSNs. Another study [108] proposes an authentication security scheme for the transport layer using the ECC algorithm. They present DTLS implementation in the context of system architecture to achieve a low overhead and high interoperability. However, the results show that the handshake mechanism consumes more energy because of the complex computation of the ECC algorithm. In contrast, the authors of [109] propose integrating DTLS with the CoAP for the IoT, a model they call Lithe. The authors propose a novel DTLS header compression scheme that aims to significantly reduce the header overhead of DTLS, leveraging the 6LoWPAN standard. The authors evaluate their scheme based on the cipher suite-based TLS-PSK-WITH-AES-128-CCM to reduce energy consumption and the round-trip time using the Contiki OS. The results show that Lithe is more efficient in many aspects than a basic plain CoAP/DTLS. However, the proposed scheme lacks security against DoS attacks due to the DTLS cookie exchange scheme, which is insufficient to handle this type of attack, as the authors of [39] reported employing DTLS as a security solution that could only serve the security issues between the UDP when running on different endpoints. It still cannot protect the IP header when packets are transmitted through the access network layer of IoT objects through the Internet. Nevertheless, they employed the DTLS protocol to reduce the number of transition bytes, rather than secure transmission through the IoT devices.

In a similar study to the work on Lithe, the authors of [110] present lightweight DTLS for the IoT called E-Lithe. They customised a DTLS packet to reduce energy consumption and communication duration to lower the DTLS computation overhead. The results show that energy and response time performance were reduced slightly more than in [109]. It is essential to provide integrity and authentication security for IoT applications. The authors of [111] propose identity-based lightweight encryption and a Diffie–Hellman encryption scheme for a smart home without using a digital certificate, in which case the public keys are only identity strings. This scheme is divided through the encryption process into data encryption and key encryption to gain efficiency and reduce communication costs. The results show that the data ciphertext is transmitted several times without assigning the ciphertext key. Likewise, the performance analysis shows that the combination of identity-based encryption (IBE) and Diffie–Hellman encryption reduced the communication overhead by nearly one third. It performed better than a

symmetric IBE scheme in speeding up encryption operations. In comparison, the authors in [112] propose “integrity security” for smart home applications based on the CoAP. They aim to add the optimal hash function to the CoAP. They implemented and evaluated their study using Contiki. The result shows that a Secure Hash Algorithm (SHA224) was the optimal algorithm for use with the CoAP when optimising power consumption and time, and the hash function algorithms do not intervene in the encryption/authentication.

Data confidentiality is crucial in smart home systems, due to the many data transactions that are necessary. One study [113] proposes a new architecture to secure the CoAP: a smart home application based on cryptography algorithms that is based on AES and SHA224, instead of the traditional security for the CoAP (DTLS). The study provides results relating to data confidentiality and integrity when considering IoT device restrictions in terms of energy and communication duration. However, the study does not compare the results with the traditional implementation of DTLS for the CoAP. It appears that much research into securing the CoAP using DTLS has been proposed, designed and implemented. Another study [114] proposes secure CoAP communication between the IoT devices in the 6LoWPAN and smartphones. The researchers evaluate the performance of DTLS using ECC and found that the use of complex computation when running the ECC algorithm led to higher energy consumption.

In conclusion, considering the shortcomings, the range of threats, cryptography-based security solutions and published research, there are still many problems to investigate in order to achieve a secure system, as highlighted in Table 2.3. There is a need to propose lightweight security algorithms by optimising algorithms such as DTLS and AES to support resource-constrained devices. Another issue mentioned above regarding IoT security research is that most authors keep their implementation source codes private. It would benefit the field of IoT security research if researchers shared their implementations with the public.

Table 2-3: Classification of IoT security solutions for transport and applications

Ref	Security Objective	Proposed Solution	Simulation/ Testbed	Advantages	Disadvantages
[104,105]	Security analysis for MQTT-TLS and CoAP with DTLS	Authentication key exchange and security privacy		Compares the security mechanisms for MQTT and CoAP protocols.	
[106]	Provides message integrity and confidentiality with low overhead and latency	DTLS based on x.509 certificate containing RSA keys	TinyOS	Provides message integrity and confidentiality with energy efficiency	
[107]	Provides data confidentiality, integrity with low overhead and high interoperability	integrated DTLS handshake and RSA key, and DTLS with ECC	Real IoT system		
[108]	Provides authentication security and secure communication	DTLS with ECC public key authentication	TinyOS		Consumes higher energy due to the complexity computation of ECC algorithm
[109]	Reduces the energy consumption for the integration of CoAP with DTLS	Using tinyOS DTLS based on pre-shared keys	Cooja-ContikiOS	significantly reduces the overhead	Does not compromise end-to-end security
[110]	Lightweight security against DoS attack	Uses Trusted Third Party between the constrained node and CoAP server	Cooja-ContikOS	Reduce the energy and respond time	

[111]	Lightweight encryption for smart-home based on public key	Cryptography solution based on stateful Diffie Hellman key and identity-based encryption scheme		Reduce the computational cost of encryption implementation	
[112]	Message integrity security for smart-home application	integrated hash functions with CoAP protocol	Cooja-ContikiOS	Provides integrity protection	Does not provide data confidentiality. In the terminology of smart-home system, the data transactions being leaked
[113]	Data confidentiality, integrity, and authentication for smart-home application	Cryptography solution based on integrated CoAP-AES and CoAP-ShA2	Cooja-ContikiOS	Provides data confidentiality, integrity, and authentication protection	Does not evaluate their solution performance with DTLS
[114]	end-to-end security for mobile devices	Integrated DTLS with ECC authentication	TniyOS		

1.6.3.1 Optimised Implementation of DTLS for the CoAP

Device authentication is a vital mechanism in verifying the entity identity of a node in a distributed sensor network and its use eliminates the impact of malicious behaviour. The CoAP, the IoT application layer protocol, is used with DTLS for reliable and secure negotiation of a session, authentication, and data communication. Although DTLS provides a complete security model for the CoAP, reducing energy consumption without compromising security is essential. Advanced metering infrastructure (AMI) data are sensitive and, if the data are stolen, this can result in significant economic loss to both users and power providers. Thus, the data transmitted over distributed sensors must be encrypted before being forwarded. The existing works mostly apply either symmetric or asymmetric encryption schemes. Symmetric encryption requires a secret key to be shared in advance, whereas asymmetric encryption does not need to set a shared key in advance. However, asymmetric encryption leads to more computational cost than a symmetric encryption technique. Therefore, an effective method is to generate a shared key and encrypt the data dynamically. Smart grid applications need to provide an effective authentication and security scheme in a lightweight pattern [115], [116]. Since smart sensors are low-energy devices, highly complex security schemes that require more energy do not apply to smart sensors.

Several works have suggested improving the performance and security of DTLS in the CoAP. In [117], an optimised implementation of DTLS for the CoAP is suggested. The DTLS protocol is integrated with CoAP and ECC optimisations, which minimises ROM utilisation. DTLS optimisation is performed with connection-oriented communication and fragmentation. The CoAP message layer provides connection-oriented communication, whereas fragmentation is attained using the block-wise transfer feature provided by the CoAP. Those mechanisms make DTLS lightweight as well as strong in its security. The optimisation applies a stateless cookie technique to reduce the resource consumption of smart sensors. This technique forces the smart sensors to re-send Hello messages with a cookie attached. According to the cookie validation, the server node can verify and decide whether to continue the handshake process. Although DTLS is used to secure and support the light weight, the CoAP lacks some security requirements. DTLS fails in satisfying important features of the CoAP, such as multicast communication. In

addition, the DTLS handshake protocol can cause exhaustion attacks. These cause the nodes to lose their roles in the network and this affects the entire IoT communication. Moreover, the security features used in DTLS cannot avoid replay attacks in all scenarios; this makes networks vulnerable to replay attacks and the loss of resources among smart sensors. Moreover, the DTLS handshake protocol does not always ensure end-to-end authentication [118].

In other work, an optimised version of DTLS attempts to reduce the complexity and cost of communication by improving cookie exchange strength during the handshake process in the CoAP [119]. During connection establishment between the IoT sensor and the server, several encryption concepts are integrated into the CoAP communication, and this provides an additional security layer. This introduces the cookie exchange technique for reducing the impact of DoS attacks. The sensor devices have to show their capabilities to the server before allocating resources to the smart sensor. This assists in reducing the energy consumption of DoS attacks in the IoT. The mechanism suggested in [120] reduces CoAP communication latency by applying forward error correction. This approach assists in compromising the packet loss and communication delay. Consequently, it shows significant improvement in the throughput of IoT communication. Moreover, several attempts have been made to improve the efficiency and security of DTLS [121], [122]. An enhanced DTLS in [121] separates the DTLS handshake and encryption phases to mitigate the delay, packet loss, and overhead of DTLS during the handshake process. Every smart sensor depends on the secure service manager to execute encryption and decryption of the data.

The primary issue associated with DTLS is the possibility of sending several Hello messages to the server. A smart gateway-based authentication and authorisation method is suggested in [122] to provide security to health data communication. However, introducing a smart gateway is not always preferable in all IoT applications due to the overhead induced in the IoT environment. In other research, a flexible ECC-based CoAP attempts to establish a secure communication session between sensors and a server [123]. This solves the issues related to key management and an insecure communication channel. Although the ECC algorithm exploits small-size cryptography keys, it increases the size of the encrypted message by the use of such keys. Moreover, the algorithm is

highly complex to implement in a resource-constrained IoT environment. The Diffie–Hellman key exchange and certificate verification implemented in DTLS make it highly complex. A simplified DTLS handshake protocol (DTLS_{Shps}) is proposed to solve this issue. It takes support from software-defined networking and mitigates the computational overhead of the smart sensors significantly. The controller generates symmetric encryption and distributes keys to the smart sensors that are seeking to communicate. A powerful controller is used in the process of certificate verification. The controller replaces the DTLS server for exchanging the cookies with the DTLS sensor device. Furthermore, it exploits BAN logic and the Scyther tool to make the DTLS more secure. In [124], the grouping of handshake messages is investigated instead of considering a single DTLS handshake message between sensors and a server. This technique drastically reduces the computational burden induced by a series of DTLS handshakes. However, there is a possibility of Hello DoS attacks being run in the IoT environment.

A payload-based mutual authentication scheme has been suggested to overcome this issue [125]. It avoids implementing DTLS through a separate control channel in the IoT and reduces the possibility of attacks in resource-constrained sensors. Although it introduces payload-based authentication to the CoAP, data confidentiality is still achieved using AES. However, traditional AES needs to share a secret key securely and utilising secret keys for a long time reduces the security of the CoAP messages in the IoT. Thus, the traditional AES algorithm needs to be improved with dynamic key sharing and lightweight encryption processes.

1.6.3.2 Lightweight Cryptography Schemes for the CoAP

Several lightweight encryption algorithms using symmetric and asymmetric techniques have been suggested for various IoT applications. For example, a lightweight encryption algorithm is proposed in [126] to enhance the secure data transmission of smart sensors. It exploits Feistel structure and a network with uniform substitution-permutation in a combinational form. Likewise, the Data Encryption Standard (DES) in [127] applies a symmetric key block cipher with Feistel structure. The plaintext of 64-bit and a key size of 56 bits are used in an encryption process with 16 rounds. However, the main drawback of the DES algorithm is its flexibility in Feistel structure and it does not support modifications for various IoT application services [128]. In [129], a symmetric encryption

algorithm exploits multi-cloud computing and provides privacy in forward and backward directions. However, there is a possibility of information leakage. The encryption process provides a faster response than DES, but it is vulnerable to various kinds of attacks. Moreover, the combination of several components in proposed scheme makes it very complex and costly.

Many-valued logic and variable block lengths are involved in the encryption algorithm in [130], and it is performed in five rounds, each round performing different operations. The first round performs the gamma and permutation procedures, and other rounds perform substitution and gamma procedures. Super-encryption cryptography in [131] combines the International Data Encryption Algorithm (IDEA) and the Word Auto Key Encryption (WAKE) algorithm. The super-encryption technique involves two or more symmetric cryptographic algorithms to ensure more security for IoT data. Although the algorithms are designed specifically for resource-constrained smart sensors, they involve complex cryptography functions to provide high-level security. The minimum memory utilisation level and feasibility in implementation have made the Tiny Encryption Algorithm (TEA) very well known. The main TEA and its numerous developed versions apply the same key for encryption rounds. However, this degrades the strength and efficiency of the security algorithm. In addition, it takes a large amount of time for the encryption and decryption process [132], [133]. A novel tiny symmetric encryption algorithm (NTSA) has been suggested to improve the security of IoT networks by providing additional key confusion in each round of encryption [134]. However, the cryptography techniques described above are limited in providing key length and security for IoT communication. The AES can handle an entire block of plain text in the form of a single matrix with the operation of substitution and permutation [135]. However, the ability to optimise the AES algorithm to support resource-constrained smart sensors is still lacking in the IoT.

2.7 Industrial IoT System Integrity

Industrial control systems (ICSs) are designed to support industrial processes. Several operations and processes, such as distributed energy resources, water treatment or transportation control, are tracked and managed by these systems. ICSs are (more formally) SCADA (supervisory control and data acquisition) systems. ICSs have moved away from being proprietary, individual systems linked only to integrated and Internet-

based standardised technology in a short period of time. Furthermore, ICS-based products are most frequently found in a range of consumer or industrial devices, such as routers or cable modems, which typically use commercial off-the-shelf software [79]. However, ICSs are increasingly vulnerable to cyberattacks and insider threats in industrial applications. Industrial applications should be protected from system and data integrity attacks in order to improve system security. Threats to systems and information integrity could disrupt critical manufacturing programs, reduce productivity, endanger safety and have a negative impact on business activities. The IIoT is gradually becoming more common due to its operational application in different areas [136]. As a result, various security vulnerabilities and threats that arise specifically with IIoT applications have been discovered. In the following subsection, the study addresses the existing solutions based on smart grid applications.

2.7.1 Case Study on IIoT Applications: Smart Grids

The term “smart grid” refers to an enhanced energy supply chain that extends from a large power plant to homes and businesses. The fundamental premise of the smart grid is to enhance a nation’s electrical distribution infrastructure with tracking, review, control and communication capabilities, in order to optimise system throughput while minimising energy consumption. Despite the advantages smart grids offer over conventional networks, the smart grid is exposed to a number of vulnerabilities and attacks [137]. This subsection highlights some of the current research being conducted to address the availability, integrity and confidentiality of the infrastructure communication and control systems of the smart grid.

The main security issues facing users connected to smart grids are consumer privacy, confidentiality and the ability to maintain the integrity of their energy consumption. A security protocol is developed for a smart grid using different double auction mechanisms and homomorphic encryption in [138]. This protocol provides authentication, security and compatibility to smart grid technologies. It assigns a pseudo-identity to each consumer in the smart grid and ensures the anonymity of consumers during communication. However, the homomorphic encryption schemes in [138] and [139] generate a lengthier cipher-suite than does plain text, resulting in a large delay for encryption and decryption. A lightweight authentication and key agreement is suggested

in [140] for a smart metering network that exploits hybrid cryptography (i.e., ECC) and a symmetric key to provide security. Before applying cryptography, both the gateway and the smart meters are mutually authenticated. The hybrid cryptography-based security scheme protects against many attacks, but it lacks a focus on a number of privacy features, such as unforgeability and undetectability.

Several ECC-based authentication schemes [141], [142] have been proposed for smart grid networks. However, they do not satisfy all the security features. A pairing scheme between smart meters and servers is an expensive operation, and it does not apply to smart meters with low power. In [143], a certificateless two-party authenticated key agreement (CL2PAKA) scheme is suggested for smart grid applications. CL2PAKA does not need to perform any pairing operation and it implements only four scalar multiplication operations based on ECC. The main disadvantage of certificateless authentication schemes [143], [144] is that the identity information cannot provide a public key for a long time. This means that the identity of consumers alone is not sufficient to provide a public and a secret key. In [145], a privacy-preserving architecture is suggested for the smart grid using a Q-learning-based optimised approach. It exploits the cryptography technique to outsource multiregional electricity data securely. It implements three dynamic protocols to perform primary operations in Q-learning: Q-value updating, Q-learning training, and knowledge replay with encrypted packet information. However, it is time-consuming in reaching an optimal value, and so is less applicable to resource-constrained smart grid applications.

It is clear that the existing works mostly apply either symmetric or asymmetric encryption schemes. Symmetric encryption has to share a secret key in advance, but asymmetric encryption does not need to set a shared key in advance. However, asymmetric encryption involves greater computational cost than the symmetric encryption technique. Therefore, an effective method is to generate a shared key and encrypt the data. For smart grid applications, it is important to provide an effective authentication and security scheme in a lightweight manner. As smart meters in AMI are low-energy devices, highly complex security schemes are not applicable in this context.

2.8 Knowledge Gap

Although other studies might not be mentioned, this thesis provides a comprehensive overview of the literature on the security included in IoT layers, the potential for cyberattacks and their solutions, limitations and countermeasures. There are numerous established methods for securing IoT applications. There are still many issues to be studied and addressed in future research. As such, this thesis addressed the following research gaps.

- Even though there have been numerous literature reviews in the field of IoT/IIoT security. However, they are limited and primarily focused on either specific IoT security aspects, attacks, layer vision or provide a limited evaluation of the implemented security solutions. There is a need for a compact literature review evaluating the attacks, countermeasures, simulations and tools and critically analysing the implemented security solutions.
- IoT devices face several obstacles in adopting effective security solutions because of their resource-constrained and restricting characteristics, such as limited memory, low cost, fewer computational capabilities, and low power consumption. Thus, developing optimised security schemes for heterogeneous industrial IoT networks is needed.
- The implementation of the IDS in IoT networks presents new challenges, as it generates a large number of false alerts. It is a challenge to provide real-time IoT-IDSs and extend the range of attack detection. Therefore, developing a novel detection system for heterogeneous industrial IoT networks needs to be developed using deep learning techniques that will make IDSs resistant to a broader range of attacks with high detection accuracy.

2.9 Chapter Summary

The new technology of the IoT is enabling the physical network connectivity and computing capability of sensors and control system to generate, exchange and consume data with minimal human interaction. The survey presented in this chapter has examined various security threats at different IoT layers, including the security challenges and solutions regarding the end-to-end IoT environment. The survey covered the security issues related to the network layer, middleware layer, communication protocols and application layer. It also provided a critical analysis of existing IoT solutions based on different security mechanisms, including cryptography and IDSs. State-of-the-art IoT security in some of the future research directions to enhance IoT security levels was also discussed. This chapter is expected to become a roadmap towards security enhancement for IIoT applications.

2.10 References

1. Al-Rubaye S., Rodriguez J., Fragonara LZ., Theron P., Tsourdos A. Unleash Narrowband Technologies for Industrial Internet of Things Services. *IEEE Network*. 2019; 33(4): 16–22. Available at: DOI:10.1109/MNET.2019.1800414
2. Al-Rubaye S., Kadhum E., Ni Q., Anpalagan A. Industrial Internet of Things Driven by SDN Platform for Smart Grid Resiliency. *IEEE Internet of Things Journal*. IEEE; 2019; 6(1): 267–277. Available at: DOI:10.1109/JIOT.2017.2734903
3. Stellios I., Kotzanikolaou P., Psarakis M., Alcaraz C., Lopez J. A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Communications Surveys and Tutorials*. IEEE; 2018; 20(4): 3453–3495. Available at: DOI:10.1109/COMST.2018.2855563
4. Alexander B., Haseeb S., Baranchuk A. Are implanted electronic devices hackable? *Trends in Cardiovascular Medicine*. Elsevier Inc.; 2019; 29(8): 476–480. Available at: DOI:10.1016/j.tcm.2018.11.011
5. Wan J., Tang S., Shu Z., Li D., Wang S., Imran M., et al. Software-Defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE Sensors Journal*. 2016; 16(20): 7373–7380. Available at: DOI:10.1109/JSEN.2016.2565621
6. Sengupta J., Ruj S., Das Bit S. A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. *Journal of Network and Computer Applications*. Elsevier Ltd; 2020; 149(April 2019): 102481. Available at: DOI:10.1016/j.jnca.2019.102481
7. Lee I., Lee K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*. ‘Kelley School of Business, Indiana University’; 2015; 58(4): 431–440. Available at: DOI:10.1016/j.bushor.2015.03.008
8. Deogirikar J., Vidhate A. Security attacks in IoT: A survey. *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC*

2017. 2017; : 32–37. Available at: DOI:10.1109/I-SMAC.2017.8058363
9. Sudeendra Kumar K., Sahoo S., Mahapatra A., Swain AK., Mahapatra KK. Security enhancements to system on chip devices for IoT perception layer. Proceedings - 2017 IEEE International Symposium on Nanoelectronic and Information Systems, iNIS 2017. 2018; 2018-Febru: 151–156. Available at: DOI:10.1109/iNIS.2017.39
 10. Kulkarni GA., Mitra M. ‘RFID Security Issues’. International Journal of Engineering Research & Technology (IJERT). 2014; (June).
 11. Wallgren L., Raza S., Voigt T. Routing attacks and countermeasures in the RPL-based internet of things. International Journal of Distributed Sensor Networks. 2013; 2013. Available at: DOI:10.1155/2013/794326
 12. Pongle P., Chavan G. A survey: Attacks on RPL and 6LoWPAN in IoT. 2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015. IEEE; 2015; 00(c). Available at: DOI:10.1109/PERVASIVE.2015.7087034
 13. Nawir M., Amir A., Yaakob N., Lynn OB. Internet of Things (IoT): Taxonomy of security attacks. 2016 3rd International Conference on Electronic Design, ICED 2016. IEEE; 2017; : 321–326. Available at: DOI:10.1109/ICED.2016.7804660
 14. Abdul-Ghani HA., Konstantas D., Mahyoub M. A comprehensive IoT attacks survey based on a building-blocked reference model. International Journal of Advanced Computer Science and Applications. 2018; 9(3): 355–373. Available at: DOI:10.14569/IJACSA.2018.090349
 15. Karlof C., Wagner D. Secure routing in wireless sensor networks: Attacks and countermeasures. Ad Hoc Networks. 2003; 1(2–3): 293–315. Available at: DOI:10.1016/S1570-8705(03)00008-8
 16. Gautam S., Malik A., Singh N., Kumar S. Recent Advances and Countermeasures against Various Attacks in IoT Environment. 2nd International Conference on Signal Processing and Communication, ICSPC 2019 - Proceedings. 2019; : 315–319. Available at: DOI:10.1109/ICSPC46172.2019.8976527

17. Sicari S., Rizzardi A., Grieco L.A., Coen-Porisini A. Security, privacy and trust in Internet of things: The road ahead. *Computer Networks*. Elsevier B.V.; 2015; 76: 146–164. Available at: DOI:10.1016/j.comnet.2014.11.008
18. Rizvi S., Kurtz A., Pfeffer J., Rizvi M. Securing the Internet of Things (IoT): A Security Taxonomy for IoT. *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*. IEEE; 2018; : 163–168. Available at: DOI:10.1109/TrustCom/BigDataSE.2018.00034
19. Hari P.B., Singh S.N. Security issues in Wireless Sensor Networks: Current research and challenges. *Proceedings - 2016 International Conference on Advances in Computing, Communication and Automation, ICACCA 2016*. IEEE; 2016; Available at: DOI:10.1109/ICACCA.2016.7578876
20. Ali W., Dustgeer G., Awais M., Shah M.A. IoT based smart home: Security challenges, security requirements and solutions. *ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing*. 2017; (September): 7–8. Available at: DOI:10.23919/ICoNAC.2017.8082057
21. Hossain M.M., Fotouhi M., Hasan R. Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things. *Proceedings - 2015 IEEE World Congress on Services, SERVICES 2015*. IEEE; 2015; : 21–28. Available at: DOI:10.1109/SERVICES.2015.12
22. Cawthra J., Ekstrom M., Lusty L., Sexton J., Sweetnam J. NIST Special Publication 1800-25, *Data Integrity: Identifying and Protecting Assets Against Ransomware and Other Destructive Events*. National Institute of Standards and Technology. 2020;
23. Singh K., Singh Tomar D.D. Architecture, enabling technologies, security and privacy, and applications of internet of things: A survey. *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*. 2019; 4(5): 642–646. Available at: DOI:10.1109/I-

SMAC.2018.8653708

24. Vashi S., Ram J., Modi J., Verma S., Prakash C. Internet of Things (IoT): A vision, architectural elements, and security issues. Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017. 2017; (December): 492–496. Available at: DOI:10.1109/I-SMAC.2017.8058399
25. Liu X., Zhao M., Li S., Zhang F., Trappe W. A security framework for the internet of things in the future internet architecture. Future Internet. 2017; 9(3): 1–28. Available at: DOI:10.3390/fi9030027
26. Andrea I., Chrysostomou C., Hadjichristofi G. Internet of Things: Security vulnerabilities and challenges. Proceedings - IEEE Symposium on Computers and Communications. 2016; 2016-Febru: 180–187. Available at: DOI:10.1109/ISCC.2015.7405513
27. Jing Q., Vasilakos A V., Wan J., Lu J., Qiu D. Security of the Internet of Things: perspectives and challenges. Wireless Networks. 2014; 20(8): 2481–2501. Available at: DOI:10.1007/s11276-014-0761-7
28. Mahmoud R., Yousuf T., Aloul F., Zualkernan I. Internet of things (IoT) security: Current status, challenges and prospective measures. 2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015. 2016; 5(4): 336–341. Available at: DOI:10.1109/ICITST.2015.7412116
29. Dragomir D., Gheorghe L., Costea S., Radovici A. A survey on secure communication protocols for IoT systems. Proceedings - 2016 International Workshop on Secure Internet of Things, SIoT 2016. IEEE; 2017; : 47–62. Available at: DOI:10.1109/SIoT.2016.012
30. Jorge Granjal, Edmundo Monteiro and JSS. Security for internet of things: A state of the art on existing protocols and open research issues. IEEE COMMUNICATION SURVEYS & TUTORIALS. IEEE; 2019; 17(3): 1294–1312. Available at: DOI:10.1145/3361570.3361622
31. Frustaci M., Pace P., Aloï G., Fortino G. Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of Things Journal. IEEE;

- 2018; 5(4): 2483–2495. Available at: DOI:10.1109/JIOT.2017.2767291
32. Jabraeil Jamali MA., Bahrami B., Heidari A., Allahverdizadeh P., Norouzi F. Towards the Internet of Things. Imrich Chlamtac, European Alliance for Innovation, Gent B (ed.) Cham: Springer International Publishing; 2020. 9–31 p. Available at: DOI:10.1007/978-3-030-18468-1
 33. Herrero R. Fundamentals of IoT Communication Technologies. El-Bawab TS, Dean P and, Engineering S of, Nigeria AU of (eds.) Springer Nature Switzerland AG; 2021. 249 p. Available at: DOI:10.1007/978-3-030-70080-5
 34. Naoui S., Elhdhili ME., Saidane LA. Enhancing the security of the IoT LoraWAN architecture. 5th IFIP International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, PEMWN 2016. IEEE; 2017; Available at: DOI:10.1109/PEMWN.2016.7842904
 35. Aras E., Ramachandran GS., Lawrence P., Hughes D. Exploring the security vulnerabilities of LoRa. 2017 3rd IEEE International Conference on Cybernetics, CYBCONF 2017 - Proceedings. 2017; Available at: DOI:10.1109/CYBConf.2017.7985777
 36. Lowne A. Securing the internet of things. ECN Electronic Component News. IEEE; 2014; 58(3): 30–31.
 37. Raza S., Duquennoy S., Voigt T., Roedig U. Demo abstract: Securing communication in 6LoWPAN with compressed IPsec. 2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS'11. IEEE; 2011; Available at: DOI:10.1109/DCOSS.2011.5982146
 38. Arfaoui A., Kribeche A., Senouci SM., Hamdi M. Game-Based Adaptive Remote Access VPN for IoT: Application to e-Health. 2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings. 2018; Available at: DOI:10.1109/GLOCOM.2018.8648064
 39. Aris, A.; Oktug, S.F.; Voigt T. Security of Internet of Things for a Reliable Internet of Services. ICPE 2017 - Companion of the 2017 ACM/SPEC International Conference on Performance Engineering. Cham, Switzerland: Springer;; 2018. iii

- p. Available at: DOI:10.1007/978-3-319-90415-3_13
40. Al-Kashoash HAA., Kharrufa H., Al-Nidawi Y., Kemp AH. Congestion control in wireless sensor and 6LoWPAN networks: toward the Internet of Things. *Wireless Networks*. 2019; 25(8): 4493–4522. Available at: DOI:10.1007/s11276-018-1743-y
 41. Khan MA., Salah K. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*. Elsevier B.V.; 2018; 82: 395–411. Available at: DOI:10.1016/j.future.2017.11.022
 42. Arabia S. 2018 1st International Conference on Computer Applications & Information Security (ICCAIS 2018) Riyadh , Saudi Arabia. 2018; (April).
 43. Bilal M. A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers. 2017; : 1–21. Available at: <http://arxiv.org/abs/1708.04560>
 44. Airehrour D., Gutierrez J., Ray SK. Secure routing for internet of things: A survey. *Journal of Network and Computer Applications*. 2016; 66: 198–213. Available at: DOI:10.1016/j.jnca.2016.03.006
 45. Santhosh Krishna B V., Gnanasekaran T. A systematic study of security issues in Internet-of-Things (IoT). *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*. 2017; (October): 107–111. Available at: DOI:10.1109/I-SMAC.2017.8058318
 46. Singh D., Tripathi G., Jara A. Secure layers based architecture for Internet of Things. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*. IEEE; 2015; : 321–326. Available at: DOI:10.1109/WF-IoT.2015.7389074
 47. Benkhelifa E., Welsh T., Hamouda W. A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE Communications Surveys and Tutorials*. IEEE; 2018; 20(4): 3496–3509. Available at: DOI:10.1109/COMST.2018.2844742
 48. Ammar M., Russello G., Crispo B. Internet of Things: A survey on the security of

- IoT frameworks. *Journal of Information Security and Applications*. Elsevier Ltd; 2018; 38: 8–27. Available at: DOI:10.1016/j.jisa.2017.11.002
49. Chernyshev M., Baig Z., Bello O., Zeadally S. Internet of things (IoT): Research, simulators, and testbeds. *IEEE Internet of Things Journal*. IEEE; 2018; 5(3): 1637–1647. Available at: DOI:10.1109/JIOT.2017.2786639
50. Ani UD., Watson JMK., Carr M., Cook A., Nurse JRC. A review of the use and utility of industrial network-based open source simulators: functionality, security, and policy viewpoints. *Journal of Defense Modeling and Simulation*. 2020; : 11–20. Available at: DOI:10.1177/1548512920953499
51. Al-Kashoash HAA., Kharrufa H., Al-Nidawi Y., Kemp AH. Congestion control in wireless sensor and 6LoWPAN networks: toward the Internet of Things. *Wireless Networks*. Springer US; 2019; 25(8): 4493–4522. Available at: DOI:10.1007/s11276-018-1743-y
52. Varga A., Hornig R. AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT. n *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. 2008.
53. Choi HY., Min SG., Han YH., Park J., Kim H. Implementation and evaluation of proxy mobile IPv6 in NS-3 network simulator. 2010 *Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications, CUTE 2010*. IEEE; 2010; Available at: DOI:10.1109/ICUT.2010.5677817
54. Siraj S., Gupta AK., Badgujar-Rinku. Network Simulation Tools Survey. *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 1, Issue 4, June 2012. 2012; 1(4): 201–210.
55. Queiroz C., Mahmood A., Tari Z. SCADASimA framework for building SCADA simulations. *IEEE Transactions on Smart Grid*. IEEE; 2011; 2(4): 589–597. Available at: DOI:10.1109/TSG.2011.2162432
56. Zikria Y Bin., Kim SW., Hahm O., Afzal MK., Aalsalem MY. Internet of things

- (IoT) operating systems management: Opportunities, challenges, and solution. *Sensors* (Switzerland). 2019; 19(8). Available at: DOI:10.3390/s19081793
57. Paul T., Kumar GS. Safe contiki OS: Type and memory safety for contiki OS. *ARTCom 2009 - International Conference on Advances in Recent Technologies in Communication and Computing*. 2009; : 169–171. Available at: DOI:10.1109/ARTCom.2009.126
58. Saadallah B., Lahmadi A., Festor O. CCNx for Contiki: implementation details. *INRIA, Tech. Rep. RT-0432*. 2012; : 52. Available at: <https://hal.inria.fr/hal-00755482%0Ahttps://hal.inria.fr/hal-00755482/document>
59. Samal N., Dalai P. A PERFORMANCE SURVEY OF OPERATING SYSTEMS IN IoT ENVIRONMENT. *International Journal of Computer Science and Mobile Applications*. 2018; (March): 23–24.
60. Avram Piltch. Raspberry pi 4 Review: The New Gold Standard for Single-Board Computing-Tom's Hardware Tom's Hardware. Online. 2020. Available at: <https://www.tomshardware.com/reviews/raspberry-pi-4>
61. best iot os, Internet Of Things, iot operating systems. 2021. Available at: <https://solaceinfotech.com/blog/top-10-iot-operating-systems-for-iot-devices-in-2021/> (Accessed: 29 April 2021)
62. Kamesh., Sakthi Priya N. A survey of cyber crimes Yanping. *Security and Communication Networks*. 2012; 5(June): 422–437. Available at: DOI:10.1002/sec
63. Ndibanje B., Lee HJ., Lee SG. Security analysis and improvements of authentication and access control in the internet of things. *Sensors* (Switzerland). 2014; 14(8): 14786–14805. Available at: DOI:10.3390/s140814786
64. Lee T-H., Wen C-H., Chang L-H., Chiang H-S., Hsieh M-C. A Lightweight Intrusion Detection Scheme Based on Energy Consumption Analysis in 6LowPAN. *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*. New York London: Springer Dordrecht Heidelberg; 2014. pp. 1205–1213. Available at: DOI:10.1007/978-94-007-7262-5_137

65. Oh D., Kim D., Ro WW. A malicious pattern detection engine for embedded security systems in the internet of things. *Sensors (Switzerland)*. 2014; 14(12): 24188–24211. Available at: DOI:10.3390/s141224188
66. Cervantes C., Poplade D., Nogueira M., Santos A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*. IEEE; 2015; : 606–611. Available at: DOI:10.1109/INM.2015.7140344
67. Raza S., Wallgren L., Voigt T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks*. Elsevier B.V.; 2013; 11(8): 2661–2674. Available at: DOI:10.1016/j.adhoc.2013.04.014
68. Deshmukh-Bhosale S., Sonavane SS. A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things. *Procedia Manufacturing*. Elsevier B.V.; 2019; 32: 840–847. Available at: DOI:10.1016/j.promfg.2019.02.292
69. Le A., Loo J., Luo Y., Lasebae A. Specification-based IDS for securing RPL from topology attacks. *IFIP Wireless Days*. IEEE; 2011; 1(1): 4–6. Available at: DOI:10.1109/WD.2011.6098218
70. Le A., Loo J., Chai KK., Aiash M. A specification-based IDS for detecting attacks on RPL-based network topology. *Information (Switzerland)*. 2016; 7(2). Available at: DOI:10.3390/info7020025
71. Shin D., Sharma V., Kim J., Kwon S., You I., Member S. Secure and Efficient Protocol for Route Optimization in PMIPv6-Based Smart Home IoT Networks. 2017; 5.
72. Airehrour D., Gutierrez JA., Ray SK. SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things. *Future Generation Computer Systems*. 2019; 93: 860–876. Available at: DOI:10.1016/j.future.2018.03.021
73. Pongle P., Chavan G. Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *International Journal of Computer Applications*. 2015; 121(9):

- 1–9. Available at: DOI:10.5120/21565-4589
74. Mirshahjafari SMH., Ghahfarokhi BS. Sinkhole+CloneID: A hybrid attack on RPL performance and detection method. *Information Security Journal*. Taylor & Francis; 2019; 28(4–5): 107–119. Available at: DOI:10.1080/19393555.2019.1658829
75. Patel HB., Jinwala DC. Blackhole Detection in 6LoWPAN Based Internet of Things: An Anomaly Based Approach. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*. IEEE; 2019; 2019-Octob: 947–954. Available at: DOI:10.1109/TENCON.2019.8929491
76. Aydogan E., Yilmaz S., Sen S., Butun I., Forsstrom S., Gidlund M. A Central Intrusion Detection System for RPL-Based Industrial Internet of Things. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. 2019; 2019-May. Available at: DOI:10.1109/WFCS.2019.8758024
77. Ghaeini HR., Tippenhauer NO. HAMIDS: Hierarchical monitoring intrusion detection system for industrial control systems. *CPS-SPC 2016 - Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and PrivaCy, co-located with CCS 2016*. 2016; : 103–111. Available at: DOI:10.1145/2994487.2994492
78. Faisal MA., Cardenas AA., Wool A. Profiling communications in industrial ip networks: Model complexity and anomaly detection. *Advanced Sciences and Technologies for Security Applications*. 2019; : 139–160. Available at: DOI:10.1007/978-3-030-12330-7_7
79. Powell M., Brule J., Pease M., Stouffer K., Tang C., Zimmerman T., et al. Protecting Information and System Integrity in Industrial Control System Environments: Cybersecurity for the Manufacturing Sector. NIST. 2020; (February 2020). Available at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-10.pdf>
80. Verma A., Ranga V. Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Personal Communications*. Springer US; 2020; 111(4):

- 2287–2310. Available at: DOI:10.1007/s11277-019-06986-8
81. Chakraborty S., Onuchowska A., Samtani S., Jank W., Wolfram B. Machine Learning for Automated Industrial IoT Attack Detection: An Efficiency-Complexity Trade-off. *ACM Transactions on Management Information Systems*. 2021; 12(4): 1–28. Available at: DOI:10.1145/3460822
 82. Sharma M., Elmiligi H., Gebali F. A Novel Intrusion Detection System for RPL-Based Cyber – Physical Systems Un nouveau système de détection d ’ intrusion pour les systèmes cyber-physiques basés sur RPL. *IEEE*; 2021; 44(2): 246–252.
 83. Foley J., Moradpoor N., Ochenyi H. Employing a Machine Learning Approach to Detect Combined Internet of Things Attacks against Two Objective Functions Using a Novel Dataset. *Security and Communication Networks*. 2020; 2020. Available at: DOI:10.1155/2020/2804291
 84. Agiollo A., Conti M., Kaliyar P., Lin TN., Pajola L. DETONAR: Detection of Routing Attacks in RPL-Based IoT. *IEEE Transactions on Network and Service Management*. 2021; 18(2): 1178–1190. Available at: DOI:10.1109/TNSM.2021.3075496
 85. Nguyen DC., Ding M., Pathirana PN., Seneviratne A., Li J., Vincent Poor H. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*. *IEEE*; 2021; 23(3): 1622–1658. Available at: DOI:10.1109/COMST.2021.3075439
 86. Chatterjee S., Hanawal MK. Federated Learning for Intrusion Detection in IoT Security: A Hybrid Ensemble Approach. 2021; Available at: DOI:10.1504/ijitca.2023.10048062
 87. Tang Z., Hu H., Xu C. A federated learning method for network intrusion detection. *Concurrency and Computation: Practice and Experience*. 2022; 34(10): 1–16. Available at: DOI:10.1002/cpe.6812
 88. K A., J S., Maurya S., Joseph S., Asokan A., M P., et al. Federated Transfer Learning for Authentication and Privacy Preservation Using Novel Supportive Twin Delayed DDPG (S-TD3) Algorithm for IIoT. *Sensors (Basel, Switzerland)*.

