

# **PRIVACY-PRESERVING AUTHENTICATION AND KEY EXCHANGE MECHANISMS IN INTERNET OF THINGS APPLICATIONS**

Thesis Submitted For the Award of the Degree of

**DOCTOR OF PHILOSOPHY**

in

**Electronics and Electrical Engineering**

By

**Gurjot Singh**

**41400192**

Supervised By

**Dr. Gulshan Kumar**

**Lovely Professional University, India**

<sup>1</sup>Co-Supervised by

**Dr. Pardeep Kumar**

**Swansea University, U.K.**

<sup>2</sup>Co-Supervised by

**Dr. Himanshu Monga**

**J.L.N. Govt. Engineering College, India**



**LOVELY PROFESSIONAL UNIVERSITY**  
**PUNJAB**  
**2021**

## Declaration

***I do hereby acknowledge that:***

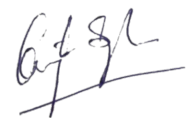
The present thesis entitled “*Privacy-Preserving Authentication and Key Exchange Mechanisms in Internet of Things Applications*” is a presentation of my original research work done under the guidance of my supervisor and co-supervisors. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

I further declare that there is no falsification or manipulation in terms of research materials. I hereby confirm that the thesis is free from any plagiarized material and does not infringe any rights of others.

I carefully checked the final version of printed and softcopy of the thesis for the completeness and for incorporation of all suggestions of doctoral committee.

I hereby submit the final version of the printed copy of my thesis as per the guidelines and the exact same content in CD as a separate PDF file to be uploaded in Shodhganga.

**Date: 13 August, 2021**



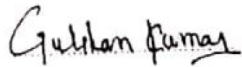
**Signature of Candidate**

## Certificate

We certify that the candidate Gurjot Singh (R.No. 41400192) has carried out his thesis work "Privacy-Preserving Authentication and Key Exchange Mechanisms in Internet of Things Applications," under our supervision. To the best of our knowledge:

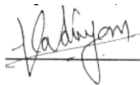
- i) The candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship, or other similar titles.
- ii) The thesis submitted is a record of original research work done by the Research Scholar during the period of study under our supervision, and
- iii) The thesis represents independent research work on the part of the Research Scholar.

### Supervised By:



Dr. Gulshan Kumar  
Associate Professor,  
Lovely Professional University, India

### Co-Supervised by:



Dr. Pardeep Kumar,  
Assistant Professor,  
Swansea University, U.K.

### Co-Supervised by:



Dr. Himanshu Monga  
Professor,  
J.L.N. Govt. Engineering College, India



# Abstract

- **Research Area:** The Internet of Things (IoT) technology has emerged as the leader in simplifying lives. The IoT is rapidly adopted by the various industries for faster data accessibility and control of machines remotely by managers. Likewise, smart cities use IoT to optimize operations across the city such as waste and traffic management, water supply management, and pollution monitoring, etc.
- **Research Problem:** Although IoT is beneficial but dangerous as the nodes communicate over the unsecured public networks. These public networks opened enormous ways for the illegitimate nodes to access the information and take control over the IoT networks despite being physically away. Moreover, in several applications, as there is no infrastructure available, an attacker can counterfeit devices to control the IoT network, and/or to collect the individuals' data over the public network. In addition, collecting individual's data can pose a serious privacy threat for the users. These threats can be thwarted by verifying the authenticity of the nodes before establishing a session, and secret key negotiation. But the challenge lies in designing a mutual authentication and secret key exchange protocol with less computation and communication complexities for resource-constrained IoT networks while ensuring security and privacy in all aspects.
- **Research Approach:** The research carried out intends to protect the wireless networks from adversarial threats. We have proposed 3 robust and lightweight mutual authentication and key agreement protocols for diverse applications of IoT. The protocols used Elliptic Curve Qu-Vanstone (ECQV) based implicit certificates, commit/open pair, Diffie Hellman key exchange algorithm, and lightweight cryptography primitives such as hash, XOR, etc.
- **Research Findings:** The messages exchanged in all the 3 proposed protocols are secured to prevent unauthorized access to information. Further, the protocols are resistant to forgery, modification, impersonation, and man-in-the-middle attacks, etc. The accomplishment of the security goals (i.e., secrecy, authentication, and message freshness) of the proposed protocols have been proven through formal (*Automated Validation of Internet Security Protocols and Applications (AVISPA)*, *Burrows–Abadi–Needham* logic) and informal analysis. We have demonstrated through the performance evaluation that proposed protocols have less computation complexities due to the utilization of light weight cryptography operations.
- **Research Implications:** The investigation revealed the superiority of proposed protocols over conventional protocols in terms of efficiency and robustness. Consequently, the recommended protocols are the most suitable alternatives to the existing protocols to secure the resource-constrained applications of IoT.



## Acknowledgement

First and foremost, I would like to thank God, 'Shri Guru Granth Sahib Ji,' for giving me strength, perseverance, and patience to achieve my goals. Further, I would extend my gratitude to his holiness 'Baba Jaswant Singh Ji' for being a true source of inspiration.

I owe my deepest gratitude to the Lovely Professional University (LPU) for giving me a platform and resources to pursue my research ambitions. My sincere thanks to Ph.D. supervisor, Dr. Gulshan Kumar for his guidance and valuable support throughout the Ph.D. I am indebted to my co-supervisor, Dr. Himanshu Monga who continued to supervise my research work despite relocating to another institution. My thesis would not have been accomplished in present shape without the guidance of Dr. Pardeep Kumar (co-supervisor). Dr. Pardeep empowered me with bundles of knowledge and gave continuous assistance despite being thousands of miles away; I dedicate my thesis to Dr. Pardeep Kumar. I am obliged to Dr. Madhusanka Liyanage for reviewing my manuscript and suggesting revisions to improve the quality of representation and work. A special gratitude to Dr. Tai-Hoon Kim for funding my research publications.

I convey my gratitude to the executive dean of Lovely faculty of technology and sciences (LFTS), Dr. Loviraj Gupta; Head of examination and division of academic affairs (DAA), Mr. Navdeep Singh Dhaliwal; Head Research Degree Programme (RDC), Dr. Ramesh Thakur; Head of Division of Research and Development (DRD), Dr. G. Geetha and Dr. Dharm Buddhi for providing all sorts of administrative and technical support required to accomplish my research objectives at Lovely Professional University (LPU). A special thanks to the heads of the School of Electronics and Electrical Engineering (SEEE), Dr. Bhupinder Verma and Dr. Gaurav Sethi for allowing me to use the lab and other facilities required to conduct the research. I must thank the head of the department (HoD), Dr. Kamal Kumar Sharma, and colleagues of SEEE, LPU for encouraging, and motivating me throughout the thesis. I would also like to thank my friends, to name a few, Dr. Rajan Miglani, Er. Sandeep Arora, Dr. Lavish Kansal, Er. Paramdeep Singh, Dr. Raj Kumar, Dr. Harmeet Singh, and Mr. Amarpreet Singh.

Last but not least, I owe the deepest debt of gratitude to my beloved family, especially to my mother, who even in her old age continues to melt like a candle to shine the light for me. A heartfelt prayer for the departed soul of my father who before passing away cultivated the skills within me that leads to the accomplishment of this program. A heartiest gratitude to my wife who stood behind me as a pillar of strength and especially my daughter who made me smile and relieve stress at times of difficulties. The love and affection given by my sister and her family motivated me to accomplish my research goals. I am thankful to the relatives who showered blessings on me. Great thanks to my in-laws for understanding me and forgiving me for all those instances where I had not attended the family events due to ongoing Ph.D. assignments.





*In honor of my dad and homeland*



# Contents

<b>Acknowledgement</b>	<b>v</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Publications</b>	<b>xvii</b>
<b>Abbreviations</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Internet of Things . . . . .	1
1.2 Problem Statement and Motivation . . . . .	3
1.3 Research Questions and Objectives . . . . .	4
1.4 Research Methodology . . . . .	6
1.5 Contribution of Thesis . . . . .	7
1.6 Thesis Outline . . . . .	9
<b>2 Preliminaries</b>	<b>11</b>
2.1 Security Fundamentals . . . . .	11
2.2 Threats . . . . .	13
2.2.1 Passive . . . . .	13
2.2.2 Active . . . . .	13
2.3 Security Requirements . . . . .	14

2.4	Implicit Certificates . . . . .	15
2.5	Mutual Authentication and Key Establishment . . . . .	17
2.6	Lightweight Cryptography . . . . .	17
<b>3</b>	<b>Related Work</b>	<b>19</b>
3.1	Overview of Conventional Schemes . . . . .	19
3.2	Summary and Comparison of Conventional Schemes . . . . .	23
<b>4</b>	<b>Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments</b>	<b>25</b>
4.1	Our contribution . . . . .	27
4.2	System Model, Adversary Model and Security Goals . . . . .	27
4.2.1	System Model . . . . .	27
4.2.2	Adversary Model . . . . .	28
4.2.3	Security Goals . . . . .	28
4.3	Proposed Scheme . . . . .	29
4.3.1	System Set-up phase . . . . .	30
4.3.2	IoT-node registration phase . . . . .	30
4.3.3	Mutual authentication and key exchange phase . . . . .	31
4.4	Security & Comparative Analysis . . . . .	33
4.4.1	Formal analysis . . . . .	33
4.4.2	Informal Proof . . . . .	37
4.5	Performance Analysis . . . . .	40
4.5.1	Experimental setting . . . . .	40
4.5.2	Evaluation of RLMA . . . . .	41
4.5.2.1	Computational costs . . . . .	41
4.5.2.2	Communication cost . . . . .	44