- 2021; 21(23). Available at: DOI:10.3390/s21237793
89. Cui L., Qu Y., Xie G., Zeng D., Li R., Shen S., et al. Security and Privacy-Enhanced Federated Learning for Anomaly Detection in IoT Infrastructures. *IEEE Transactions on Industrial Informatics*. 2022; 18(5): 3492–3500. Available at: DOI:10.1109/TII.2021.3107783
 90. Rey V., Sánchez Sánchez PM., Huertas Celdrán A., Bovet G. Federated learning for malware detection in IoT devices. *Computer Networks*. Elsevier B.V.; 2022; 204(April 2021): 108693. Available at: DOI:10.1016/j.comnet.2021.108693
 91. Nguyen TD., Rieger P., Miettinen M., Sadeghi A-R. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. 2021; (February). Available at: DOI:10.14722/diss.2020.23003
 92. Li Y., Li Y., Xu H., Ren S. An Adaptive Communication-Efficient Federated Learning to Resist Gradient-Based Reconstruction Attacks. *Security and Communication Networks*. 2021; 2021. Available at: DOI:10.1155/2021/9919030
 93. Wu Q., He K., Chen X. Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge based Framework. *IEEE Computer Graphics and Applications*. 2020; 1(August): 35–44. Available at: DOI:10.1109/OJCS.2020.2993259
 94. Rahman SA., Tout H., Talhi C., Mourad A. Internet of Things intrusion Detection: Centralized, On-Device, or Federated Learning? *IEEE Network*. 2020; 34(6): 310–317. Available at: DOI:10.1109/MNET.011.2000286
 95. Chen Z., Lv N., Liu P., Fang Y., Chen K., Pan W. Intrusion Detection for Wireless Edge Networks Based on Federated Learning. *IEEE Access*. 2020; 8: 217463–217472. Available at: DOI:10.1109/ACCESS.2020.3041793
 96. Man D., Zeng F., Yang W., Yu M., Lv J., Wang Y. Intelligent Intrusion Detection Based on Federated Learning for Edge-Assisted Internet of Things. *Security and Communication Networks*. 2021; 2021. Available at: DOI:10.1155/2021/9361348
 97. Attota DC., Mothukuri V., Parizi RM., Pouriyeh S. An Ensemble Multi-View

- Federated Learning Intrusion Detection for IoT. IEEE Access. IEEE; 2021; 9: 117734–117745. Available at: DOI:10.1109/ACCESS.2021.3107337
98. Zhang W., Yang D., Wu W., Peng H., Zhang H., Shen XS. Spectrum and Computing Resource Management for Federated Learning in Distributed Industrial IoT. 2021 IEEE International Conference on Communications Workshops, ICC Workshops 2021 - Proceedings. IEEE; 2021; 39(12): 3688–3703. Available at: DOI:10.1109/ICCWorkshops50388.2021.9473515
99. Yilmaz S., Aydogan E., Sen S. A Transfer Learning Approach for Securing Resource-Constrained IoT Devices. IEEE Transactions on Information Forensics and Security. 2021; 16: 4405–4418. Available at: DOI:10.1109/TIFS.2021.3096029
100. Zhang P., Sun H., Situ J., Jiang C., Xie D. Federated Transfer Learning for IIoT Devices with Low Computing Power Based on Blockchain and Edge Computing. IEEE Access. IEEE; 2021; 9: 98630–98638. Available at: DOI:10.1109/ACCESS.2021.3095078
101. Khoa TV., Hoang DT., Trung NL., Nguyen CT., Quynh TTT., Nguyen DN., et al. Deep Transfer Learning: A Novel Collaborative Learning Model for Cyberattack Detection Systems in IoT Networks. 2021; : 1–12. Available at: <http://arxiv.org/abs/2112.00988>
102. Liu X., Yu W., Liang F., Griffith D., Golmie N. Toward Deep Transfer Learning in Industrial Internet of Things. IEEE Internet of Things Journal. IEEE; 2021; 8(15): 12163–12175. Available at: DOI:10.1109/JIOT.2021.3062482
103. Guan J., Cai J., Bai H., You I. Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems. International Journal of Machine Learning and Cybernetics. Springer Berlin Heidelberg; 2021; 12(11): 3351–3365. Available at: DOI:10.1007/s13042-021-01415-4
104. Raza A., Tran KP., Koehl L., Li S. Designing ECG monitoring healthcare system with federated transfer learning and explainable AI. Knowledge-Based Systems. Elsevier B.V.; 2022; 236: 107763. Available at:

DOI:10.1016/j.knosys.2021.107763

105. Otoum Y., Wan Y., Nayak A. Federated Transfer Learning-Based IDS for the Internet of Medical Things (IoMT). 2021 IEEE Globecom Workshops, GC Wkshps 2021 - Proceedings. IEEE; 2021; (MI): 1–6. Available at: DOI:10.1109/GCWkshps52748.2021.9682118
106. Kothmayr T., Schmitt C., Hu W., Brunig M., Carle G. A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. Proceedings - Conference on Local Computer Networks, LCN. 2012; : 956–963. Available at: DOI:10.1109/LCNW.2012.6424088
107. Kothmayr T., Schmitt C., Hu W., Brünig M., Carle G. DTLS based security and two-way authentication for the Internet of Things. Ad Hoc Networks. Elsevier B.V.; 2013; 11(8): 2710–2723. Available at: DOI:10.1016/j.adhoc.2013.05.003
108. Granjal J., Monteiro E. End-to-end transparent transport-layer security for Internet-integrated mobile sensing devices. 2016 IFIP Networking Conference (IFIP Networking) and Workshops, IFIP Networking 2016. 2016; : 306–314. Available at: DOI:10.1109/IFIPNetworking.2016.7497235
109. Raza S., Shafagh H., Hewage K., Hummen R., Voigt T. Lithe: Lightweight secure CoAP for the internet of things. IEEE Sensors Journal. 2013; 13(10): 3711–3720. Available at: DOI:10.1109/JSEN.2013.2277656
110. Haroon A., Akram S., Shah MA., Wahid A. E-lithe: A lightweight secure DTLS for IoT. IEEE Vehicular Technology Conference. 2018; 2017-Sept: 1–5. Available at: DOI:10.1109/VTCFall.2017.8288362
111. Al Salami S., Baek J., Salah K., Damiani E. Lightweight encryption for smart home. Proceedings - 2016 11th International Conference on Availability, Reliability and Security, ARES 2016. IEEE; 2016; : 382–388. Available at: DOI:10.1109/ARES.2016.40
112. Halabi D., Hamdan S., Almajali S. Enhance the security in smart home applications based on IOT-CoAP protocol. 6th International Conference on Digital Information, Networking, and Wireless Communications, DINWC 2018. IEEE;

- 2018; : 81–85. Available at: DOI:10.1109/DINWC.2018.8357000
113. Abosata NRA., Kemp AH., Razavi M. Secure Smart-Home Application Based on IoT- CoAP protocol. Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE; 2019. pp. 13–17.
 114. Granjal J., Monteiro E., Silva JS. End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication. 2013 IFIP Networking Conference, IFIP Networking 2013. IFIP; 2013; : 1–9.
 115. Kumar P., Lin Y., Bai G., Paverd A., Dong JS., Martin A. Smart Grid Metering Networks: A Survey on Security, Privacy and Open Research Issues. IEEE Communications Surveys and Tutorials. IEEE; 2019; 21(3): 2886–2927. Available at: DOI:10.1109/COMST.2019.2899354
 116. Abosata N., Al-Rubaye S., Inalhan G., Emmanouilidis C. Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. Sensors. MDPI AG; 2021. Available at: DOI:10.3390/s21113654
 117. Capossele A., Cervo V., De Cicco G., Petrioli C. Security as a CoAP resource: An optimized DTLS implementation for the IoT. IEEE International Conference on Communications. IEEE; 2015; 2015-Sept: 549–554. Available at: DOI:10.1109/ICC.2015.7248379
 118. Alghamdi TA., Lasebae A., Aiash M. Security analysis of the constrained application protocol in the Internet of Things. 2nd International Conference on Future Generation Communication Technologies, FGCT 2013. IEEE; 2013; : 163–168. Available at: DOI:10.1109/FGCT.2013.6767217
 119. Maleh Y., Ezzati A., Belaissaoui M. An enhanced DTLS protocol for Internet of Things applications. Proceedings - 2016 International Conference on Wireless Networks and Mobile Communications, WINCOM 2016: Green Communications and Networking. IEEE; 2016; : 168–173. Available at: DOI:10.1109/WINCOM.2016.7777209

120. Herrero R., Hernandez D. Forward error correction in real-time Internet of things CoAP-based wireless sensor networks. *IET Wireless Sensor Systems*. 2019; 9(1): 42–47. Available at: DOI:10.1049/iet-wss.2018.5077
121. Park J., Kang N. Lightweight secure communication for CoAP-enabled Internet of Things using delegated DTLS handshake. *International Conference on ICT Convergence*. IEEE; 2014; : 28–33. Available at: DOI:10.1109/ICTC.2014.6983078
122. Hummen R., Ziegeldorf JH., Shafagh H., Raza S., Wehrle K. Towards Viable Certificate-based Authentication for the Internet of Things. *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*. 2013. pp. 37–41. Available at: <https://dl.acm.org/doi/pdf/10.1145/2463183.2463193>
123. Kim H., Lee EA. Authentication and Authorization for the Internet of Things. *IT Professional*. 2017; 19(5): 27–33. Available at: DOI:10.1109/MITP.2017.3680960
124. Kumar PM., Gandhi UD. Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application. *Journal of Supercomputing*. Springer US; 2020; 76(6): 3963–3983. Available at: DOI:10.1007/s11227-017-2169-5
125. Jan MA., Khan F., Alam M., Usman M. A payload-based mutual authentication scheme for Internet of Things. *Future Generation Computer Systems*. Elsevier B.V.; 2019; 92: 1028–1039. Available at: DOI:10.1016/j.future.2017.08.035
126. Park CS., Park WS. A Group-Oriented DTLS Handshake for Secure IoT Applications. *IEEE Transactions on Automation Science and Engineering*. IEEE; 2018; 15(4): 1920–1929. Available at: DOI:10.1109/TASE.2018.2855640
127. Usman M., Ahmed I., Aslam MI., Khan S., Shah UA. SIT: A Lightweight Encryption Algorithm for Secure Internet of Things. *International Journal of Advanced Computer Science and Applications*. 2017; 8(1): 402–411. Available at: DOI:10.14569/IJACSA.2017.080151
128. Ren W., Miao Z. A hybrid encryption algorithm based on DES and RSA in Bluetooth communication. *Proceedings - 2010 2nd International Conference on*

- Modeling, Simulation, and Visualization Methods, WMSVM 2010. IEEE; 2010; 0: 221–225. Available at: DOI:10.1109/WMSVM.2010.48
129. Gao C., Lv S., Wei Y., Wang Z., Liu Z., Cheng X. M-SSE: An effective searchable symmetric encryption with enhanced security for mobile devices. IEEE Access. IEEE; 2018; 6: 38860–38869. Available at: DOI:10.1109/ACCESS.2018.2852329
 130. Zhdanov ON., Sokolov A V. Block symmetric cryptographic algorithm based on principles of variable block length and many-valued logic. Far East Journal of Electronics and Communications. 2016; 16(3): 573–589. Available at: DOI:10.17654/EC016030573
 131. Abdullah D., Rahim R., Utama Siahaan AP., Ulva AF., Fitri Z., Malahayati M., et al. Super-Encryption Cryptography with IDEA and WAKE Algorithm. Journal of Physics: Conference Series. 2018; 1019(1). Available at: DOI:10.1088/1742-6596/1019/1/012039
 132. Rachmawati D., Sharif A., Jaysilen., Budiman MA. Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm. IOP Conference Series: Materials Science and Engineering. 2018; 300(1). Available at: DOI:10.1088/1757-899X/300/1/012042
 133. Novelan MS., Husein AM., Harahap M., Aisyah S. SMS Security System on Mobile Devices Using Tiny Encryption Algorithm. Journal of Physics: Conference Series. 2018; 1007(1). Available at: DOI:10.1088/1742-6596/1007/1/012037
 134. Rajesh S., Paul V., Menon VG., Khosravi MR. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry. 2019; 11(2). Available at: DOI:10.3390/sym11020293
 135. Al-Zubaidie M., Zhang Z., Zhang J. Efficient and secure ECDSA algorithm and its applications: A survey. International Journal of Communication Networks and Information Security. 2019; 11(1): 7–35.
 136. Stouffer K. DETECTING AND PROTECTING AGAINST DATA INTEGRITY ATTACKS IN INDUSTRIAL CONTROL Cybersecurity for the Manufacturing Sector. NIST. 2019; (June 2019).

137. European Union Agency for Network and Information Security. Smart Grid Security Recommendations for Europe and Member States. European Network and Information Security Agency. 2012; : 1–69.
138. Sarenche R., Salmasizadeh M., Ameri MH., Aref MR. A secure and privacy-preserving protocol for holding double auctions in smart grid. *Information Sciences*. Elsevier Inc.; 2021; 557: 108–129. Available at: DOI:10.1016/j.ins.2020.12.038
139. Khan AA., Kumar V., Ahmad M., Rana S. LAKAF: Lightweight authentication and key agreement framework for smart grid network. *Journal of Systems Architecture*. Elsevier B.V.; 2021; 116(October 2020): 102053. Available at: DOI:10.1016/j.sysarc.2021.102053
140. Abdallah A., Shen X. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*. IEEE; 2018; 9(1): 396–405. Available at: DOI:10.1109/TSG.2016.2553647
141. Abbasinezhad-Mood D., Nikooghadam M. An Anonymous ECC-Based Self-Certified Key Distribution Scheme for the Smart Grid. *IEEE Transactions on Industrial Electronics*. IEEE; 2018; 65(10): 7996–8004. Available at: DOI:10.1109/TIE.2018.2807383
142. Grover HS., Adarsh., Kumar D. Cryptanalysis and improvement of a three-factor user authentication scheme for smart grid environment. *Journal of Reliable Intelligent Environments*. Springer International Publishing; 2020; 6(4): 249–260. Available at: DOI:10.1007/s40860-020-00113-2
143. Deng L., Gao R. Certificateless two-party authenticated key agreement scheme for smart grid. *Information Sciences*. Elsevier Inc.; 2021; 543: 143–156. Available at: DOI:10.1016/j.ins.2020.07.025
144. Chaudhry SA., Alhakami H., Baz A., Al-Turjman F. Securing Demand Response Management: A Certificate-Based Access Control in Smart Grid Edge Computing Infrastructure. *IEEE Access*. 2020; 8: 101235–101243. Available at: DOI:10.1109/ACCESS.2020.2996093

145. Wang Z., Liu Y., Ma Z., Liu X., Ma J. LiPSG: Lightweight Privacy-Preserving Q-Learning-Based Energy Management for the IoT-Enabled Smart Grid. IEEE Internet of Things Journal. IEEE; 2020; 7(5): 3935–3947. Available at: DOI:10.1109/JIOT.2020.2968631

CHAPTER 3: Efficient encryption scheme for IoT based on distributed sensor networks

3.1 Abstract

The Internet of Things (IoT) is an embedded technology that enables physical objects to connect with the Internet in order to exchange and collect data. In practice, these objects rely on protocols to exchange data between the server and the Sensor. One of these protocols is the Constrained Application Protocol (CoAP), which enables sensors to send and receive data from a server with appropriate energy efficiency. However, these objects are not secure and are vulnerable to attacks, as well as being subject to restricted energy and limited resources. In this chapter, message integrity and authentication are added to the CoAP. Two different efficient algorithms are used, the 128-bit Advanced Encryption Standard (AES) and the secure hash algorithm (SHA)-256, to secure the communication over CoAP and investigate the impact of our schemes in terms of communication duration and energy consumption. The proposed schemes are evaluated using the Contiki OS that is included in the Cooja simulator tool based on an application environment that uses distributed sensors. The results show that the communication duration of the AES is six times faster than the SHA and was stable and energy efficient when the data size increased.

3.2 Introduction

The IoT is a vision that enables a wide range of wireless electronic devices (aka ‘things’) to communicate with each other using different addressing mechanisms [1]. IoT devices are equipped with sensors and processing power that allow them to be deployed in many environments [2]. Common IoT applications are home area networks (HANs) and smart cities, smart grid, and other industrial applications. Although the number of the IoT connected devices to the Internet increasing rapidly [3], however, only 10% of IoT devices were estimated to have sufficient security features [1]. The number of IoT devices in an IoT network might range from the tens, such as in health care and smart HANs, to the thousands in industrial automation and agriculture, all relying on (typically) software applications. Hence, efficient security must be considered when designing security for such networks. The IoT CoAP is an application-layer protocol and was predicted to

become the one most commonly used of all application protocols [4]. The CoAP is a new version of the Hypertext Transfer Protocol (HTTP) and runs over the User Datagram Protocol (UDP) for transactions. In contrast, the HTTP runs over the Transmission Control Protocol (TCP) and is inappropriate for use in constrained environments [5], whereas the CoAP is designed to be lightweight and to support resource-constrained environments [6].

Security is important for wireless devices as they communicate in an insecure channel. For example, the objects (the sensors) of a smart IoT application (AMI), such as an electronic heater or an air-conditioning unit, will connect to a server for information regarding temperature resources. The server is responsible for the information provided to the sensor and each sensor might identify a certain condition for notifying a temperature reading [6]. Therefore, Datagram Transport Layer Security (DTLS) is proposed for use with the CoAP and to be responsible for data encryption and integrity protection, key management, and authentication. Similar to the Hypertext Transfer Protocol Secure (HTTPS), the CoAP with DTLS is termed a secure CoAP (CoAPs) [7]. However, DTLS is a resource-heavy protocol for IoT devices, which are restricted in terms of time and energy consumption, because it requires many message exchanges to initiate a secure session [8].

In this chapter, a new secure CoAP is proposed based on a HAN of a smart grid application and tested using Contiki OS that is included in Cooja simulation. As DTLS is inefficient for use with IoT networks, our proposed scheme is to secure the CoAP without using DTLS. We provide data integrity and authentication for a HAN application using the CoAP, as these are the most important security issues for this kind of application. It is important to provide integrity and authentication, rather than confidentiality, since it is not important if someone obtains the room temperature inside a home; however, it is crucial not to allow the manipulation of the temperature or to take the place of controlling the air-conditioning system.

In this chapter, the CoAP is shown to be secured using two different schemes: the secure hash algorithm (SHA)-256 and the Advanced Encryption Standard (AES). Both schemes are tested and compared in terms of their communication duration and energy consumption.

The main contributions of this work are as follows:

- A new security approach architecture has been proposed to secure the data exchange using lightweight algorithm such the AES-128.
- The new scheme provides integrity and authentication security mechanisms for the integrated security AES/CoAP.
- The new scheme executed in Contiki to evaluate its performance in terms of communication duration and energy consumption.

The rest of the chapter is organised as follows. Section 3.3 discusses the CoAP properties. Section 3.4 outline the cryptography algorithms that used to secure the communication of the IoT network. Section 3.5 proposes a new stack for the CoAP without using a DTLS layer. Our implementation, network setup, and proposed schemes are outlined in Sections 3.6 and 3.7. Section 3.8 discusses the evaluation results. Section 3.9 concludes the chapter.

3.3 The Constrained Application Protocol (CoAP)

CoAP remains a lightweight network communication protocol designed from the beginning to support the transmission of sensor data in Low power and lossy network (LLN). The protocol is a version of the HTTP and was created to match the IoT requirements for low overhead and multicast support. The CoAP depends on REST (i.e., representational state transfer), a principle adopted from the HTTP, and is embedded in the UDP for transactions. The initial reason for developing this protocol was to match the high demands of the IoT and the need for a low-rate and light protocol [6], [9]. Overall, the main features of the CoAP are that it supports machine-to-machine (M2M) requirements in constrained environments, UDP binding with optional support of unicast and multicast requests, asynchronous message exchanges, low header overhead and parsing complexity, the Universal Resource Identifier (URI) and content type, and has simple proxy and caching capabilities [6]. As shown in figure 3.2, the CoAP architecture is split into two layers: the message layer and the request/response layer. The message layer is responsible for controlling message exchange over the UDP between two endpoints. The CoAP uses a short fixed-length binary header and components and messages are encoded in a simply binary format.

The CoAP header and message format is shown in table 3.1. The first row depicts the 4-byte CoAP fixed header, which consists of the version field (2 bits), the message type field (2 bits), the token length field (4 bits), the code field (8 bits), and the message ID (16 bits). The token permits a CoAP entity to do request/response matching, and the message ID field can enable duplicate and optional dependency. Each option instance within a CoAP message includes the option number, the option value length, and the option value itself. CoAP options are used to support application-layer mechanisms and more options can be added in the future to support new functionality [10].

Table 3-1: CoAP message format

0		1				2				3														
O		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	6	7	8	9	0	1	
Ver	T	TKL			Code			MessageID																
Token (If any, TKL Bytes) ...																								
Options (if any)																								
Payload																								

3.4 Cryptography Algorithms

To add a layer of protection to network-shared data, we propose using two distinct cryptographic algorithms: the AES and the SHA-256. The AES, a symmetric cryptography algorithm, encrypts and decrypts data using a single key. This feature makes AES encryption more efficient than asymmetric cryptography in contexts with limited resources. AES encryption tries to establish the authenticity of data and to demonstrate that they were not altered or transformed during transmission. The AES is a security algorithm that is commonly used for a range of applications due to the high level of security it provides and its unique mechanism for encrypting and decrypting data [11]. The AES keys employed in this investigation are 128 bits in length. The number of processes required for encryption and decryption are determined by the size of a block cipher and the key. Figure 3.1 demonstrates that the encryption procedure of the AES

algorithm contains four types of byte transformation: SubBytes, ShiftRows, MixColumns, and AddRoundKey [12].

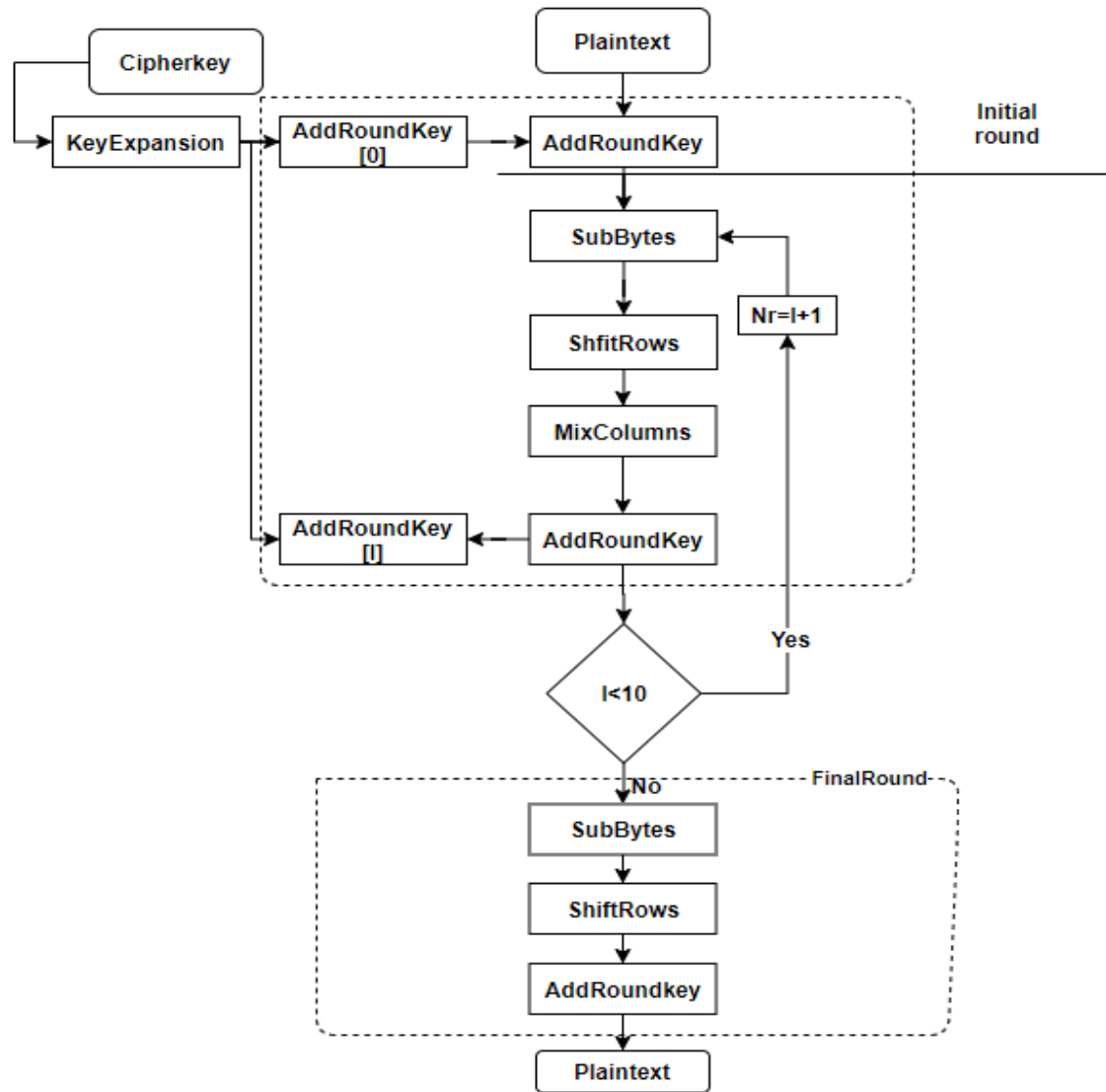


Figure 3-1: AES algorithm process

The SHA-256 makes it hard to recover the original data from message digests created by the hash algorithm. The National Institute of Standards and Technology (NIST) developed the SHA-256 in 2002. The SHA-256 algorithm received 264-bit input data and generates output length of 256-bit message digest or hash value. The hashing algorithm employs multiple logical operations, including AND, bitwise OR, eXclusive OR (XOR), SHIFT right, and ROTATE right [18]. The data are hashed by first providing bits so that

the bits achieve a total length of 256 bits, and then splitting the bits into blocks of 256 bits [13].

3.5 Architecture of the Proposed Work vs. Basic Architecture

Most CoAP implementations secure the data exchange among physical objects (e.g., sensors and smart devices) that are relayed on DTLS. The CoAP stack presents a separate layer for the DTLS to provide security features for exchanging data. The drawback of DTLS is that it increases the cost of communication and computation. As shown in figure 3.2, instead of using the DTLS layer, our proposed mechanism uses an authentication scheme in order to secure the data communication between the sensor and the server.

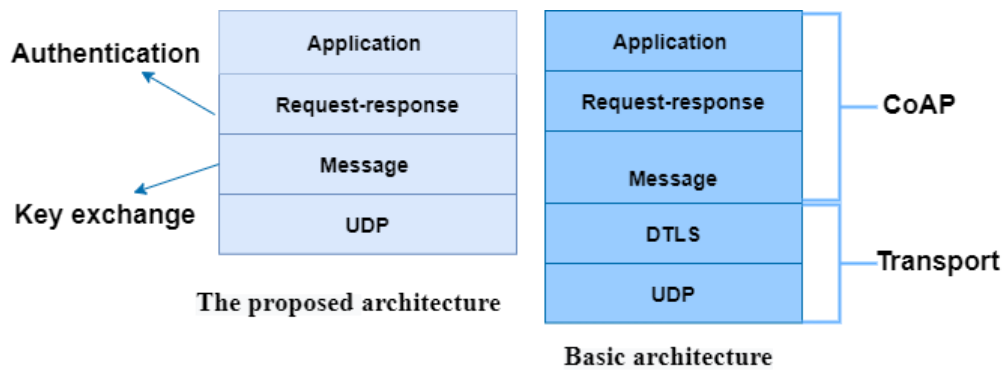


Figure 3-2: CoAP with our proposed architecture vs. Basic CoAP-DTLS stack

3.6 System Model and Network design

We implemented our scenario in the open-source Contiki operating system (OS) and evaluated it using wireless sensor network motes, WiSMote and Zolertia z1 motes microcontrollers provided by Contiki, (See Appendix A). WiSMote was employed based on its MSP430-gcc version 4.7.0, which uses 20-bit registers and a 20-bit address space. The Z1 mote has a second-generation MSP430F2617 low-power microcontroller with a 16-bit RISC CPU @16MHz clock speed, 8KB RAM, and 92KB Flash memory. Includes the CC2420 transceiver, which operates at 2.4GHz and 250Kbps. Z1 hardware ensures efficiency, resilience, and reduced energy costs. Z1 has built-in digital sensors: an accelerometer and a temperature sensor. As shown in figure 3.3, the network consists of several sensor nodes that simulate a HAN of AMI application. These sensors are connected to the CoAP server via the border router.

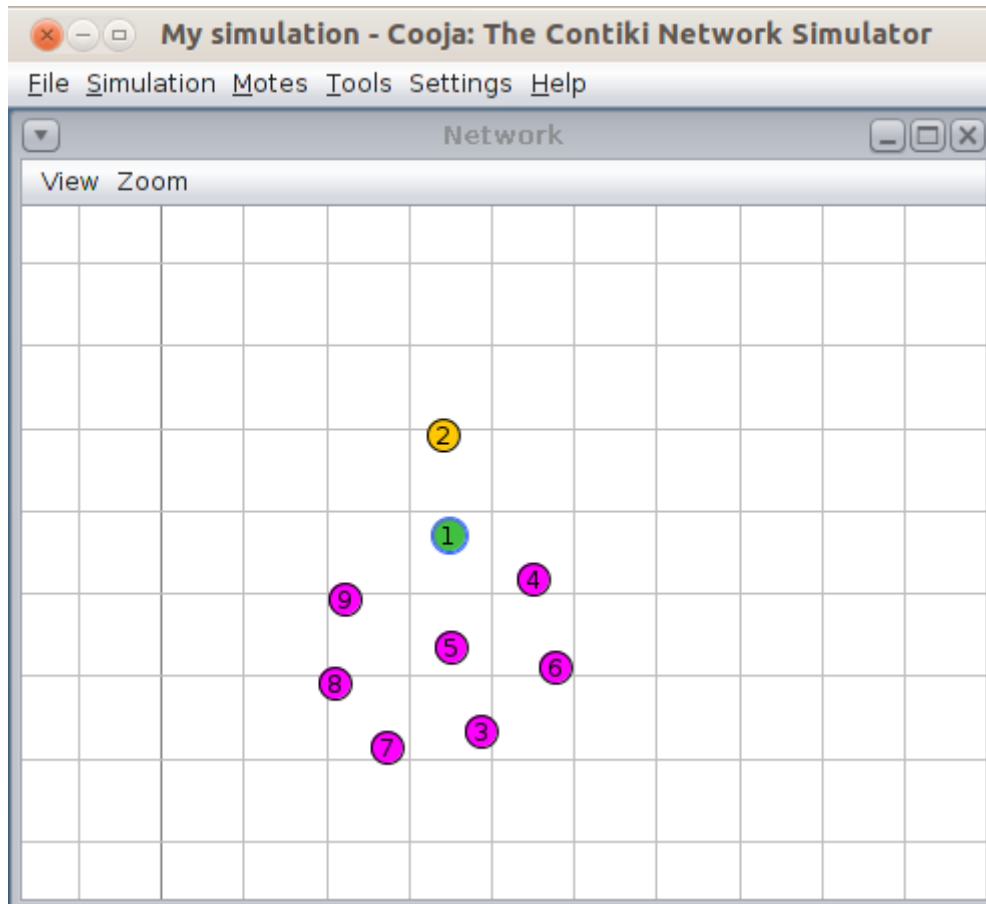


Figure 3-3: IoT network distributed sensors based

Home area network sensors ●

Border router ●

Server ●

We modified the CoAP implementation in Contiki to enable the protocol to connect with the integration module of the AES algorithm and the SHA-256. All the outgoing messages are protected by the AES algorithm and the SHA-256. The focus of our scheme is to add integrity and authentication to the CoAP by using lightweight algorithms, such as the AES, for sharing a symmetric key with the use of the SHA-256. We evaluated the performance of the implemented scenarios in terms of communication duration and energy consumption. As mentioned earlier, instead of using a separate CoAP-DTLS channel, our proposed scenario removes one channel and integrates the 128-bit CoAP-AES and SHA-256 algorithms.

3.6.1 Mechanism

The proposed scheme is implemented between a wireless sensor node (the sensor) and the server. This is illustrated in figure 3.3. In the scheme proposed in this scenario, both the sensor and the server verify identities after communicating with each other in order to avoid a replay attack. The sensors and the server then agree on a security mechanism to secure the CoAP. The data transmission will be protected by both the SHA-256 and the AES. The proposed mechanism for authenticating the data consists of a two-way authentication handshake between the sensors and the server. At this point, the process of creating a data encryption system that uses the AES encryption algorithm for the application is implemented. The encryption process in the constructed apps is a system procedure that encrypts plaintext data. If the data are read by the CoAP server, it will convert both the key and the plaintext into a bit array before sending it to the encryption operation. In the process of data encryption, the AES algorithm generates the SHA-256 key security. The results are then in the form of cypher text, which can be delivered to chat applications, such as WhatsApp, in a more secure format. Figure 3.4 depicts the procedure for encrypting and sending files.

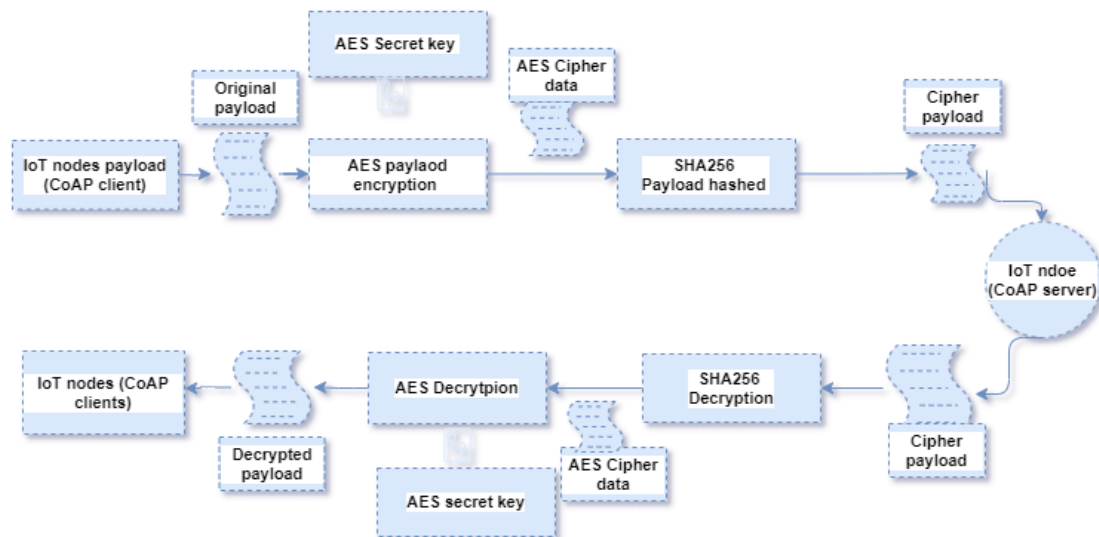


Figure 3-4: AES-SHA-256 encryption/decryption mechanism

3.6.2 Procedure

We added a security feature to that CoAP that provides authentication and integrity to secure the protocol against replay and malicious node attacks. Our proposed scheme initiates a secure connection for exchanging messages between the sensor and the server (see figure 3.5). Figure 3.5 shows the process of the handshake scheme among the sensors and the server. The sensors and server communicate with each other to exchange the key. The sensor sends an encrypted message with a symmetric key to the server. The symmetric key is only known to the sensor and the server. The server receives the message request from the sensor and verifies it using the sensor unique ID and the pre-shared key. Then, the server verifies the encrypted message to confirm that it matches and then the server replies with an acknowledgement (ACK) message and the sensor receives a response. If it does not match, the server does not respond and discards the message. The SHA-256-based authentication is as follows. When the server receives the sensor's message, the server will calculate the hash value of the message. If the calculated hash value of the message matches the sent hash value, the server responds with an ACK message. Otherwise, the server will discard the message. The process of initiating secure communication is summarised below.

Definition 1 (Session Initiation). *During the registration phase, a gateway provides a unique and fixed-bit-length ID, and a pre-shared secret key to the sensors separately. Both server and sensors must dynamically generate the pre-shared secret key and session key for each communication process using the hash value (hvalue) and pre-shared secret values.*

Definition 2 (Server Challenge). *After receiving the CoAP request message from the sensors, the gateway compares the hvalue and secret value of the corresponding sensor node. If not match, the gateway believes it is an authorised sensor device.*

Definition 3 (Smart Sensor Response and Challenge). *The proposed authentication scheme authenticates the sensors on both sides using the AES and SHA-256 algorithms.*

At this stage, a smart sensor generates encrypted data. An XOR operation is performed on the message using the AES.

Definition 4 (Server Response). *The gateway node needs to decipher the encrypted data to retrieve the pseudonym ID (P-IDi) value and generate the session key dynamically.*

Definition 5 (Secure Communication in the IoT). *The AES, SHA-256 integrated with CoAP (DF-CoAP) exploit several messages to secure the communication in the distributed sensors with the server.*

Table 3-2: Algorithm 1. Secure Authentication and Communication

```

1: Session Initiation:
(a) Each Sensor  $N_i$  is provided with a unique P-IDi, hvalue,  $\psi_i$ 
(a) A gateway  $\delta$  is provided with all P-IDi, hvalue,  $\psi_i$ , stored in an array  $A[ ][ ]$ 
2: Step 1 [Session Initiation]: [Input: (P-IDi, , hvalue,  $\psi_i$ )]
3: for  $i = 1: N$  do (Nested For loop generates a three-column server table)
4: for  $j = 1: 3$  do
5: input ( $A[i][j][k]$ ) (P-IDi,  $\psi_i$ , hvalue of  $N_i$  are stored in the array)
6: end for
7: end for
8:  $N_i$  sends a CON message containing P-IDi, hvalue,  $\psi_i$ , in the payload to  $\delta$ 
9: Step 2 [Server Challenge]:  $\delta$  retrieves P-IDi, hvalue,  $\psi_i$ , to authenticate the received CON message
10: if P-IDi, hvalue,  $\psi_i$ ,  $= A[i][j][k]$  then a node  $i$  is authenticated successfully
11: Session Created 2.01
12: Step 3 [Sensor Response and Challenge]:  $\delta$  responds with an encrypted payload, with P-ID
13: else
14:  $N_i$  Unauthorised 4.01
15: end if
16: Step 4 [Sensor Response & Challenge]:  $N_i$  deciphers challenge and responds with an encrypted payload, after verifying the  $\delta$  using hvalue and  $\psi_i$ 
17: Step 7:  $\delta$  responds as Sensor Authenticated
    [Data Exchange]: Mutual data exchange between  $\delta$  and  $N_i$  take place.
18: else
18: [Access Denied]- $N_i$  Unauthenticated
19: end if

```

3.6.3 Performance metrics

We conducted simulations to evaluate the following metrics with the proposed security algorithms AES and SHA-256 in DF-CoAP on varying the number of smart sensors and the data sizes. The time complexity of the proposed schemes was estimated using the

metric of handshake duration and communication. Handshake duration is defined as the sum of the time taken by two round-trip messages, which includes the session initiation requests and the challenge and response between the smart sensor and the server. The smart sensor acknowledges the session initiation request through a smart sensor response, and the server acknowledges the smart sensor response and challenge through a server response. Time complexity, $Time_{Comp}$, is computed at the smart sensor end using the following equation:

$$Time_{Comp} = Hd + T_{Communication} \quad (1)$$

In the above equation, Hd is estimated using the summation of the round-trip time taken by the session initiation request, the round-trip time taken by the smart sensor response and request, and the processing time of the request at the smart sensor. CoAP messages are exchanged over a UDP socket. Each sensor keeps track of the transmitted CON (Confirmable) message request through the network. When the sensor receives an ACK message from the server, the message exchange is considered successful. Thus, a sensor node can send and receive secure CoAP messages using the proposed security scheme.

Communication Duration: Communication duration is the time a smart sensor node takes to deliver a data packet successfully. This includes the handshake process, mutual authentication, and secure communication with the server.

Communication Duration (milliseconds)

$$= \left(\frac{\text{Total Number of Transmitted Packets}}{\text{Simulation Time} * \text{Node Count}} \right) * 1000 \quad (2)$$

Energy Consumption: This is the amount of joules (J) consumed by a smart sensor to deliver data packets to the gateway.