---

<b>5</b>	<b>Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0</b>	<b>47</b>
5.1	Introduction . . . . .	47
5.1.1	Our Contribution . . . . .	49
5.2	System Model, Adversary Model, and Security and other goals . . . . .	50
5.2.1	System Model . . . . .	50
5.2.1.1	WSN-IIoT network . . . . .	50
5.2.1.2	Gateway . . . . .	51
5.2.1.3	Certification Authority . . . . .	51
5.2.2	Adversary Model . . . . .	51
5.2.3	Security and other goals . . . . .	52
5.2.3.1	Security Goals . . . . .	52
5.2.3.2	Other Goals . . . . .	53
5.3	Proposed Scheme . . . . .	53
5.3.1	System Set-up and Registration Phase . . . . .	55
5.3.2	Certificate and Node Key Generation Phase . . . . .	55
5.3.3	Light-weight Key Establishment Phase . . . . .	58
5.3.4	Certificate Renewal Phase . . . . .	59
5.4	Security Analysis . . . . .	61
5.4.1	Formal Analysis . . . . .	61
5.4.2	Informal Analysis . . . . .	64
5.5	Performance and Comparative Analysis . . . . .	67
<b>6</b>	<b>Secure Device-to-Device Communications for 5G enabled Internet of Things applications</b>	<b>73</b>
6.1	Introduction . . . . .	73
6.2	Background . . . . .	75
6.2.1	WiFi Direct Overview . . . . .	76
6.2.2	Short-Authentication-String (SAS) based Key Agreement Protocol	77

6.2.3	Limitations of WiFi WPS . . . . .	79
6.2.4	Cryptanalysis of SAS based Key Agreement Protocol . . . . .	79
6.2.5	Research Contribution . . . . .	82
6.3	Notations and Security Goals . . . . .	83
6.3.1	Notations . . . . .	83
6.3.2	Security Goals . . . . .	83
6.4	Proposed Protocol . . . . .	84
6.5	Performance, Security and Comparative Analysis . . . . .	90
6.5.1	Performance Analysis . . . . .	90
6.5.2	Security Investigation . . . . .	91
6.5.3	Comparative Analysis . . . . .	96
<b>7</b>	<b>Conclusions</b>	<b>99</b>
7.1	Summary . . . . .	99
7.2	Immediate Impact . . . . .	100
7.3	Future Directions . . . . .	100
7.4	Ethical Considerations . . . . .	101
	<b>References</b>	<b>103</b>
	<b>Appendix A AVISPA Installation</b>	<b>119</b>
	<b>Appendix B Burrows–Abadi–Needham (BAN) logic</b>	<b>121</b>
	<b>Appendix C Elliptic Curve Cryptography (ECC)</b>	<b>123</b>
	Appendix C.1 Elliptic Curve Arithmetic (ECA) . . . . .	123
	Appendix C.2 Diffie Hellman Key Exchange (DHKE) Algorithm . . . . .	124
	Appendix C.3 ECC-Diffie Hellman Key Exchange Algorithm (ECDH) . . . . .	125
	<b>Alphabetical Index</b>	<b>127</b>

## List of Figures

1.1	Year by Year market growth of IoT . . . . .	2
1.2	Global IoT Market Share by Sub-Sector . . . . .	2
1.3	Year by Year increase in the IoT devices . . . . .	2
1.4	Research goals and outcomes . . . . .	6
1.5	Process flow of Research Methodology . . . . .	7
2.1	Key Length Comparison. . . . .	16
4.1	Distributed smart environments: IoT home area network; IoT-airport; IoT wireless sensor networks. . . . .	26
4.2	System Model for Authentication of IoT devices in Smart Homes. . . . .	28
4.3	Generation of Implicit Certificate. . . . .	30
4.4	Mutual Authentication and pair wise key establishment. . . . .	32
4.5	Specification of the Node U role. . . . .	34
4.6	Specification of the Node V role. . . . .	35
4.7	Specification of the Session role. . . . .	35
4.8	Specification of the goal and environment for the proposed RLMA. . . . .	36
4.9	RLMA results using OFMC and CL-AtSe backend. . . . .	36
4.10	Time Elapsed for Mutual Authentication and Key Establishment. . . . .	42
4.11	Communication cost comparisons in terms of the number of message exchanges. . . . .	44
5.1	Industrial revolution. . . . .	48
5.2	System Model for mutual authentication and key exchange between IoT devices in Industry 4.0. . . . .	50
5.3	Certificate and Node Key Generation Phase. . . . .	56
5.4	Key Establishment between Node U and Gateway. . . . .	58
5.5	Certificate Renewal Phase. . . . .	60
5.6	Role Specification of the Node U and Gateway. . . . .	62
5.7	Specification of the session, environment and goal for the proposed LKE. . . . .	62
5.8	LKE results using OFMC and CL-AtSe backend. . . . .	64
5.9	Communication Cost Comparison. . . . .	70
6.1	Smart Cities: Connecting everything, comforting lives . . . . .	74
6.2	WiFi Direct Protocol . . . . .	77
6.3	WiFi Direct application scenario: one device shares its cellular connection. . . . .	77
6.4	WiFi Direct application scenario: multiple devices form an ad hoc network . . . . .	78
6.5	SAS based key agreement protocol . . . . .	78
6.6	SAS based key agreement scheme is subjected to MITM attack . . . . .	80

---

6.7	Proposed security protocol for WiFi Direct based smart city applications	86
6.8	Mutual Authentication and Secret Key Establishment between Device A and B . . . . .	88
6.9	Initiation of communication between Device 1 and 2 . . . . .	90
6.10	Case of forged MAC address . . . . .	91
6.11	Bogus request with same credentials . . . . .	91
6.12	Case of new neighbouring device . . . . .	91
6.13	Computation and communication by Device A for performing MAKE .	92
6.14	Computation and communication by Device B for performing MAKE . .	92
6.15	Discovery phase comparison of proposed protocol with conventional protocols . . . . .	97
6.16	MAKE phase comparison of proposed protocol with conventional protocols . . . . .	97
Appendix C.1	Key Length Comparison. . . . .	123
Appendix C.2	Example of elliptic curve: $y^2 = x^3 + x + 1$ . . . . .	124



# List of Tables

1.1	Contribution, applied primitives, and schemes for comparison . . . . .	8
2.1	Generation of Security Keys from Implicit Certificates . . . . .	16
3.1	Review of existing schemes . . . . .	24
4.1	Symbols and descriptions . . . . .	29
4.2	Analysis and Comparison of Protocols based on protection against attacks and security properties . . . . .	41
4.3	Execution time and Energy costs . . . . .	42
4.4	Computation Cost Comparisons . . . . .	43
4.5	Communication Energy Costs . . . . .	44
5.1	Symbols, Abbreviations and Operators description . . . . .	54
5.2	Storage Cost of Proposed Algorithm . . . . .	68
5.3	Analysis and Comparison of Protocols based on protection against attacks and security goals . . . . .	69
5.4	Computation Cost of Scheme LKE for various phases of Operation . . . . .	69
5.5	Computation Cost Comparison for Key Establishment Phase: Between Smart Node and Gateway . . . . .	70
5.6	Energy Cost for communication: Considering Resource Constrained Smart device (Key Establishment Phase) . . . . .	71
6.1	Notations and Denotations . . . . .	83
6.2	Filtering of malicious requests at discovery phase to protect from DoS attack . . . . .	85
Appendix C.1	Diffie-Hellman Key Exchange process . . . . .	125
Appendix C.2	ECC Diffie-Hellman Key Exchange process . . . . .	125



# List of Publications

Publication 1 **Gurjot Singh Gaba**, Gulshan Kumar, Himanshu Monga, Tai-Hoon Kim, Pardeep Kumar, “*Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments*,” in IEEE Access, vol. 8, pp. 69722 – 69733, 2020. {SCIE Indexed, IF = 4.098}; DOI: 10.1109/ACCESS.2020.2986480

Publication 2 **Gurjot Singh Gaba**, Gulshan Kumar, Himanshu Monga, Tai-Hoon Kim, Madhusanka Liyanage, Pardeep Kumar, “*Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0*,” in IEEE Access, vol. 8, pp. 132808 – 132824, 2020. {SCIE Indexed, IF = 4.098}; DOI: 10.1109/ACCESS.2020.3010302

Publication 3 **Gurjot Singh Gaba**, Gulshan Kumar, Tai-Hoon Kim, Himanshu Monga, Pardeep Kumar, “*Secure Device-to-Device Communications for 5G enabled Internet of Things applications*,” in Computer Communications (Elsevier), vol. 169, pp. 114 – 128, March 2021. {SCIE Indexed, IF = 2.816}; DOI: 10.1016/j.comcom.2021.01.010



# Abbreviations

<i>AES</i>	Advanced Encryption Standard
<i>AP</i>	Access Point
<i>AVISPA</i>	Automated Validation of Internet Security Protocols and Applications
<i>BAN</i>	Burrows–Abadi–Needham logic
<i>CA</i>	Certification Authority
<i>CL – AtSe</i>	Constraint-Logic-based Attack Searcher
<i>CPS</i>	Cyber-Physical Systems
<i>DHCP</i>	Dynamic Host Configuration Protocol
<i>DHKE</i>	Diffie–Hellman Key Exchange
<i>D2D</i>	Device-to-Device
<i>DoS</i>	Denial of Service
<i>DY</i>	Dolev-Yao attack model
<i>ECC</i>	Elliptic Curve Cryptography
<i>ECDH</i>	Elliptic-curve Diffie–Hellman
<i>ECQV</i>	Elliptic Curve Qu-Vanstone
<i>5G</i>	Fifth generation technology standard for broadband cellular networks
<i>HAN</i>	Home Area Network
<i>HLPSL</i>	High Level Protocol Specification Language
<i>IoT</i>	Internet of Things
<i>IIoT</i>	Industrial Internet of Things
<i>IWSN</i>	Industrial Wireless Sensor Network
<i>I4.0</i>	Industry 4.0
<i>KDF</i>	Key Derivation Function
<i>LT</i>	Lifetime
<i>LKE</i>	Lightweight Key Exchange
<i>MAC</i>	Media Access Control Address
<i>MAKE</i>	Mutual Authentication and Key Establishment
<i>MITM</i>	Man-in-the-middle
<i>M2M</i>	Machine to Machine
<i>NIST</i>	National Institute of Standards and Technology
<i>NS2/NS3</i>	Network Simulator
<i>OFMC</i>	on-the-fly model-checker

<i>PIN</i>	Personal Identification Number
<i>PUF</i>	Physically Unclonable Functions
<i>P2P</i>	Peer to Peer
<i>RoR</i>	Real-or-Random
<i>RLMA</i>	Robust and Lightweight Mutual Authentication
<i>RSSI</i>	Received Signal Strength Indicator
<i>SAS</i>	Short Authentication String
<i>SHA</i>	Secure Hash Algorithm
<i>TS</i>	Timestamp
<i>WiFi</i>	Wireless Fidelity
<i>WFD</i>	WiFi Direct
<i>WPA</i>	WiFi Protected Access
<i>WPS</i>	WiFi Protected Setup

# Chapter 1

## Introduction

### Chapter Overview

1.1	Internet of Things . . . . .	1
1.2	Problem Statement and Motivation . . . . .	3
1.3	Research Questions and Objectives . . . . .	4
1.4	Research Methodology . . . . .	6
1.5	Contribution of Thesis . . . . .	7
1.6	Thesis Outline . . . . .	9

This chapter gives an overview of the internet of things, problem statement and motivation, research questions and objectives, research methodology followed by an outlining of the thesis.

### 1.1 Internet of Things

Internet of Things (IoT) is an amalgam of 'internet' and 'things' wherein *things* refer to the physical objects and *internet* refers to the interconnection of networks. The IoT is a combination of software (operating system, etc.), and hardware (sensors, communication modules, etc.) and is used to collect, analyze, and disseminate the information in real-time for more efficient and effective utilization. IoT has enabled humans to interact with the machines that in turn have enhanced productivity [1, 2, 3]. The communication technologies such as WiFi, 4G, etc. have played a big role in the evolution of the IoT networks.

IoT is making an impact in every sector, for example, Industrial IoT provides a real-time analysis of the performance of the machines, supply chain, and logistics operations. Due to immense benefits, IoT has achieved enormous growth in the market. As shown in fig. 1.1, the market value of IoT by 2025 is estimated as 1567 billion US\$ [4].

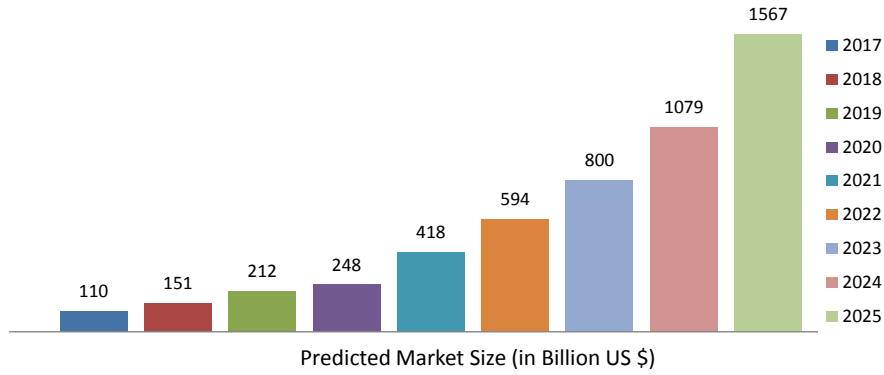


FIGURE 1.1: Year by Year market growth of IoT

Additionally, fig. 1.2 provides the sector wise market share of IoT wherein smart cities, industrial IoT, and connected Health holds the major chunk of the market share [5].

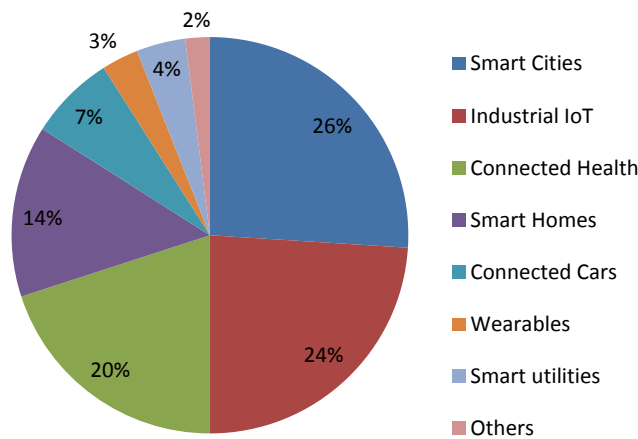


FIGURE 1.2: Global IoT Market Share by Sub-Sector

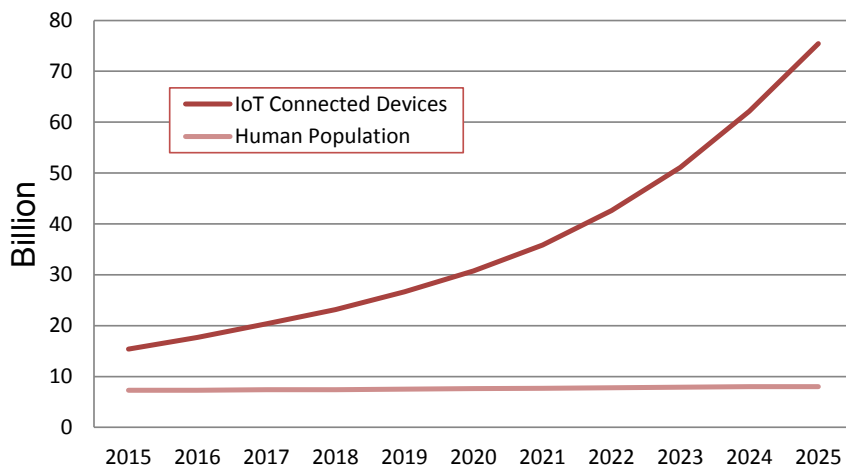


FIGURE 1.3: Year by Year increase in the IoT devices

The fig. 1.3 depicts the growth in the number of IoT devices that may grow from 15 billion to 75 billion by 2025 [6]. The total amount of data expected to be generated by the end of the year 2020 would be around 500 zettabytes which is magnificent in



size [7]. The IoT offers various benefits like collecting the human body information through wearable sensors, enabling the managers to diagnose the fault of the machines remotely, traffic lights are controlled automatically based on the traffic conditions, doctors can monitor the health of the patient from their office following physical distancing, etc.

## 1.2 Problem Statement and Motivation

IoT constitutes of physical systems embedded with sensors, software, and other technologies to facilitate the exchange of information with other devices over the internet. The usability and benefits of IoT has influenced every sector to converge their traditional systems to cyber-physical systems [8, 9]. The significance of the IoT is mainly due to real-time connectivity, mobility, small size, scalable, and flexible framework, etc. Due to enormous benefits, IoT finds its applications in almost all the sectors [10], few are summarized as follows.

1. Internet of Health Care – The doctors can collect the medical information of the patients remotely, etc.
2. Internet of Vehicles – The vehicles can communicate with each other to suggest congestion free routes, etc.
3. Industrial IoT – The engineers can monitor and control the machines remotely.
4. Internet of Smart Things – The user can examine self health conditions using wearable sensors, e.g., smart watch, etc.
5. Smart Cities – The garbage collection staff can optimize their routes according to the intensity of garbage in bins, etc.
6. Smart farming – The activation of water pumps to irrigate the farms can be done remotely by the farmer through internet or automatically based on soil moisture.
7. Smart grids – The theft of electricity can be traced along with notifying the grid manager about the incident, etc.
8. Supply chain management – The items can be tracked easily and more precise information can be provided to customers, etc.

Despite numerous features and benefits, IoT could not successfully influence much due to these issues: unguarded wireless channel, limited memory, less computing power, low battery capacity, and small bandwidth. Due to limited resources, the administrators in the past did not incorporated adequate security measures to protect the networks from cyber attacks. Besides, the vulnerabilities in the employed data security

mechanisms pave an effortless way for cyber attacks especially when identity privacy is compromised. Attackers are always on the lookout to steal precious identity-related information as it can be used to impersonate to get unauthorized access to IoT nodes. Disclosure of identity-related information during the exchange of authentication messages or privacy-leak resulting from authentication failure messages could compromise data and location privacy. Unlike previous attacks, the new generation privacy attacks can even monitor the activities of the IoT-enabled 4G/5G nodes which could further devastate the privacy of the communications. The attackers can harness the intercepted information to get access to unauthorized services, classified data, etc. [11].

In 2018, attackers hacked the complete building automation units because of absence of inadequate authentication procedures [12]; similarly, attackers captured 33,420 patients' medical records through unauthorized access at Barnes-Jewish/Christian health care [13]. There are many identical cases where attackers have exploited the vulnerabilities of the IoT network to fulfill their malicious desires. The possible consequences of these cyber attacks are loss of reputation, financial loss, business suspension, disclosure of confidential information, loss of lives, and loss to infrastructure, etc.