Energy Consumption (Micro Joules)

$$= \left(\frac{\text{Total amount of consumed energy in the network over simulation time}}{\text{Node Count}} \right) * 1,000,000 \quad (3)$$

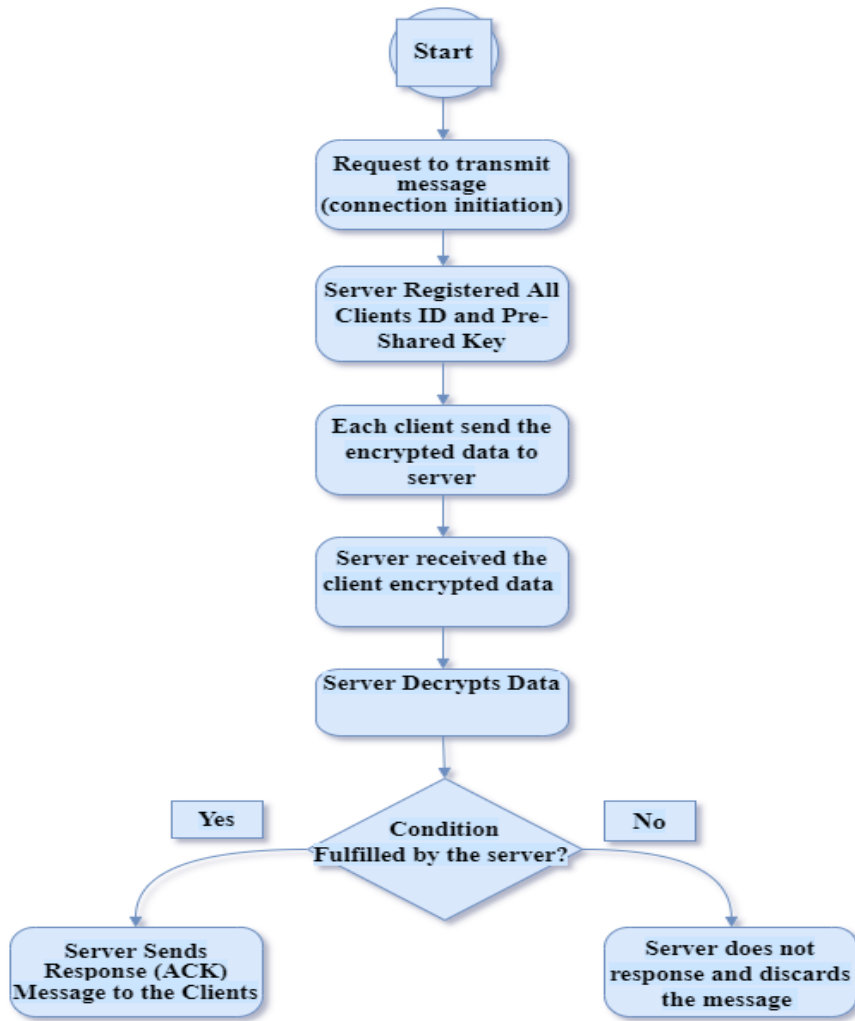


Figure 3-5: Flowchart of the proposed AES-based authentication

3.7 Performance and Experimental Results

Our experimental study is aimed at assessing the impact of adding security to critical resources on computational time and energy consumption. The use of such resources has a direct impact on essential aspects such as the rate at which sensing devices are able to communicate or the lifetime of the sensing applications. Our run-time performance evaluation is based on the impact of the SHA-256 and AES algorithms on a sensor node's time and energy consumption. We measured the time and energy consumption for processing the CoAP request and response messages. The measurement starts when the sensor sends the request and stops after the response is received and processed by the server. Table 3.2 outlines the impact of the proposed security schemes on performance in terms of the cost complexity, including communication duration and energy consumption.

3.8 Results and Discussion

3.8.1 Communication duration

Figure 3.6 shows the results concerning communication duration for both algorithm schemes (i.e., the AES-128 and SHA-256). The communication duration of AES increases slightly with increases in the data size, while the number of request-response messages of the SHA256 algorithm has a significant impact on the communication duration. It is clear that the AES response time is considerably less than that of the SHA-256, and the payload length has a moderate impact on the response time. For instance, with a data size of 16 bytes or under, the AES/CoAP encryption schemes successfully transmit the data packets in 3 seconds. However, the basic SHA-256/CoAP scheme delivers the packets in 1851 msec with a data size of 4 bytes and 2866 msec for 128 bytes. Figure 3.6 shows that the AES communication duration increases slightly with the increase in data size. At the same time, the number of request-response messages for the SHA-256 algorithm significantly impacts the communication duration by 14202 msec in 4 bytes and 12334 msec in 128 bits, respectively. Overall, the AES-128 bit is more efficient in providing security and the best performance compared to SHA256.

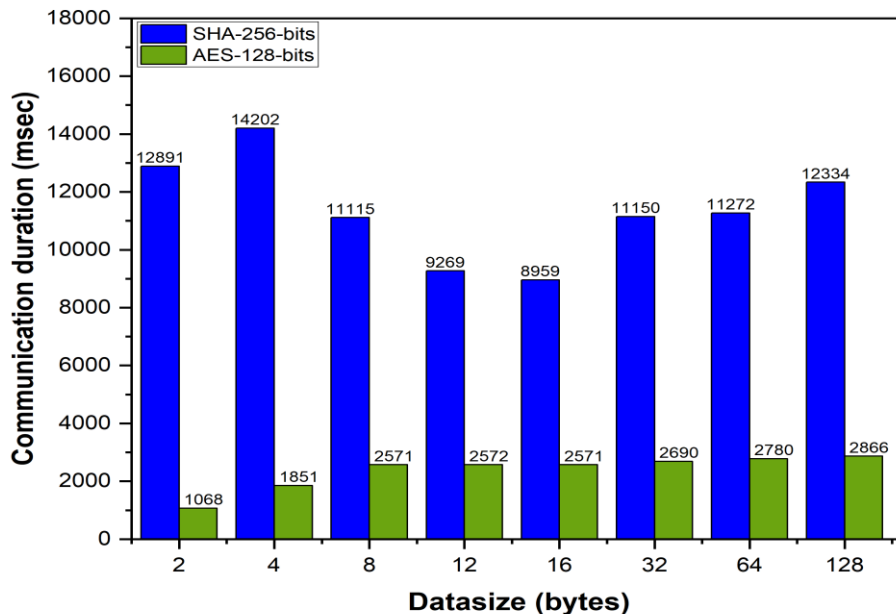


Figure 3.6: Communication duration for different payload sizes

3.8.2 Energy Consumption

In our evaluation, we focus on the energy consumption of the sensor node. Contiki has a module called Energest, which estimates the power consumption, as shown in figure 3.8. This module gives the usage time of the: (i) CPU, (ii) transmission and listening communication, and (iii) low power mode (LPM) [12]. Figure 3.7 shows the energy consumption performance for the AES-128 and SHA-256 schemes on the CoAP for data sizes 2 to 128 bytes. The AES energy consumption increases slightly in accordance with the data size. For instance, the AES consumes the same energy with data sizes of 8 to 16 at 1263 microjoules. In contrast, the number of request-response messages of the SHA-256 has a significant impact on the energy consumption among the different data sizes. For example, there was a significant increase with the increase in data size, with 2 bytes of the energy consumed by 979 microjoules and 1360 microjoules for 128 bytes. The overall energy consumed is less than for the AES. The AES consumes more energy than the SHA-256 for data sizes 2 to 32 due to the AES encryption and decryption.

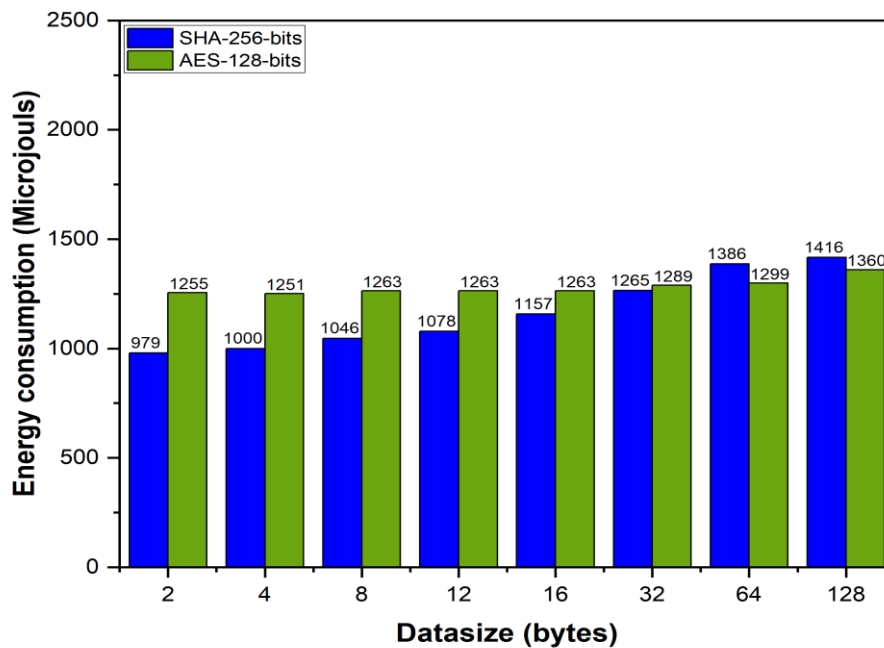


Figure 3.7: Energy consumption for different plaintext sizes

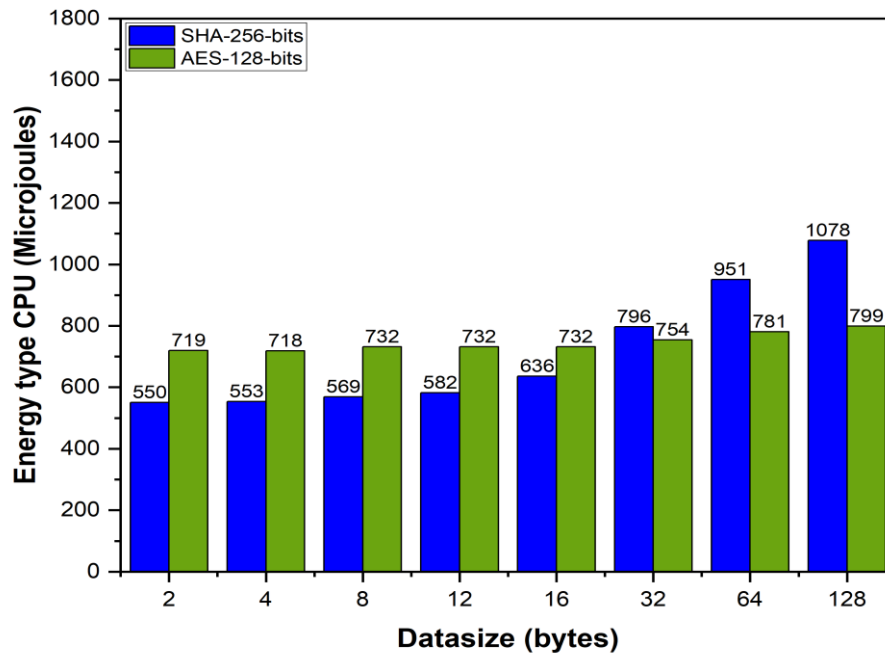


Figure 3.8: Energest type CPU

Table 3-3: Average communication duration for data transmission during the request and response messages, the average energy consumption, and the Energest type CPU for different payload sizes from 2 to 16 bytes

Data size (Bytes)	SHA-256			AES		
	Average response time (Milliseconds)	Average energy consumption (Microjoules)	Energy type CPU (Microjoules)	Average response time (Milliseconds)	Average energy consumption (Microjoules)	Energy type CPU (Microjoules)
2	12891	979.08	550.35	1068	1255.2	719.06
4	14202	999.65	553.21	1851	1250.83	718.04
8	11115	1046.24	569.17	2571	1263.47	732.06
12	9269	1078.48	582.28	2572	1263.48	732.07
16	8959	1157.24	635.56	2571	1263.47	732.06
32	11150	1264.65	796.46	2690	1289.15	754.30
64	11272	1386.45	950.76	2780	1298.60	780.66
128	12334	1416.20	1078.18	2866	1360.45	798.85

3.9 Chapter Summary

The real-world deployment of the CoAP enables devices and servers to communicate with each other, which is why the protocol requires robust security solutions. Our proposed schemes use a pre-shared symmetric key between the sensor and server to verify the identities of the sensors. This chapter presents an authentication and integrity security mechanism for securing communication through the CoAP protocol. The proposed scheme provides authentication and integrity security for communication using AES-128 and SHA-256. Furthermore, we addressed the impact of our security schemes in terms of cost computation, Communication duration and energy consumption. The results show that the AES performed better and required less communication duration and energy consumption than the SHA-256. Compared to SHA256, using AES-128 bit to secure smart-home applications significantly improves security and performance. The next chapter will be an extension to this chapter by addressing some limitation, including comparing the results with basic DTLS performance for various payload data sizes. A lightweight approach will be built based on the proposed architecture by optimising DTLS-AES with the CoAP. The nonce mechanism will also be implemented to reduce threats, such as replay attacks.

3.10 References

1. Banerjee U., Juvekar C., Fuller SH., Chandrakasan AP. EeDTLS: Energy-efficient datagram transport layer security for the internet of things. 2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings. 2017; 2018-Janua: 1–6. Available at: DOI:10.1109/GLOCOM.2017.8255053
2. Frustaci M., Pace P., Aloï G., Fortino G. Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of Things Journal. IEEE; 2018; 5(4): 2483–2495. Available at: DOI:10.1109/JIOT.2017.2767291
3. Alwarafy A., Al-Thelaya KA., Abdallah M., Schneider J., Hamdi M. A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things. IEEE Internet of Things Journal. 2021; 8(6): 4004–4022. Available at: DOI:10.1109/JIOT.2020.3015432
4. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings. 2017; Available at: DOI:10.1109/SysEng.2017.8088251
5. Kothmayr T., Schmitt C., Hu W., Brünig M., Carle G. DTLS based security and two-way authentication for the Internet of Things. Ad Hoc Networks. Elsevier B.V.; 2013; 11(8): 2710–2723. Available at: DOI:10.1016/j.adhoc.2013.05.003
6. Rahman RA., Shah B. Security analysis of IoT protocols: A focus in CoAP. 2016 3rd MEC International Conference on Big Data and Smart City, ICBDS C 2016. IEEE; 2016; : 172–178. Available at: DOI:10.1109/ICBDSC.2016.7460363
7. Raza S., Shafagh H., Hewage K., Hummen R., Voigt T. Lithe: Lightweight secure CoAP for the internet of things. IEEE Sensors Journal. 2013; 13(10): 3711–3720. Available at: DOI:10.1109/JSEN.2013.2277656
8. Haroon A., Akram S., Shah MA., Wahid A. E-lithe: A lightweight secure DTLS for IoT. IEEE Vehicular Technology Conference. 2018; 2017-Sept: 1–5. Available at: DOI:10.1109/VTCFall.2017.8288362
9. Chen X. Constrained Application Protocol for Internet of Things Abstract : Table

of Contents : Wireless and Mobile Networking. 2014; 857: 1–12. Available at: <http://www.cse.wustl.edu/~jain/cse574-14/index.html> Back

10. Jorge Granjal, Edmundo Monteiro and JSS. Application-Layer Security for the WoT: Extending CoAP to Support End-to-End Message Security for Internet-Integrated Sensing Applications. Tsaoussidis V, Kassler AJ, Koucheryavy Y, Mellouk A (eds.) Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. Available at: DOI:10.1007/978-3-642-38401-1
11. Panagiotou P., Sklavos N., Darra E., Zaharakis ID. Cryptographic system for data applications, in the context of internet of things. Microprocessors and Microsystems. Elsevier B.V.; 2020; 72: 102921. Available at: DOI:10.1016/j.micpro.2019.102921
12. Rothke B. A look at the Advanced Encryption Standard (AES). Information Security Management Handbook, Sixth Edition. 2007; : 1151–1158. Available at: DOI:10.1201/9781439833032.ch89
13. Publication F. Secure Hash Standard, Federal Inf. Process. National Institute of Standards and Technology (NIST); 2015. Available at: <http://csrc.nist.gov/publications/PubsFIPS.html#fips180-4>.

Chapter 4: Efficient Payload Encryption-based Authentication Scheme for IIoT based Sensor Networks

4.1 Abstract

The Internet of Things (IoT) connects billions of sensors to share and collect data at any time and place. Advanced metering infrastructure (AMI) is one of the most critical IoT applications. IoT supports AMI in collecting data from smart sensors and measuring and analysing abnormalities in the energy consumption pattern of those sensors. However, two-way communication distributed among sensors is sensitive and tends towards security and privacy issues. Before deploying distributed sensors, data confidentiality, privacy and message authentication for sensor devices and control messages are major security requirements. Several authentication and encryption protocols have been developed to provide confidentiality and integrity. However, the number of sensors in distributed systems, the presence of resource-constrained smart sensors, and the adaptability of the IoT communication protocols in sensors necessitate designing an efficient and lightweight security authentication scheme. A payload encryption-based optimisation scheme (PEOS) for lightweight authentication on distributed sensors is proposed in this chapter. PEOS integrates and optimises important features of Datagram Transport Layer Security (DTLS) in Constrained Application Protocol (CoAP) architecture instead of implementing the DTLS in a separate channel. The proposed work designs a payload encryption scheme and an optimised Advanced Encryption Standard (OP-AES). PEOS modifies the DTLS handshaking and retransmission processes using payload encryption and non-acknowledgement (NACK) messages, respectively. It also removes the duplicate features of the protocol version and sequence number without having an impact on the performance of the CoAP. Moreover, PEOS attempts to improve the CoAP over distributed sensors from the aspect of optimised AES operations, such as the parallel execution of S-Boxes in SubBytes and delayed MixColumns. The efficiency of PEOS authentication is evaluated on the Conitki OS using the Cooja simulator for lightweight security and authentication. The proposed scheme attained better throughput while minimising the message size overhead by 9% and 23% compared to the existing payload-based mutual authentication and basic CoAP/DTLS scheme in random network topologies with fewer than 50 nodes.

4.2 Introduction

The Internet of Things (IoT) has developed the conventional grid system into a modernised grid, called the smart grid. The traditional grid system established one-way communication from the grid to the user, whereas the smart grid implements bidirectional communication between the service provider and distributed sensors. The IoT serves as a bridging component between the sensing devices and the data plane in implementing the smart grid [1]. Figure 4.1 illustrates the growing awareness of the need for smart application cybersecurity, which necessitates implementing security and privacy assurance measures. The difficult work of meeting these demands has been delegated to the security and privacy level. As indicated in figure 4.1, each individual level of this architecture requires complete and effective security and privacy level. One of the main applications of the smart grid with two-way communication is advanced metering infrastructure (AMI) [2]. An important criterion in AMI application is to ensure that the security serves several entities, such as domestic or non-domestic users and governments [3], [4]. The absence of an effective security scheme makes communication between legal entities vulnerable to communication attacks [5], [6]. One of the major constraints in providing security for the IoT protocols used in sensor networks is the restricted resource availability of smart sensors.

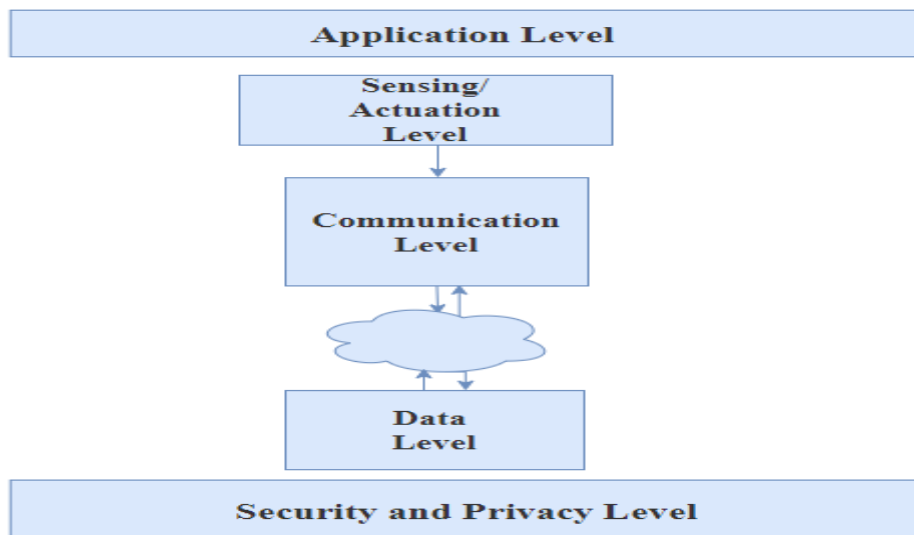


Figure 4-1: Block diagram architecture of smart grid applications

The Constrained Application Protocol (CoAP) is designed for resource-constrained sensors in the IoT application layer [7]. Datagram Transport Layer Security (DTLS) is

recommended to make the CoAP secure [8]. DTLS offers authentication, key exchange, and the protection of communication between legal entities. However, DTLS is not specifically designed for IoT applications, especially the smart grid. Therefore, there are several problems when connecting it with the CoAP/DTLS over AMI directly. The CoAP/DTLS has to exchange six flight handshake messages between sensors. It fragments data packets into 127 bytes, but this leads to data loss and delays in communication [8], [9]. The small packet size increases the number of packets, leading to network traffic congestion due to the need for frequent handshaking and packet loss. Another issue associated with the DTLS protocol is the likelihood of transmitting several Hello messages to the server, which could lead to denial-of-service (DoS) attacks against the server. The possibility of a DoS attack in CoAP/DTLS architecture increases bandwidth usage and the allocation of resources for malicious Hello messages. Moreover, the periodic request message size in the CoAP/DTLS is 32 bytes. The 32 bytes of the packet header and the energy reading information in AMI tend to have large packet sizes. If congestion occurs during the communication, the gateway may receive a delayed packet or only a few packet fragments. In such a case, the gateway requests that the smart sensors retransmit the packets. This contributes to communication delays, unnecessary packet retransmission, network collisions, and high energy consumption.

Thus, the work proposed in this chapter plans to integrate the important features of DTLS into CoAP instead of using DTLS in a separate channel. The DTLS feature is integrated with the CoAP architecture as the DF-CoAP. Malicious nodes could, however, misuse the DF-CoAP. Moreover, although AMI sensors expect complete anonymity for their information, AMI is a critical structure and does not require complete anonymity to be provided. For instance, law authorities must be able to track users who attack a smart grid. Thus, only conditional privacy preservation and concealment of the identity information of a smart sensor is desirable in the sensor network. After considering the issues above in respect of AMI, providing a lightweight security and authentication scheme is one of the research focuses.

A widely used cryptography technique to provide end-to-end security in the CoAP is the Advanced Encryption Standard (AES). The AES, with a 128-byte key size as a basic cipher suite of DTLS, helps solve the issues of packet fragmentation, loss, and delay in

the IoT environment. However, there is a limitation in applying the AES technique directly for secure data communication in DF-CoAP architecture for resource-constrained smart sensors. The AES generates different data using the original key, but the possibility of deriving the relationship between those words reduces the security level of AES. Furthermore, biased inputs in the AES keyspace create the possibility of malicious sensors observing differences between the data in the ciphertext. Thus, the proposed work plans to optimise security using a lightweight and strong cryptography technique, the AES, and, at the same time, minimise its complexity.

From the above discussion, it can be seen that this chapter proposes a lightweight and secure authentication scheme for distributed sensor networks. The main contributions of this chapter include the following:

- The proposed work aims to integrate optimised DTLS features in a default CoAP (DF-CoAP) and implement a lightweight authentication mechanism with the DF-CoAP to ensure secure communication between authorised entities in a distributed sensor network.
- The proposed approach uses a payload encryption-based handshaking process to reduce the handshaking process from three to two round trips. This ensures a lightweight and secure authentication scheme for the application layer.
- By adding a NACK-based retransmission scheme to the DF-CoAP, the exact missing data in the flight is communicated to the sender with the received byte to reduce the communication load.
- The proposed work introduces a buffer size for intimating the exact size to the server and avoiding the buffer overflow in DTLS. This will optimise the performance of the DF-CoAP without affecting its security level.
- The proposed work reduces system complexity drastically by removing the duplicate features of the protocol version and sequence number component of the explicit nonce field from the packet header in the DF-CoAP and introducing the parallel execution of S-Boxes in the SubBytes of the AES.
- The proposed scheme introduces nonce and seed values to generate the secret and session keys dynamically, which significantly improves DTLS.

The remaining part of this chapter is organised as follows. Section 4.3 provides the related case study on the authentication schemes for a smart grid application. Section 4.4 provides the problem formulation, system (model) and attack model. Section 4.5 explains the proposed method with the figures and algorithm. Attacks and cost analysis outlines in section 4.6. Section 4.7 shows the performance results with detailed descriptions. Section 4.8 concludes the chapter.

4.3 Case Study: Authentication Schemes for a Smart Grid Application

Device authentication is a vital mechanism in verifying the entity identity of a node in a distributed sensor network and eliminates the impact of malicious behaviour. The IoT application-layer protocol, the CoAP, is used with DTLS for the reliable and secure negotiation of a session and for data authentication and communication. Although DTLS provides a complete security model for the CoAP, reducing energy consumption without compromising security is essential. AMI data are sensitive and, if the data are stolen, it can result in significant economic loss to both users and power providers. Thus, the data transmitted over distributed sensors must be encrypted before being forwarded. Existing studies have mostly applied either symmetric or asymmetric encryption schemes: the former must share a secret key in advance, but the latter does not need to do this. However, asymmetric encryption leads to more computational cost than symmetric encryption. Therefore, generating a shared key and encrypting data dynamically is a more effective method. Smart grid applications need to provide an effective authentication and security scheme in a lightweight pattern [10], [11]. Since smart sensors are low-energy devices, highly complex security schemes that require a lot of energy are not practicable.

One security protocol was developed for the smart grid using double auction mechanisms and homomorphic encryption [12]. This protocol provides authentication, security, and compatibility to smart grid technologies. It assigns a pseudo-identity to each consumer in the grid and ensures the anonymity of sensors during communication. Moreover, the Paillier cryptosystem is used for encrypting bids/asks. However, the homomorphic encryption schemes in [12] and [13] generate a lengthier cipher suite than plaintext, resulting in a large delay for encryption and decryption. A lightweight authentication and key agreement is suggested in [14] for a smart sensor network that exploits hybrid cryptography to provide security, i.e., elliptic-curve cryptography (ECC) and a symmetric

key. Before applying cryptography, both the gateway and the smart sensors are mutually authenticated. The hybrid cryptography-based security scheme protects against many attacks, but it lacks focus on many privacy features, such as unforgeability and undetectability. Several ECC-based authentication schemes [15], [16] have been proposed for sensor networks but they do not satisfy all the security features. A self-certified key distribution scheme eliminates the necessity for certificate management overhead [15]. However, using small key sizes does not always ensure security.

A secure key agreement model for smart metering communications is developed in [17]. The model eliminates the necessity for a secure channel during the registration of entities and prevents the possibility of DoS attacks. However, the encrypted message is very large and increases the system complexity. The Password-based Anonymous Lightweight Key (PALK) agreement framework is developed and improved in [18] and [19], respectively. The framework was first developed using ECC and symmetric hash functions. The superficial ECC operation performed in PALK with the multiplication of two points over the curve leads to incorrect authentication. Thus, PALK was improved using block cipher-based encryption/decryption operations. An improved authentication protocol using ECC is developed in [18] for the smart grid environment and utilises the advantages of biometric information to improve the strength of ECC. However, the small keys sizes for the various security levels affected the performance of the authentication scheme.

A pairing scheme between smart sensors and a server is expensive and does not apply to smart sensors with low power. In [20], a certificateless two-party authenticated key agreement (CL2PAKA) scheme is suggested for smart grid application. The CL2PAKA does not need to perform pairing operations and implements only four scalar multiplication operations in ECC. The main disadvantage of certificateless authentication schemes [20], [21], however, is that the identity information cannot provide a public key for a long time. The identity of sensors alone is not sufficient to provide a public and secret key. In [22], a privacy-preserving architecture is suggested for the smart grid using a Q-learning-based optimised approach that exploits the cryptography technique to outsource multiregional electricity data securely. It implements three dynamic protocols to perform primary operations in Q-learning: Q-value updating, Q-learning training, and knowledge replay with encrypted packet information. However, it requires a lot of time

to reach an optimal value, which is less applicable to resource-constrained smart grid applications.

4.4 Preliminaries

The AMI network is responsible for providing data on the quality of the power and the quantity of the resource utilisation on the smart sensor side. The smart sensors in AMI are embedded, with low computational and storage capabilities. Moreover, they communicate with each other and the server using low data rates and lossy radio channels. The AMI network needs a redesigned and appropriate CoAP/DTLS architecture to offer lightweight, low-latency, and highly reliable communication from both the meters to the gateway and the gateway to the meters. However, the unreliable communication links in AMI still suffer from adversarial activities. The application-layer protocol deployed must cope with the DTLS security and overhead to satisfy the AMI low-latency and high-reliability requirements. Figure 4.2 provides a general overview of smart grid AMI. The figure shows an AMI network that is implemented using a wireless sensor network (WSN). Wireless sensors are embedded into smart meters and work in industrial and scientific bands. A traffic concentrator receives traffic from the power meters and acts as a communication gateway between the WSN and the server.

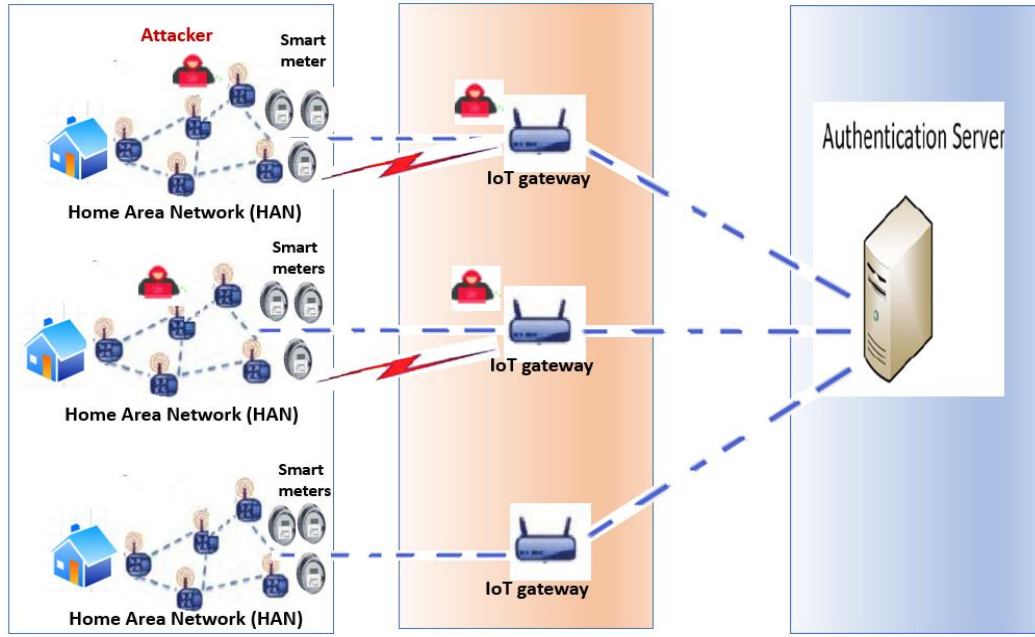


Figure 4-2: System model architecture of the proposed scenario

4.4.1 Problem Statement

The design of DTLS is not suitable for resource-constrained smart sensors due to the expensive handshaking and lengthy cipher suite agreement process. Moreover, complex handshake processes are not always protected from attacks. Sending a handshake request message in DTLS to low-memory and low-capacity sensors resembles a DoS attack in the IoT. An attacker could send several Hello messages to the server and this scenario would cause a DoS attack against the server. The loss or delay of a single fragment in DTLS forces the whole flight to be retransmitted. In DTLS, the recording layer incurs 13 bytes of overhead. Some duplicate parameters are maintained in the recording layer, such as the protocol version and sequence number. DTLS allows the server to specify the maximum fragment size of smart sensors, but the server cannot specify the fragment size based on the available buffer size. Thus, the proposed work plans to solve the issues referred to above and improve DTLS while minimising its complexity.

4.4.2 Problem Formulation

To construct a CoAP security model based on DTLS architecture, the sensor output (y) can be formulated as

$$y = \text{DTLS}(s) - \text{DTLS}(o) \quad (1)$$

Where y represents the quality of the CoAP/DTLS architecture in terms of both security (s) and overhead (o). Basic DTLS performs a handshake between communicating sensors for a long time. The notation H_d below denotes the handshake duration:

$$H_d = T_{\text{session}} + T_{\text{challenge}} + \Xi_{\text{Processing}} \quad (2)$$

From the above equation, H_d is estimated using the summation of round-trip time taken by a session initiation request, T_{session} , the round-trip time spent by the smart sensor response and request, $T_{\text{challenge}}$, and the processing time of a request at the smart sensor, $\Xi_{\text{Processing}}$.

The processing time increases with the number of redundant features appended in the DTLS messages. This may exceed the response timeout value and initiate the handshaking process again. Moreover, the delay of a single fragment in DTLS forces the whole flight to be retransmitted. This is another factor behind the increase in DTLS (o).

$$\Xi_{\text{Processing}} \propto \text{DTLS(o)} \quad (3)$$

The handshake phase explores either a symmetric key- or asymmetric key-based cipher suite to ensure DTLS (s). The cipher suite selection makes a large difference to the value of y . If a symmetric key-based cipher suite is used, for example the AES, performance must be improved. However, sharing a secret key in advance and having to protect it against several attacks during communication is a notable limitation. In addition, the delay induced by the AES algorithm makes smart sensors complex and increases energy consumption. Thus, the proposed work designs a lightweight CoAP authentication scheme by appending the most important DTLS features and optimising the AES algorithm.

4.5 System Model

The system model of the distributed sensor consists of the server, δ , which is situated in the corner of the IoT network, and N numbers of smart sensors. The server δ is involved in collecting and analysing data from N , so is called the gateway. Based on the data collected, δ provides predictive information to the sensors. The N smart sensors are

located in different positions in the network to mimic the sensors. Two-way wireless communication has led to several potential vulnerabilities. For instance,

- (i) There is a possibility of unauthorised users compromising the smart sensors connected to the server.
- (ii) By injecting false data or implementing a DoS into AMI, malicious users could trip the energy supply to various locations from the smart grid.
- (iii) Smart home energy sensors need to send consumption usage periodically to δ , but malicious users could leak consumption data by eavesdropping on the communication channel, and this affects the consumer's privacy.

To avoid such attacks, the proposed lightweight CoAP authentication scheme involves the following processes. The below authentication scheme processes are also highlighted in an algorithm format in *table 4-1, Algorithm 1. Secure Authentication and Communication*.

Definition 1 (Session Initiation). *During the registration phase, a gateway, δ , provides a unique and fixed bit-length identity (ID) along with the pre-shared secret key (ψ) and seed value, Se_i , to N sensors separately. Both the server δ and N sensors are required to generate the ψ and session key, Y , dynamically for each communication process using pre-shared secret values. Thus, the gateway δ shares the ψ_i and Se with the corresponding sensor N_i via a secure channel. Although it is essential to provide an efficient security scheme for sensor network communication, law authorities need to track who performs malicious activities in a smart sensor. Thus, conditional privacy preservation, the concealment of the identity information of a smart sensor, is desirable. The randomly chosen pseudonym ID ($P-ID_i$) replaces the original sensor ID. The secret values, ψ_i , Se , and $P-ID_i$ are known to the consumer N_i and δ ; other users and attackers cannot obtain the real ID_i . Therefore, the proposed lightweight security scheme provides identity privacy without compromising communication security.*

Definition 2 (Server Challenge). *After receiving the CoAP request message from N_i , the δ compares the ψ_i value of the corresponding node, N_i . If the secret values match, the gateway, δ , is assured that N_i is an authorised sensor device, and it sends the server a challenge message with the nonce value 1 , and both entities, N_i and δ , derive the secret values dynamically.*

Definition 3 (Smart Sensor Response and Challenge). *The proposed lightweight authentication scheme authenticates the sensors on both sides using two nonce values, ψ and Υ .*

Dynamic ψ , $D\psi = \psi \parallel \text{Nonce 1} \parallel \text{Sei}$ for mutual authentication.

At this stage, the smart sensor generates an encrypted payload. An eXclusive OR (XOR) operation is performed on the message using $D\psi$. Here, the message consists of Nonce 2.

Definition 4 (Server Response). *The gateway node, δ , needs to decipher the encrypted payload to retrieve the nonce value 2 and generate the session key dynamically.*

Dynamic Υ , $D\Upsilon = \Upsilon \parallel \text{Nonce 2} \parallel \text{Sei}$ for secure communication.

Definition 5 (Secure and Optimised Communication in IoT). *The DTLS feature integrated with the CoAP (DF-CoAP) exploits six flight messages, leading to communication delay and energy inefficiency. Thus, the proposed work applies a payload-based mutual authentication (PbMA) scheme and successfully reduces the number of flight messages from six to four. During secure communication, the packet header removes the protocol version and the sequence number component of the explicit nonce field. Moreover, the acknowledgement (ACK) message includes the exact missing data in the current flight together with the right-most received byte.*

4.5.1 Overview of the Proposed Scheme

The proposed lightweight PEOS authentication scheme for the DF-CoAP verifies the identities of the sensors and the server in two-way communication. The proposed lightweight handshake mechanism, which includes only two round-trip message exchanges, is used for two-way authentication in the sensor network. As shown in figure 4.3, the challenge messages in the sensors and in the server for two-way authentication generate encrypted payloads with some optimised DTLS features. This assists in improving the performance and overhead of the payload-based authentication scheme. Moreover, the Advanced Encryption Standard-Counter with Cipher block chaining Message authentication code (AES-CCM) encryption scheme is optimised by choosing methods from design choices of AddRoundkey, SubBytes, ShiftRows, and MixColumns

and dynamically selecting a secret key using a nonce value, a seed value, and the secret master key.

Lightweight DF-CoAP for a Sensor Network: The DF-CoAP and AES-CCM provide flexibility in cipher suite usage and design choices. Using the DF-CoAP features, the payload-based four-way handshaking and missing data retransmission using NACK means that the DF-CoAP is sufficiently lightweight for sensor networks.

Payload Encryption-based Four-Way Handshaking with Unique Packet Features: This explores the encrypted PbMA scheme but the limited payload space in the sensors makes it unsuitable for larger payloads. Existing works have attempted to reduce the CoAP overhead using four-way authentication schemes. However, the nature of DTLS flight messages means that the exact missing data in the current flight is unknown to the sender and the loss or delay of a single fragment in DTLS forces the whole flight to be retransmitted. Thus, the proposed scheme aims at optimising the PbMA process using exact buffer size intimation and missing data retransmission schemes.

Optimised and Secure AES-CCM Cipher Suite for Distributed Sensors: Applying the existing AES-CCM for secure data communication in sensor networks is inadequate for resource-constrained sensors. Design choices exist for the processes used in AES-CCM. As per the features of smart sensors, selecting appropriate methods for such a cryptographic scheme improves the performance of the proposed lightweight DF-CoAP scheme.

Moreover, parallel execution of SubBytes and multiple small S-Boxes is effective in reducing the time delay and system complexity. A dynamically generated secret key in two-way authentication is used as the symmetric key for optimised AES-CCM. Thus, the proposed mutual authentication scheme reduces the round-trip time of CoAP smart sensor challenge messages without compromising the communication security in distributed sensors.