One of the possible measures to protect the network is to embed privacy preserving mutual authentication and key agreement protocol [14, 15, 16]; it enables only the legitimate entities to communicate, thus preventing attacks and consequences. However, the substantial challenge is to perform the mutual authentication and key exchange securely over the vulnerable wireless channel with limited computation and communication complexities. The term 'limited computation complexities' indicates the use of minimal lightweight cryptography operations, whereas 'limited communication complexities' point out the exchange of minimum small size messages. The existing security protocols are not directly applicable because they are resource expensive [17, 18]. Therefore, it is important and urgent to evolve new security protocols to ensure the safe continuity of IoT applications.

### 1.3 Research Questions and Objectives

The main goal of this thesis is to develop robust and lightweight mutual authentication and key exchange protocols for the IoT environment. Following the standard practice, research questions have been framed on the following aspects: an investigation of vulnerabilities and privacy issues in the existing protocols, exploring lightweight cryptography primitives that can be used to develop the protocols, and how the robustness and lightweightness can be examined. The research goals have been carefully prepared to address the research questions. The research questions and the corresponding research goals are discussed as follows.

- **Research Question 1 (RQ1):** What are the various vulnerabilities and privacy issues in the existing IoT security protocols?
  - **Research Goal 1 (RG1):** *To investigate the vulnerabilities and privacy issues in existing IoT protocols.* The goal is to inspect the vulnerabilities in the existing paradigm and framework of IoT security protocols. The identification of the vulnerabilities (e.g., insecure channel, inadequate authentication procedures, etc.) and the corresponding threats (e.g., disclosure, replay, unauthorized access, man-in-the-middle attack, etc.) would be primarily useful for understanding the exact security requirements to protect the IoT networks from various cyber-attacks.
  
- **Research Question 2 (RQ2):** What are the various lightweight and robust cryptography primitives? How these primitives can be used to form privacy preserving, robust and light-weight mutual authentication and secret key establishment protocols for resource-constrained IoT environments?
  - **Research Goal 2 (RG2):** *To design and develop privacy-preserving, robust, and lightweight mutual authentication and key exchange protocols for distributed IoT applications.* The goal is to construct lightweight and robust security protocols for the various specific environments (smart home, industrial IoT, etc.) of the IoT that can overcome the vulnerabilities discovered in *RG1*. The protocol is desired to protect the unauthorized access and also enable the legitimate devices to exchange the secret key for protecting their communications from attackers. However, the existing security protocols are not suitable because they make use of X.509 explicit certificates and other cryptography primitives that are expensive for a resource-constrained IoT node. Therefore, this study involves the identification of lightweight cryptography primitives like ECC, hash, and implicit certificates, etc. that can be used to develop the security protocols. *RG2* intends to disclose the assumptions, system model, adversary model, and the process of combining the cryptography primitives to prepare novel and efficient protocols.
  
- **Research Question 3 (RQ3):** What are the various security analyzer tools and methodologies that can be used to verify the robustness of the proposed protocols against the possible security threats. How one can verify through the identified tools, and how to make inferences from the results obtained?
  - **Research Goal (RG3):** *Examining the robustness of the proposed protocols through verification tools.* The protocols developed as an outcome of *RG2* should be validated in context to robustness through some scientific tools (e.g., Automated Validation of Internet Security Protocols and Applications (AVISPA)) and methods (e.g., Burrows–Abadi–Needham (BAN) logic). The security

protocols have hidden underlying vulnerabilities that sometimes developers could not see, therefore security analyzer tools and methods must be used before implementing the protocol for practical applications.

- **Research Question 4 (RQ4):** How and on what factors should the protocols be compared to be considered as efficient?
  - **Research Goal (RG4):** *Performance and comparative analysis of the proposed protocols.* The devised protocols must be cost-effective. After the accomplishment of *RG3*, the proposed protocol must be compared with state of the art on various aspects such as computation and communication cost. The analysis should reveal the suitability of the protocol in resource-constrained IoT environments. Since the IoT is resource-constrained, it is expected that the devised protocols should use the resources wisely.

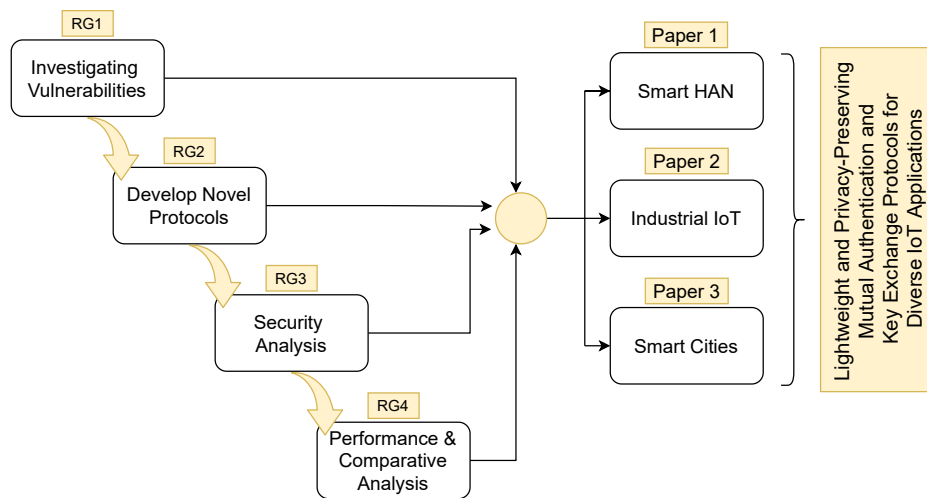


FIGURE 1.4: Research goals and outcomes

Fig. 1.4 links the research goals with the outcomes of this thesis.

## 1.4 Research Methodology

The standard procedure of research methodology has been adopted while designing the mutual authentication and key establishment protocols for diverse applications of IoT. The research methodology can be primarily classified into three sectors: analytical, theoretical, and experimental. The overall flow of the research methodology is illustrated in Fig. 1.5 and discussed as follows:

- *Analytical* refers to reading and understanding the literature to identify gaps in the present security frameworks. It further includes understanding of adversary model and setting up security goals.

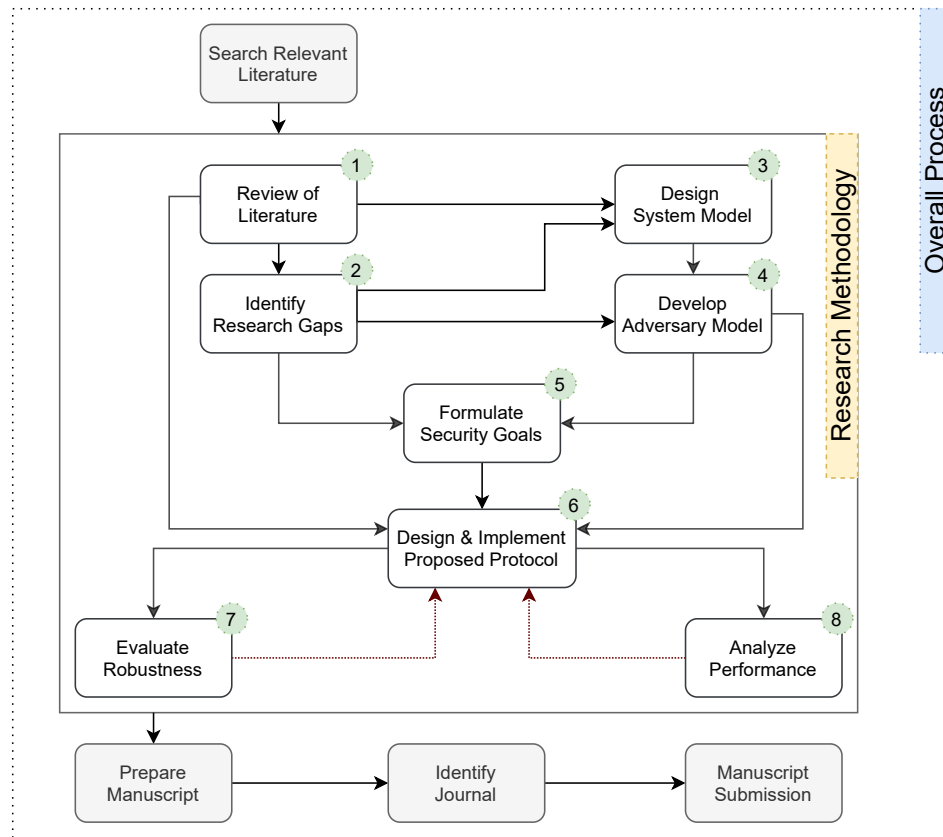


FIGURE 1.5: Process flow of Research Methodology

- *Theoretical* refers to designing of system model and proposing the security protocols using lightweight cryptography primitives. The proposed model should not be prone to attacks despite the attacker having more computation capabilities.
- *Experimental* refers to the implementation of the approach to evaluate its performance through estimations, simulations, or real-time implementations together with robustness evaluation using scientific tools. Eventually, the proposed scheme is compared with the conventional schemes to establish superiority.

## 1.5 Contribution of Thesis

The thesis constitutes three distinct and novel contributions in the form of security protocols. These protocols are developed to prevent the IoT networks from security threats defined in section 2.2.

Paper-I [17] presents a lightweight mutual authentication scheme for distributed smart environments, such as smart homes, smart buildings, etc. The main contribution of this paper is to protect the IoT network and devices from unauthorized abuses. The proposed protocol utilizes symmetric and asymmetric ciphering, digital certificates, and ECC point multiplication to implement the secure mutual authentication and key

TABLE 1.1: Contribution, applied primitives, and schemes for comparison

	Research Contributions	Applied primitives	Vs.
I	Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments	ECQV implicit certificates, Message digest, and symmetric encryption	[19], [20], [21], [22], [23]
II	Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0	ECQV implicit certificates, asymmetric key cryptography, symmetric key cryptography, and keyed-hash	[1], [15], [24], [25], [26], [27], [28]
III	Secure Device-to-Device Communications for 5G enabled Internet of Things applications	commit/open pair, symmetric-key cryptography, message authentication code, diffie hellman key exchange	[29]

agreement framework for IoT networks. The investigation of the proposed technique on AVISPA and comparative analysis with conventional schemes reveals its robustness and superiority.

Paper-II [18] discusses the impact of cyber-physical systems on Industries. Besides, it discusses the various threats on the Industrial IoT due to insecure communication channels. A novel protocol is proposed in this paper to enable the IoT devices to exchange a secret key with the gateway securely over the insecure communication channel. The protocol uses lightweight ECQV certificates to prevent unauthorized access and attain mutual authentication and secret key establishment. The proposed protocol is examined using AVISPA and found secure against all potential attacks, to name a few, MITM, modification, impersonation, forgery, replay, etc.

Paper-III [30] introduces the novel approach developed to secure the 5G enabled D2D communications for smart city IoT applications. The motivation behind this work is the rapid transformation of traditional cities to smart cities. The smart cities collect the information in real-time from various IoT applications through the vulnerable wireless channel, thus making the network and devices prone to adversarial attacks. The main contribution to the paper is to prevent DoS attacks during the discovery phase and negotiating keys for the secure session. The proposed technique uses lightweight cryptography primitives such as the DHKE algorithm, commit/open pair, etc. The proposed protocol has numerous other benefits over conventional schemes and can be integrated easily into the existing D2D framework.

All these contributions together accomplish the objectives of the thesis stated in section 1.3. Table 1.1 provides the details of the cryptography primitives applied in the individual contributions along with the details of the schemes taken into consideration for comparison purposes.

## 1.6 Thesis Outline

The remainder of the thesis is organized as follows:

**Chapter 2** contains preliminary studies on security fundamentals, threats, and security requirements. The chapter also includes discussion on implicit certificates, mutual authentication and key establishment, and lightweight cryptography.

**Chapter 3** discusses the various conventional mutual authentication and key exchange approaches developed by the researchers for the IoT applications. The chapter ends with a summary of the research gaps identified after studying the literature.

**Chapter 4** elaborates the system model, adversary model, security goals, and the proposed scheme of the first research contribution, “Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments”. Besides, the chapter covers the security, performance, and comparative analysis of the proposed approach.

**Chapter 5** uncovers the second research contribution, “Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0”. The methodology from registration to authenticated key agreement has been discussed in this chapter. The last few sections of the chapter include the security analysis through AVISPA and performance estimations considering TelosB mote.

**Chapter 6** reveals the significance of IoT in smart cities and the security complications in the IoT D2D communication model. The chapter covers the cryptanalysis of the conventional approach and also discusses the new lightweight approach with strong immunity against attacks (third research contribution). In addition, simulation results and security analysis (BAN logic) are presented.

**Chapter 7** summarizes the research gaps and the security requirements in the IoT framework and how it has been addressed by the proposed protocols. Besides, it provides how the proposed research would benefit society, science, and researchers in the current era and future.

Lastly, appendices cover the AVISPA installation guide, BAN logic rules, and briefing of various ECC based techniques (DHKE, ECDH, ECQV) used in the thesis work.





## Chapter 2

# Preliminaries

### Chapter Overview

2.1	Security Fundamentals . . . . .	11
2.2	Threats . . . . .	13
2.3	Security Requirements . . . . .	14
2.4	Implicit Certificates . . . . .	15
2.5	Mutual Authentication and Key Establishment . . . . .	17
2.6	Lightweight Cryptography . . . . .	17

## 2.1 Security Fundamentals

There are various security fundamentals whose knowledge is essential to understand the thesis work. The few important security fundamentals [31] are discussed as follows:

- **Asymmetric Key Cryptography:** It is a cryptography system that makes use of different keys (public, private) for ciphering and deciphering. The public key is disseminated publically by the owner whereas the private key is kept secret. In one instance, the sender can use the receiver's public key to encrypt the information. Since the private key is available with the receiver, no one else would be able to decrypt the information. Elliptic Curve Cryptography (ECC) is the most significant scheme that works on the principle of asymmetric key cryptography (also known as public-key cryptography).
- **Symmetric Key Cryptography:** In this type of cryptography, only a single key is used to encrypt and decrypt the information. Therefore, the sender and receiver should negotiate the security key before the start of the real communication. As

we know that establishing the secret key via an insecure wireless channel is susceptible to interception, therefore in some applications, asymmetric key cryptography is used to exchange the symmetric keys. Advanced encryption standard (AES) is an example of symmetric key cryptography.

- **Digital Certificates:** Digital certificates enable the receiver to verify the authenticity of the sender and also ensure that the message is not altered during the transition. It is one of the required elements of the cryptography suite as it helps to detect forgery and tampering. It can be considered as an equivalent to handwritten signatures but digital signatures are comparatively more difficult to forge. Digital signatures are constructed using asymmetric cryptography. The examples of digital signatures are explicit certificates (X.509), and implicit certificates (Elliptic Curve Qu-Vanstone (ECQV)).
- **Hash Algorithms:** Cryptographic hash functions are one-way functions that transform the arbitrary sized message to a fixed size message digest. The hash algorithm takes into consideration every single bit of message while execution, hence a minor variation in the message produces a completely different output. There are many variants of hash algorithms, a few of them are secure hash algorithms (SHA-128, SHA-256), and message-digest (MD5).

The hash algorithm should exhibit the following properties:

1. Two different input messages should not produce an identical message-digest.
  2. Hash algorithms must process the inputs quickly with little burden on the computing processor.
  3. It should be computationally infeasible to retrieve the message from a given hash.
- **Message Authentication Code (MAC):** MAC is also known as a keyed-hash. MAC uses a hash algorithm and two inputs namely, message and secret key. The secret key must be known to the sender and receiver before the actual communication. MAC enables the receiver to verify the integrity of the received message together with verification of sender authenticity. One of the widely used MAC is hash-based MAC (HMAC).
  - **Random Number Generator (RNG):** RNG uses a complex deterministic algorithm to produce a string of numbers that appears random. The RNG uses a secret seed to initiate the algorithm. Post initialization, RNG continues to generate the random numbers within a finite field. The strength of the RNG depends upon the algorithm, the size of the seed, and the secrecy of the seed, etc. RNG's are divided into Pseudo-RNG (PRNG) and True-RNG (TRNG). TRNG extracts a seed from the natural phenomenon and cannot be reiterated. For these reasons, PRNG is used in cryptography for various applications such as session key generation, key distribution, authentication, nonces, and one-time pad, etc.

The random number generator should exhibit the following properties:

1. **Randomness:** The total number of binary 0 and 1 should be uniformly and randomly distributed.
  2. **Unpredictability:** It should be computationally infeasible to predict the future random number based on the current set of random numbers produced by the generator.
- **Commit/Open Pair:** The commit/open pair is designed in such a way that a commitment value alone does not disclose any information about the secret value. A commit/open pair intends to provide secure mutual authentication between the parties involved in communication.

Let us consider the following commitment scheme wherein  $F$  is a cryptographic hash function:

- **Commit:** Given  $w$ , randomly choose  $p \leftarrow \{0, 1\}^n$ , compute  $c = F(w, p)$ .
- **Open:** Let  $d = (w, p)$ . Output  $w$  if  $c = F(w, p)$ .

In simple terms, the receiving device accepts the secret value  $w$  if  $F(d) \equiv c$  i.e., earlier received commit value.

## 2.2 Threats

Threats refer to the exploitation of vulnerabilities by the attacker to accomplish malicious goals. The consequences of the threat range from benign to severe. The attacks are classified into two categories [32]:

### 2.2.1 Passive

Passive attacks are silent attacks where the adversary intends to eavesdrop on the messages exchanged in the network with no intentions to modify or harm the network resources. During eavesdropping, the attacker can capture the messages to retrieve the unauthorized information. The message may contain sensitive information whose disclosure to an attacker can pose serious threats.

### 2.2.2 Active

In the active attacks, the attacker modifies the network messages and resources to harm the routine network operations. Few active attacks are summarized as follows:

- **Denial of Service (DoS):** DoS attacks are a form of attack where the adversary has intentions to hinder the services provided to legitimate entities. DoS can be performed by flooding the victim device with malicious requests; the legitimate device exhausts all its resources while processing these fake requests and eventually fails to provide the services to the legitimate user/devices.
- **Man-in-the-middle (MITM):** In this attack, the attacker silently intercepts, modifies, and relays the messages. The process is so smooth that the communicating parties never realize that they have initiated a session with the attacker rather than the legitimate counterparty. MITM attack is only possible if the attacker has sufficient information to impersonate the identities or the security protocol has loopholes.
- **Modification:** Considering the vulnerabilities of the wireless medium, it is believed that the attacker can intercept the communication. Therefore, the attacker can modify the received message to get privileged access, confidential information, and maybe to generate a secret session.
- **Impersonation:** Attacker can intercept the messages to collect the private information of devices/users such as key, password, etc. for impersonating the identity of legitimate entities.
- **Known Key:** The attacker can try to generate new keys from the old expired (known) keys.
- **Replay:** The attacker can capture the messages to replay later for getting access to unauthorized privileged information or account.

## 2.3 Security Requirements

To overcome the threats mentioned in section 2.2, the security administrator must ensure the attainment of the following security properties [14, 15, 24]:

- **Confidentiality:** To prevent the disclosure of information to unauthorized parties.
- **Integrity:** To protect the data from unauthorized modifications.
- **Availability:** To ensure the disruption-free network services.
- **Anonymity:** To keep the identity of the communicating entities private.
- **Untraceability:** To prohibit the trace of the message journey, i.e., origination and destination.