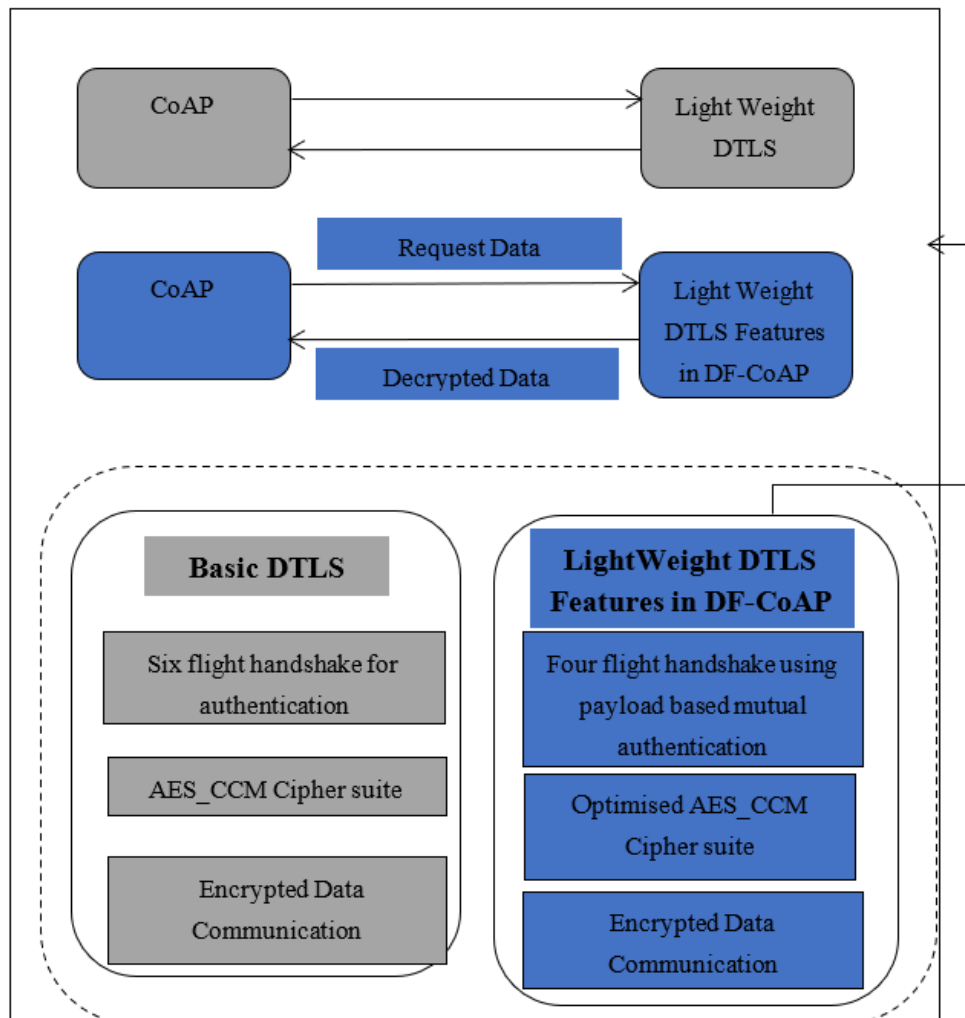


Figure 4-3: Block diagram of the proposed work

4.5.2 Lightweight DF-CoAP for Distributed Sensor Networks Using Payload Encryption-based Four-Way Handshaking

The proposed lightweight PEOS scheme in the DF-CoAP architecture provides authentication using the payload of messages exchanged between the sensors and the server. Both the sensor and the server challenge each other during the authentication process. The two-way authentication scheme uses four handshake messages with the improved DTLS features messages with an exact buffer size and missing data retransmission scheme using the NACK method.

As shown in figure 4.4, four steps are involved in the proposed work:

1. Session Initiation

2. Server Challenge
3. Smart Sensor Response and Challenge
4. Server Response and Secure Communication

Session Initiation and Server Challenge: Session initiation is a provisioning phase. It is a prerequisite offline phase, in which the sensor, N , shares a 128-bit AES pre-shared secret key (ψ) and a seed value, S_e , with the server, δ . The pre-shared secret values are known only to the corresponding sensors and the server. Each sensor has a unique identifier (P-ID), and the P-ID _{i} is used for identity verification on the server side. Upon successful verification, both parties communicate with each other to exchange the session key. It is assumed that the sensors used on the sensor side are tamper-safe, and they can alert the data by generating an alarm if an attacker tries to tamper with the sensor. Each sensor sends a request message to the server. To create a session, this request message acts as a confirmable message (CON). A challenge message is created at the server with a specific token for correlating the request with a corresponding challenge when the secret values are matched. The server challenge message consists of a nonce value 1, and it assists both entities, N_i and δ , in deriving the secret values dynamically.

$$\begin{aligned} \text{PAYLOAD}(< P - ID >, < NONCE 1 >) \\ &= \text{OP_AES} - \text{CCM} \{ D\psi, (\Psi_{\text{resultant}}) \} \end{aligned} \quad (4)$$

where

$$\Psi_{\text{resultant}} = \text{XOR}(\psi, \text{NONCE } 1, S_e) \quad (5)$$

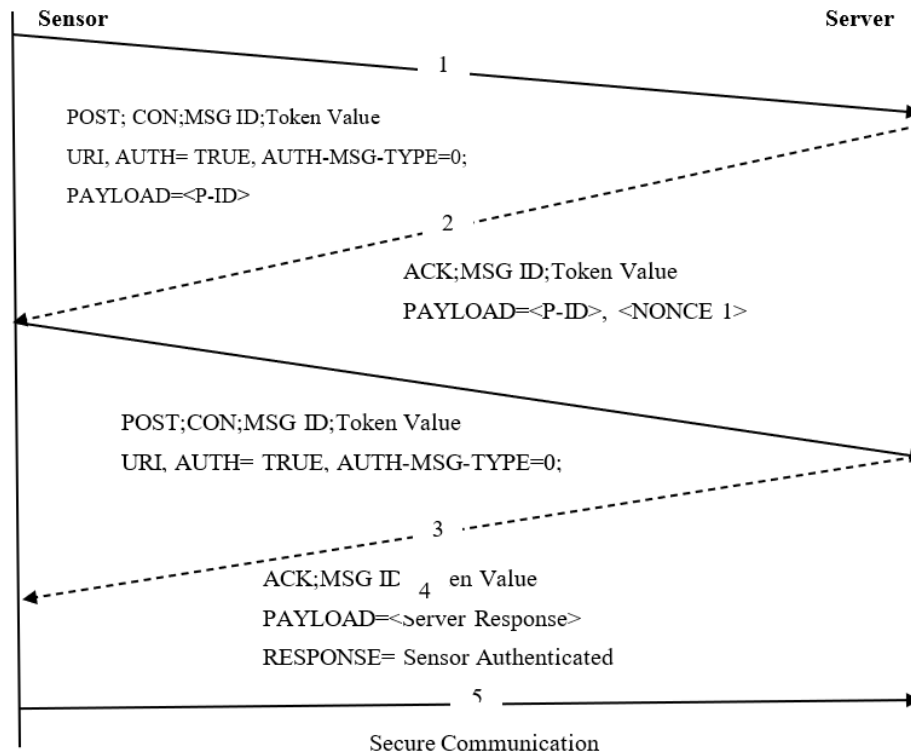


Figure 4-4: Two-way authentication scheme using payload encryption

Exact Buffer Size in Hello Message Transmission: The server specifies maximum fragment lengths for each smart sensor and allows data transmission within a specified fragment length. An undefined buffer size incurs buffer overflow and data loss. If a buffer with a fixed length overflows in a node, the data stored in the memory are overwritten. An attacker can use this to send frequent requests to the nodes and destroy the network. Moreover, the DTLS maximum fragment length mechanism is not symmetrical, since the server cannot state the exact buffer size of the nodes. Instead of a server, allowing resource-constrained sensors to advertise their maximum fragment length lowers the possibility of data loss and buffer overflow issues. The proposed work allows nodes to append the Exact Buffer Size (EBS) field in a Hello message to avoid this problem.

Smart Sensor Response and Challenge: The nonce value 2 is the message payload during a smart sensor response and challenge. Two options, i.e., Auth and Auth-Msg-Type, denote the type of operation executed on a resource at δ . During the second phase, the server, δ , obtains and verifies the object ID from the message payload. If it verifies the identity, the δ responds with an encrypted payload using the optimised AES-CCM

algorithm. It supports creating a new session with a response code of 2.01 (Created). This phase generates the dynamic secret key, $D\psi$, using the nonce value 1, ψ , and Se_i . Using $D\psi$, a payload with a nonce value 2 is created at the smart sensor side.

$$\begin{aligned} \text{PAYLOAD}(< P - ID >, < NONCE 1 >) \\ &= \text{OP_AES} - \text{CCM} \{ DY \{ DY, (\Psi_{\text{resultant 1}}) \} \} \end{aligned} \quad (6)$$

where

$$\Psi_{\text{resultant 1}} = \text{XOR}(\psi, \text{NONCE 2}, Se) \quad (7)$$

Although the DTLS overhead is not suitable for resource-constrained sensors (the DTLS record layer header appends 13 bytes of overhead), some of the fields carried in the header are unavoidable and other parameters are redundant but are included for backward-compatibility reasons. This overhead becomes substantial for resource-constrained networks. The DTLS feature integrated with CoAP architecture (DF-CoAP) header fields that are not strictly needed can be removed to reduce the burden. The proposed work removes the fields of the protocol version and the sequence number component of the explicit nonce field from the OP_AES-CCM layer. The second field is a duplicate of the sequence number used in the DF-CoAP and assists the proposed work to eliminate 8 bytes per record. An algorithm for CoAP message exchange using the proposed work is given in Algorithm 1.

Table 4-1: Algorithm 1. Secure Authentication and Communication

```

1: Session Initiation:
(a) Each Sensor  $N_i$  is provided with a unique P-IDI,  $\psi_i$ , and seed value,  $Se$ 
(a) A gateway  $\delta$  is provided with all P-IDI,  $\psi_i$ , and seed value,  $Se$  stored in an array  $A[ ][ ]$ 
2: Step 1 [Session Initiation]: [Input: (P-IDI,  $\psi_i$ , and seed value,  $Se$ )]
3: for  $i = 1: N$  do (Nested For loop generates a three-column server table)
4: for  $j = 1: 3$  do
5: input ( $A[i][j][k]$ ) (P-IDI,  $\psi_i$ , and seed value,  $Se$  of  $N_i$  are stored in the array)
6: end for
7: end for
8:  $N_i$  sends a CON message containing P-IDI,  $\psi_i$ , and  $Se$  in the payload to  $\delta$ 
9: Step 2 [Server Challenge]:  $\delta$  retrieves P-IDI,  $\psi_i$ , and  $Se$  to authenticate the received CON message
10: if P- IDi,  $\psi_i$ , and  $Se == A[i][j][k]$  then a node  $i$  is authenticated successfully
11: Session Created 2.01
12: Step 3 [Sensor Response and Challenge]:  $\delta$  responds with an encrypted payload, with P-ID and NONCE 1 value.
13: else
14:  $N_i$  Unauthorised 4.01
15: end if
16: Step 4 [Sensor Response & Challenge]:  $N_i$  deciphers challenge and responds with an encrypted payload, OP_AES-CCM {NONCE 2} after verifying the  $\delta$  using  $D\psi$ 
17: Step 5:  $N_i$  compares generated  $D\psi$  using NONCE 1 and retrieved  $D\psi$  from  $N_i$ 
18: if Both matches then
19:  $S$  is authenticated
20: Step 6 [Server Response]:  $\delta$  generates  $DY$  and verifies the sensor challenge
21: if Both matches then
22: [Access Granted]-  $N_i$  Authenticated
23: Step 7:  $\delta$  responds as Sensor Authenticated
24: else
25: [Access Denied]- $N_i$  Unauthenticated
26: end if
27: Step 8 [Data Exchange]: Mutual data exchange between  $\delta$  and  $N_i$  take place.
28: else
29:  $\delta$  is unauthenticated
30: end if

```

Server Response and Secure Communication: Finally, the server deciphers the encrypted payload using the same session key generated at the smart sensor side in the server response phase. The derived session key is used to implement secure communication between sensor entities. The communication between the server and the smart sensors should consider the EBS field to reduce buffer overrun and data loss.

Missing Data Retransmission using NACK without Encouraging DoS Attacks: CoAP data transactions use the observation model in figure 4.5. The CoAP is modelled by improving the 6LoWPAN fragmentation in the IoT [42]. Fragments result from fragmenting network layer packets. Fragmentation breaks packets into smaller pieces in an adaptation layer that allows IPv6 datagrams to meet the requirements of the IEEE 802.15.4. As per the proposed work, data fragmentation is also adopted and improved by the application-layer protocol. A single fragment loss or delay forces the DF-CoAP to retransmit the whole flight in resource-constrained sensors, as per DTLS. The proposed lightweight authentication utilises the NACK packet to avoid retransmission of the entire flight in the case of losing a single fragment. An ACK message is used to confirm the receipt of all flights in the DF-CoAP.

In the case of indicating that a packet has been lost, corrupted, or delayed and to resend it, the DF-CoAP sends a NACK report to the transmitter nodes. The DTLS handshake associates every fragment with a unique identifier. The proposed work utilises this identification in the NACK report and obtains the exact identity of the missing data in the current flight along with the right-most received byte. The nodes retransmit only the lost or delayed fragment with this identity information, instead of sending the whole flight. There are two fields in the NACK report: The Exact IDentity of the missing data in the Current Flight (EID_CF) and the Right-Most Received Flights (RM_RF). If the EID_CF is assigned as null, the retransmission does not occur on the consumer side. In this case, the RM_RF denotes the identity of entire flights. Otherwise, only a mentioned identity of a flight in EID_CF is retransmitted from the consumer node. Utilising NACK messages, the proposed lightweight authentication scheme avoids unnecessary retransmission of entire flight messages in response to losing a single fragment. Figure 4.5 compares the retransmissions of the default DTLS and the proposed scheme. Case 1 considers that a server cannot obtain the entire flight message due to the loss of the first fragment. After

sending the NACK with EID_CF = 1, and RM_RF = 2,3,4, the consumer node resends only fragment 1 of the in-flight message. However, the basic DF-CoAP resends an entire message if it does not receive an ACK message from the server. The second case differs from the first. The ACK message is lost due to network conditions. In this case, the smart sensor resends the packet as per the basic DTLS, but the proposed scheme does not resend the packet without receiving a NACK message.

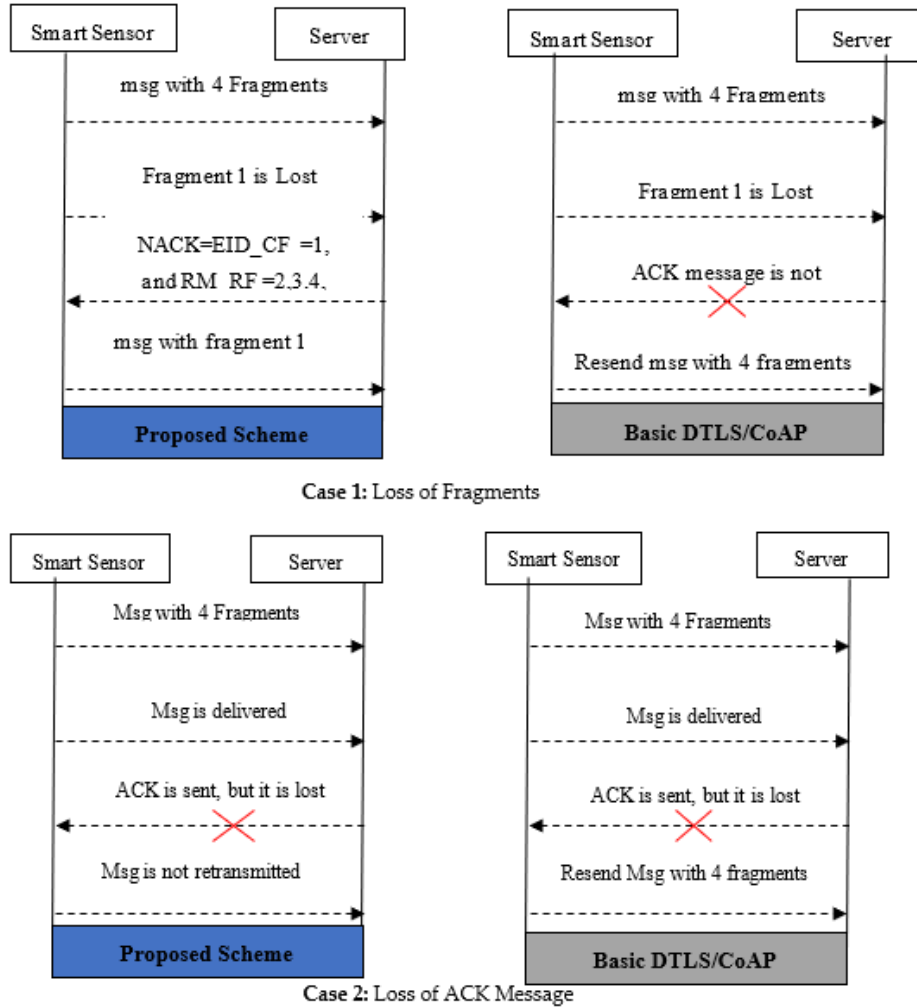


Figure 4-5: Communication between server and smart sensors

4.6 Attack and Cost Analysis

The proposed PEOS for lightweight authentication ensures secure communication when implementing the CoAP over the IoT. It can handle various attacks and is able to secure communication by mutual authentication between the server and the smart sensors. The

possibility of security provision against different attackers in IoT applications is discussed below.

Security against Replay Attacks: An attacker generally fails to change the messages between the smart sensors and the server in a strong cryptographic mechanism. However, an attacker could copy a valid message with a wireless communication protocol and resend it to the server. If an attacker node, A_i , traces a session initiation request for a legitimate IoT sensor device, it can later gain network access. A replay attack would fail to attack the proposed PEOS because all the entities would generate nonce values to prevent the attacker from sending the packets again. The nonce values are generated randomly and are varied over the communication. If a node replays an old packet, the server and the sensors can identify the attack packet as it will be using expired nonce values.

Security against Guessing Attacks: Another main type of network attack is a secret key-guessing attack. Legitimate smart sensors, which have access rights to a server, are compromised when they identify the identity and secret key of another legitimate smart sensor. Secret key-guessing attacks can be classified into brute force attacks and dictionary attacks. A brute force attack attempts to analyse every possible code, overheard message and secret key until the correct one is found. This type of attack takes a long time to complete. A dictionary attack explores a dictionary of common secret keys to identify the exact secret key of legitimate smart sensors. If attacker A_i penetrates the server challenge between S_i and δ over a long period of time, it will attempt to guess ψ . If it is traced, an attacker can access the network as a legitimate sensor device. In our proposed PEOS, an attacker would fail to identify the secret key, ψ , of any IoT sensor device unless it knows the randomly chosen nonce value and seed value. Moreover, dynamically generated passwords using random generators periodically cannot be identified using brute force or dictionary attacks. The proposed PEOS ensures security against guessing attacks on smart sensors in IoT applications.

Security against DoS Attacks: Denial of service is accomplished by flooding the target smart sensor with superfluous CoAP requests and overloading the server to prevent some or all legitimate requests from being transmitted to the server. In a distributed DoS (DDoS) attack, the incoming traffic flooding originates from different sources.

An attacker has to overhear a CON message to implement a DoS attack against the PEOS because DoS attacks send the same request multiple times in order to destroy the target sensor. In accordance with the PEOS, the smart sensor uses a token to identify fresh request messages and rejects login requests that contain the same token value and nonce values again.

Security against Traceability Attacks: A traceability attacker traces several requests to identify an IoT sensor device identity and secret data. However, the PEOS explores the pseudonym ID instead of the original identity and ensures security against traceability attacks. Moreover, the proposed scheme ensures better data confidentiality, even in the worst case of initial key compromise, because it uses the AES with a dynamic key-generation scheme.

4.7 Experimental Evaluation

The proposed PEOS for lightweight authentication was implemented on the DTLS feature and integrated with the CoAP architecture (DF-CoAP) to demonstrate its performance using the Contiki Cooja network simulator. A communication process was conducted between the smart sensors and the gateway. Adversarial activities were made to disturb the sensors' data communication and PEOS was applied to the DF-CoAP to ensure secure communication between the smart sensors and the gateway. This section presents the experimental results of the PEOS, an existing PbMA [23][24], and the CoAP/DTLS [7][8]. As shown in Figure 4.6, the experiment was conducted on 30 and 50 compatible smart sensors, one border router, and one server working under the control of the Contiki OS for 300 s. Two types of motes were used in the simulation: z1 was used for the border-router, and WiSMote for the sensor and server nodes. The simulation was conducted on various numbers of smart sensors to identify the impact of network scalability on the efficiency of the lightweight authentication scheme. The simulation was performed in a $100 \times 100 \text{ m}^2$ area for both the proposed and the existing works. In the distributed sensor network, the payload size varied from 64 to 1024 bytes. Various scenarios were created for the payload since it plays an important role in the efficiency of lightweight authentication. The communication range of each node was set to 50 m. Table 4.2 presents the experimental parameters. (See Appendix B&C).

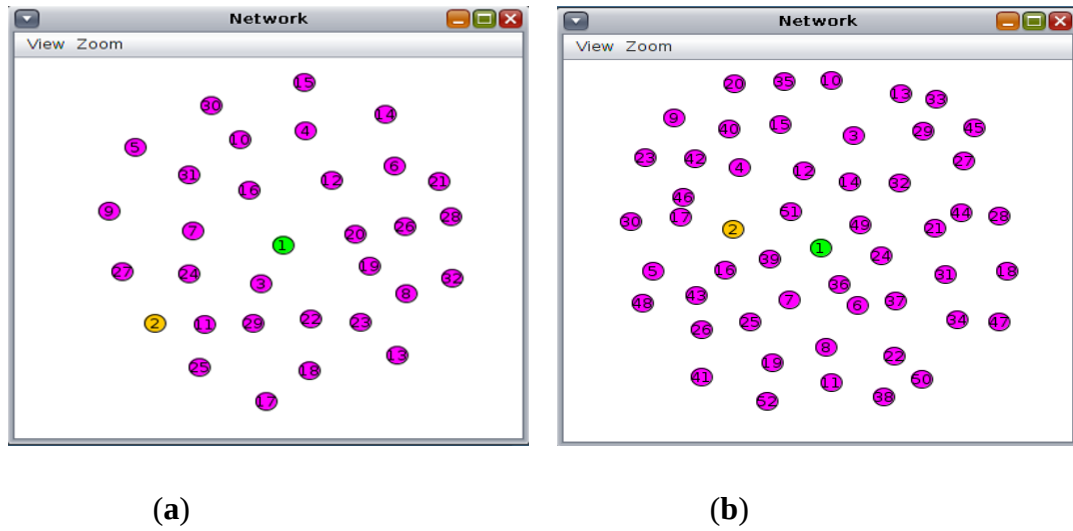


Figure 4-6: Experiment scenarios based on the Contiki OS. (a) Network design diagram for 30 nodes; (b) Network design diagram for 50 nodes

Table 4-2: Experimental parameters

Parameter	Value
Application Layer	CoAP
Transport Layer	UDP
Network Layer	6LoWPAN
MAC Layer	IEEE 802.15.4
Total Number of Nodes	30 and 50
Total Number of Nodes	30 and 50
Simulation Area (m)	100 × 100
Transmission Range (m)	50
Simulation Time (minutes)	5
Payload Size (bytes)	64, 127, 256, 512, 1024
Algorithms	CoAP/DTLS, PbMA, and PEOS on DF/CoAP

4.7.1 Time, Computation, and Storage Complexity

In addition to providing a security scheme, the lightweight authentication scheme used in PEOS was evaluated in terms of time, storage, and computational complexity.

4.7.1.1 Time Complexity

The time complexity of PEOS was estimated using the metric of handshake duration and communication. Handshake duration can be defined as the sum of time taken by two round-trip messages, and includes the session initiation requests, the smart sensor response and the challenge between the smart sensors and the server. A smart sensor acknowledges the session initiation request through a smart sensor response, whereas the server acknowledges a smart sensor response and challenge through a server response. The time complexity, $\text{Time}_{\text{Comp}}$, was computed at the smart sensor end using Equation (8):

$$\text{Time}_{\text{Comp}} = \text{Hd} + T_{\text{Communication}} \quad (8)$$

From the above equation, Hd is estimated using the summation of the round-trip time taken by a session initiation request, the round-trip time taken by a smart sensor response and request, and the processing time of a request at the smart sensor. The summation of Hd and the time taken to deliver the data denotes the time complexity of the proposed work. Instead of sending messages through a control channel, both the existing and the proposed schemes implement PbMA. This reduces the time complexity significantly in all the schemes [30]. However, the proposed work shows a noticeable reduction in time complexity compared to the existing schemes. Since the proposed work applies NACK-based message transmission, this assists PEOS in informing the smart sensor of the exact data missing from the current flight with the right-most received byte. The number of retransmissions as well as the $\text{Time}_{\text{comp}}$ are reduced in PEOS.

4.7.1.2 Computational and Storage Complexity

Computational and storage complexity is decided based on the number of encryption/decryption operations, signature/verification operations, and random number generation. The proposed scheme shows a significant reduction in the computational cost of PEOS by secret key generation at the server and the smart sensor side individually. However, the existing payload-based encryption scheme needs periodically to share the secret key with each smart sensor [30]. Thus, the proposed work shows a significant reduction in computational complexity when compared to the existing work. The notation α in the following equation shows a reduction in the computational complexity of the

proposed work compared to the payload-based encryption scheme since it explores the AES-based scheme. The parallel execution of S-Boxes in SubBytes of the AES reduces the complexity of the proposed work drastically. Moreover, the proposed scheme introduces nonce and seed values, but this does not have a significantly greater impact on complexity than the generation of a secret key for each smart sensor at the server, where α denotes the interval for refreshing the secret key for each sensor device at the server.

$$\text{Comp Complexity}_{\text{Existing}} = O(\alpha N) \quad (9)$$

$$\text{Comp Complexity}_{\text{PEOS}} = O(\alpha N \log N) \quad (10)$$

The storage cost is decided based on the number of parameters used for the mutual authentication and secure communication processes. The storage cost of the proposed scheme on the smart sensor side is lower than for the existing work due to the incorporation of dynamic secret and session key generation. On the server side, both works show similar storage complexity. Thus, the complexity of PEOS is significantly reduced. Table 4.3 demonstrates that the smart sensor side storage cost is 384 bits, whereas the storage cost of the server is 640 bits.

Table 4-3: Storage cost of the proposed PEOS and existing payload-based encryption schemes

Type	Parameter	Proposed Work		Existing Work	
		Used in a Smart Sensor	Used in the Gateway	Used in a Smart Sensor	Used in the Gateway
Handshake Features	Nonce 1	Yes	No	Yes	No
	Nonce 2	No	Yes	No	Yes
	Token	Yes	Yes	Yes	Yes
AES-related Features	Secret Key	Yes	No	Yes	Yes
	Session Key	Yes	No	Yes	Yes
	Seed Value	Yes	No	No	No
	Potential Session Key	No	No	Yes	Yes
Total Storage Cost (bits)		640 bits	384	640	640

4.7.1.3 Performance Metrics

A simulation was conducted to evaluate the following metrics for lightweight PbMA (for PEOS on DF-CoAP, PbMA [24], and CoAP/DTLS [8]). This was carried out by varying the number of smart sensors and the payload size.

Throughput: Number of bits transmitted successfully to a server per second.

$$\text{Throughput } \left(\frac{\text{bits}}{\text{sec}} \right) = \text{Total Number of Transmitted Packets} * 8 / \text{Simulation Time} \quad (11)$$

In Equation (11), the packets are given in bytes; this is converted into bits by multiplying by 8. When the total number of bits is divided by the simulation time, the equation gives the number of bits transmitted to a server per second.

Communication Duration: The communication duration is the time a smart sensor node takes to deliver a data packet successfully. This includes the handshake process, mutual authentication, and secure communication with the server.

$$\begin{aligned} &\text{Communication Duration (milliseconds)} \\ &= \left(\text{Total Number of Transmitted Packets} / \text{Simulation Time} * \text{Node Count} \right) \\ &\quad * 1000 \end{aligned} \quad (12)$$

Energy Consumption: The number of joules consumed by a smart sensor to deliver data packets to the gateway.

$$\begin{aligned} &\text{Energy Consumption (Micro Joules)} \\ &= \left(\text{Total amount of consumed energy in the network over simulation time} / \text{Node Count} \right) \\ &\quad * 1,000,000 \end{aligned} \quad (13)$$

Message Size Overhead: This is defined as the total length of the header in the packets transmitted and measured in bytes.

$$\text{Message Size Overhead (Bytes)} = \text{Total length of the Packet} - \text{Length of Payload} \quad (14)$$

4.7.2 Experiment Results for a 30-Node Topology

In the IoT topology, 30 nodes were positioned to simulate CoAP communication and to compare the performance of the proposed and existing schemes. With a reduced payload

size, more packets can be transmitted in the network. High network traffic interrupts the device process and tends towards high packet loss, which is the reason for the throughput improvement in all the works with an increase in payload size, as shown in Figure 4.7. For instance, the proposed scheme attains nearly 2 bits/s of throughput with 127 bytes, and this is increased to 6.4 bits/s and 11 bits/s with 512 and 1024 bytes of payload, respectively. Likewise, the CoAP/DTLS improves the throughput from 0.8 to 8 bits/s, and the existing payload-based scheme improves the throughput from 1.8 to 6 bits/s when the payload is increased from 64 to 1024 bytes. The proposed scheme outperforms PbMA and basic DTLS under various payload sizes, as shown in Figure 4.7. The proposed authentication scheme decides the exact buffer size for each smart sensor and reduces unnecessary packet loss. However, the basic DTLS scheme [8] allows the server to decide the buffer size randomly, reducing the throughput compared to the other schemes. When increasing the payload size from 127 bytes to 1024 bytes, the difference in network throughput between the proposed scheme and the basic DTLS starts to improve, from 0.1 to 0.5 bits/s. Figure 4.7 shows that the proposed scheme performs better than both the existing schemes.

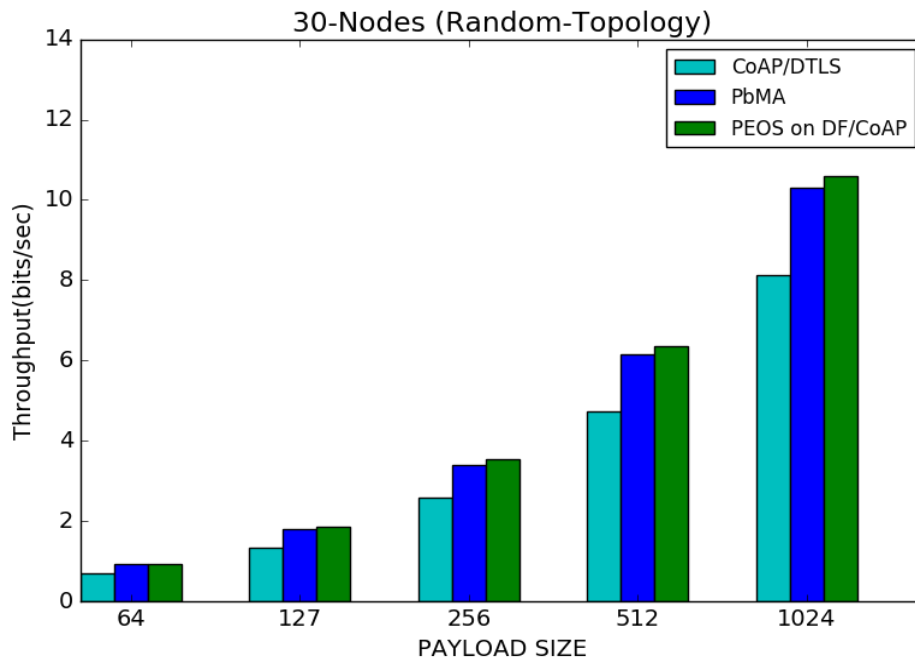


Figure 4-7: Payload size vs. throughput for a 30-node topology

Figure 4.8 shows the energy consumption performance for the proposed and existing schemes on the CoAP. The PEOS scheme outperforms the PbMA [31] and basic DTLS [8] when the network sends data packets in various payload sizes. As with the payload sizes of 64 and 127 bytes, the proposed scheme achieves a reduction in network traffic and no significant change in energy consumption compared to the basic CoAP/DTLS. The energy consumption in the proposed scheme increases slightly with the payload size. For instance, the proposed scheme increases the energy consumption from 1830 to 2200 microjoules with an increase in payload size of 64 to 1024 bytes. The proposed and existing payload-based schemes spend less energy than the basic CoAP/DTLS. Moreover, the difference in average energy consumption between the proposed and the existing payload-based encryption schemes is minimal. For instance, the difference between the PEOS and the existing PbMA is 25 microjoules with a payload size of 512 bytes. The proposed PEOS explores unique features of basic DTLS, resulting in a small amount of energy consumption. The concepts of exact fragment sending and delayed MixColumns reduce the requirement for additional storage registers and slightly reduce the energy consumption level compared to the existing payload scheme. However, the difference between the proposed scheme and the basic CoAP/DTLS is 230 microjoules with a payload size of 512 bytes.

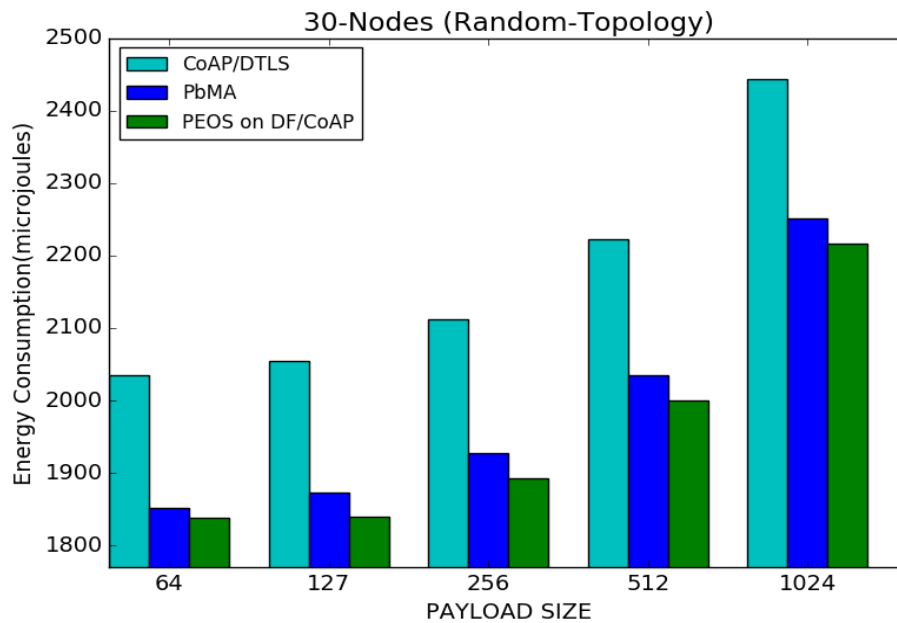


Figure 4-8: Payload size vs. average energy consumption for a 30-node topology

Figure 4.9 shows the results for communication duration for both the proposed and the existing schemes. The proposed and existing payload-based encryption schemes (i.e., PEOS and PbMA) have approximately equal duration for delivering data packets. For instance, with a payload size of 512 bytes or under, the proposed and existing payload-based encryption schemes successfully transmit the data packets in 1200 milliseconds (ms) and 1500 ms, respectively. However, the basic CoAP/DTLS delivers the packets in 1350 ms. High packet loss in the basic CoAP/DTLS is the main reason for the short communication duration. As observed, since the proposed scheme can deliver many transmitted packets due to the network traffic reduction using exact buffer size declaration and fragment transmission, the delay difference in the proposed scheme is considered to ensure low packet loss and retransmission. In comparison, the existing schemes attain less throughput and, at the same time, take longer to deliver data packets. For instance, in a payload size of 64 bytes, the proposed scheme takes 1710 ms. However, the PbMA and CoAP/DTLS take 1680 ms and 1480 ms, respectively.

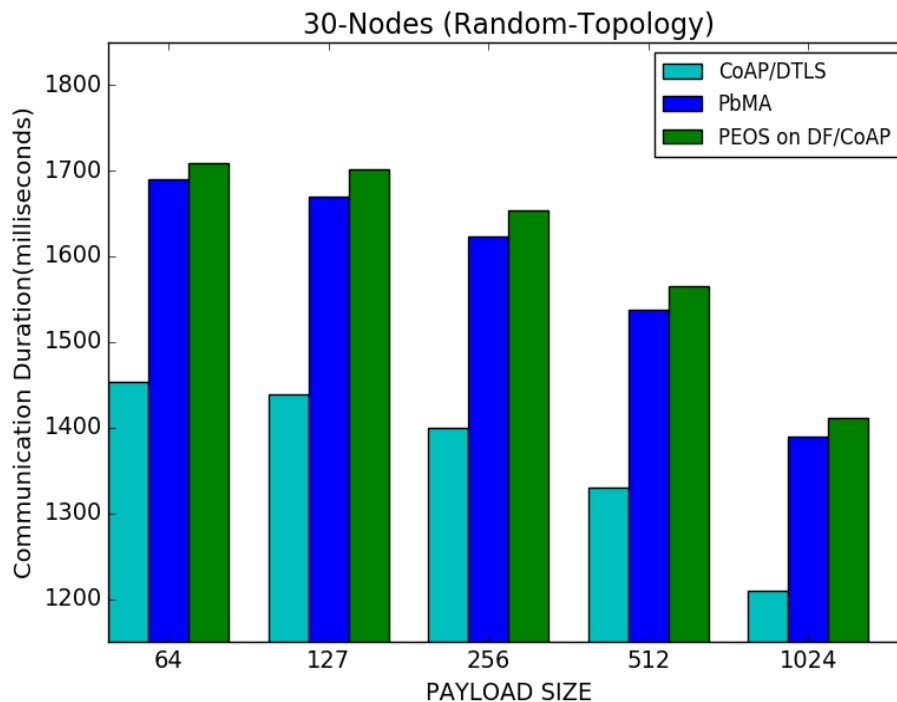


Figure 4-9: Payload size vs. communication duration for a 30-node topology

Generally, CoAP CON messages increase the network delay. However, the proposed work can maintain the delay. The CoAP retransmission scheme in the proposed method

is optimised using NACK-based retransmission. The payload encryption-based handshaking process reduces the handshaking process from three to two round trips. Thus, the delay is reduced considerably in the proposed scheme.

Figure 4.10 shows the performance of the proposed PEOS and existing security schemes (PbMA and basic CoAP/DTLS) when the payload size is varied from 64 to 1024 bytes. Both payload-based encryption schemes (PEOS and PbMA) reduce the message size overhead due to the utilisation of unique DTLS packets. The proposed PEOS considerably reduces the packet header size and unnecessary delays in communication. This improves the availability of network resources and the possibility of successful packet transmissions to the server. This reduces the message size drastically compared to the other protocols. For a payload size of 127 bytes, PEOS experiences a message size overhead of 120 bytes, whereas the existing payload-based authentication scheme (PbMA) and the basic CoAP/DTLS demonstrate a message size overhead of 130 bytes and 170 bytes, respectively. With incremented payload sizes, all the authentication schemes reduced the number of messages. Therefore, the message size overhead and the total length of the header in all the packets transmitted is also reduced. In figure 4.10, the proposed scheme experiences 120 bytes of message size overhead, whereas the message size overhead is reduced to 30 bytes with a payload size of 512 bytes. For a payload size of 127 bytes, the basic CoAP/DTLS scheme experiences 170 bytes, which is reduced to 40 bytes when the payload size is 512 bytes.