- **Freshness:** To ensure newness in the information.
- **Key Security:** To keep the session keys confidential.
- **Mutual Authentication:** To verify the authenticity of each other before establishing a secret key.

These properties can be accomplished at the cost of implementing some cryptography tools like ciphering, keyed hash functions, message authentication code, and digital certificates, etc. [19, 33] However, these mechanisms are difficult to execute in IoT devices due to limited resources (storage space, computing power, battery, etc.) [34]. One of the possible solutions is to process the computation in an external node and post-processing store the information like keys, etc. in the IoT. But this solution may not be practical because it is not possible to store a large amount of information due to limited storage space as well as the threat of side-channel attack. Moreover, it would be infeasible to bring the devices every time to the administrator for updating the secret information upon expiration. The other possible and feasible solution is to make use of cryptography protocols to secure the communication in IoT networks. The IoT devices can perform mutual authentication and secret key establishment remotely before the exchange of precious information [20, 35, 36]. The cryptography protocols are robust enough to exchange and renew the keys, and certificates, etc. over the vulnerable wireless medium.

## 2.4 Implicit Certificates

Digital certificates have a globally unique identifier that can be used by the communicating parties to verify the authenticity of each other before connection establishment. Identical to X.509 explicit certificates, implicit certificates are composed of (a) *identification data*, (b) *a public key*, and (c) *a digital signature* [19]. The public key is tied to the user's identification data and the binding between them can be verified easily by the trusted third parties. In explicit certificates, digital signature and public key are two different elements whereas they are included within implicit certificates to reduce the bandwidth requirements.

The significant advantages of implicit certificates over conventional X.509 certificates are rapid computation, tiny certificate size, and low computational power [37]. Fig. 2.1 depicts the comparison of symmetric and asymmetric key size requirements to attain the same level of security. The curves in the graph validates the fact that ECC based approaches have less key size requirements in contrary to symmetric ciphers, e.g., RSA [38]. The *Elliptic Curve Qu-Vanstone* (ECQV) implicit certificates are fundamentally derived from the ECC to cater to the needs of resource-constrained networks. ECQV has a

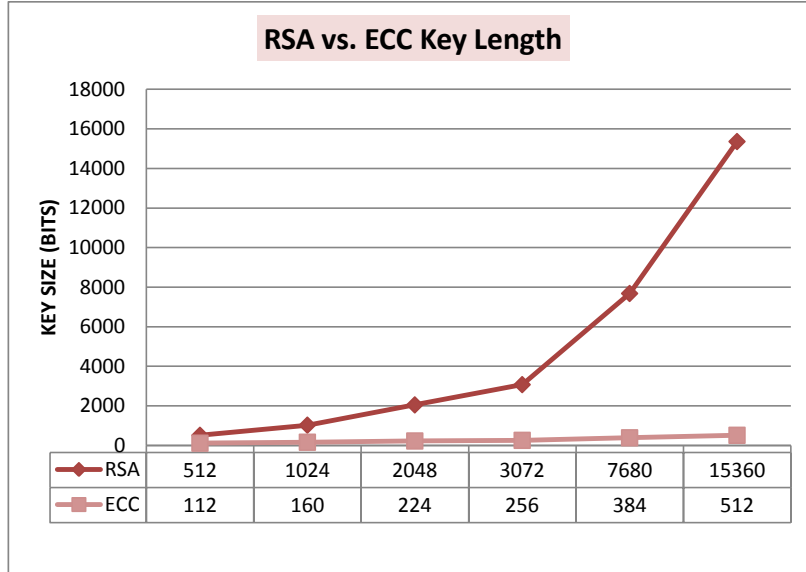


FIGURE 2.1: Key Length Comparison.

TABLE 2.1: Generation of Security Keys from Implicit Certificates

Action	Alice	CA
Generate	$r_A \in [1, n - 1]$	
Compute	$R_A = r_A * G$	
Alice transmits $R_A$ to CA		
Generate		$r_{CA} \in [1, n - 1]$
Compute		$Cert_A = R_A + r_{CA} * G$ $e = H(Cert_A)$ $s = er_{CA} + d_{CA} \pmod{n}$
CA transmits $Cert_A$ and $s$ to Alice		
Calculate private	$d_A = er_A + s \pmod{n}$	
Calculate public	$Q_A = eCert_A + Q_{CA}$	
Public and Private Keys are established successfully		

smaller size, low computation requirements, and require lesser bandwidth; these benefits make the ECQV a better choice over RSA based X.509 explicit certificates [37]. As the proposed schemes are intended for IoT resource-constrained networks, ECQV implicit certificates have been used instead of X.509 explicit certificates. Table 2.1 elaborates the whole process, including the process of certificate request by IoT node, certificate and signature generation by Certification Authority (CA), and a public-private key pair calculation from the signature and certificate by the IoT node.

## 2.5 Mutual Authentication and Key Establishment

Mutual authentication is a process to protect the network resources from malicious nodes. It is performed before the key establishment to allow only the legitimate nodes to negotiate a security key and access the authorized resources. The mutual authentication must be robust enough to protect from attacks; besides, it should use lightweight cryptography primitives to prevent excessive drainage of precious resources of nodes.

The existing key agreement protocols incur high computation costs which makes them unfit in present form for resource-constrained nodes. The requirements to be kept in mind while designing the security protocol are authentication, confidentiality, integrity, and anonymity, to name a few. These security properties can be attained with the use of symmetric and asymmetric cryptography, digital certificates, and hash algorithms, and so forth. The nodes complete the keys agreement with each other during the initialization phase and use them unless expired. As the IoT nodes may be deployed in a hostile environment, it may not be easily possible to replace the battery. Therefore, it is always essential to construct efficient schemes to extend the lifetimes of the nodes [39].

## 2.6 Lightweight Cryptography

The existing protocols need to be altered to suit the resource-constrained nature of the networks. The resource-deprived nodes suffer from computational capability, memory space, and battery power. Conventional security protocols were constructed to secure the mains powered wired computer networks. However, the use of these security protocols is not possible for low powered wireless networks. The main challenge is to choose the cryptography primitives that provide acceptable security while consuming minimum cost. There can be two possibilities:

1. Use of conventional protocols with shrunk security parameters like reduced key size, lesser iterations, etc. It reduces the strength of the security and makes the network more prone to attacks.
2. Designing of new protocols with specific IoT network requirements.

The latter one is more preferred. Lightweight computations [40] can rely on symmetric and asymmetric key cryptography; however, binary calculations through XOR are the lightest operation that can be put into use. But standalone XOR does not ensure the accomplishment of all the security properties like key management, digital signatures, etc. Therefore, researchers have started using discrete logarithm problems (DLP) to construct lightweight cryptography protocols based on ECC. Implicit certificates (ECQV) are an example of lightweight cryptography that helps to verify the authenticity of other devices at minimal cost.





## Chapter 3

# Related Work

### Chapter Overview

3.1 Overview of Conventional Schemes . . . . .	19
3.2 Summary and Comparison of Conventional Schemes . . . . .	23

To provide security in smart environments, several schemes have been proposed. Each scheme has its own merits and demerits. This chapter briefly covers the paradigm, primitives, complexity, and deficits of conventional schemes. Lastly, a tabular summary is provided for a clear understanding of the research gaps that need to be filled through the development of new security protocols.

### 3.1 Overview of Conventional Schemes

Esfahani et al. in [1] presented an authentication model for M2M communications in IIoT environment. The authors stated that traditional schemes cannot be used in IIoT due to excessive overheads which may drain the node resources. Therefore, authors have devised a new security model which computes only *hash* and *ex-or* operations during authentication. Due to use of few cryptography operations, the authors declared their scheme as computation efficient. The authors further claimed that their scheme exhibits the security properties such as session key agreement and is also resistant against replay, man-in-the-middle (MITM) attacks, etc. Indeed the scheme is providing many security benefits, the authors have not performed vulnerability assessment and formal analysis, thus behaviour of the scheme is unpredictable under compromised conditions. In addition, the scheme spends a lot of energy in communicating large sized mutual authentication and key exchange messages, which makes the scheme energy inefficient. Therefore, the proposed scheme is not suitable for IIoT networks due to its unpredictable behaviour and high energy consumption.

In another research [14], the authors have elaborated that IoT networks have become a honeypot for attackers, thereby turning the privacy of the individuals under threat. The session key in their protocol is continuously renewed to prevent replay attacks. However, the authors have introduced several cryptosystem operations which made it bulky e.g. eight times hashing operations.

Das et al. in [15] raised a concern on security and privacy of Industrial IoT networks due to use of open channel for communication. The authors believed that existing schemes may not be suitable in IIoT specific environment due to excessive overheads. The authors formulated a Biometric-based privacy preserving authentication scheme to combat against unauthorized intrusions with limited overheads. The scheme makes use of biometric and smart card as a 2 factor authentication process. The protocol has been simulated on NS2 to verify its behaviour. The authors performed formal and informal security analysis and declared their scheme as robust against various attacks. In spite of 2 factor authentication, the scheme fails to ensure privacy and protection against known key attacks.

Gope et al. [16] have not only emphasized on vulnerability of IoT devices at public places but also realized a need of robust IoT device authentication strategy. The authors proposed an authentication model using PUF to make IoT devices invulnerable to physical and cloning attacks. Authors claimed that their scheme is resilient to impersonation, achieves untraceability and also exhibit security properties e.g., mutual authentication, protection against physical attacks etc. However, their scheme may incur high computation requirements due to massive use of hash operations and high communication complexities. Thus, the scheme may not be pertinent for the resource constrained and sensitive applications of IoT.