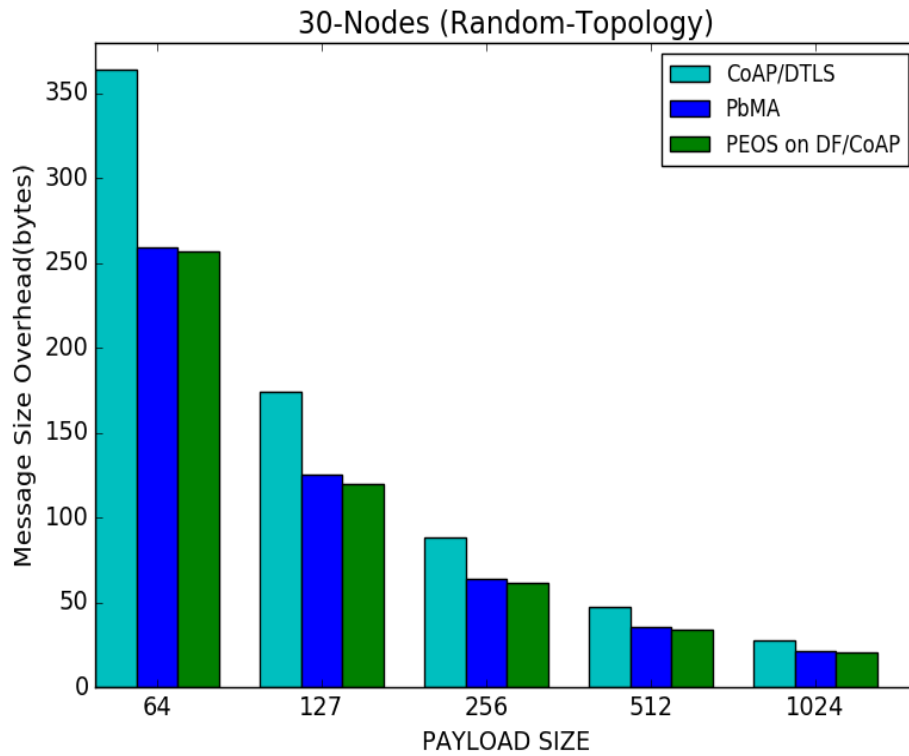


Figure 4-10: Payload size vs. message size overhead for a 30-node topology

4.7.3 Simulation Results for a 50-Node Topology

Figure 4.11 presents the throughput for various schemes under a 50-node topology. The network throughput is highly related to network traffic and congestion. The lowest value of payload size increases the network traffic and network congestion unnecessarily, leading to packet loss at the routers, which is the main reason for the throughput improvement with the payload size increment. The proposed and existing payload-based schemes append only the important features of DTLS and reduce the packet length, which improves the network throughput and reduces packet loss. For instance, the proposed PEOS attains nearly 0.9 bits/s of throughput with 64 bytes, increasing to 8.7 bits/s with 1024 bytes of payload. Likewise, the CoAP/DTLS improves the throughput from 1.2 to 4.1 bits/s, and the existing payload-based scheme improves the throughput from 0.5 to 6.3 bits/s when the payload is increased from 64 to 1024 bytes. The proposed scheme improves the network throughput tremendously compared to the existing PbMA scheme and the CoAP/DTLS. The existing schemes implement the whole flight retransmission scheme, which means that a single fragment loss in those schemes requires sending the

whole flight again. Moreover, both the existing schemes explore pre-shared keys for a long time, increasing the possibility of attacker activities and traffic. For instance, the proposed scheme works better 8.7% than the existing payload-based scheme (PbMA) when the payload size is 1024 bytes.

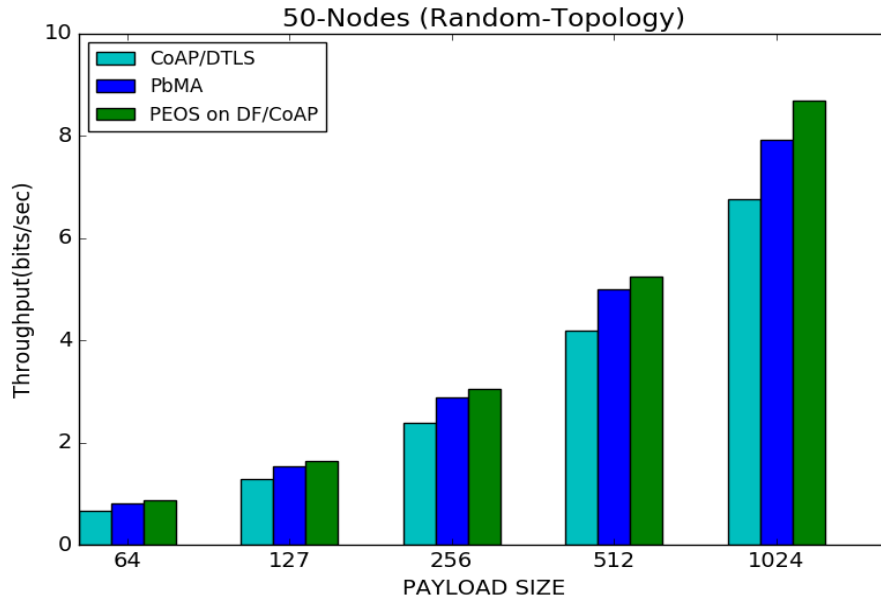


Figure 4-11: Payload size vs. throughput for a 50-node topology

Figure 4.12 shows the influence of payload size on the performance of the proposed and existing security schemes with the CoAP, PEOS, PbMA, and basic CoAP/DTLS under a 50-node topology. The proposed scheme demonstrates its performance with minimum energy consumption, especially with a small payload size. The proposed scheme also reduces the number of transmitted packets with a large payload size, but the packet length is high. This increases the energy consumption slightly with an increment in payload size.

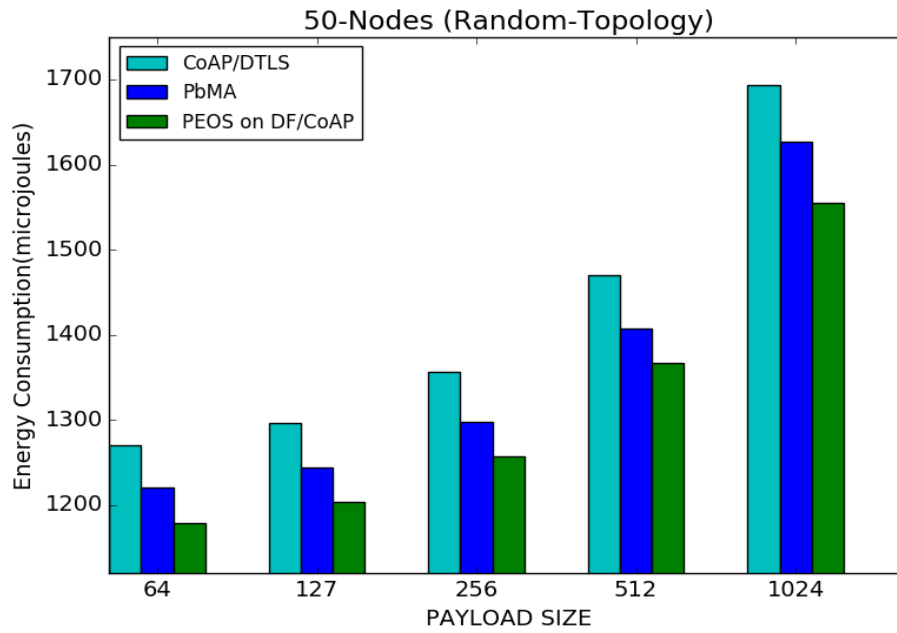


Figure 4-12: Payload size vs. average energy consumption for a 50-node topology

For instance, the proposed PEOS spends 1170 microjoules with a payload size of 64 bytes, which is increased to 1550 microjoules for a payload size of 1024 bytes. Figure 4.12 shows the performance of the proposed scheme over the existing security schemes (PbMA and basic CoAP/DTLS) when the payload size is varied from 64 to 1024 bytes. The proposed scheme reduces the header size and delay of communication due to the proper measurement of buffer size, which considerably reduces unnecessary packet loss. Moreover, the increased MixColumns and S-Boxes used in the AES scheme for the existing payload-based authentication scheme increases the number of registers and average energy consumption compared to the proposed scheme. For a payload size of 64 bytes, the proposed PEOS spends 1170 microjoules, whereas the existing payload-based PbMA and basic CoAP/DTLS expend 1220 and 1270 microjoules, respectively.

Figure 4.13 shows the communication duration performance for the PEOS, PbMA, and basic CoAP/DTLS schemes. It shows that the basic CoAP/DTLS outperforms both the existing payload-based authentication scheme and the proposed scheme in all the scenarios. As with payload size, improved performance in communication duration is achieved by all the schemes. The number of transmitted packets is reduced with a larger payload size, and the communication duration is reduced compared to the scenario with a small payload size. For instance, with a payload of 512 bytes, the proposed and existing

payload-based encryption schemes, PEOS and PbMA [31], successfully transmit data packets in 2300 ms and 2350 ms, respectively. However, the basic CoAP/DTLS [8] delivers packets in 2100 ms. Moreover, identifying the exact buffer size and an intimation of the loss of exact fragments avoids unnecessary packet loss and communication duration in the proposed scheme. For instance, in a payload of 64 bytes, the proposed PEOS delivers a packet in 2700 ms. However, the existing payload-based scheme, PbMA, and the basic CoAP/DTLS take 2620 ms and 2390 ms, respectively.

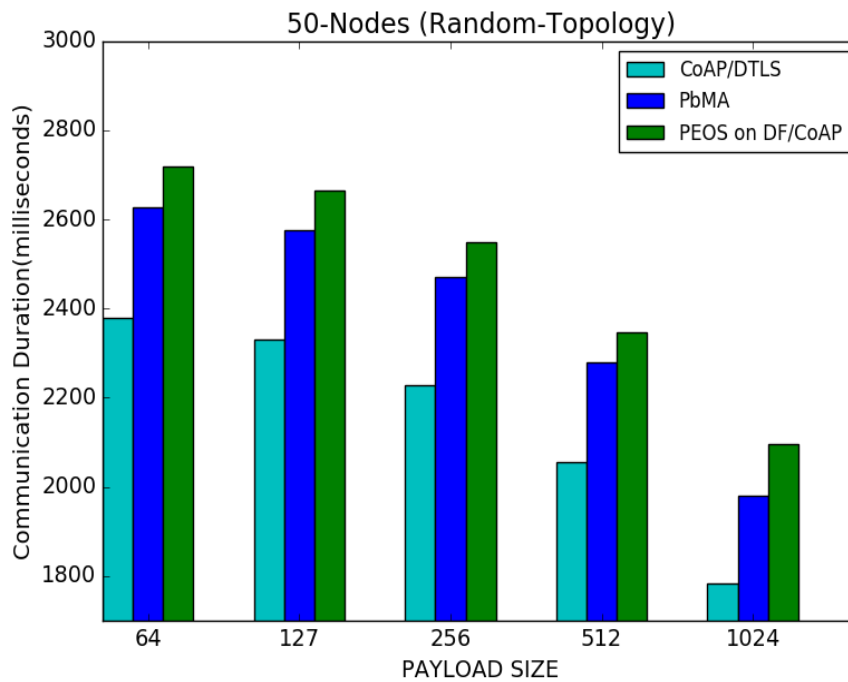


Figure 4-13: Payload size vs. communication duration for a 50-node topology

Figure 4.14 shows the results of the message size overhead simulation for the proposed PEOS and the existing PbMA and basic CoAP/DTLS schemes under various payload sizes. As observed, since the proposed scheme delivers data in a few packets with a large payload size, the message size overhead of all the schemes is considerable compared to the scenario of having a small payload size. For a payload size of 1024 bytes, PEOS experiences a message size overhead of 50 bytes, whereas the existing PbMA and the basic CoAP/DTLS experience a message size overhead of 55 bytes and 65 bytes, respectively. In figure 4.14, the proposed scheme is seen to experience 230 bytes of message size overhead when the payload size is 127 bytes, whereas the message size overhead of the proposed scheme is reduced to 50 bytes for the maximum payload size.

For a payload size of 127 bytes, the CoAP/DTLS scheme experiences 310 bytes, which is reduced to 65 bytes for a payload of 1024 bytes.

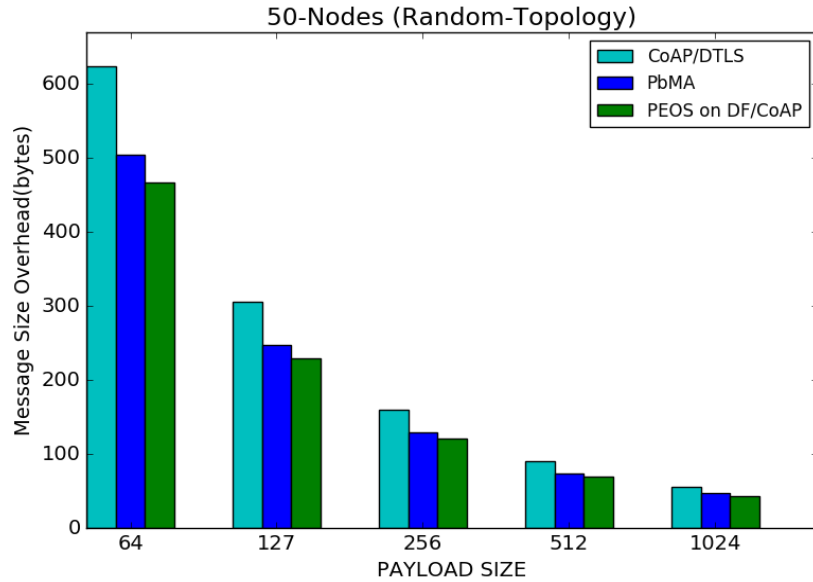


Figure 4-14: Payload size vs. message size overhead for a 50-node topology

4.8 Chapter Summary

The proposed work designed a mutual authentication scheme, a Payload Encryption-based Optimisation Scheme (PEOS), for lightweight authentication on the CoAP. Most of the existing works on CoAP security do not consider the DTLS overhead for resource-constrained smart sensors. The proposed PEOS integrates and optimises important features of DTLS in the CoAP architecture to overcome this issue and avoids the need to implement DTLS in a separate channel. Moreover, incorporating payload encryption and NACK messages improves the DTLS handshaking and retransmission processes in PEOS. By removing the duplicate features of the protocol version and the sequence number, the overhead is reduced significantly without this having an impact on CoAP performance over distributed sensors. The proposed PEOS exploits an OP-AES. PEOS implemented the parallel execution of S-Boxes in SubBytes and delayed MixColumns in the AES and successfully reduced the necessity for additional storage registers. Moreover, PEOS implemented a dynamic key-generation process and avoided offering the opportunity for key traceability and guessing attacks in the CoAP over distributed sensors. The utilisation of token and dynamic key-generation processes in PEOS protects

against DDoS and confidentiality attacks. The efficiency of PEOS over distributed sensors in lightweight security and authentication was evaluated on the Contiki OS, which supported the Cooja simulator. Moreover, the proposed work was compared with an existing payload-based scheme and basic DTLS in a sensor network scenario. From the results, with a large payload size, the throughput of the proposed scheme was improved by 8.7% compared to the existing PbMA by consuming 1550 microjoules under a 50-node random topology.

4.9 References

1. Ghasempour A. Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges. *Inventions*. 2019; 4(1). Available at: DOI:10.3390/inventions4010022
2. Senior FMC., Ieee M. Cyber Securiry Issues for Advanced Metering Infrastructure ((AMI)). IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century. Pittsburgh, PA, USA; 2008.
3. Katsikas S., Cuppens F., Cuppens N., Lambrinoudakis C., Kalloniatis C., Mylopoulos J., et al. *Computer Security*. 2020.
4. Huseinović A., Mrdović S., Bicakci K., Uludag S. A survey of denial-of-service attacks and solutions in the smart grid. *IEEE Access*. 2020; 8: 177447–177470. Available at: DOI:10.1109/ACCESS.2020.3026923
5. Gunduz MZ., Das R. Cyber-security on smart grid: Threats and potential solutions. *Computer Networks*. Elsevier B.V.; 2020; 169: 107094. Available at: DOI:10.1016/j.comnet.2019.107094
6. Mohamad Noor M binti., Hassan WH. Current research on Internet of Things (IoT) security: A survey. *Computer Networks*. Elsevier B.V.; 2019; 148: 283–294. Available at: DOI:10.1016/j.comnet.2018.11.025
7. Arvind S., Narayanan VA. An Overview of Security in CoAP: Attack and Analysis. 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019. IEEE; 2019; : 655–660. Available at: DOI:10.1109/ICACCS.2019.8728533
8. Surendran S., Nassef A., Beheshti BD. A survey of cryptographic algorithms for IoT devices. 2018 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2018. IEEE; 2018; : 1–8. Available at: DOI:10.1109/LISAT.2018.8378034
9. Mrabet Z El., Kaabouch N., Ghazi H El., Ghazi H El. Cyber-security in smart grid: Survey and challenges. *Computers and Electrical Engineering*. Elsevier Ltd; 2018;

- 67: 469–482. Available at: DOI:10.1016/j.compeleceng.2018.01.015
10. Kumar P., Lin Y., Bai G., Paverd A., Dong JS., Martin A. Smart Grid Metering Networks: A Survey on Security, Privacy and Open Research Issues. *IEEE Communications Surveys and Tutorials*. IEEE; 2019; 21(3): 2886–2927. Available at: DOI:10.1109/COMST.2019.2899354
 11. Abosata N., Al-Rubaye S., Inalhan G., Emmanouilidis C. Internet of things for system integrity: A comprehensive survey on security, attacks and countermeasures for industrial applications. *Sensors*. MDPI AG; 2021. Available at: DOI:10.3390/s21113654
 12. Sarenche R., Salmasizadeh M., Ameri MH., Aref MR. A secure and privacy-preserving protocol for holding double auctions in smart grid. *Information Sciences*. Elsevier Inc.; 2021; 557: 108–129. Available at: DOI:10.1016/j.ins.2020.12.038
 13. Abdallah A., Shen X. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*. IEEE; 2018; 9(1): 396–405. Available at: DOI:10.1109/TSG.2016.2553647
 14. Khan AA., Kumar V., Ahmad M., Rana S. LAKAF: Lightweight authentication and key agreement framework for smart grid network. *Journal of Systems Architecture*. Elsevier B.V.; 2021; 116(October 2020): 102053. Available at: DOI:10.1016/j.sysarc.2021.102053
 15. Abbasinezhad-Mood D., Nikooghadam M. An Anonymous ECC-Based Self-Certified Key Distribution Scheme for the Smart Grid. *IEEE Transactions on Industrial Electronics*. IEEE; 2018; 65(10): 7996–8004. Available at: DOI:10.1109/TIE.2018.2807383
 16. Grover HS., Adarsh., Kumar D. Cryptanalysis and improvement of a three-factor user authentication scheme for smart grid environment. *Journal of Reliable Intelligent Environments*. Springer International Publishing; 2020; 6(4): 249–260. Available at: DOI:10.1007/s40860-020-00113-2
 17. Braeken A., Kumar P., Martin A. Efficient and provably secure key agreement for

- modern smart metering communications. *Energies*. 2018; 11(10): 1–18. Available at: DOI:10.3390/en11102662
18. Khan AA., Kumar V., Ahmad M., Rana S., Mishra D. PALK: Password-based anonymous lightweight key agreement framework for smart grid. *International Journal of Electrical Power and Energy Systems*. Elsevier; 2020; 121(November 2019): 106121. Available at: DOI:10.1016/j.ijepes.2020.106121
 19. Chaudhry SA. Correcting “PALK: Password-based anonymous lightweight key agreement framework for smart grid”. *International Journal of Electrical Power and Energy Systems*. Elsevier Ltd; 2021; 125(June 2020): 106529. Available at: DOI:10.1016/j.ijepes.2020.106529
 20. Deng L., Gao R. Certificateless two-party authenticated key agreement scheme for smart grid. *Information Sciences*. Elsevier Inc.; 2021; 543: 143–156. Available at: DOI:10.1016/j.ins.2020.07.025
 21. Chaudhry SA., Alhakami H., Baz A., Al-Turjman F. Securing Demand Response Management: A Certificate-Based Access Control in Smart Grid Edge Computing Infrastructure. *IEEE Access*. 2020; 8: 101235–101243. Available at: DOI:10.1109/ACCESS.2020.2996093
 22. Wang Z., Liu Y., Ma Z., Liu X., Ma J. LiPSG: Lightweight Privacy-Preserving Q-Learning-Based Energy Management for the IoT-Enabled Smart Grid. *IEEE Internet of Things Journal*. IEEE; 2020; 7(5): 3935–3947. Available at: DOI:10.1109/JIOT.2020.2968631
 23. Kumar PM., Gandhi UD. Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application. *Journal of Supercomputing*. Springer US; 2020; 76(6): 3963–3983. Available at: DOI:10.1007/s11227-017-2169-5
 24. Jan MA., Khan F., Alam M., Usman M. A payload-based mutual authentication scheme for Internet of Things. *Future Generation Computer Systems*. Elsevier B.V.; 2019; 92: 1028–1039. Available at: DOI:10.1016/j.future.2017.08.035

Chapter 5: Customised Intrusion Detection for IIoT based on Machine learning algorithms called FTL-CID

5.1 Abstract

Technological breakthroughs in the Internet of Things (IoT) easily promote humans toward smart lives by connecting everything through the internet. The de-facto standardized IoT routing strategy is the Routing Protocol for Low Power and Lossy Networks (RPL) applied in various heterogeneous IoT applications. Hence, the swell in reliance on IoT demands a high focus on the security of RPL protocol. The top defense layer is Intrusion Detection System (IDS), and the heterogeneous IoT characteristics and variety of novel intrusions make the design of the RPL-IDS significantly complex. Most of the existing IDS solutions are unified models and fail to detect novel RPL intrusions. Therefore, the RPL requires a global attack knowledge-based customised IDS model to identify both existing and novel intrusions for its security enhancement. Federated Transfer Learning (FTL) is a trending topic that paves the way to designing a customised RPL-IoT IDS security model over a heterogeneous IoT environment. This chapter proposes a Federated Transfer learning-assisted Customised Distributed IDS (FT-CID) model to detect RPL intrusion on heterogeneous IoT. The design process of FT-CID includes three steps which are dataset collection, FTL-assisted edge IDS learning, and intrusion detection. Initially, the central server initialises the FT-CID with a predefined learning model and observes the unique features of different RPL-IoT to construct the local model. Secondly, the edge IDSs are trained using the local parameters and the globally shared parameters generated by the central server through federation and aggregation of different local parameters of various edges. Hence, transfer learning is exploited to update the server's and edges' local and global parameters based on relational knowledge. Also, it builds the customized IDS model with partial retraining at local learning based on globally shared server knowledge. Finally, the customised IDS in the FT-CID model enforces the detection of intrusions in heterogeneous IoT networks. Moreover, the FT-CID model accomplishes high RPL security by implicitly utilising the local and global parameters of different IoT with the assistance of FTL. Thus, the FT-CID detects the RPL intrusions with an accuracy of 85.52% while testing on a heterogeneous IoT network.

5.2 Introduction

The Internet of Things (IoT) is a collection of smart devices, such as mobile phones and smart sensors, cameras and laptops, which are connected with each other through the Internet [1]. A variety of technological breakthroughs in the industrial IoT (IIoT) has recently brought numerous applications, ranging from small industries to smart cities, which have become an integral part of several aspects of human life [2]. The Routing Protocol for Low-Power and Lossy Networks (RPL) is a standardised IIoT routing strategy. It is well suited to establishing communication infrastructure among tiny resource constrained IIoT devices. However, the resource-limited nature of IoT devices and critical smart infrastructures motivate hackers to launch various attacks within the RPL-IIoT scenario, and RPL security is of paramount importance. A common method of RPL-IIoT attack detection is to employ a machine learning (ML)-based intrusion detection system (IDS) [3], [4]. However, most traditional ML-based IDS solutions are centralised and do not have the flexibility to detect the variety and full range of RPL attacks on IIoT applications [5]. Heterogeneous IIoT networks perform a range of network functions and present a challenge to intrusion detection as there is a variety of data and attack types. ML-based solutions are typically evaluated with small datasets, resulting in a lack of detection of novel attacks in real time. In reality, the RPL is mainly utilised in heterogeneous IIoT environments and it is crucial to evaluate the IDS security model with a variety of IIoT intrusions presented in diverse RPL-IIoT networks.

Federated learning (FL)-based IDS models are currently effective for IIoT intrusion detection, as they can train a deep learning model with global knowledge and ensure a high degree of security while protecting user data [6]. Depending on the data sample and feature distribution, FL is categorised into three types: horizontal FL, vertical FL, and federated transfer learning (FTL) [7], [8]. FL-based IDS models enhance learning efficiency, with reduced overhead, and protect user information by achieving distributed learning at the edges. However, the insufficiency of unique data labelling and the dissimilarity of data features make FL-based IDS systems a challenge. Transfer learning (TL) solves such issues by transferring the abundant labelled source domain knowledge to the edges. This avoids the lengthy learning time of particularised RPL-IIoT for designated applications. Moreover, an FLT algorithm is a novel concept that achieves an

effective learning model by implicitly integrating the local and global knowledge of multiple IIoT networks.

This chapter presents an FTL-assisted customised IDS model, the FT-CID, to detect intrusions in a heterogeneous RPL-IIoT network. The main contributions of the FT-CID are as follows.

- To the best of our knowledge, the FT-CID is the first model to apply FTL to a heterogeneous RPL-IIoT IDS security model. This approach aggregates the data for IIoT application domains and accomplishes a customised attack detection model with global knowledge for each IIoT network by transferring and federating the learning knowledge between the server and edges.
- Firstly, the FT-CID uses FL to accumulate the local learning models of IIoT network and generate a global shared parameter for edges to detect different attack types of existing and unknown intrusions.
- Secondly, the utilisation of relational knowledge transfer learning in the FT-CID achieves a personalised IDS by the efficient transfer of distributed local and global parameters between the server and IIoT network or sensors with a high degree of security.
- Thirdly, the FT-CID detects and classifies intrusions under multiple classes using the customised IDS model and ensures a rich level of routing security for diverse IIoT scenarios without data leakage.
- Finally, the FT-CID reflects the security of a heterogeneous IIoT environment by carrying out experimental evaluation of IIoT dataset: Advanced Metering Infrastructure (AMI).

The remainder of the chapter is organised as follows. Section 5.2 and 5.3 outlines the network architecture, system model, and attack model. Section 5.4 presents the proposed FT-CID design process, with appropriate figures, equations, and algorithms. Section 5.5 shows the experimental setup and evaluation of FT-CID. Section 5.6 presents the experimental results. Finally, section 5.5 summarises this chapter.

5.2.1 Problem Statement

ML-based IDS models are commonly used approaches in RPL intrusion detection. However, in such models, the learning is performed in a centralised manner, in which the entirety of the data is transmitted to a centralised server, resulting in a high burden and low detection accuracy. To rectify these issues, later works propose FL-based IDS models in which a distributed IDS model is achieved based on federation and aggregation. However, such models lack the depth of knowledge of a local learning model transferred to a central server and this minimises the performance of the IDS. Transfer learning solves the issue of insufficient knowledge of unknown attacks at the edge by bridging the knowledge transfer between servers and edges. Also, they are not proposing a federated transfer learning based RPL security protocol to the heterogeneous IIoT environment. IIoT applications are heterogeneous in real time and it is crucial to design an effective IDS model for the IIoT environment. Therefore, the main intention of the proposed FT-CID is to detect intrusions into heterogeneous IIoT network and improve the security of the network layer. The proposed FT-CID is intended to achieve a very high level of intrusion detection accuracy without compromising network performance.

5.3 Preliminaries

This section explains the heterogeneous IIoT architecture model. It also describes the system model of the proposed FT-CID.

5.3.1 Network Architecture

The architecture of the proposed FT-CID is shown in Figure 5.1. The design of the FT-CID model assists in intrusion detection for heterogeneous IIoT network, Advanced Metering Infrastructure (AMI). The edge devices of the corresponding IIoT network are denoted as E-AMI. Firstly, the edge devices collect data from different IIoT devices through the RPL routing process. The FT-CID then instructs ‘N’ different edges of IIoT network to obtain traffic data and construct the datasets.

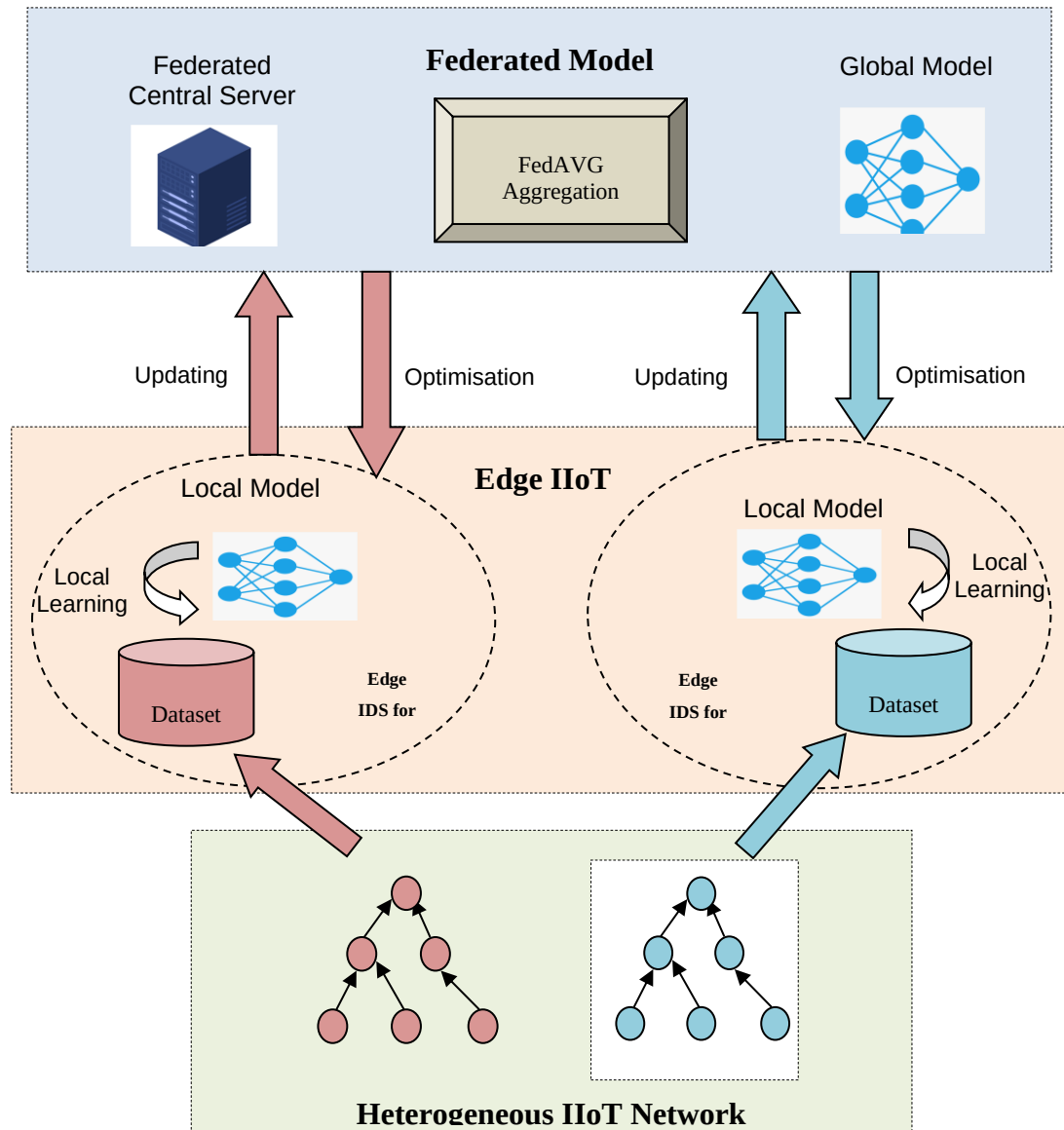


Figure 5-1: Architecture of heterogeneous IIoT network

5.4 System Model

An IIoT network refers to heterogeneous networks in which each IIoT network consists of a set of IoT devices, such as smartphones and smart sensors, cameras and meters. The FT-CID is composed of different aspects: IIoT devices, edges of the IIoT network or sensors, and the central server. The devices in an IIoT network are tiny and have constrained energy, storage space, and bandwidth. Compared with resource constrained IIoT devices, edge nodes have enormous resource capabilities. Therefore, the FT-CID places a signature-based IDS on the edge of the IIoT networks. Unlike resource limited IIoT devices and edges, such as smart watches, fire alarms, security systems and medical

sensors, the central server has enormous processing and storage capability. In FL, the central or global server gains knowledge from heterogeneous IIoT network, instead of storing or processing the data that belong to the sensors, and thus facilitates a distributed IDS at the edge of those networks. The IIoT network considered in the FT-CID is represented as a set of Heterogeneous IIoT ($HIIoT$)= $\{IIoT1\}$, where $IIoT1=AMI$. The edges of various IIoT network are denoted by $\{E-AMI\}$, and the traffic data collected from the heterogeneous IIoT network is represented by $\{D1\}$. By building a local learning model, the edge devices are responsible for learning the patterns from the local dataset gathered from the distributed IIoT networks. The central server exploits a traditional dataset to construct a predefined global learning model and partially retrains the global learning model with the assistance of the local model parameters provided through the TL process. The data collected from the heterogeneous IIoT network comprise both normal and abnormal data. Finally, a convolutional neural network (CNN)-based learning model in the FT-CID classifies the data at the edges as normal or abnormal based on the global learning parameters provided by the central server.

Attack Model: The RPL is exploited in heterogeneous IIoT environments and the attacks faced in such environments are also different. To experiment with the intrusion detection model, this work builds a dataset with the design of intrusion patterns from RPL protocol-related attacks, such as rank, version, hello flood, denial of service (DoS), blackhole, Sybil, selective forwarding, sinkhole, theft, and wormhole attacks. In the FT-CID, all the aforementioned attacks are modelled as intrusions, which addresses the binary classification problem of having only normal samples and intrusion samples. In a blackhole attack, a malicious intruder may attempt to throw packets away instead of forwarding them. A sinkhole attack is performed by making changes in the RPL ranks and obtaining a large amount of traffic from the network in order to alter its RPL topology. In a rank attack, a malicious intruder disseminates false ranks to enjoy the network benefits. Wormhole attackers employ the off-band communication of two different entities to perform attacks. During hello flooding, the malicious node exploits the DODAG Information Object (DIO) messages of Destination Oriented Directed Acyclic Graphs (DODOG) to launch the attack activity. The clone ID attacker mimics the legitimate nodes to diminish the network performance. A clone ID attacker mimics legitimate node to diminish network performance. Traffic analysis attackers exploit the

features and data patterns of communication and launch harmful activities in a network. During a sniffing attack, a malicious node notices the network traffic exchanges between two entities and performs attacking activities. Hence, an efficient learning model is crucial to identify the intrusions across normal samples.

Learning Model: An FL model contains a single server and multiple sensors, in which the server refers to the remote cloud server and the sensors refer to the IIoT network. The main objective of FL is to enable a range of knowledge to be gained from multiple IoT networks by utilising the parameters of the learning model instead of directly storing and processing the entire data of the distributed sensors on a central server. In an FL model, the edge of the IIoT network trains the local datasets generated from the heterogeneous IIoT applications with the deep learning model independently. The federated model then provides the local learning model parameters as the input to the global server through the edge network, and the remote server provides a global shared model with the updated parameters in the FL. To model intrusion detection for an IIoT network, FTL is employed to establish the communication across the IoT networks and obtain the different intrusion patterns of the local training data distributed across different silos and the various samples. Moreover, to address the issue of learning from a small number of attack behaviours in the IIoT network, this work models FTL for RPL-IIoT intrusion detection. The CNN model is adopted as the local and global learning model in the FTL with two different transfer scenarios: parameter transfer and relational knowledge transfer. Finally, the customised CNN model detects intrusions in heterogeneous IIoT networks.

5.5 Proposed System Design Overview

This work aims to design a novel RPL-IIoT security protocol with intrusion detection and introduces an optimised FTL-assisted customised IDS model: the FT-CID. The proposed FT-CID model is intended to achieve a very high level of intrusion detection accuracy in heterogeneous RPL-IIoT network by adopting an FTL approach. The FTL model shares global attack knowledge among the various IIoT application scenarios, which are generated based on the unique local learning parameters of a distributed IDS model. The design overview of the FT-CID is shown in Figure 5.2. The FT-CID design process can be explained according to three steps. Firstly, the centralised server initiates the network

and sends a predefined learning model to the edges to trigger the local learning process on the data. Secondly, the edges collect unique IIoT application data from application specific IIoT network and generate a local copy using a CNN classifier. Thirdly, the optimised FTL-based customised IDS model detects various RPL-IIoT intrusions based on local learning data, optimised using a global learning model. Instead of transferring raw input data or results from the local learning model, the FT-CID transfers only the non-normalised prediction output as the logits outcome of a fully connected CNN with an IIoT network or application-specific weight values to optimise the global model generation. After receiving the logits from the local learning model, the centralised server initiates federated aggregation on local model logits, considering the degree of importance in terms of the weights of local learning model parameters and logits for different IIoT networks. The FT-CID model can then generate novel logit-based global shared parameters, such as the number of hidden layers and neurons, as well as activation functions, to optimise the local learning model of the different RPL edge devices. Inspired by the idea of an attention mechanism [12], the FT-CID minimises the number of communications rounds and maximises the learning convergence. With the federation of different local RPL-IIoT network parameters, the FT-CID achieves strong security and generalisation attack detection ability in heterogeneous IIoT networks. Finally, the central server sends back the globally shared learning parameters by utilising TL among the different IIoT network to build customised learning models. Hence, the FT-CID model focuses on stochastic gradient descent (SGD)-based TL optimisation in order to improve the learning accuracy of each edge IDS without increasing the training time, which is repeated until the learning process converges. Intrusions are identified based on the global federated and unique attack knowledge acquired during testing. Thus, the distributed learning-based detection model overcomes the shortcomings of centralised intrusion detection models and improves RPL security when compared with a heterogeneous IIoT network, without degrading network efficiency.

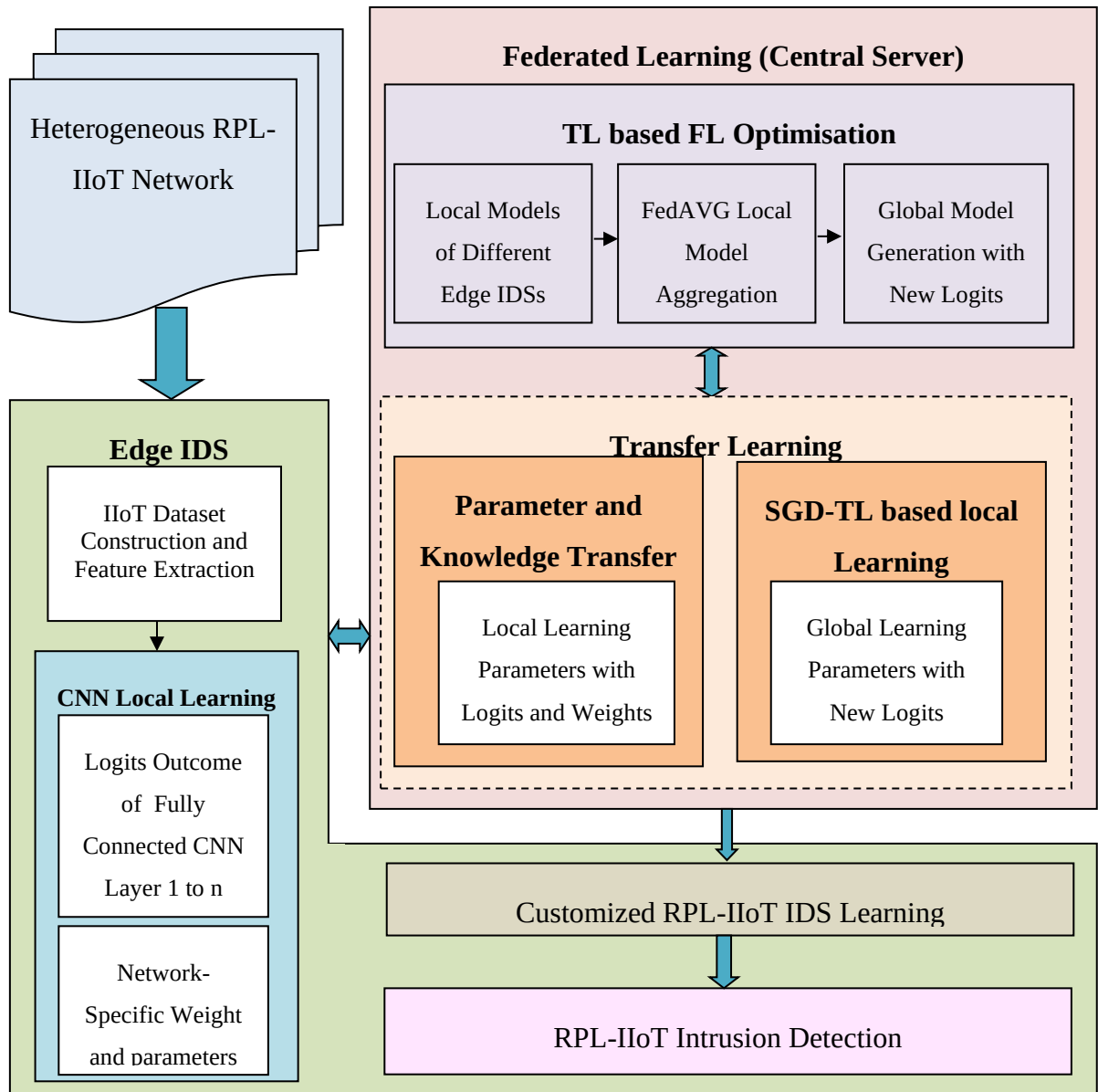


Figure 5-2: Overview design of the proposed FT-CID model

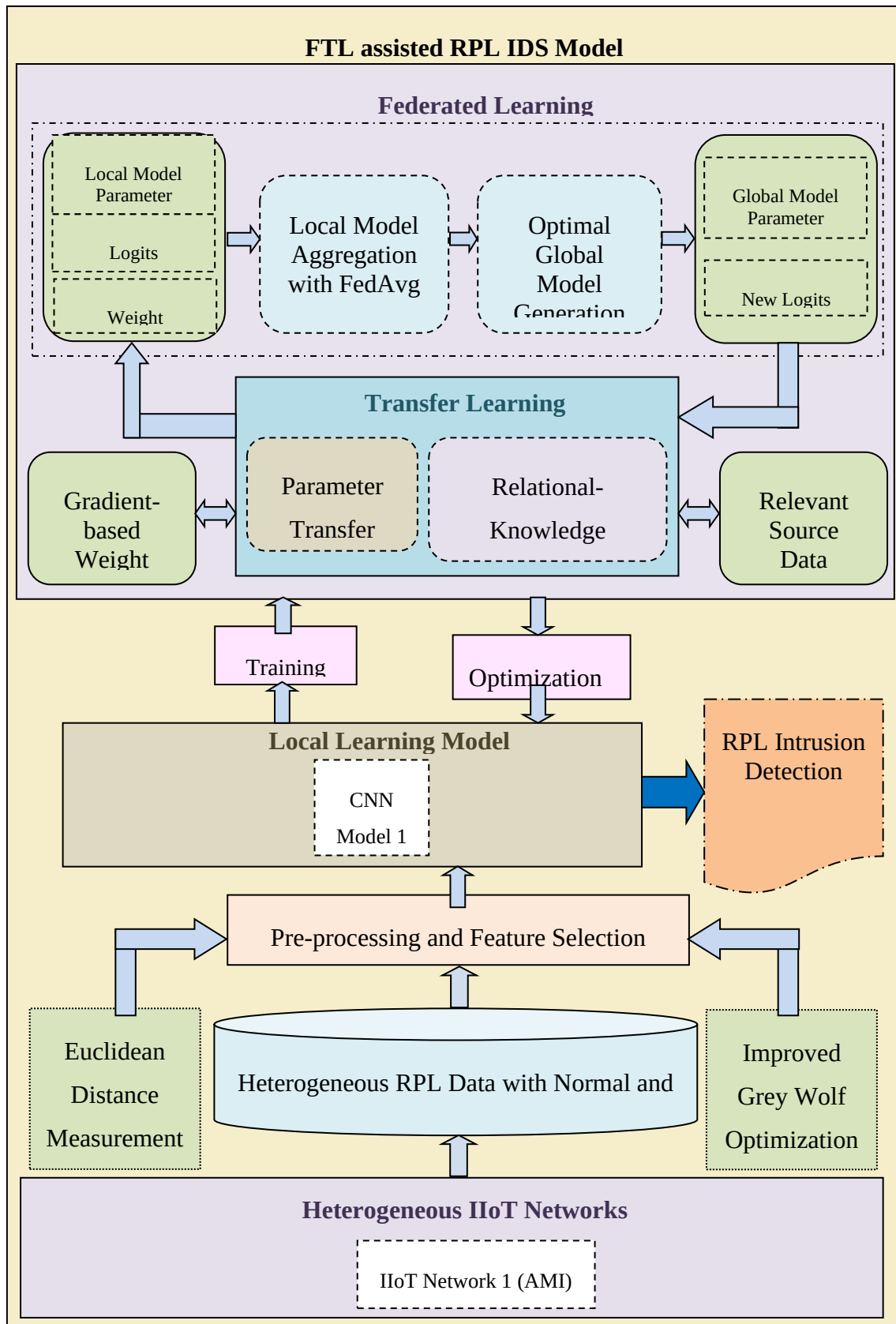


Figure 5-3: Process Flow of Proposed FT-CID Model

Figure 5.3 shows the process involved in the proposed FT-CID model. Firstly, the FT-CID model collects RPL data traffic with normal intrusion samples from the heterogeneous IIoT networks. The model pre-processes the collected traffic samples using Euclidean distance measurement to remove noisy data and then selects potential features from the pre-processed data from each IIoT network with the assistance of the improved grey wolf optimisation (IGWO) algorithm. The model then adopts the CNN model for local learning and generates logits for a local dataset of each IIoT network. The FT-CID model uses TL to establish communication between the local learning model in the sensor and the global learning in the server. After training the local model in the edge of the IIoT networks, the FT-CID model utilises the TL model with parameter transfer and relational knowledge transfer to update the global model. In the FT-CID, the TL model is associated with gradient-based weight generation for each local learning model and transfers the local learning model parameters, logits, and weights to the global learning model with the help of Federated Averaging (FedAVG)-based aggregation. As a consequence of training the global model using the transferred parameters, the FT-CID model optimises each local learning model by transferring knowledge, including the global parameters and logits, through TL associated with parameter transfer and knowledge transfer. Moreover, it selects the relevant source data to train the TL model with respect to the target domain of each IIoT network and thus optimises the local learning model with the assistance of the SGD model to detect intrusion samples in the heterogeneous IIoT networks.

5.5.1 Dataset Construction

The RPL-IIoT has a distributed and hierarchical system model that necessitates personalised and distributed security solutions instead of a centralised security model. Hence, the proposed FT-CID model relies heavily on the RPL data collected from the different IIoT application domains to build a distributed security protocol. As a consequence of the ethical, technical, and economic complications of modelling a real IIoT network infrastructure with real-time operation, building or generating simulation datasets is vital for enabling intrusion detection in an IIoT environment [9], [10]. Hence, this work designs IIoT network architecture through simulation on the Contiki-NG OS. In the simulation design, the Cooja environment emulates IIoT network with real CPU

and memory power as the host workstation to generate normal and intrusion samples. The simulation design of the IIoT network is based on the considerations of portability and applicability in a real-world scenario. The data collection process is no different for a heterogeneous IIoT network and generates the same feature space due to the hardware and protocols used by all the nodes in the IIoT network remaining the same. In the FT-CID model, the data are collected from the IIoT network by involving AMI. However, as the FT-CID builds IIoT dataset with various data samples and similar features, this may result in redundant and non-informative data. Inputting the collected data straight to the learning process escalates the time and maximises the algorithm complexity. Hence, to ensure the quality of decision making and reduce training time and algorithm complexity, it is crucial to preprocess the data and select optimal feature subsets in the IDS.

5.5.2 Pre-processing and Feature Selection

As a result of the enormous amount of erroneous and redundant data in the IIoT datasets collected, removing such data from the datasets improves the accuracy of the IDS and reduces the burden on the IDS learning process. Firstly, the FT-CID model applies Euclidean distance measurement to the raw dataset to remove erroneous and redundant data. The FT-CID model then employs IGWO on the dataset (D_{HIIoT}) to select the optimal feature subsets for each IIoT network [11]. IGWO is a type of swarm intelligence algorithm in which optimisation is based on the three best solutions for food searching and hunting behaviours. According to the IGWO algorithm, the proposed FT-CID model obtains the best solutions for three wolves from the pre-processed data based on the hunting behaviour using Equation (1).

$$X(t + 1) = \frac{(x_\alpha - A_1 * |C_1 * X_\alpha - X|)(x_\beta - A_2 * |C_2 * X_\beta - X|)(x_\delta - A_3 * |C_3 * X_\delta - X|)}{3} \quad (1)$$

In Equation (1), A_1, A_2, A_3, C_1, C_2 , and C_3 are all random vectors. Where, α, β , and δ are the best, second-best and third-best positions of the three wolves, respectively. In every round, δ represents the worst positions of neglected features. Finally, the FT-CID model selects the optimal feature subset based on the values of search optimisers α, β , and δ , and the selected feature subset is provided as the input to the CNN model of IIoT1. Furthermore, it repeats the IGWO-based feature selection process on all the

heterogeneous IIoT network to enforce the learning process of the deep learning model on the local dataset: AMI.

5.5.3 Optimised FTL assisted Edge IDS Learning

Training the learning model with an enormous amount of centralised data is not possible in distributed IIoT domains due to the privacy constraint. Although FL-based models address the privacy constraint effectively, an inadequate number of standard datasets and a lack of considering the unique characteristics of the IIoT have led to inaccurate intrusion detection. Hence, the FT-CID model integrates the concept of TL with FL to build accurate IDS models with minimum learning time, with the advantage of utilising the learned knowledge of a source domain in various target domains in the TL model. The FT-CID model adopts FTL to resolve the security issues associated with a heterogeneous IIoT environment. To handle the heterogeneous data in the distributed IIoT networks, the FT-CID model exploits the FL and TL models to aggregate the attack data of the various RPL-IIoT network and ensure the design of customised edge IDS models, respectively. Federated learning consists of two learning models: server-side and edge-side. During network initialisation, the central security server trains its global model on a publicly available source dataset and initialises the global learning model.

The edge devices execute the distributed local learning models for the IIoT network dataset. With the assistance of TL, the edge devices transfer local models in terms of model parameters, such as the number of hidden layers, hidden neurons, and activation functions, logits in terms of probability values (the vector of raw predictions), and weights from the learning model to the central security server. Instead of sending the entire data to the central server for global model generation, the proposed FT-CID model transfers only the logit outputs of the specified CNN layer of the local learning models and sends the local information obtained as knowledge to the server. Moreover, the proposed model optimises the parameters of the local learning model in terms of the number of iterations based on the gradient obtained from the local learning results. During aggregation in the FT-CID model, the server aligns the local parameters in the local model by employing the degree of importance of factors with the attention mechanism and applies FedAVG [13] to the received models of different edges to generate a global model. The FedAVG method is responsible for updating the model multiple times before sending the learning

model back to the central or global server for aggregation. In the FL model, it ensures that the central server aggregates the updated parameters from the sensors to update the global model in each communication round. With the centralised characteristics of the FedAVG model, there is a need for a very high number of communication rounds between the sensor and the server in order to collect the model parameters of the sensors and distribute the parameters to each sensor. Finally, the global parameters are sent back to the edge of the IIoT network to optimise the local learning models and detect intrusions.

5.5.3.1 Distributed Local Model Training with Parameter Transfer

The FT-CID model first builds the local learning model at the edges of heterogeneous IIoT network for training the IIoT data samples with the optimal feature set. In the FT-CID model, the IDS located at the edge of the IIoT network exploits the CNN model to perform the local learning process on the data samples of the different IIoT networks. With the advantage of a learning model with a minimal number of parameters in the CNN, the proposed FT-CID adopts the CNN deep learning model for intrusion detection. The network traffic of the optimal features of the HIIoT is provided as input to the CNN. In the FT-CID, the CNN structure comprises the following layers: input, convolution, pooling, flattening, dropout, full connection, softmax, and output. In the concept behind the TL model, layers are frozen during learning, whereas the flattening, dropout, and full connection layers are used to transfer parameters along with the logits and weights. To retain the optimal set of features, the FT-CID model freezes the convolution and pooling layers in the CNN model. Furthermore, the softmax layer is used for optimising the local learning process.

In the FT-CID model, the local learning model results in output in the form of logits. The FT-CID builds CNN local learning models, CNN1, using the datasets D1. Further, the updated local learning outputs of the fully connected layers with weights are fed as input to the global learning process through TL. The FT-CID model optimises the parameters of the local learning model with the optimal number of iterations based on the gradient of the learning results. The optimisation of the local learning model relies on weighted aggregated-based globally shared parameters and attention mechanisms. By monitoring the gradient of each local and global learning model iteratively, the FT-CID model customises the local learning model with the optimal number of iterations. Moreover, it

transfers the parameter of the local learning model by the degree of importance of each model, as well as transferring the logits and weights to generate the global model.

5.5.3.2 Global Model Generation with Relational Knowledge Transfer

Following local model learning, it is crucial to transfer or update the specified local parameters to the server in order to build a global shared model in the FTL-based IDS. The TL model aims to transfer the local parameters generated with gradient-based weights from the edges to the central server. The TL partially retrains the predefined learning of the central server, which limits the complexity of the algorithm and increases the accuracy of the global model generation. In the FT-CID model, the local learning models comprise local parameters with the logits, which are between 0 and 1, considered as relational knowledge value. In the local learning model, the parameter transfer involves the transfer of the learning model parameters, logits, and weights. The local learning model parameters and logits mainly refer to the patterns learned from the known and unknown types of intrusions of every RPL-IIoT domain, and the weights denote the importance of the local learning model parameter for the results. The logit of each local model is updated to the security sensor-server using a TL algorithm. The TL process with the parameter transfer and the relational-knowledge transfer during the local-to-global updating and global-to-local optimisation processes is shown in Figure 5-4. After the global parameter modelling, the FT-CID model carries the global intrusion patterns in terms of new logits and weights to the edges for local learning model optimisation with the assistance of the TL model.

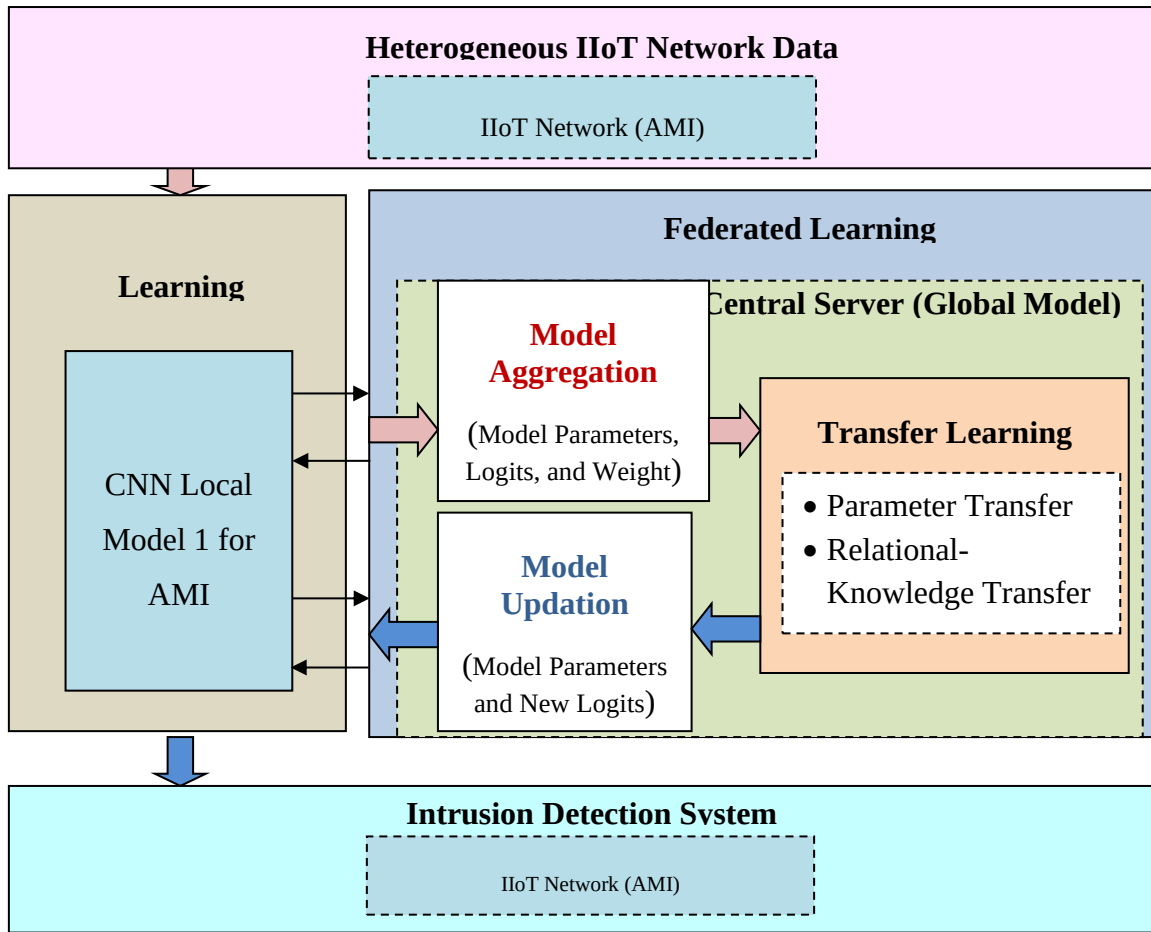


Figure 5-4: Federated Transfer learning in FT-CID

The FT-CID model builds the global shared model with the weighted average of the local learning models using FedAVG for the heterogeneous IIoT edges. The FedAVG computes the differing importance of weight degree values to each local model. Hence, the edge of each IIoT network has a different importance value in the FT-CID model based on the impact of the local learning model parameters on the global model and the gradient. Hence, the FedAVG computes the standard importance degree for each edge of the IIoT network based on the local model contribution parameter weights and gradients and thus optimises the updating of the global shared model. The local model receives high weights when the contribution is high with the assistance of the attention weight modelling. As a result, the FT-CID server not only aggregates the local parameters, it also considers the weights of the local learning model to accelerate the global model convergence. Let the central server receive local model updates from ‘n’ different edges through TL, and it starts to retrain its predefined model based on the significance of the

local models. For instance, the local models of different edges are denoted as $LM=\{CNN1\}$, the corresponding weights are denoted as $WLM=\{W1\}$, and the global model on the centralised server is modelled using Equation (2) across ‘M’ number of IIoT networks.

$$W_{GM} = \sum_{E|LM=1}^M [|N|_{LM} * h_E(i-1) * W_{LM}] \quad (2)$$

Where, ‘N’ refers to the total number of samples in each local learning model or IIoT network, and W_{GM} refers to the weight of the global learning model. In the FT-CID model, the weight of the global model relies on the weight of the local learning model determined by the contribution level of each local model on global level optimisation through attention weight modelling. By applying Equation (3), the FT-CID model measures the influence of each local learning model on the global learning model.

$$d(W_{GM}(i), W_{LM}(i)) = \sqrt{\sum_{i=1}^n ((W_{GM})_i - (W_{LM})_i)^2} \quad (3)$$

Where, $d(W_{GM}, W_{LM})$ refers to the difference between the weights of the global shared model and each local model during iteration (ith), where ‘i’ is varied from 1 to n. Further, ‘d’ is used to estimate the contribution of the local model parameters of each sensor to the global model optimisation. FL requires multiple iterations to generate a global model. Hence, the FT-CID model optimises the iterations with the help of attention mechanism-based global model updation.

$$h_E(i) = \frac{M \times w_E(i)}{\sum_{E=1}^M w_E(i)} \quad (4)$$

The FT-CID model minimises the communication rounds for convergence with the influence of the attention mechanism using Equation (4). Where, ‘M’ denotes the number of local models or network environments and $w_E(i)$ refers to the weight of the local learning model parameters from the environment (E) for each ith iteration. In the FT-CID model, the gradient and dimensional difference-based attention mechanism limits the number of communication rounds and enhances the convergence speed without degrading

the aggregation efficiency. The global shared model is generated with new logits and sent to each edge IDS to customise the local model.

$$w_E(i) = f_{\text{sigmoid}} \left[\frac{d(W_{GM}(i), W_{LM}(i))}{\sigma_E(i)} \right] \quad (5)$$

The FT-CID model formulates Equation (5) to compute the gradient ($\sigma_E(i)$) and dimensional difference-based normalised result to update the attention weight of the learning model at each i th iteration. In Equation (5), the gradient refers to the results of the local learning model on the local datasets, which assists in aggregating the local model parameters within the optimal number of iterations until reaching the optimal convergence.

5.5.3.3 SGD-TL-based Local Model Optimisation

The FT-CID model uses the global model knowledge and logits to optimise the local learning model with partial retraining on the local learning model. During the local model optimisation, the new logits comprise the global knowledge of unknown attacks of the various RPL-IIoT networks. Moreover, the FT-CID model selects the relevant source domain for building the pre-trained FTL model with the aid of the SGD method. The FT-CID inputs the new logits to the softmax layer of each local CNN and partially retrains the local learning model based on the global knowledge, resulting in high intrusion detection accuracy. The TL model of the FT-CID between the server and sensor edge of the IIoT network utilises the SGD model to optimise the local learning model. To address the constraints of bandwidth-limited networks, the FT-CID aims to apply the standard optimisation methods of the SGD on the local models, rather than using the SGD iterations to update the global model. In the FT-CID model, the sensor updating relies on the results from the multiple SGD iterations on its local datasets and then establishes communication with the server with reference to the local models, which is intended to reduce the number of communication rounds. SGD is an iterative method that optimises the local learning model based on the measure of variance between the server and the specified edge. The variance (D_V) reflects the difference between the source and target domain datasets in the concept of TL, in which the data pre-trained on the server refer to the source domain, and the local data on the IIoT network refer to the target domain. In Equation (6), Lg_S and $IIoT_T(E)$ denote the new logits in the S th source domain and local

data in the Tth target domain of the IIoT network. Finally, the FT-ID appends the distance value in the computation of the loss function of RPL IDS learning (L_{LIDS}) to build the customised learning model for the IDS.

$$L_{LIDS}(E) = L(LM_E) + \lambda D^2_v(Lg_S, IIoT_T(E)) \quad (6)$$

The FT-CID model customises each local learning model for the IIoT network based on the loss function or gradient of the learning model during the adaptation of the new logits of the global learning model in the local learning model. In equation (6), ‘ λ ’ balances the task of the intrusion detection between the distance of the source and the target data and $L(LM_E)$ refers to the loss of the individual Eth local learning model, which is used to update the loss value of the learning model. The FT-CID can then optimise the learning process of the local CNN of IIoT network with the relevant global knowledge. The learning process of the FTL-enabled IDS is outlined in the Process Flow diagram shown in Table 5.1.

Table 5-1: FTL assisted RPL IDS Learning

//FTL-assisted RPL IDS Learning Process//	
Input:	Local parameters of different IIoT network and global shared knowledge
Output:	Efficient Learning
The Edge IDS do {	
	Collects data samples from each RPL-IIoT network;
	Performs CNN local learning;
	Optimises iterations with attention
	Obtains the local parameters of a fully connected CNN layer;
	Initiates Transfer Learning Process;
TL {	
	Transfers the relational knowledge from edge to server;
Server do {	
	Estimates degree of importance of each IIoT local parameter;
	Aggregates the data using equations (2) and (3);
	Generates attention weight of global model using equations (4) and (5)
	Optimises iterations with attention mechanism;
	Generates a global shared model with new logits;
	Sends back to the edges through transfer learning;
	}
	Transfers the new logits to the softmax of local edge IDS;
	Optimises the local learning process using equation (6);
}}	

In the subsequent CNN local model optimisation, the FT-ID model initiates the intrusion detection task for the corresponding local datasets in the heterogeneous IIoT network. By learning the patterns of the upcoming testing traffic data with the customised local learning model, the FT-CID model first classifies the data as normal and abnormal samples as intrusions. By implicitly exploiting the global attack knowledge in customised IDS learning, the proposed FT-CID model detects and classifies the intrusions over a heterogeneous IIoT environment. With the potential advantage of the global knowledge of various unknown attacks on different RPL-IIoTs, the FT-CID model significantly improves the quality of intrusion detection without compromising the RPL network performance.

5.6 Experimental Evaluation

To demonstrate the performance improvement of the proposed FT-CID model, the experimental model evaluates the proposed FT-CID model through the comparison of baseline models of different learning algorithms based on an IDS. It selects the baseline models, logistic regression (LR), multilayer perceptron (MLP), and CNN-based RPL-IIoT IDS due to the lack of existing research work in the area of FTL-based RPL-IIoT IDSs.

5.6.1 Experimental Setup

To implement an RPL-IIoT intrusion detection algorithm, the experimental model conducts experiments using the Python programming language. The FT-CID model runs on Ubuntu 18.04 LTS with an Intel i5 3.00GHZ CPU and 16 GB memory. The CNN-based FTL-IDS model is implemented using an FL model with the Keras API. Using heterogeneous IIoT networks, the experimental model demonstrates the intrusion detection process AMI network. The experimental model implements the classification algorithms on a centralised and federated environment setup. The centralised environment setup refers to the training and decision making by the local learning model on the local IIoT network themselves, whereas the federated environment setup refers to the training and collaborative decision making by the local and global learning models on the local IIoT networks. The centralised setup restricts several of the learning models trained on the entire heterogeneous RPL-IIoT dataset. The experimental model conducts experiments on the centralised setup for the LR, MLP, and CNN. In the centralised setup,

the aggregator function is the arithmetic mean of the prediction probabilities of each learning model on the two heterogeneous RPL-IIoT networks (AMI). Moreover, the experimental model runs the CNN algorithm in the federated setup associated with the procedures involved in the FT-CID model.

5.6.2 Dataset Description

To evaluate the RPL-IIoT intrusion detection in heterogeneous IIoT networks, the experimental model generates RPL-IIoT dataset with normal and abnormal traffic for IIoT network using Cooja Simulator. The dataset generation process is for a heterogeneous IIoT network (i.e., AMI) with different samples and the same feature space. With the support of Contiki-NG, the RPL-IIoT dataset samples comprise the intrusion and normal samples, in which the intrusions involve rank, version, hello flood, DoS, blackhole, Sybil, selective forwarding, sinkhole, theft, and wormhole attacks.

5.6.3 Performance Metrics

To validate the performance of the FT-CID model, the experimental model employs the widely used evaluation metrics described below.

Precision: This is the ratio of the total number of samples correctly classified as intrusions to the total number of samples detected as intrusions expressed as a percentage.

$$\text{Precision} = \left[\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \right]$$

Recall: This is the ratio of the total number of samples correctly classified as intrusions to the total number of actual intrusions expressed as a percentage.

$$\text{Recall} = \left[\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \right]$$

F-measure: This is the harmonic mean of precision and recall expressed as a percentage.

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Intrusion detection accuracy: This is the overall accuracy of correctly identified normal and abnormal traffic samples expressed as a percentage.

$$\text{Accuracy} = \left[\frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \right]$$

False Positive Rate: This is the ratio of legitimate samples incorrectly classified as intrusions to intrusions incorrectly classified in the corresponding attack category expressed as a percentage.

$$\text{False Positive Rate} = \left[\frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \right]$$

Where,

- True positive (TP) = Number of samples correctly detected as intrusion samples.
- True negative (TN) = Number of samples correctly detected as normal samples.
- False positive (FP) = Number of samples incorrectly detected as intrusion samples.
- False negative (FN) = Number of samples incorrectly detected as normal samples.

5.7 Experimental Results

To illustrate the performance of the proposed FT-CID model against the baseline IDS models, the experimental model evaluates the RPL-IIoT AMI dataset while varying the scenarios of the training data size and the number of communication rounds. To validate the results, we compared our FT-CID model in terms of accuracy with different algorithms such as MLP, LR, and CNN on the centralised setup, to show the significant improvements in the intrusion detection accuracy. This section also considers the influence of the communication rounds on the performance of intrusion detection.

5.7.1 Training Data Size Vs. Precision

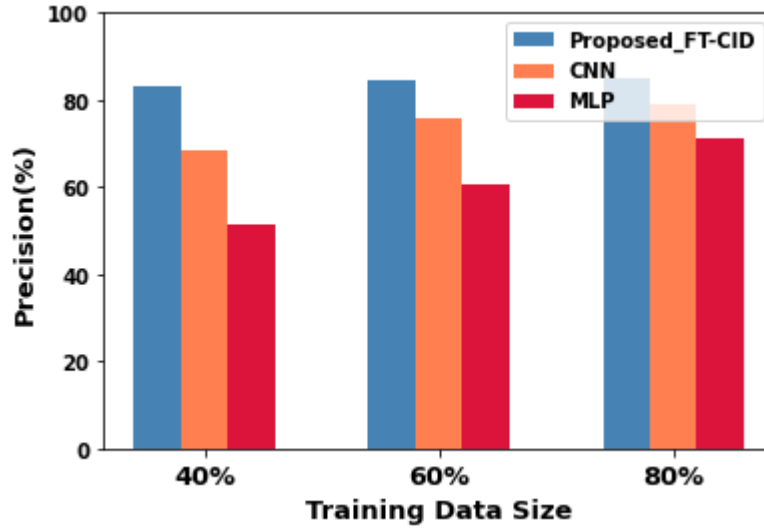


Figure 5-5: Performance of Precision for FT-CID, CNN, and MLP

Figure 5.5 illustrates the effect of a varying number of training samples on the precision performance of the FL setup of the proposed FT-CID model by comparing the CNN and MLP algorithms in the centralised setup with the average score. The performance of the intrusion detection model improves with the increases size of training samples. Precision performance is obtained for 20 communication rounds in the AMI associated federated environment. The proposed FT-CID achieves the precision value of 85.18%, and the precision values of the CNN and MLP models are 78.92% and 71.09%, respectively. As the size of the training data increases, the precision of both the proposed intrusion detection model and the baseline algorithm-based intrusion detection model increases. Compared with the CNN in the centralised setup, the CNN model in the proposed federated setup shows a 14.33 percentage point improvement in precision, even when there is only 40% of the local learning model with the assistance of the parameter transfer and relational knowledge transfer-associated transfer learning model. By inherently analysing the multiple training data. The proposed intrusion detection model learns the attack patterns from the heterogeneous IIoT network with the updating of the global model and optimisation of the local learning model with the assistance of the parameter transfer- and relational knowledge transfer-associated TL model. By analysing the multiple attack patterns with the global and local knowledge, the FT-CID model shows greater accuracy in distinguishing the intrusion samples from the normal samples and and

has higher precision than the existing CNN and MLP models. Moreover, the FT-CID model customises the learning model based on the gradient of the local learning model and the knowledge from the global model, which tends to improve the performance of the comparative models. As a result, the customised modelling of IIoT network tends to yield better TPs in the IDS even when there is a lack of available intrusion patterns in the training data of the local IIoT network. Table 5.2 presents the numerical points of Figure 5.5.

Table 5-2: Comparative Experimental Results of Precision

Training Data Size (%)	Precision (%)		
	Proposed FT-CID	CNN	MLP
40	82.97	68.64	51.25
60	84.32	75.63	60.83
80	85.18	78.92	71.09

5.7.2 Training Data Size Vs. Recall

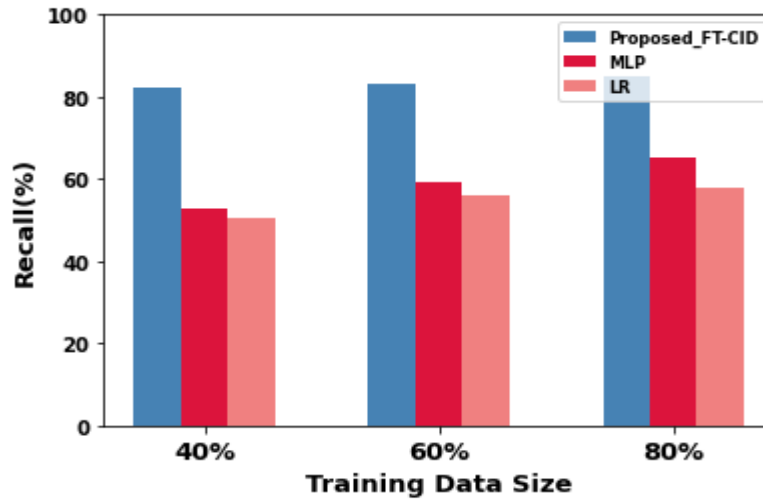


Figure 5-6: Performance of Recall for FT-CID, MLP, and LR

As shown in Figure 5.6 and Table 5.3, the influence of training data size on recall performance is significantly higher in the RPL IDS. The FT-CID achieves higher recall performance on all three different training data sizes in the federated setup (40%, 60%,

and 80%) and for 20 communication rounds in the AMI heterogeneous IIoT network. MLP is ranked second, and LR recall is third, 7.45 percentage points lower than the MLP and 27.03 percentage points lower than the FT-CID. The proposed FT-CID model improves the intrusion detection performance, particularly in small training samples. From Figure 5.6, it can be observed that the recall obtained by the proposed FT-CID model is better than the MLP and LR models for IIoT dataset for all sizes of training data. By modelling the federated environment for the heterogeneous IIoT application, the performance of the FT-CID is better than the ML algorithms of the MLP and LR on IIoT dataset on average results. Moreover, the FT-CID yields 23.7 and 27.9 percentage points higher recall, respectively, than the baseline MLP and LR models when there is 60% of training data, due to the advantage of gaining knowledge from other IIoT datasets, as well as the source dataset, of the attack patterns during the global model training and updation.

Table 5-3: Comparative Experimental Results of Recall

Training Data Size (%)	Recall (%)		
	Proposed FT-CID	MLP	LR
40	82.12	52.95	50.62
60	83.08	59.38	55.99
80	84.94	65.36	57.91

5.7.3 Number of Communication Rounds Vs. F-Measure

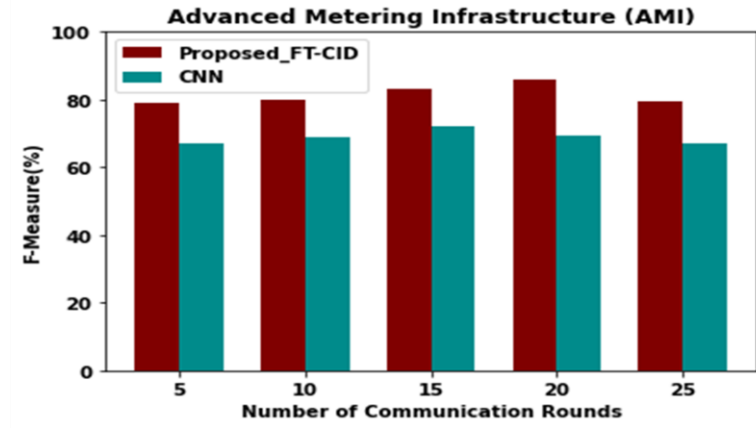


Figure 5.7 (a): Performance of F-Measure in AMI for CNN and FT-CID

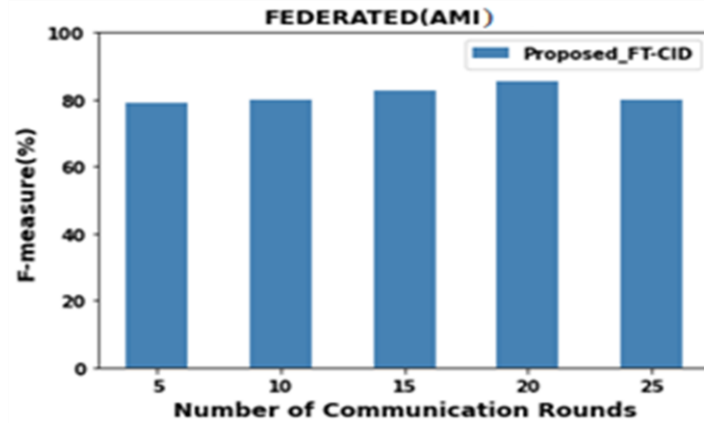


Figure 5.7 (b): Performance of F1-Measure

Figure 5-7: Performance of F-Measure

Figure 5.7 compares the F-measure of the proposed FT-CID model with the neural network models on local IIoT environments (i.e., AMI). Figure 5.7 (b) shows the F-measure of the proposed intrusion detection model, the FT-CID, with various communication rounds (5, 10, 15, 20, and 25) and for 80% of the training data size. From the analysis of the results in Figure 5.7 (a)-(b), it can be determined that both the centralised and federated intrusion detection models tend to converge after a certain

number of communication rounds. Thus, the proposed intrusion detection model has a 16.83% higher F-measure than the baseline of the CNN on the independent AMI IIoT environment due to the advantage of collaboratively utilising the knowledge from the heterogeneous IIoT environments for 20 communication rounds. As the number of communication rounds increases from 5 to 25, the F-measure performance of each intrusion detection model increases and gradually stabilises, and the performance is reduced after reaching a sufficiently large number of iterations or communication rounds. Moreover, Figure 5.7 (b) shows the F-measure performance of the FT-CID model with the average score of the AMI heterogeneous IIoT environment for the various numbers of communication rounds. The data values from the implementation F-measure results are provided in Table 5.4.

Table 5-4: Comparative Experimental Results of F-Measure

Number of Communication Rounds	F-Measure (%)		
	Centralised		Federated
	AMI		AMI
	Proposed FT-CID	CNN	Proposed FT-CID
5	79.23	67.23	78.97
10	80.16	68.98	80.12
15	83.296	72.15	82.72
20	85.91	69.15	85.75
25	79.56	67.16	79.98

5.7.4 Intrusion Detection Accuracy

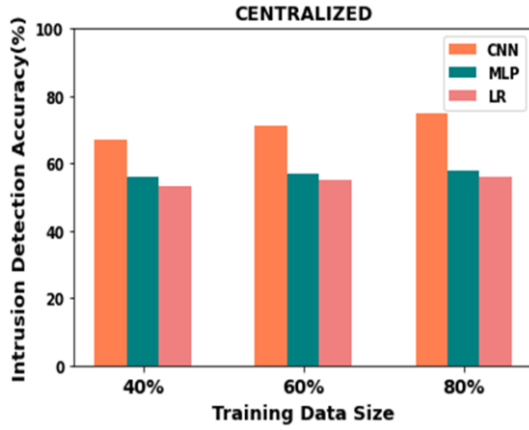


Figure 5.8 (a): Intrusion Detection Accuracy for Centralized Setup - CNN, MLP, and LR

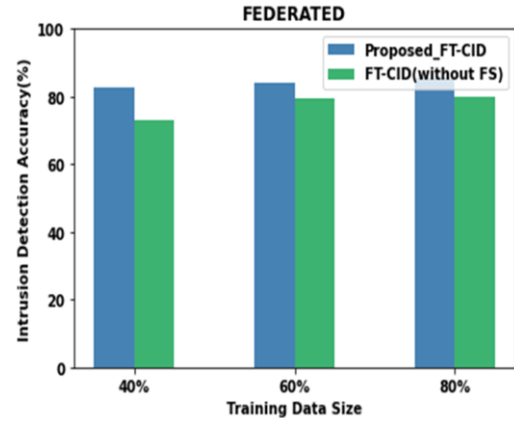


Figure 5.8 (b): Intrusion Detection Accuracy for Federated Setup – FT-CID with FS and FT-CID Without FS

Figure 5-8: Performance of Intrusion Detection Accuracy

Compared with the CNN and MLP neural network models, the FT-CID significantly improves the detection accuracy while being tested in heterogeneous IIoT environment. Furthermore, from the experimental results, it is recognised that the neural network models have stronger intrusion detection performance than the ML, MLP and LR algorithms. The FT-CID obtains the best detection performance even when there is a minimum number of training data sizes. Figure 5.8 and Table 5.5 show the implementation results of the performance of the centralised and federated intrusion detection models in terms of accuracy under scenarios of different percentages of training data size and for 20 communication rounds. From Figure 5.8 (b), it can be seen that the proposed FT-CID model with the integration of IGWO-based feature selection (FS) outperforms the FT-CID model without FS in terms of 5.49% higher intrusion detection accuracy when there is 80% of training data. As evident from Figure 5.8, the performance accomplished by the FT-CID is better than that of the neural network and ML models of the CNN, MLP, and LR in centralised settings. By applying the IGWO algorithm, the FT-CID model selects the optimal set of features from the raw IIoT dataset, and thus ensures

improved detection accuracy when compared to the detection of intrusions by the MLP and LR models without FS of 22.7% and 24.69% 60% of training data, respectively.