Sciancalepore et al. [19] have implemented a public key based authentication and key agreement protocol for IoTnet. The authors have employed ECDH with ECQV implicit certificates for achieving authentication. However, in their scheme, 2 different keys are needed to operate the protocol and the efficiency of key generation depends upon the key derivation function. Hence, malfunctioning of KDF may lead to connection abortion between entities. In their scheme, future keys are generated with the use of master key and any disclosure of related information may lead to loss of forward secrecy.

In [20], the authors proposed an authentication model for IoT enabled smart home. The authors claimed that their scheme is lightweight and secured against vulnerabilities. The basic idea of this scheme is to utilize the concept of temporary identity, keyed-hash chain mechanism and fog computing to achieve mutual authentication and identity assurance. Nevertheless, the scheme may fail to provide complete confidentiality and protection against DoS, known key attacks etc. Moreover, the communication and computation cost in [20] is a hindrance to its acceptance as an authentication model for resource limited devices of smart homes.

Patel et al. have described an authentication and access control protocol for IoT [21]. The scheme has used ECC based mutual authentication (EMA) and capability based access control (CBAC) for operation. Elliptic Curve Discrete Logarithm Problem (ECDLP) and ECDH are used for generating and sharing the common secret keys for authentication. In order to do this, the protocol utilized a plethora of cryptosystem operations which make it compute expensive.

Hossain et al. [22] have proposed an authentication technique, which is based on hardware and software co-verification for IoT. The authors have pointed out that since inception of IoT, targeting devices through cloning of hardware has become easy. To address cloning issue, they have proposed a physical unclonable function (PUF) based security protocol. The proof-of-concept is implemented on Contiki operating system. This method is claimed to be very first attempt to prevent the IoT devices from cloning and reprogramming attacks.

Dey et al. [23] developed a model of authentication for smart homes. The authors emphasized the need of a new security model for smart homes as distinct devices with different computational abilities work altogether. Their scheme exploited the Diffie Hellman Key Exchange (DHKE) protocol for achieving the mutual authentication and sharing of key. The security strength of their scheme is evaluated on protocol security analyzer tool *AVISPA* (automatic verification of internet protocols and applications). However, in spite of emphasizing on the computation and communication cost, the scheme still incurred high complexities, fails to ensure message freshness and may not withstand with known-key attack.

Gope et al. in [24] focused on the realization challenges of Industrial WSN (IWSN). Considering the security as the most significant challenge, the authors devised a new bilateral authentication scheme for the real-time IWSN. The authors applied exclusive-or, one way hash, and physically unclonable functions (PUF), to name a few, in their algorithm. The main strength highlighted in the paper is the security of the credentials even if the sensor nodes are physically captured by the adversary. The scheme ensures mutual authentication, and integrity, etc. Despite the benefits, the scheme exchanges 6 messages to accomplish session key which is itself a challenge for resource constrained devices. The number of bits exchanged in those messages is quite much in quantity which further escalates the energy consumption bar. This immense energy consumption can deplete the energy reserves of IIoT nodes quickly. Moreover, the behavior of the schemes [24, 28] under the influence of the DoS attack is not observed, therefore adversaries can exploit the hidden vulnerabilities to attack the IIoT networks.

Li et al. in [25] discussed the challenges in implementing security protocols i.e., open nature of wireless medium and resource constrained nodes. The authors proposed a 3 factor user authentication protocol for WSN-IIoT environment while keeping these

challenges into consideration. The three factors used to authenticate are *user's identity*, *password* and *biometric*. The user is only able to access the sensor's data if all the factors generate positive results. The authors declare that their scheme is resistant to impersonation, replay attack, etc. however validation using formal analysis is found missing. The scheme is communication inefficient as the resource constrained node *transmits* and *receives* a total of 2688 bits for the key exchange process. Consequently, the scheme is unfit for resource constrained applications of IIoT.

ECC based authentication protocol for IIoT has been presented by Li et al. in [26]. The authors emphasized on the need of authentication mechanism to prevent from unauthorised access due to unsecured nature of medium in wireless sensor networks. Their scheme makes use of biometrics to identify the legitimacy of the entity. The authors simulated their scheme on NS3 to determine the performance. Regardless of the claimed advantages, it is found that authors have not considered Denial of Service (DoS) and MITM attacks during security analysis which may pose threats to network existence. It is evident that scheme fails to provision privacy and message freshness for all exchanged messages due to absence of ciphering and nonce, respectively.

Paliwal in [27] has expressed his concern over integrity and confidentiality of data in IIoT networks. The author emphasized that sensitive information collected by the sensor nodes in Wireless Sensor Networks (WSN) should be accessible to intended recipients only. The article briefs the various existing authentication schemes along with their vulnerabilities. The scheme makes use of hash to achieve mutual authentication and key establishment whilst ensuring anonymity of identities. The scheme is claimed as lightweight and efficacious due to limited computations and resistance against many significant attacks. The author has affirmed that the scheme has undergone formal and informal analysis and is declared secured to be used in IIoT environment. Despite the fact the scheme is asserted robust, the scheme does not ensure privacy. Though scheme does not make use of any ciphering model but extensive use of hash and large size of exchanged messages over burdens the overall scheme.

Chang et al. in [28] introduced an authentication scheme for WSN to prevent unauthorised penetrations. Though claimed as efficient and secured but complex as it operates in twin modes. The authors have tried to overcome the deficiencies of existing authentication protocols by introducing a smart card based authentication scheme. Their protocol works on 2 different algorithms and attains 2 different set of security properties accordingly. The authors have performed formal security analysis using Real-Or-Random (RoR) model to prove the robustness of their protocol. It is observed that their first protocol ( $P_1$ ) does not offer complete security solutions whereas the second one ( $P_2$ ) is resource expensive. Since the WSN-IoT devices are resource-constrained, using these approaches can lessen the lifetime of the nodes. Therefore, their protocol cannot be deployed readily unless modified to suit the requirements of IoT networks.

Kumar et al. have suggested a lightweight session key establishment protocol for smart home environments [35]. A session key is produced using a short authentication token, which uses the silicon chip-identity. The authors claimed their scheme is efficient in terms of computation and communication and capable of protecting against attacks e.g., DoS, eavesdropping, masquerade, etc. In addition, their scheme satisfies the property of mutual authentication, session key establishment, confidentiality, integrity, and freshness. However, the scheme may not resist time synchronisation attacks. For instance, if clock loses synchronisation, then the scheme is vulnerable to replay attack. Moreover, anonymity and unlinkability issues are not addressed in the scheme [41].

In [36], Sciancalepore et al. have proposed a key management protocol for IoT networks (IoTnet). The scheme is based on the concept of Elliptic Curve Diffie Hellman (ECDH) and Elliptic Curve Qu-Vanstone (ECQV) implicit certificates. The authors have claimed that their proposed scheme is lightweight and secure against security attacks. However, the threat model designed in this paper does not include many popular attacks, such as impersonation, MITM, etc. As a consequence, their scheme may be incompetent to protect against impersonation and MITM attacks. In addition, to execute the scheme (e.g., mutual authentication phase), the system incurs high time complexities. Therefore, this scheme may not be practical for resource-constrained devices.

In [37], Porambage et al. have introduced introduced a pair-wise key establishment scheme for wireless sensor networks (WSNs). The scheme uses ECC based implicit certificates for pair-wise key establishment. The authors first performed the bootstrapping followed by establishment of pair-wise key between nodes. However, physical capturing of a node may lead to disclosure of authentication key, which may emanate high security risks to other non-compromised IoT applications.

## 3.2 Summary and Comparison of Conventional Schemes

In summary, most of the schemes are insufficient as they are either not considering reasonable threat model that might cause some security issues or incurring high complexities at resource constrained nodes. It is evident from Table 3.1 that conventional schemes are vulnerable to the attacks such as MITM, Known Key, and DoS etc. Additionally, Table 3.1 provides detailed insights on various aspects including the employed cryptography primitives, area of application, computation and communication complexities, verification using formal analysis, and non-accomplished security properties, etc. These concerns make the conventional schemes unfit for the resource constrained sensitive applications of IoT. Thus, there is a necessity of an authentication scheme (while providing privacy) which can protect the distributed smart environments from unauthorized abuses with less complexity and more robustness against attacks.

TABLE 3.1: Review of existing schemes

Scheme	Cryptography Primitives										A	T	CPE	CME	Y	FA	NASP
	SC	AC	PUF	BM	PW	ID	Hash	IC	SC/T								
[1]							✓				IIoT	1	L	H	2019		I, II, III
[14]	✓						✓				IoT-SHN	1	M	L	2017	✓	IV
[15]				✓			✓	✓			IIoT	3	M	M	2018	✓	I
[16]			✓				✓				IoT	1	H	M	2019	✓	I
[19]		✓					✓				IoT	1, 3, 5, 8	M	H	2017		I, III, IV
[20]	✓				✓		✓				IoT	1, 3	L	H	2019	✓	I
[21]	✓	✓					✓				IoT	3, 4, 5	L	M	2016	✓	III, IV
[22]		✓	✓				✓				IoT	1, 2, 3	H	L	2017		I, III, IV
[23]	✓						✓	✓			IoT-SHN	1, 3	H	H	2019	✓	I, III, IV, V
[24]			✓	✓			✓				IWSN	1	M	H	2019	✓	I
[25]	✓			✓	✓		✓				IIoT	1, 2, 3	H	H	2018		I, II
[26]				✓			✓				IIoT	1, 2	M	H	2018	✓	I, V
[27]				✓			✓				IIoT	1	H	H	2019	✓	I, III
[28]				✓			✓	✓			IoT-WSN	1, 2	M	H	2015	✓	I, III
[35]	✓				✓		✓	✓			IoT-SHN	6, 7	M	L	2016	✓	III, IV
[36]		✓					✓				IoT	3, 4	M	M	2015	✓	III, IV
[37]		✓					✓	✓			IoT-WSN	1, 2, 3	L	M	2013		I, II, III, IV