Table 5-5: Comparative Experimental Results of Intrusion Detection Accuracy

Training Data Size (%)	Intrusion Detection Accuracy (%)				
	Centralised			Federated	
	CNN	MLP	LR	Proposed FT-CID	Proposed FT-CID (Without FS)
40	67.16	55.86	53.25	82.83	73.24
60	71.23	56.95	54.96	84.17	79.65
80	74.89	58.04	56.12	85.52	80.03

5.7.5 Training Data Size Vs. False Positive Rate

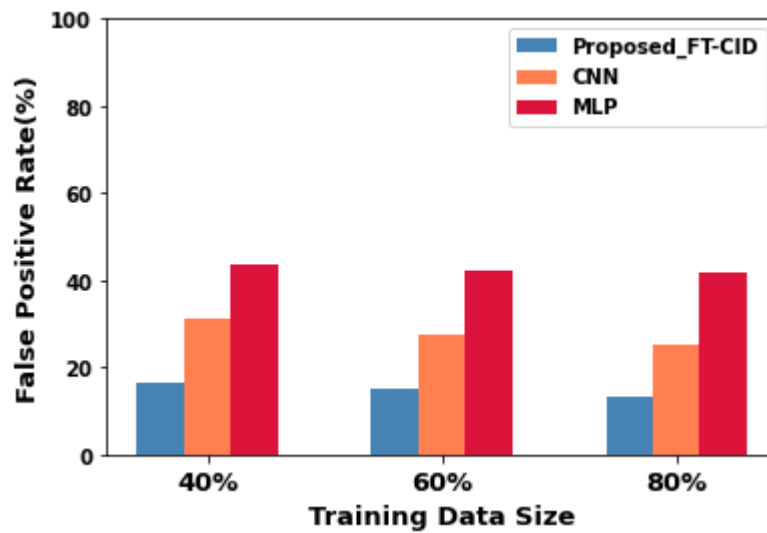


Figure 5-9: Performance of False Positive Rate for FT-CID, CNN, and MLP

Figure 5.9 shows the FP rate of the proposed FT-CID and baseline CNN and MLP models under different training data sizes and 20 communication rounds in AMI IIoT environment. As the size of the training data increases, the FP rate of the FT-CID, CNN and MLP decreases. When the training data size is 80%, the FP rate of the FT-CID is still below 16%, but the FP rates of the CNN and MLP are higher at 24.98% and 41.76%, respectively. The principle behind the above phenomenon is that the proposed FT-CID learns the related attack patterns from a similar intrusion detection task using FL and TL methods. Moreover, the FT-CID optimises the local learning model with the assistance

of SGD and relevant knowledge learned from the global model. Consequently, the FT-CID improves the intrusion detection performance with a reduced number of FPs in the target task and maintains higher intrusion detection accuracy even with a small training data size. The numerical points obtained from the experiments for the FP rate are shown in Table 5.6.

Table 5-6: Comparative Experimental Results of False Positive Rate

Training Data Size (%)	False Positive Rate (%)		
	Proposed FT-CID	CNN	MLP
40	16.57	31.251	43.74
60	15.01	27.36	42.17
80	13.29	24.98	41.76

5.8 Chapter Summary

This chapter presented the design of the proposed customised IDS for heterogeneous RPL-IIoT network with the assistance of a deep FTL model. Firstly, the proposed FT-CID model removes the noise and filters irrelevant features from the simulated RPL-IIoT dataset by modelling the IGWO algorithm. In the FT-CID model, the deep FTL algorithm relies heavily on the CNN algorithm to build local models on IIoT network: global model updating and local model optimisation. The logit output of each local model greatly influences the updating of the global model in the federated server, and thus the logits in the global model assisted the optimisation of the local models in the sensors or the IIoT networks. In the FT-CID model, the TL model involves parameter transfer and relational knowledge transfer during the global model updating and local model optimisation with the SGD model. Thus, the experimental results illustrate that the proposed FT-CID model yields 10.63% higher intrusion detection accuracy than the centralised CNN model on the heterogeneous IIoT network or the AMI applications.

5.9 References

1. Nižetić S., Šolić P., López-de-Ipiña González-de-Artaza D., Patrono L. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*. 2020; 274. Available at: DOI:10.1016/j.jclepro.2020.122877
2. Varga P., Peto J., Franko A., Balla D., Haja D., Janky F., et al. 5G Support for Industrial Iot Applications – Challenges, Solutions, and Research Gaps. *Sensors (Switzerland)*. 2020; 20(3). Available at: DOI:10.3390/s20030828
3. Thakkar A., Lohiya R. A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. *Archives of Computational Methods in Engineering*. Springer Netherlands; 2021. 3211–3243 p. Available at: DOI:10.1007/s11831-020-09496-0
4. Seyfollahi A., Ghaffari A. A Review of Intrusion Detection Systems in RPL Routing Protocol Based on Machine Learning for Internet of Things Applications. *Wireless Communications and Mobile Computing*. 2021; 2021. Available at: DOI:10.1155/2021/8414503
5. Simoglou G., Violettas G., Petridou S., Mamatas L. Intrusion detection systems for RPL security: A comparative analysis. *Computers and Security*. Elsevier Ltd; 2021; 104. Available at: DOI:10.1016/j.cose.2021.102219
6. Imteaj A., Thakker U., Wang S., Li J., Amini MH. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet of Things Journal*. IEEE; 2022; 9(1): 1–24. Available at: DOI:10.1109/JIOT.2021.3095077
7. Khan LU., Saad W., Han Z., Hossain E., Hong CS. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys and Tutorials*. IEEE; 2021; 23(3): 1759–1799. Available at: DOI:10.1109/COMST.2021.3090430
8. Wang KI., Zhou X., Liang W., Yan Z., She J. Cross-Domain Prediction for Smart Manufacturing. *IEEE*; 2022; 18(6): 4088–4096.

9. Al-Hadhrami Y., Hussain FK. Real time dataset generation framework for intrusion detection systems in IoT. *Future Generation Computer Systems*. Elsevier B.V.; 2020; 108: 414–423. Available at: DOI:10.1016/j.future.2020.02.051
10. Ullah I., Mahmoud QH. A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. 2020; 2020-October(May): 134–140. Available at: DOI:10.1109/SMC42975.2020.9283220
11. Nadimi-Shahraki MH., Taghian S., Mirjalili S. An improved grey wolf optimizer for solving engineering problems. *Expert Systems with Applications*. Elsevier Ltd; 2021; 166(August 2020): 113917. Available at: DOI:10.1016/j.eswa.2020.113917
12. Man, D., Zeng, F., Yang, W., Yu, M., Lv, J., & Wang, Y, “Intelligent intrusion detection based on federated learning for edge-assisted internet of things”, *Security and Communication Networks*, 2021
13. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A, “Communication-efficient learning of deep networks from decentralised data”, In *Artificial intelligence and statistics*, pp.1273-1282, 2017

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This chapter provides a summary of the thesis findings alongside the goal and objectives of the research. This thesis established a comprehensive literature analysis of all aspects of cybersecurity for IoT and industrial IoT application environments, which included its high-level architecture and a categorisation of IoT layers, potential threats, and security countermeasures. A lightweight and secure authentication system for the IoT based on distributed sensors was also developed and presented. Finally, a novel approach was executed for heterogeneous IIoT networks to detect intrusions using machine learning.

It can be concluded that the research findings aligned with the research objectives as follows.

Objective 1. To examine the knowledge and conduct critical analysis of the existing academic literature on IoT cybersecurity to define the basis and requirements of the research.

- **Related conclusion:** Many of the studies in our published review paper focus mainly on reviewing specific IoT security aspects, attacks and layers, and providing only a limited evaluation of the existing security solutions. Therefore, there is still a need for a comprehensive survey of most types of IoT/IIoT security attacks, solutions, and countermeasures. This thesis provided a roadmap for researchers that offers a complete picture and understanding of the state of the art in IoT/IIoT security attacks. It classified the types of attacks and possible security solutions regarding IoT-layer architecture. Each type of attack is connected to one or more layers of the architecture, and the review of the literature was accompanied by an analysis of various IoT security countermeasures. The review also carried out a critical analysis of existing IoT/IIoT solutions based on their different security mechanisms, including the communications protocols, networking, cryptography, and intrusion detection systems.

Objective 2. To implement a new lightweight authentication scheme using optimised datagram transport layer security (DTLS) with advanced encryption standard (AES) algorithms.

- **Related conclusion:** A new optimised DTLS features were integrated with a default CoAP (DF-CoAP) and a lightweight authentication mechanism was implemented with the protocol to ensure secure communication between authorised entities in a distributed sensor network. Secondly, to provide a lightweight and secure authentication scheme, a payload encryption-based handshake process was employed to reduce the number of round trips from three to two. Thirdly, a NACK-based retransmission scheme was added to the DF-CoAP for the exact missing data in the flight, which enabled a reduction in the communication load. The proposed scheme was then able to reduce system complexity drastically by removing the duplicate features of the protocol version and sequence number component of the explicit nonce field from the packet header in the DF-CoAP and introducing the parallel execution of S-Boxes in the SubBytes of the AES. Lastly, the scheme introduced nonce and seed values to generate secret and session keys dynamically, significantly improving DTLS security.

Objective 3. To develop a novel detection system using a federated learning approach.

- **Related conclusion:** A novel detection approach, called customised intrusion detection (CID), was proposed for heterogeneous IIoT network using federated transfer learning (FTL). The design process of the FT-CID contained three steps: dataset collection, FTL-assisted edge IDS learning, and intrusion detection. First, the central server initialises the FT-CID using a predefined learning model and observes the unique features of different RPL-IIoTs to construct the local model. Secondly, the edge IDSs are trained using the local parameters and the globally shared parameters generated by the central server through the federation and aggregation of the different local parameters of various edges. Hence, transfer learning is exploited to update the local and global parameters between the server and the edges based on relational knowledge. It also builds the customised IDS model through partial retraining through partial using local learning based on the globally shared knowledge of the server. Finally, the customised IDS in the FT-CID model enforces the detection of intrusions in heterogeneous IIoT networks. Moreover, the FT-CID model accomplishes a high level of RPL security by

implicitly utilising the local and global parameters of different IIoTs with the assistance of FTL.

Objective 4. To evaluate the impact of the applied authentication and detection schemes by building simulations and implementing the schemes in terms of system performance.

- **Related conclusions:** First, the efficiency of PEOS authentication was evaluated using the Cooja simulator in the Contiki OS for lightweight security and authentication. The proposed scheme attained better throughput while minimising the message size overhead by 9% and 23% compared with an existing payload-based mutual authentication (PbMA) scheme and a basic DTLS/CoAP in random network topologies with fewer than 50 nodes.

Second, The FT-CID is able to detect RPL intrusions with an accuracy of 85.52% when tested on two heterogeneous IIoT network (AMI) through a customised local learning model.

Objective 5. To validate the proposed system and methods and disseminate the outcome in conference and journal papers.

- **Related conclusion:** This thesis consists of a collection of chapters based on published works. Chapters 2, 4 and 5 have been published in the MDPI *Sensors* journal, and chapter 3 has been published in IEEE Xplore. All published articles were authored by the initial author and examined and approved for publication by two primary supervisors and one associate supervisor, as well as by MDPI *Sensors* and IEEE Xplore reviewers.C

6.2 Contribution to Knowledge

This study contributed to designing cryptography and detection schemes on several performance aspects of heterogeneous industrial IoT networks in accomplishing resilient security systems. Specifically, this contribution offered to develop integrated, efficient security for the industrial IoT network application. Therefore, the knowledge added by this thesis can be summed up as follows:

- To provide a roadmap for researchers, a comprehensive survey on state-of-the-art IoT/IIoT security attacks, solutions and countermeasures has been proposed.

- Considering the cost computation of any proposed security is crucial. From the results, lightweight security help reduce the cost of calculation while dealing with resource-constrained devices. Developed a new mechanism to CoAP integrated with DTLS and AES for providing security and IoT applications with low computation. This will give the development field and industry an integrated algorithm that enables such applications' security features.
- Proposed a novel intrusion detection using a machine learning approach based on federated transference learning: this will give the researcher in the field of cybersecurity to investigate the proposed novel detection by integrating it with other detection approaches such as anomaly detection and hybrid detection

6.3 Future Work

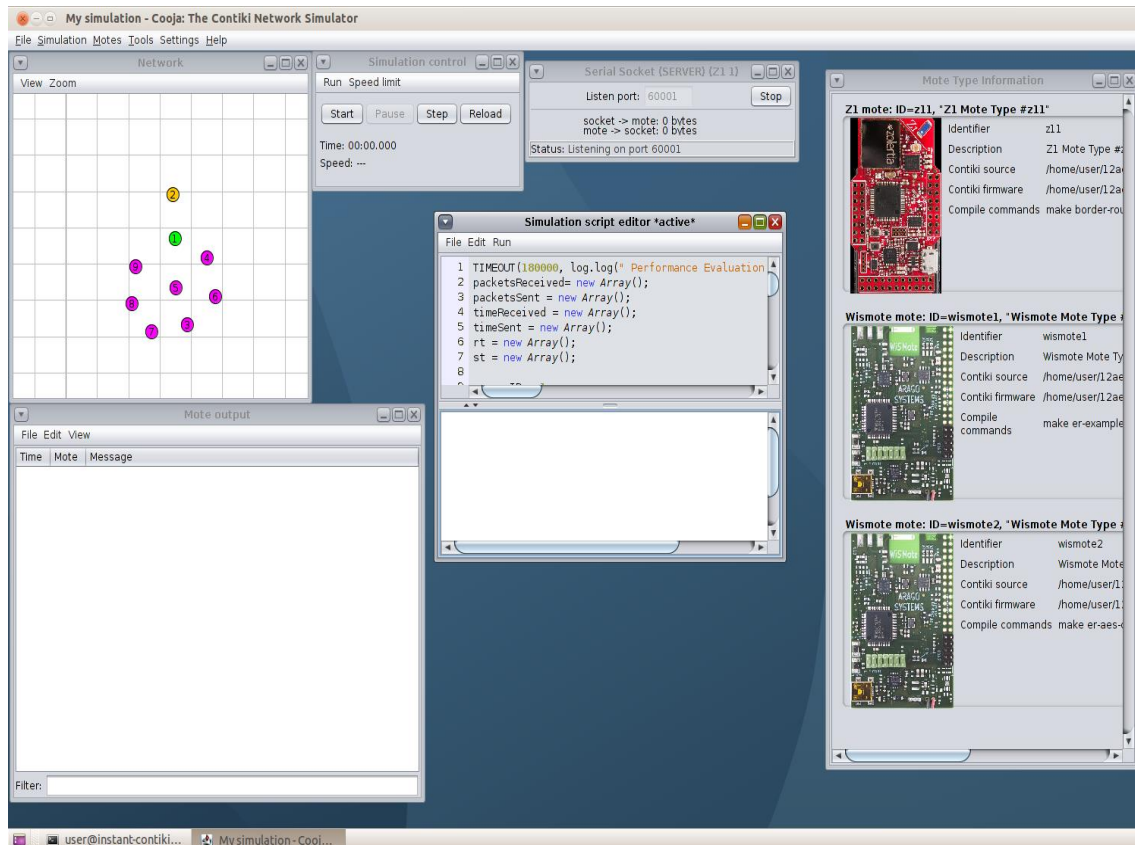
This thesis presented a new architecture to minimise the performance complexity of executing security-based authentication and detection schemes. According to those mentioned above, other research areas within this research domain may provide opportunities for additional study. The following is a presentation of the suggested future research.

- Future research is needed to assess further the robustness of the proposed security scheme in terms of threat models or other possible threats, such as malicious code injection or sleep deprivation attacks.
- Although the comprehensive detection approach developed in this study resulted in signature-based performance patterns over heterogeneous IIoT networks, this was designed to detect attacks but not to prevent them. Hence, future studies could integrate other approaches with our detection method to complement the detection of specific attack types. In addition, there is a need for complete cyber-detection and prevention systems that provides resilience, scalability, precision, and security against all forms of hostile attacks.
- For the detection of hostile attacks in IoT systems, semi-supervised machine and reinforcement learning algorithms have not been thoroughly researched.

APPENDICES

Appendix A : Encryption and authentication

A.1 System model and network architecture



Simulation Details:

Node-count: 32 (1 Border router, 1 Server and 30 Sensor nodes)

-  - Border-router
-  - Server
-  - Sensor

Advanced Encryption Standard (AES) Cryptography Algorithm

Encryption Process

```
test_aes x
53 00:10.305 ID:7 Sending Data Time=10 NodeID= 7
54 00:10.311 ID:7 client_key:2b7e151628aed2a6abf7158809cf4f3c
55 00:10.324 ID:7 encrypted text:4bbb3a58114d37a032636694eba5a37e376e0c69a008e157e344e9c70ef64b0b
56 00:10.336 ID:7 encrypted text in string format:K:XM72cf~7niWdK+~( 0<6<
57 00:10.347 ID:7 Data Encrypted Time 10 ID 7 Encr K:XM72cf~7niWdK+~( 0<6<
58 00:10.351 ID:7 Requested #0 (MID 39742)
59 00:10.581 ID:5 --Toggle timer--
60 00:10.585 ID:5 Sending Data Time=10 NodeID= 5
61 00:10.591 ID:5 client_key:2b7e151628aed2a6abf7158809cf4f3c
62 00:10.604 ID:5 encrypted text:5aa0d438790ceb592d55ceaf618c38d5efb406be5fde0adc2a8526545e34d2ae
63 00:10.612 ID:5 encrypted text in string format:Z8yY-Ua8_
64 00:10.616 ID:5 *&T^4+~( 0<6<
65 00:10.624 ID:5 Data Encrypted Time 10 ID 5 Encr Z8yY-Ua8_
66 00:10.628 ID:5 *&T^4+~( 0<6<
67 00:10.632 ID:5 Requested #0 (MID 39742)
68 00:10.954 ID:4 --Toggle timer--
69 00:10.958 ID:4 Sending Data Time=10 NodeID= 4
70 00:10.961 ID:6 --Toggle timer--
71 00:10.964 ID:4 client_key:2b7e151628aed2a6abf7158809cf4f3c
72 00:10.965 ID:6 Sending Data Time=10 NodeID= 6
73 00:10.971 ID:6 client_key:2b7e151628aed2a6abf7158809cf4f3c
74 00:10.977 ID:4 encrypted text:9c6ae654877245a4c33bf98f0fee99661be8011d81c47024c4ca289e12786667
75 00:10.984 ID:6 encrypted text:961c14a64ea8ac0cbeefa76608f6a291be8011d81c47024c4ca289e12786667
76 00:10.989 ID:4 encrypted text in string format:jTrE;fp$(xfg+~( 0<<
77 00:10.996 ID:6 encrypted text in string format:Nv`j)p$(xfg+~( 0<<
78 00:11.000 ID:4 Data Encrypted Time 10 ID 4 Encr jTrE;fp$(xfg+~( 0<<
79 00:11.005 ID:4 Requested #0 (MID 39742)
80 00:11.008 ID:6 Data Encrypted Time 10 ID 6 Encr Nv`j)p$(xfg+~( 0<<
81 00:11.012 ID:6 Requested #0 (MID 39742)
82 00:11.014 ID:3 --Toggle timer--
83 00:11.019 ID:3 Sending Data Time=10 NodeID= 3
84 00:11.025 ID:3 client_key:2b7e151628aed2a6abf7158809cf4f3c
85 00:11.038 ID:3 encrypted text:ba6ce071a49b46ed16b6b73cc2eeea791be8011d81c47024c4ca289e12786667
86 00:11.049 ID:3 encrypted text in string format:lqF<yyp$(xfg+~( 0<<
87 00:11.061 ID:3 Data Encrypted Time 10 ID 3 Encr lqF<yyp$(xfg+~( 0<<
88 00:11.065 ID:3 Requested #0 (MID 39742)
```

Each node sends encrypted data to server via birder router

Decryption Process

```
test_aes x
86 00:11.049 ID:3 encrypted text in string format:lqF<yyp$(xfg+~( 0<<
87 00:11.061 ID:3 Data Encrypted Time 10 ID 3 Encr lqF<yyp$(xfg+~( 0<<
88 00:11.065 ID:3 Requested #0 (MID 39742)
89 00:13.890 ID:2 handle_incoming_data(): received uip_dataalen=68
90 00:13.900 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0007]:5683
91 00:13.902 ID:2 Length: 68
92 00:13.908 ID:2 Parsed: v 1, t 0, tkl 0, c 2, mid 39742
93 00:13.916 ID:2 URL: K:XM72cf~7niWdK+~( 0<5m
94 00:13.919 ID:2 Payload: Smart-home!
95 00:13.931 ID:2 received cipher
text:4bbb3a58114d37a032636694eba5a37e376e0c69a008e157e344e9c70ef64b0b
96 00:13.945 ID:2 received cipher text in string format:K:XM72cf~7niWdK+~( 0<5m node_id:2
97 00:13.951 ID:2 server_key:2b7e151628aed2a6abf7158809cf4f3c
98 00:13.962 ID:2 Server Decrypted Plaintext:camera systems from node_id:2
99 00:13.966 ID:2 Data Decrypted Time 13 ID 2
100 00:13.978 ID:7 handle_incoming_data(): received uip_dataalen=4
101 00:13.988 ID:7 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0002]:5683
102 00:13.990 ID:7 Length: 4
103 00:13.996 ID:7 Parsed: v 1, t 2, tkl 0, c 132, mid 39742
104 00:13.997 ID:7 URL:
105 00:13.999 ID:7 Payload:
106 00:14.001 ID:7 Received ACK
107 00:14.004 ID:7 Received #0 (0 bytes)
108 00:14.004 ID:7 |
109 00:14.009 ID:7 Got Response Data= (null) NodeID= 7
110 00:14.015 ID:7 ENERGY CONSUMPTION IS 2479 ID 7
111 00:14.015 ID:7
112 00:14.016 ID:7 --Done--
113 00:14.170 ID:2 handle_incoming_data(): received uip_dataalen=68
114 00:14.180 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0005]:5683
115 00:14.182 ID:2 Length: 68
116 00:14.188 ID:2 Parsed: v 1, t 0, tkl 0, c 2, mid 39742
117 00:14.192 ID:2 URL: Z8yY-Ua8_
118 00:14.196 ID:2 *&T^4+~( 0<5m
119 00:14.199 ID:2 Payload: Smart-home!
120 00:14.211 ID:2 received cipher
text:5aa0d438790ceb592d55ceaf618c38d5efb406be5fde0adc2a8526545e34d2ae
```

```
test_aes x
90 00:13.945 ID:2 received cipher text in string format:K:XM72cf~7niWDK+~( 0<Sm node_id:2
97 00:13.951 ID:2 server_key:2b7e151628aed2a6abf7158809cf4f3c
98 00:13.962 ID:2 Server Decrypted Plaintext:camera systems from node_id:2
99 00:13.966 ID:2 Data Decrypted Time 13 ID 2
100 00:13.978 ID:7 handle_incoming_data(): received uip_dataalen=4
101 00:13.988 ID:7 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0002]:5683
102 00:13.990 ID:7 Length: 4
103 00:13.996 ID:7 Parsed: v 1, t 2, tkl 0, c 132, mid 39742
104 00:13.997 ID:7 URL:
105 00:13.999 ID:7 Payload:
106 00:14.001 ID:7 Received ACK
107 00:14.004 ID:7 Received #0 (0 bytes)
108 00:14.004 ID:7 |
109 00:14.009 ID:7 Got Response Data= (null) NodeID= 7
110 00:14.015 ID:7 ENERGY CONSUMPTION IS 2479 ID 7
111 00:14.015 ID:7
112 00:14.016 ID:7 --Done--
113 00:14.170 ID:2 handle_incoming_data(): received uip_dataalen=68
114 00:14.180 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0005]:5683
115 00:14.182 ID:2 Length: 68
116 00:14.188 ID:2 Parsed: v 1, t 0, tkl 0, c 2, mid 39742
117 00:14.192 ID:2 URL: Z8yY-Ua8_
118 00:14.196 ID:2 *&T^4+~( 0<Sm
119 00:14.199 ID:2 Payload: Smart-home!
120 00:14.211 ID:2 received cipher
text:5aa0d438790ceb592d55ceaf618c38d5efb406be5fde0adc2a8526545e34d2ae
121 00:14.220 ID:2 received cipher text in string format:Z8yY-Ua8_
122 00:14.225 ID:2 *&T^4+~( 0<Sm node_id:2
123 00:14.231 ID:2 server_key:2b7e151628aed2a6abf7158809cf4f3c
124 00:14.243 ID:2 Server Decrypted Plaintext:air conditioning from node_id:2
125 00:14.247 ID:2 Data Decrypted Time 13 ID 2
126 00:14.258 ID:5 handle_incoming_data(): received uip_dataalen=4
127 00:14.269 ID:5 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0002]:5683
128 00:14.271 ID:5 Length: 4
129 00:14.277 ID:5 Parsed: v 1, t 2, tkl 0, c 132, mid 39742
130 00:14.278 ID:5 URL:
131 00:14.279 ID:5 Payload:
132 00:14.281 ID:5 Received ACK
133 00:14.289 ID:5 Received #0 (0 bytes)
134 00:14.289 ID:5 |
135 00:14.294 ID:5 Got Response Data= (null) NodeID= 5
136 00:14.300 ID:5 ENERGY CONSUMPTION IS 2508 ID 5
137 00:14.300 ID:5
138 00:14.301 ID:5 --Done--
139 00:20.301 ID:7 --Toggle timer--
140 00:20.305 ID:7 Sending Data Time=20 NodeID= 7
141 00:20.311 ID:7 client_key:2b7e151628aed2a6abf7158809cf4f3c
142 00:20.324 ID:7 encrypted text:4bbb3a58114d37a032636694eba5a37e376e0c69a008e157e344e9c70ef64b0b
143 00:20.336 ID:7 encrypted text in string format:K:XM72cf~7niWDK+~( 0<
144 00:20.347 ID:7 Data Encrypted Time 20 ID 7 Enc Key:XM72cf~7niWDK+~( 0<Sm
```

Plain Text Tab Width: 4 Ln 121, Col 68 INS

```
test_aes x
109 00:14.009 ID:7 Got Response Data= (null) NodeID= 7
110 00:14.015 ID:7 ENERGY CONSUMPTION IS 2479 ID 7
111 00:14.015 ID:7
112 00:14.016 ID:7 --Done--
113 00:14.170 ID:2 handle_incoming_data(): received uip_dataalen=68
114 00:14.180 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0005]:5683
115 00:14.182 ID:2 Length: 68
116 00:14.188 ID:2 Parsed: v 1, t 0, tkl 0, c 2, mid 39742
117 00:14.192 ID:2 URL: Z8yY-Ua8_
118 00:14.196 ID:2 *&T^4+~( 0<Sm
119 00:14.199 ID:2 Payload: Smart-home!
120 00:14.211 ID:2 received cipher
text:5aa0d438790ceb592d55ceaf618c38d5efb406be5fde0adc2a8526545e34d2ae
121 00:14.220 ID:2 received cipher text in string format:Z8yY-Ua8_
122 00:14.225 ID:2 *&T^4+~( 0<Sm node_id:2
123 00:14.231 ID:2 server_key:2b7e151628aed2a6abf7158809cf4f3c
124 00:14.243 ID:2 Server Decrypted Plaintext:air conditioning from node_id:2
125 00:14.247 ID:2 Data Decrypted Time 13 ID 2
126 00:14.258 ID:5 handle_incoming_data(): received uip_dataalen=4
127 00:14.269 ID:5 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0002]:5683
128 00:14.271 ID:5 Length: 4
129 00:14.277 ID:5 Parsed: v 1, t 2, tkl 0, c 132, mid 39742
130 00:14.278 ID:5 URL:
131 00:14.279 ID:5 Payload:
132 00:14.281 ID:5 Received ACK
133 00:14.289 ID:5 Received #0 (0 bytes)
134 00:14.289 ID:5 |
135 00:14.294 ID:5 Got Response Data= (null) NodeID= 5
136 00:14.300 ID:5 ENERGY CONSUMPTION IS 2508 ID 5
137 00:14.300 ID:5
138 00:14.301 ID:5 --Done--
139 00:20.301 ID:7 --Toggle timer--
140 00:20.305 ID:7 Sending Data Time=20 NodeID= 7
141 00:20.311 ID:7 client_key:2b7e151628aed2a6abf7158809cf4f3c
142 00:20.324 ID:7 encrypted text:4bbb3a58114d37a032636694eba5a37e376e0c69a008e157e344e9c70ef64b0b
143 00:20.336 ID:7 encrypted text in string format:K:XM72cf~7niWDK+~( 0<
144 00:20.347 ID:7 Data Encrypted Time 20 ID 7 Enc Key:XM72cf~7niWDK+~( 0<Sm
```

Plain Text Tab Width: 4 Ln 134, Col 22 INS

Appendices: A, B, and C

Server node received encrypted data from Sensor node. Server decrypts the encrypted data. Then server node sends response to Sensor node.

SHA256 Hashing Algorithm

```
sha256.c  er-sha256-client.c  test_sha.c
64 00:10.392 ID:5 Sending Data Time=10 NodeID= 5
65 00:10.394 ID:6 --Toggle timer--
66 00:10.398 ID:6 Sending Data Time=10 NodeID= 6
67 00:10.406 ID:5 client side final digester:AAvvQQQ33qq plain text:air conditioning
68 00:10.411 ID:6 client side final digester:QGQGQQFFll5353 plain text:lightbulbs
69 00:10.417 ID:5 Data Hashing Time 10 ID 5 Encr air conditioning+AAvvQQQ33qq+
70 00:10.421 ID:5 Requested #0 (MID 39742)
71 00:10.422 ID:6 Data Hashing Time 10 ID 6 Encr lightbulbs+QGQGQQFFll5353+
72 00:10.426 ID:6 Requested #0 (MID 39742)
73 00:10.434 ID:1 sinsisnsinSIN: 48
74 00:10.439 ID:1 clt_id clt_id clt_id clt_id 1
75 00:10.507 ID:4 --Toggle timer--
76 00:10.512 ID:4 Sending Data Time=10 NodeID= 4
77 00:10.524 ID:4 client side final digester:^^b'b'G/G/FF[[ plain text:heating
78 00:10.534 ID:4 Data Hashing Time 10 ID 4 Encr heating+^^b'b'G/G/FF[[+
79 00:10.538 ID:4 Requested #0 (MID 39742)
80 00:10.794 ID:7 --Toggle timer--
81 00:10.798 ID:7 Sending Data Time=10 NodeID= 7
82 00:10.812 ID:7 client side final digester:11..CC}}WM66 plain text:camera systems
83 00:10.823 ID:7 Data Hashing Time 10 ID 7 Encr camera systems+11..CC}}WM66+
84 00:10.827 ID:7 Requested #0 (MID 39742)
85 00:10.857 ID:3 --Toggle timer--
86 00:10.862 ID:3 Sending Data Time=10 NodeID= 3
87 00:10.874 ID:3 client side final digester:U>U>qq66aa plain text:lightning
88 00:10.885 ID:3 Data Hashing Time 10 ID 3 Encr lightning+U>U>qq66aa+
89 00:10.889 ID:3 Requested #0 (MID 39742)
90 00:13.961 ID:2 handle_incoming_data(): received uip_dataalen=68
91 00:13.972 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0005]:5683
92 00:13.973 ID:2 Length: 68
93 00:13.979 ID:2 Parsed: v 1, t 0, tkl 0, c 1, mid 39742
94 00:13.987 ID:2 URL: air conditioning+AAvvQQQ33qq+
95 00:13.990 ID:2 Payload: Smart-home!
96 00:14.005 ID:2 Received hash value:AAvvQQQ33qq client_id:5 original message:air
conditioning
97 00:14.020 ID:2 Generated hash value:AAvvQQQ33qq client_id:5 original message:air
conditioning
Plain Text  Tab Width: 4  Ln 82, Col 21  INS
```

Sensor-Side Hash Generation. Each node sends hashed data to server via border router.

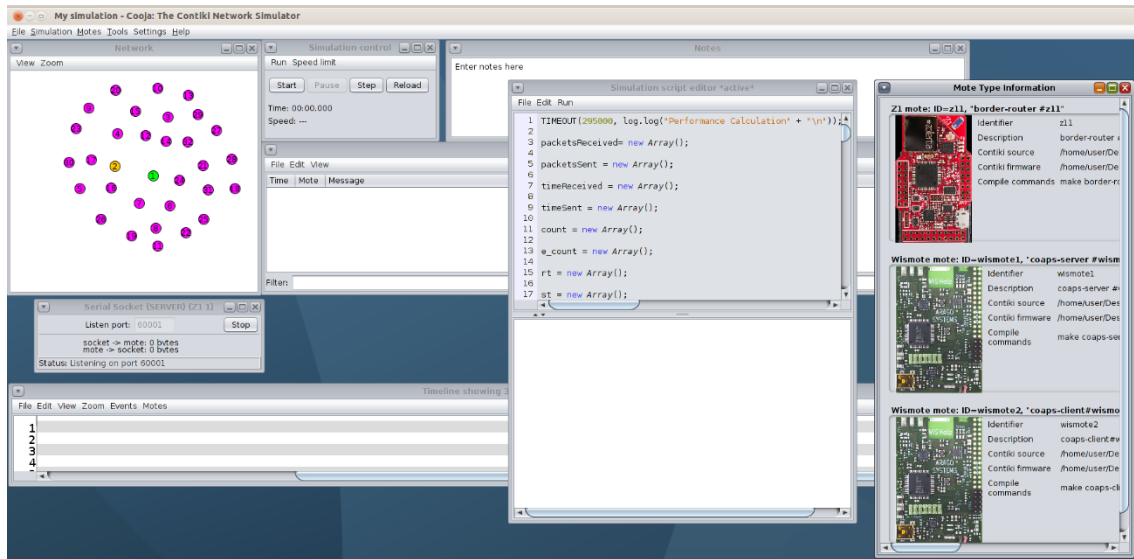
Hash Verification Process

```
sha256.c  er-sha256-client.c  test_sha.c
89 00:10.889 ID:3 Requested #0 (MID 39742)
90 00:13.961 ID:2 handle_incoming_data(): received uip_dataalen=68
91 00:13.972 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0005]:5683
92 00:13.973 ID:2 Length: 68
93 00:13.979 ID:2 Parsed: v 1, t 0, tkl 0, c 1, mid 39742
94 00:13.987 ID:2 URL: air conditioning+AAvvQQQ33qq+
95 00:13.990 ID:2 Payload: Smart-home!
96 00:14.005 ID:2 Received hash value:AAvvQQQ33qq client_id:5 original message:air
conditioning
97 00:14.020 ID:2 Generated hash value:AAvvQQQ33qq client_id:5 original message:air
conditioning
98 00:14.025 ID:2 The Hashes are equal & verified
99 00:14.028 ID:2 Data Verified Time 13 ID 2
100 00:14.040 ID:5 handle_incoming_data(): received uip_dataalen=4
101 00:14.050 ID:5 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0002]:5683
102 00:14.052 ID:5 Length: 4
103 00:14.058 ID:5 Parsed: v 1, t 2, tkl 0, c 132, mid 39742
104 00:14.059 ID:5 URL:
105 00:14.061 ID:5 Payload:
106 00:14.063 ID:5 Received ACK
107 00:14.066 ID:5 Received #0 (0 bytes)
108 00:14.066 ID:5 |
109 00:14.071 ID:5 Got Response Data= (null) NodeID= 5
110 00:14.072 ID:2 handle_incoming_data(): received uip_dataalen=60
111 00:14.077 ID:5 ENERGY_CONSUMPTION IS 2382 ID 5|
112 00:14.078 ID:5
113 00:14.083 ID:5 --Done--
114 00:14.085 ID:2 receiving UDP datagram from: [fe80:0000:0000:0000:0200:0000:0000:0004]:5683
115 00:14.085 ID:2 Length: 60
116 00:14.091 ID:2 Parsed: v 1, t 0, tkl 0, c 1, mid 39742
117 00:14.097 ID:2 URL: heating+^^b'b'G/G/FF[[
118 00:14.101 ID:2 Payload: Smart-home!
119 00:14.114 ID:2 Received hash value:^^b'b'G/G/FF[[ client_id:4 original message:heating
120 00:14.128 ID:2 Generated hash value:^^b'b'G/G/FF[[ client_id:4 original message:heating
121 00:14.132 ID:2 The Hashes are equal & verified
122 00:14.136 ID:2 Data Verified Time 13 ID 2
Plain Text  Tab Width: 4  Ln 111, Col 52  INS
```

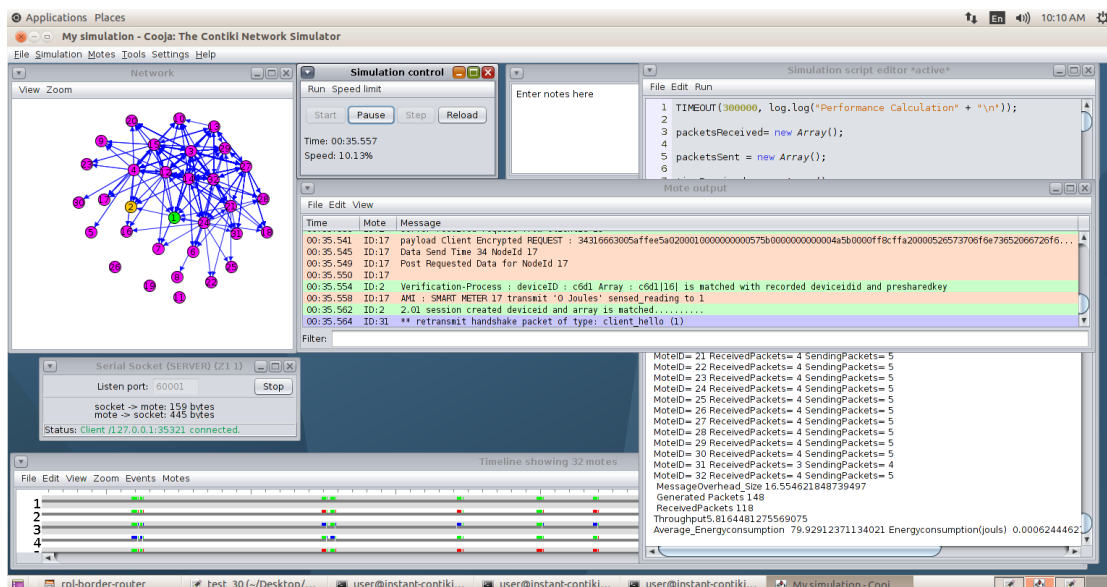
Server nodes receive the hash value from Sensor. Then server node recalculates the hash value for original data. The server node verifies the rehash value and received hash value. If it is equal server send response to Sensor node. Otherwise discard it.

Appendix B Payload based Light Weight Authentication
COAP/AMI: 30 nodes

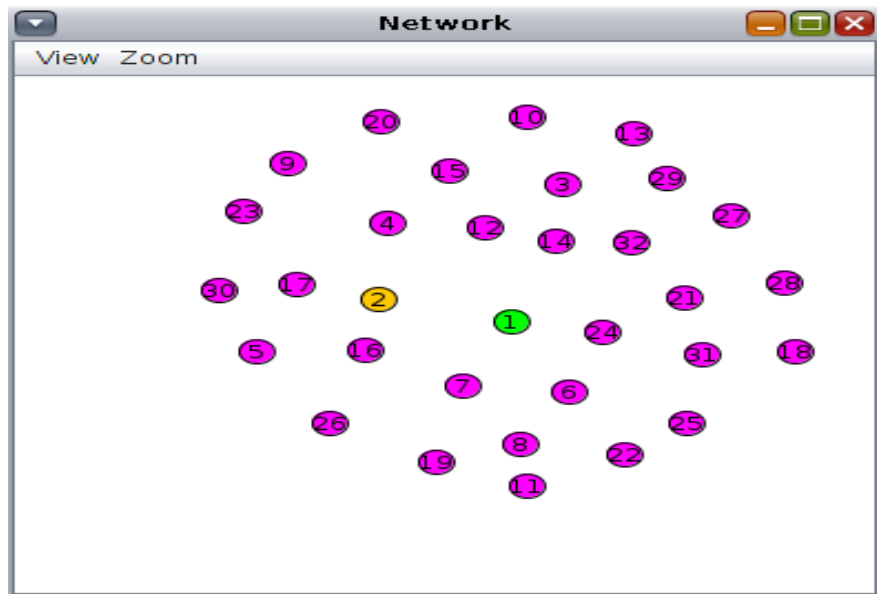
Appendices: A, B, and C



Network architecture:



Network Diagram:



Simulation Details:

Node-count: 32 (1 Border router, 1 Server and 30 Sensor nodes)



- Border-router



- Server



- Sensor

Simulation time: 300 seconds

Topology: Distributed sensors network

Modules:

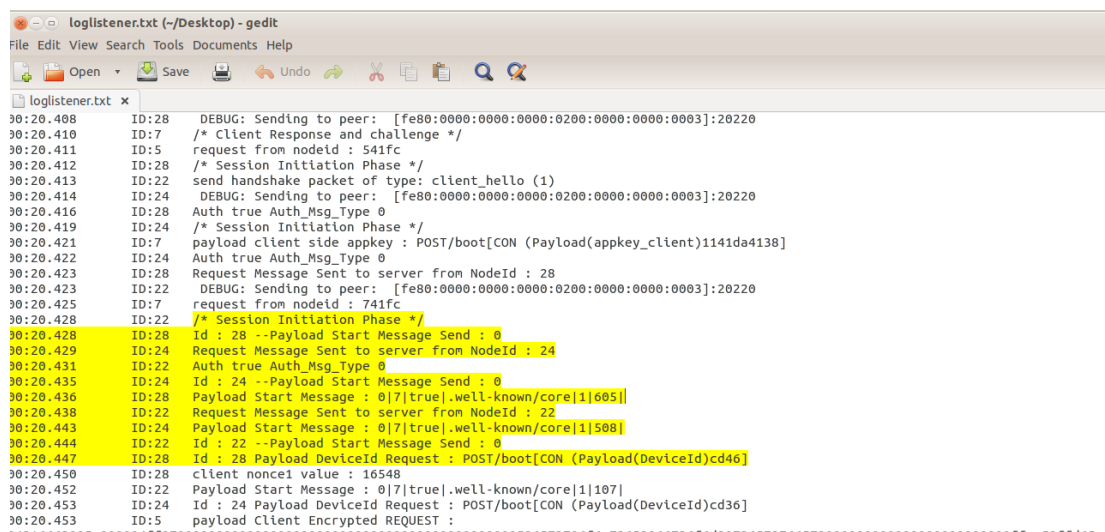
- 1.Session Initiation
- 2.Server Challenge
- 3.Sensor Response and Challenge

4. Server Response, and
5. Secure and Optimised Communication in SG
6. Performance analysis

Algorithm Explanation

1. Session Initiation

Sensor node send Device id request message (set of fields) to server for authentication purpose. Sensor request consist of mid, token, uri, Auth, Auth_msg_type as well as device id and presharedkey. For session initiation phase Auth_Msg_Type should be zero.



```
loglistener.txt x
30:20.408 ID:28 DEBUG: Sending to peer: [fe80:0000:0000:0000:0200:0000:0000:0003]:20220
30:20.410 ID:7 /* Client Response and challenge */
30:20.411 ID:5 request from nodeid : 541fc
30:20.412 ID:28 /* Session Initiation Phase */
30:20.413 ID:22 send handshake packet of type: client_hello (1)
30:20.414 ID:24 DEBUG: Sending to peer: [fe80:0000:0000:0000:0200:0000:0000:0003]:20220
30:20.416 ID:28 Auth true Auth_Msg_Type 0
30:20.419 ID:24 /* Session Initiation Phase */
30:20.421 ID:7 payload client side appkey : POST/boot[CON (Payload(appkey_client)1141da4138]
30:20.422 ID:24 Auth true Auth_Msg_Type 0
30:20.423 ID:28 Request Message Sent to server From NodeId : 28
30:20.423 ID:22 DEBUG: Sending to peer: [fe80:0000:0000:0000:0200:0000:0000:0003]:20220
30:20.425 ID:7 request from nodeid : 741fc
30:20.428 ID:22 /* Session Initiation Phase */
30:20.428 ID:28 Id : 28 --Payload Start Message Send : 0
30:20.429 ID:24 Request Message Sent to server From NodeId : 24
30:20.431 ID:22 Auth true Auth_Msg_Type 0
30:20.435 ID:24 Id : 24 --Payload Start Message Send : 0
30:20.436 ID:28 Payload Start Message : 0[7|true|.well-known/core|1|605|
30:20.438 ID:22 Request Message Sent to server From NodeId : 22
30:20.443 ID:24 Payload Start Message : 0[7|true|.well-known/core|1|508|
30:20.444 ID:22 Id : 22 --Payload Start Message Send : 0
30:20.447 ID:28 Id : 28 Payload DeviceId Request : POST/boot[CON (Payload(DeviceId)cd46]
30:20.450 ID:28 client nonce1 value : 16548
30:20.452 ID:22 Payload Start Message : 0[7|true|.well-known/core|1|107|
30:20.453 ID:24 Id : 24 Payload DeviceId Request : POST/boot[CON (Payload(DeviceId)cd36]
30:20.453 ID:5 payload Client Encrypted REQUEST :
```

2. Server Challenge

After receiving device id request from the Sensor, server verifies the device id and Preshared-key with pre-stored data. If server successfully verifies the user, then it will send the authentication response to Sensor. Here the comparison made sure to ensure communication. The acknowledgement message include server nonce value and session appkey.

[illegible]

Appkey-Server

[illegible]

Server generate application session key based on Psk value,nonce of server value.

3. Sensor Response and Challenge

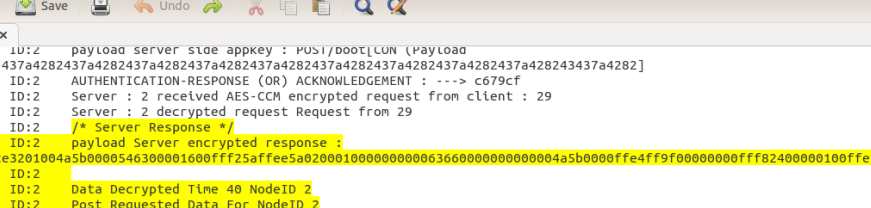
After receiving authentication response message from server, the Sensor generate Sensor side appkey based on Sensor and server nonce value with psk value. During response phase Auth_msg_Type should be one. The clinet side request message encrypted using AES-CCM-128 algorithm, using appkey. So that it will ensure the Sensor side message security. Because appkey is differ from each and every session. Therefore, it will hard to crack the messages.

The screenshot displays a Wireshark packet capture of a communication session. The packet list pane on the left shows 22 packets. The packet details pane in the center shows the structure of the selected packet (Packet 22), which is a POST request for NodeId 22. The packet bytes pane at the bottom shows the raw data of the selected packet, which is a hexadecimal representation of the encrypted payload. The interface includes a menu bar (File, Edit, View, Search, Tools, Documents, Help) and a toolbar with icons for opening, saving, undo, redo, and other standard network analysis functions.

No.	Time	Source	Destination	Protocol	Length	Info
0	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
1	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
2	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
3	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
4	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
5	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
6	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
7	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
8	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
9	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
10	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
11	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
12	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
13	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
14	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
15	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
16	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
17	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
18	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
19	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
20	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1
21	0.000000	192.168.1.1	192.168.1.100	HTTP	1024	200 OK
22	0.000000	192.168.1.100	192.168.1.1	HTTP	1024	GET / HTTP/1.1

4. Server Response

After receiving Sensor encrypted request message, the server will decipher using nonce value. If the nonce value should be same, the Sensor and server will mutually be authenticated itself. Then data packet will be sent based on session key.



loglistener.txt (-/Desktop) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Find Replace

loglistener.txt x

90:41.424 ID:2 payload server side appkey : POST|boot|CON (Payload
 (appkey_server)437a4282437a4282437a4282437a4282437a4282437a4282437a4282437a428243437a4282]
 90:41.432 ID:2 AUTHENTICATION-RESPONSE (OR) ACKNOWLEDGEMENT : ---> c679cf
 90:41.441 ID:2 Server : 2 received AES-CCH encrypted request from client : 29
 90:41.447 ID:2 Server : 2 decrypted request Request from 29
 90:41.450 ID:2 /* Server Response */
 90:41.494 ID:2 payload Server encrypted response :
 56320a000000ffce3201004a5b0000546300001600ffff25affee5a020001000000000036600000000000004a5b0000ffe4ff9f00000000ff82400000100ffef5affe02a0000390
 90:41.495 ID:2
 90:41.499 ID:2 Data Decrypted Time 40 NodeID 2
 90:41.504 ID:2 Post Requested Data For NodeID 2
 90:41.509 ID:2 server received request from clientID 31
 90:41.525 ID:2 Verification-Process : deviceId : c681 Array : c681|38| is matched with recorded deviceId and presharedkey
 90:41.533 ID:2 2.01 session created deviceId and array is matched.....
 90:41.543 ID:2 AMI : UTILITY CENTER 2 received sensed_reading '86 from SMART METER ID
 90:41.565 ID:2 payload server side appkey : POST|boot|CON (Payload
 (appkey_server)437a4282437a4282437a4282437a4282437a4282437a4282437a4282437a428243437a4282]
 90:41.573 ID:2 AUTHENTICATION-RESPONSE (OR) ACKNOWLEDGEMENT : ---> c68138
 90:41.582 ID:2 Server : 2 received AES-CCH encrypted request from client : 31
 90:41.588 ID:2 Server : 2 decrypted request Request from 31
 90:41.591 ID:2 /* Server Response */
 90:41.636 ID:2 payload Server encrypted response :
 56320a000000ffce3201004a5b0000546300001600ffff25affee5a020001000000000033300000000000004a5b0000ffe4ff9f00000000ff82400000100ffef5affe02a0000390
 90:41.636 ID:2
 90:41.640 ID:2 Data Decrypted Time 40 NodeID 2
 90:41.645 ID:2 Post Requested Data For NodeID 2
 90:50.826 ID:5 ** retransmit handshake packet of type: client_hello (1)
 90:50.836 ID:5 DEBUG: Sending to peer: [fe80:0000:0000:0000:0200:0000:0000:0003]:20220

Plain Text ▾ Tab Width: 8 ▾ Ln 4067, Col 25 INS

5. Secure and Optimised Communication in SG

**B.1 PERFORMANCE ANALYSIS OF CoAP/DTLS(NO SECURITY)
IMPLEMENTATION FOR VARIOUS PAYLOAD SIZE**

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.010934	0.010934	0.010934	0.010935	0.010936
AverageEnergyConsumption(microJoule)	2011.769	2011.769	2011.718	2011.667	2011.566
Communication Duration	1469.790	1469.790	1469.827	1469.864	1469.939
MessageOverhead	2847.571	2847.571	2847.428	2847.285	2847

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.685836	0.686935	0.686935	0.685683	0.686905
AverageEnergyConsumption(microJoules)	2034.606	2032.775	2032.775	2034.861	2032.826
Communication Duration	1453.277	1454.588	1454.588	1453.096	1454.551
MessageOverhead	363.964	363.321	363.321	364.053	363.339

PAYLOAD SIZE - 127BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.33822543	1.33606402	1.33623901	1.33618067	1.33834246
AverageEnergyConsumption(microJoule)	2054.036	2055.918	2055.765	2055.816	2053.934
Communication Duration	1439.518	1438.199	1438.306	1438.270	1439.589
MessageOverhead	174.487394	174.798319	174.773109	174.781512	174.470588

PAYLOAD SIZE - 256BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	2.5706309	2.57309681	2.5717024	2.5742778	2.5725603
AverageEnergyConsumption(microJoules)	2111.460	2110.290	2110.951	2109.731	2110.544
Communication Duration	1400.334	1401.111	1400.672	1401.483	1400.942
MessageOverhead	88.2782258	88.1854838	88.237903	88.1411290	88.2056451

PAYLOAD SIZE 512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	4.7242445	4.7267781	4.7246123	4.7262410	4.72358541
AverageEnergyConsumption(microJoule)	2221.984	2221.272	2222.290	2221.832	2222.442
Communication Duration	1330.625	1331.052	1330.442	1330.717	1330.351
Message Overhead	47.628968	47.601190	47.623015	47.605158	47.6349206

PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	8.130762	8.139880	8.135587	8.131566	8.129958
AverageEnergyConsumption(microJoule)	2444.051	2442.321	2443.135	2443.898	2444.203
Communication Duration	1209.639	1210.496	1210.093	1209.715	1209.564
MessageOverhead	27.777	27.744	27.759	27.774	27.780

B.2 PERFORMANCE ANALYSIS OF PAYLOA AUTHENTICATION USING COAP/DTLS FOR VARIOUS PAYLOAD SIZE

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.0147892	0.0147892	0.0147892	0.0147892	0.0147892
AverageEnergyConsumption(microJoule)	1826.605	1826.605	1826.605	1826.605	1826.605
Communication Duration	1712.794	1712.794	1712.794	1712.794	1712.794
Message Overhead	2006.857	2006.857	2006.857	2006.857	2006.857

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.920532	0.920477	0.920477	0.920532	0.920477
AverageEnergyConsumption(microJoules)	1851.453	1851.508	1851.508	1851.453	1851.508
Communication Duration	1689.781	1689.732	1689.732	1689.781	1689.732
MessageOverhead	259.035	259.053	259.053	259.035	259.053

PAYLOAD SIZE - 127BYTES

Perforamnce	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.78628638	1.78544929	1.78544929	1.78565849	1.78544929
AverageEnergyConsumption(microJoule)	1873.372	1873.806	1873.806	1873.697	1873.806
Communication Duration	1669.991	1669.604	1669.604	1669.701	1669.604
MessageOverhead	125.042016	125.109243	125.109243	125.092436	125.109243

PAYLOAD SIZE - 256BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	3.4046954	3.40488417	3.40375240	3.40375240	3.40488417
AverageEnergyConsumption(microJoule)	1928.114	1928.059	1928.385	1928.385	1928.059
Communication Duration	1622.529	1622.575	1622.300	1622.300	1622.575
MessageOverhead	63.9475806	63.9435483	63.9677419	63.9677419	63.9435483

PAYLOAD SIZE -512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	6.15045798	6.15138165	6.15045798	6.1526136	6.15107373
AverageEnergyConsumption(microJoules)	2034.450	2034.288	2034.450	2034.071	2034.342
Communication Duration	1537.637	1537.760	1537.637	1537.924	1537.719
MessageOverhead	35.2956349	35.2896825	35.2956349	35.281746	35.2916666

PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	10.301666	10.301666	10.299939	10.301235	10.299508
AverageEnergyConsumption(microJoules)	2250.596	2250.596	2250.813	2250.651	2250.868
Communication Duration	1389.832	1389.832	1389.698	1389.799	1389.665
MessageOverhead	21.312	21.312	21.315	21.312	21.316

**B.3 PERFORMANCE ANALYSIS OF PAYLOAD LIGHTWEIGHT
AUTHENTICATION USING AMI CoAP/DTLS FOR
VARIOUS PAYLOAD SIZE**

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.01544898	0.01544898	0.01544898	0.01544898	0.01544898
AverageEnergyConsumption(microJoule)	1789.008	1789.008	1789.008	1789.008	1789.008
Communication Duration	1748.833	1748.833	1748.833	1748.833	1748.833
MessageOverhead	1907.857	1907.857	1907.857	1907.857	1907.857

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.9352098	0.9619640	0.8869691	0.962265	0.9524776
AverageEnergyConsumption(microJoules)	1838.184	1813.910	1874.195	1813.639	1826.074
Communication Duration	1708.606	1724.799	1664.667	1725.057	1723.773
MessageOverhead	256.821	246.142	272.839	246.053	251.482

PAYLOAD SIZE - 127BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.85812115	1.85812115	1.85846103	1.85834772	1.85846103
AverageEnergyConsumption(microJoules)	1839.029	1839.029	1838.867	1838.921	1838.867
Communication Duration	1701.216	1701.216	1701.367	1701.316	1701.367
MessageOverhead	119.470	119.470	119.445	119.453	119.445

PAYLOAD SIZE - 256BYTES

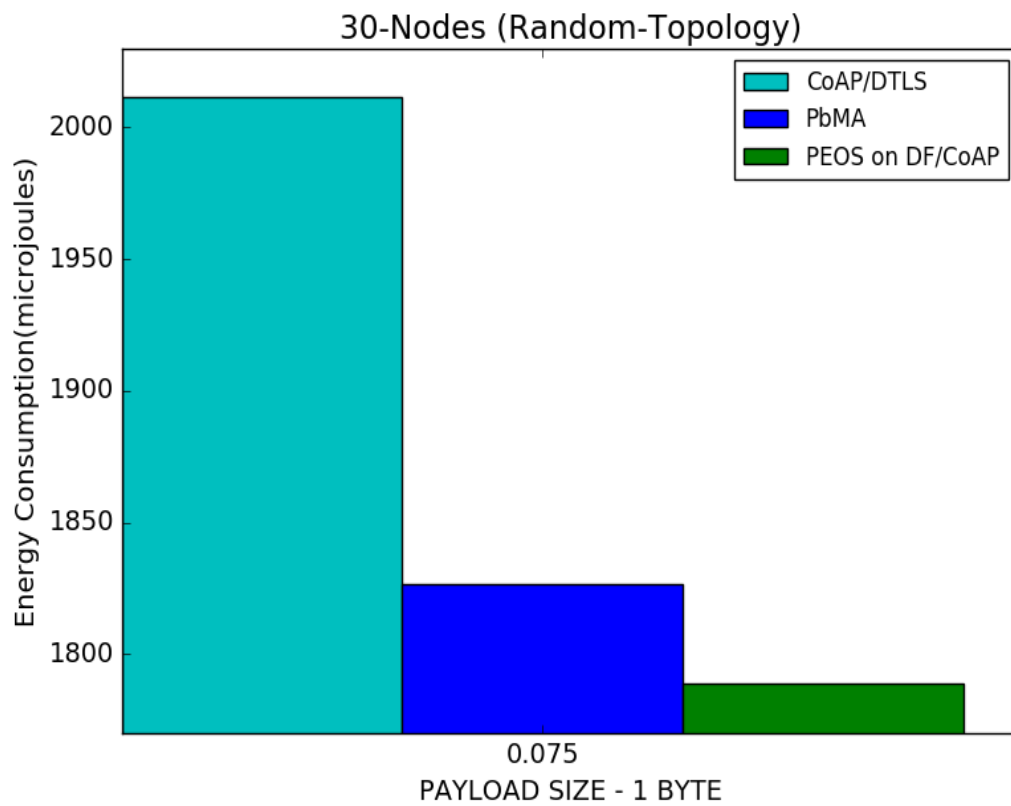
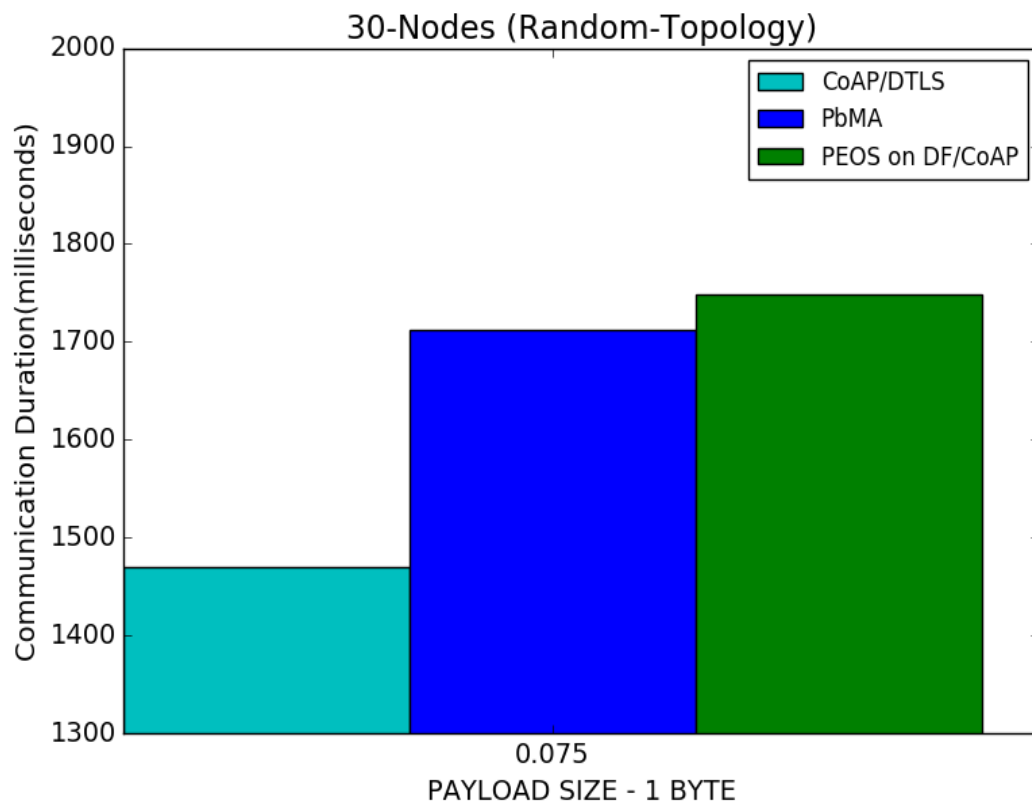
Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	3.53496221	3.53536903	3.53394558	3.53496221	3.53435216
AverageEnergyConsumption(microJoules)	1892.035	1891.927	1892.306	1892.035	1892.198
Communication Duration	1653.503	1653.598	1653.266	1653.503	1653.361
MessageOverhead	61.266	61.258	61.286	61.266	61.278

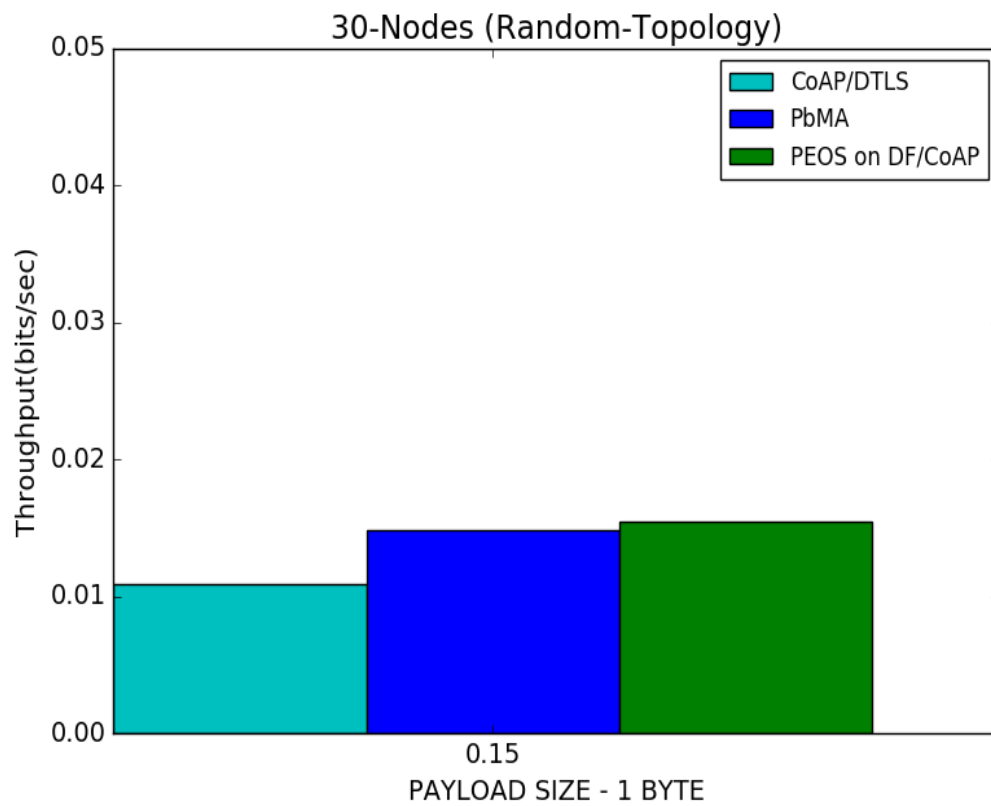
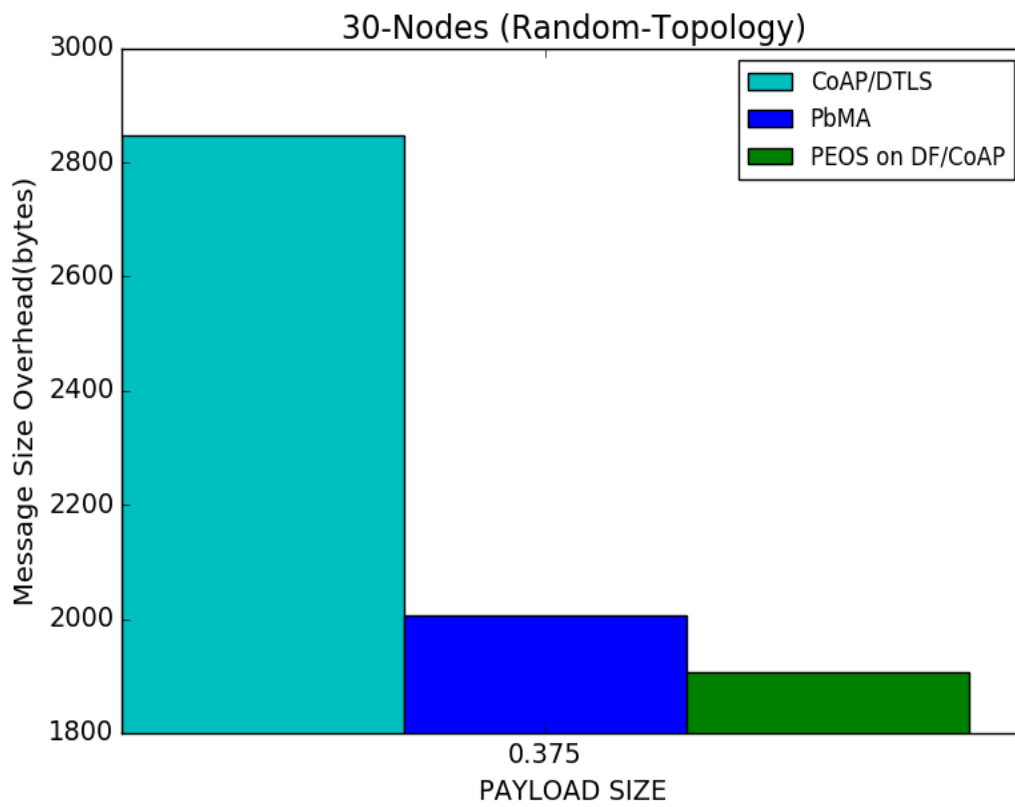
PAYLOAD SIZE - 512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	6.36343235	6.36343235	6.36310284	6.36573994	6.36541018
AverageEnergyConsumption(microJoules)	1999.620	1999.620	1999.674	1999.240	1999.294
Communication Duration	1564.451	1564.451	1564.409	1564.749	1564.706
MessageOverhead	33.962	33.962	33.964	33.948	33.950

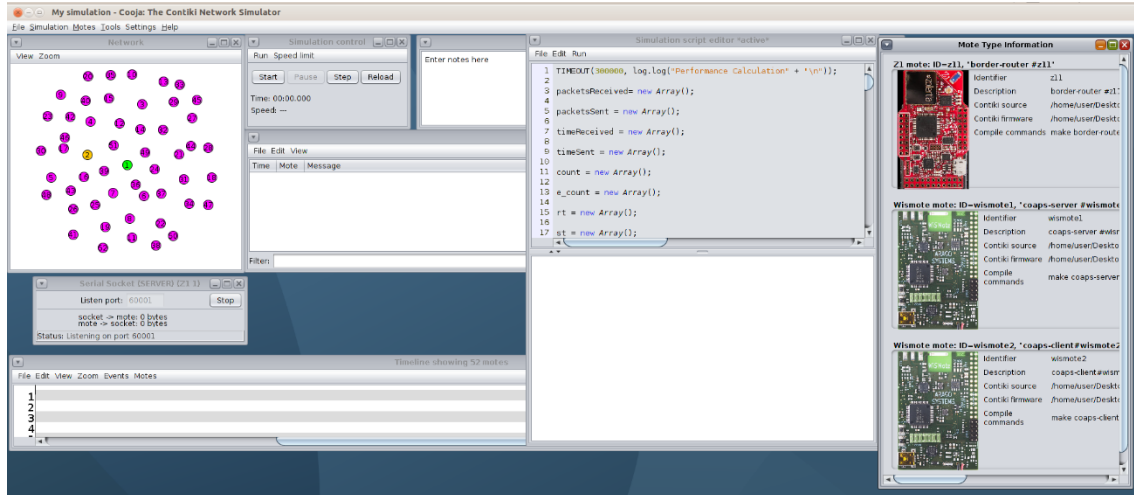
PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	10.591683	10.589553	10.590922	10.589502	10.588691
AverageEnrgyConsumption(microJoules)	2216.796	2216.905	2216.742	2216.959	2216.959
Communication Duration	1411.046	1410.977	1411.081	1410.942	1410.942
MessageOverhead	20.665	20.670	20.667	20.670	20.672

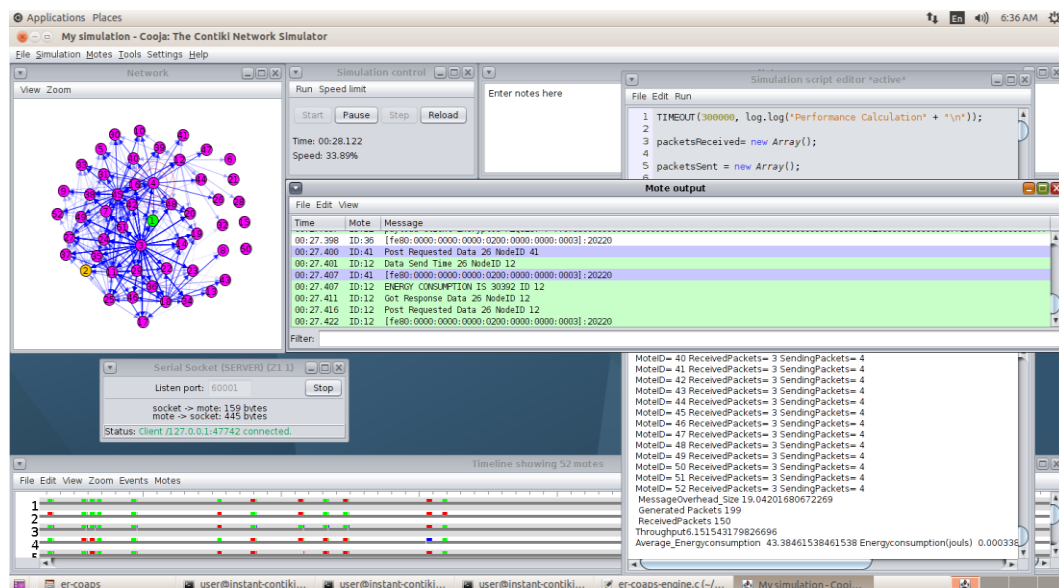




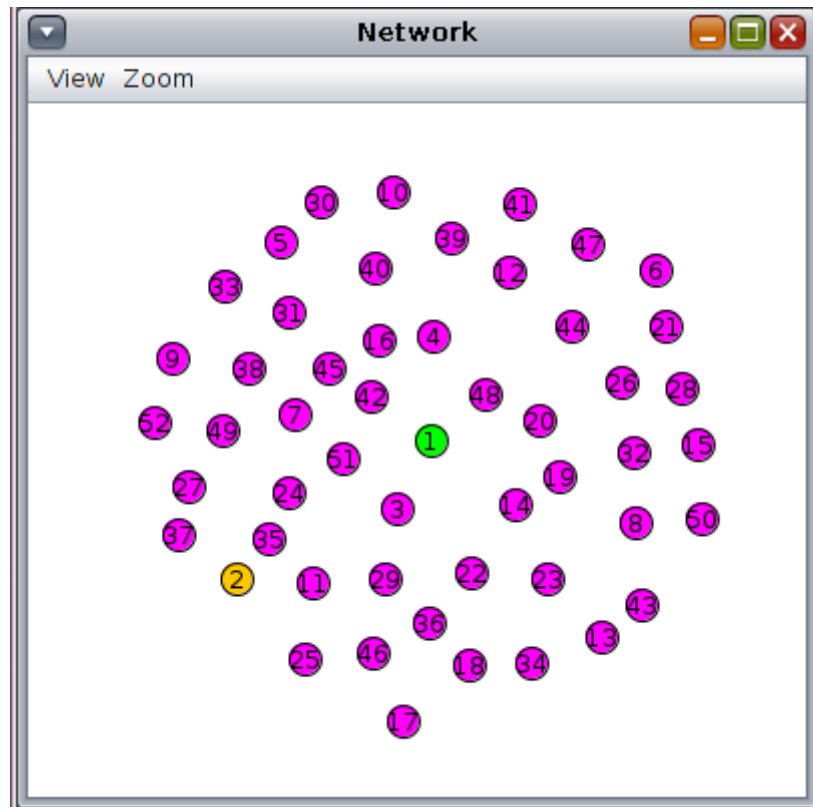
Appendix C Payload based Light Weight Authentication COAP/AMI: 50 nodes



Network Architecture: 50 nodes



Network Diagram:



Simulation Details:

Node-count: 52 (1 Border router, 1 Server and 50 Sensor nodes)

-  - Border-router
-  - Server
-  - Sensor

Simulation time: 300 seconds

Topology: Random topology

1. Sensor Response

Sensor Send encrypted Request message to server.

**C.1 PERFORMANCE ANALYSIS OF CoAP/DTLS (NO SECURITY)
IMPLEMENTATION FOR VARIOUS PAYLOAD SIZE**

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.01078111	0.01078111	0.01078111	0.01078111	0.01078111
AverageEnergyConsumption(microJoules)	1245.755	1245.755	1245.755	1245.755	1245.755
Communication Duration	2425.443	2425.443	2425.443	2425.443	2425.443
MessageOverhead	4804.857	4804.857	4804.857	4804.857	4804.857

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.6670189	0.66707099	0.66696697	0.66703631	0.66700164
AverageEnergyConsumption(microJoules)	1269.812	1269.756	1269.869	1269.794	1269.831
Communication Duration	2379.467	2379.572	2379.361	2379.502	2379.431
MessageOverhead	623.482	623.428	623.535	623.464	623.5

PAYLOAD SIZE - 127BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.27650454	1.27624863	1.2765685	1.27628061	1.27692063
AverageEnergyConsumption(microJoules)	1296.480	1296.630	1296.443	1296.612	1296.236
Communication Duration	2330.497	2330.227	2330.564	2330.260	2330.936
MessageOverhead	305.336	305.403	305.319	305.394	305.226

PAYLOAD SIZE - 256BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	2.38304457	2.38227035	2.38133090	2.38116519	2.38598013
AverageEnergyConsumption(microJoules)	1356.238	1356.501	1356.820	1356.877	1355.243
Communication Duration	2227.760	2227.327	2226.803	2226.711	2229.396
MessageOverhead	159.342	159.399	159.467	159.479	159.129

PAYLOAD SIZE 512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	4.18647860	4.19035913	4.19040740	4.19279589	4.19029394
AverageEnergyConsumption(microJoules)	1470.552	1469.482	1469.538	1469.181	1469.613
Communication Duration	2054.507	2056.004	2055.925	2056.425	2055.820
MessageOverhead	90.210	90.123	90.121	90.063	90.123

PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	6.76305262	6.76171621	6.76806883	6.76572700	6.76661893
AverageEnergyConsumption(microJoules)	1693.340	1693.565	1692.495	1692.889	1692.739
Communication Duration	1784.097	1783.859	1784.988	1784.572	1784.730
MessageOverhead	56.217	56.229	56.173	56.193	56.186

**C.2 PERFORMANCE ANALYSIS OF PAYLOAD
AUTHENTICATION USING COAP/DTLS FOR
VARIOUS PAYLOAD SIZE**

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.01306662	0.01306662	0.01306662	0.01306662	0.01306790
AverageEnergyConsumption(microJoules)	1195.492	1195.492	1195.492	1195.492	1195.432
Communication Duration	2682.752	2682.752	2682.752	2682.752	2682.887
MessageOverhead	3854.142	3854.142	3854.142	3854.142	3853.714

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.80396118	0.80401168	0.80423900	0.80393593	0.80398643
AverageEnergyConsumption(microJoules)	1220.132	1220.092	1219.911	1220.152	1220.112
Communication Duration	2628.542	2628.628	2629.017	2628.498	2628.585
MessageOverhead	503.732	503.696	503.535	503.75	503.714

PAYLOAD SIZE - 127BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.53874631	1.53855990	1.53855990	1.53851330	1.53869971
AverageEnergyConsumption(microJoules)	1244.491	1244.571	1244.571	1244.591	1244.511
Communication Duration	2577.062	2576.896	2576.896	2576.855	2577.021
MessageOverhead	246.848	246.882	246.882	246.890	246.857

PAYLOAD SIZE - 256BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	2.87491499	2.8757223	2.87475357	2.87459217	2.87443079
AverageEnergyConsumption(microJoules)	1297.556	1297.355	1297.596	1297.636	1297.676
Communication Duration	2471.610	2471.992	2471.533	2471.457	2471.381
MessageOverhead	128.927	128.887	128.935	128.943	128.951

PAYLOAD SIZE -512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	4.98892972	4.98872718	4.98838965	4.98967252	4.98795767
AverageEnergyConsumption(microJoules)	1407.411	1407.391	1407.411	1407.271	1407.572
Communication Duration	2278.583	2278.616	2278.583	2278.810	2278.324
MessageOverhead	74.206	74.210	74.216	74.194	74.222

PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	7.9295875	7.9286688	7.9306596	7.93020012	7.92790349
AverageEnergyConsumption(microJoules)	1627.303	1627.423	1627.163	1627.223	1627.524
Communication Duration	1980.411	1980.265	1980.582	1980.509	1980.143
MessageOverhead	47.426	47.432	47.419	47.422	47.437

**C.3 PERFORMANCE ANALYSIS OF PAYLOAD LIGHTWEIGHT
AUTHENTICATION USING AMI CoAP/DTLS FOR
VARIOUS PAYLOAD SIZE**

PAYLOAD SIZE - 1BYTE

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.01400494	0.01400494	0.01400494	0.01400494	0.01400494
AverageEnergyConsumption(microJoules)	1154.407	1154.407	1154.407	1154.407	1154.407
Communication Duration	2778.296	2778.296	2778.296	2778.296	2778.296
MessageOverhead	3561.142	3561.142	3561.142	3561.142	3561.142

PAYLOAD SIZE - 64BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	0.86059419	0.86059419	0.86059419	0.86059419	0.86059419
AverageEnergyConsumption(microJoules)	1179.046	1179.046	1179.046	1179.046	1179.046
Communication Duration	2720.200	2720.200	2720.200	2720.200	2720.200
MessageOverhead	466.214	466.214	466.214	466.214	466.214

PAYLOAD SIZE - 127BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	1.64133345	1.64117438	1.64128042	1.6417047	1.64133345
AverageEnergyConsumption(microJoules)	1204.046	1204.106	1204.066	1203.906	1204.046
Communication Duration	2663.686	2663.553	2663.642	2663.997	2663.686
MessageOverhead	229.462	229.487	229.470	229.403	229.462

PAYLOAD SIZE - 256BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	3.04549405	3.04630945	3.04612821	3.04603760	3.04640008
AverageEnergyConsumption(microJoules)	1257.592	1257.411	1257.451	1257.471	1257.391
Communication Duration	2550.202	2550.567	2550.486	2550.445	2550.608
MessageOverhead	120.883	120.846	120.854	120.858	120.842

PAYLOAD SIZE - 512BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	5.25403630	5.25498000	5.25376673	5.25349720	5.25255403
AverageEnergyConsumption(microJoules)	1366.666	1366.526	1366.706	1366.746	1366.887
Communication Duration	2346.556	2346.797	2346.488	2346.419	2346.178
MessageOverhead	70.087	70.073	70.091	70.095	70.109

PAYLOAD SIZE - 1024BYTES

Performance	Testcase 1	Testcase 2	Testcase 3	Testcase 4	Testcase 5
Throughput	8.68139126	8.73899596	8.73994624	8.7380256	8.68316708
AverageEnergyConsumption(microJoules)	1554.967	1550.780	1550.659	1550.922	1554.825
Communication Duration	2094.956	2103.453	2103.618	2103.260	2095.147
MessageOverhead	43.006	42.699	42.694	42.704	42.997