SC: Symmetric Cryptography, AC: Asymmetric Cryptography, PUF: Physically Unclonable Function, BM: Biometric, PW: Password based, ID: Identity based, IC: Implicit Certificate, S C/T: Smart Card/Tag, A: Application, IIoT: Industrial IoT, WSN: Wireless Sensor Networks, SHN: Smart Home Network, T: Threats, CPE: Computation Expenses, CME: Communication Expenses, 1: Denial of Service, 2: Man-in-the-middle, 3: Known Key, 4: Impersonation, 5: Modification of Messages, 6: Time Synchronisation, 7: Replay, 8: Node Compromise, L: Low, M: Moderate, H: High, Y: Year, FA: Formal Analysis, NASP: Non-accomplished Security Properties, I: Privacy, II: Session Key Security, III: Identity Anonymity, IV: Untraceability, V: Message Freshness

## Chapter 4

# Robust and Lightweight Mutual Authentication Scheme in Distributed Smart Environments

### Chapter Overview

4.1	Our contribution . . . . .	27
4.2	System Model, Adversary Model and Security Goals . . . . .	27
4.3	Proposed Scheme . . . . .	29
4.4	Security & Comparative Analysis . . . . .	33
4.5	Performance Analysis . . . . .	40

A smart environment is one of the emerging trends that allow people and objects to stay connected via the information and communication technologies. Smart environments (also known as IoT) include smart homes [42], smart healthcare [43], smart car and cities [44] and many more. Note that smart environments/objects and IoT applications/objects are interchangeably used. In a recent research report, it is estimated that the “things” in connected smart environments to grow tremendously and is anticipated to reach up to billions of devices by 2025 [45].

In smart environment, IoT objects are computationally constraint devices that can sense, compute, and extend connectivity between the last miles systems and users via the Internet in a ubiquitous manner. Fig. 4.1 shows a typical network of distributed smart environments, where several heterogeneous objects/nodes are installed to control and monitor the applications through the IoT cloud. All the sensors, objects or nodes collect data within their respective environments and send it to the cloud via the networking technologies, e.g., Zwave, ZigBee, and other IoT protocols. The collected data can be



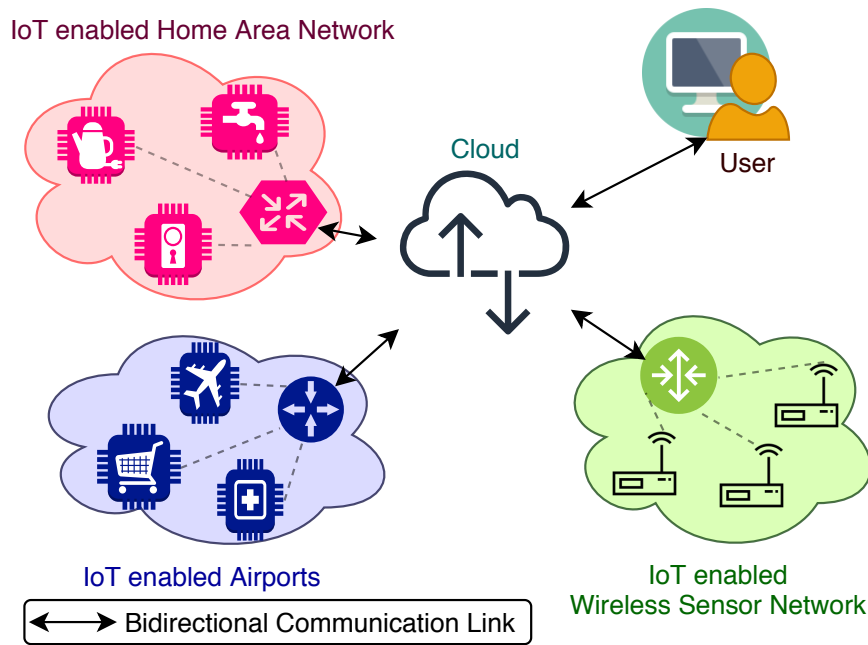


FIGURE 4.1: Distributed smart environments: IoT home area network; IoT-airport; IoT wireless sensor networks.

used for many purposes which depend on an application of interest e.g., health monitoring, data analytics for smart homes and cities [46], faults reporting in a flight system, leakage alarm of chemical in a factory etc.

As data from IoT objects is precious, inadequate security measures in IoT devices may invite various security threats to the applications. An unauthorized data access may cause harm to an application where the end-users are directly involved. An attacker may exploit vulnerabilities in IoT devices to collect data through eavesdropping, and may gain financial profit by selling collected data. Moreover, recently security researchers have pointed out several vulnerabilities in smart cities technologies, few of them are attributed to *authentication flaws*, thus leaving IoT applications unsecured [47]. Ali-Awad have pointed out various vulnerabilities including *lack of sufficient authentication* in the smart home technologies, and have claimed that these vulnerabilities may pose many risks to the individuals [48].

In [49], security researchers have claimed that an attacker can access several home routers (i.e., 1,700 IoT devices) by exploiting a list of default login credentials on the IoT devices. Stellios et al. have shown verified cyberattacks on various IoT enabled domains, e.g., smart grid, intelligent transport network, industrial control system, medical IoT, and smart homes, etc. [50]. The authors have also claimed that the vulnerabilities (e.g., *design flaws in authentication mechanism*) in a smart light may lead to many threats in a smart home. Moreover, a Dyn Attack is carried out by the IoT Botnet named 'Mirai' which has seriously affected many of IoT devices as claimed in [51]. Nevertheless, such *lack of sufficient authentication* and/or *design flaws in authentication* mechanisms



in IoT devices leads to sensitive information or data breach which may be misused. Resultant, security has been one of the main challenges in the success of distributed smart environments and applications.

## 4.1 Our contribution

- We propose a robust and lightweight mutual-authentication scheme (RLMA) for the distributed smart environments.
- To achieve the efficiency and lightweightness at resource constrained nodes, elliptic curve cryptography (ECC), implicit certificates, and symmetric encryption are used.
- The proposed scheme exhibits several security properties, such as mutual authentication, session key agreement, message freshness and anonymity and/or untraceability. Besides security properties, security analysis also shows that the proposed scheme is secure against many security attacks, e.g., replay, message modification, node compromise, key compromise, impersonation, known key, denial of service (DoS), and man-in-the-middle (MITM).
- Performance evaluation (including energy efficiency) and comparison demonstrates its high computational and communicational efficiency as compared to the state-of-the-art schemes.

## 4.2 System Model, Adversary Model and Security Goals

### 4.2.1 System Model

Fig. 4.2 depicts a high level system model in distributed smart environment. The system model mainly consists of following entities, such as IoT nodes, bi-directional communication channel, certification authority, etc.

1. *WSN-IoT Network*: In a smart network, the resource-constrained sensor nodes collect the data (e.g., humidity, light, etc.) from their respective environments and send the data wirelessly to the sink node via utilizing low-powered technologies, e.g., ZigBee. More precisely, sensors data is easily available from anywhere in an ad-hoc manner. From the security perspective, the IoT nodes request security credentials from the certificate authority. These security credentials are later utilized to perform the mutual authentication.

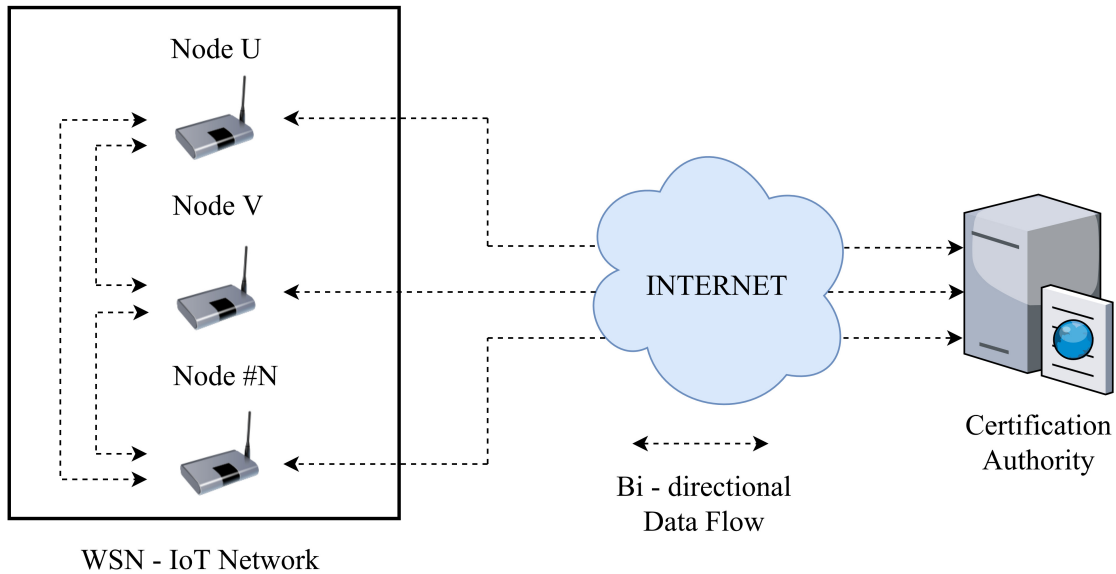


FIGURE 4.2: System Model for Authentication of IoT devices in Smart Homes.

2. *Certificate Authority (CA)*: The CA is a trusted entity, and is responsible for generating and distributing implicit certificates to the entities. Moreover, it is considered to be a tamper proof entity.
3. *Communication Link*: In the distributed IoT applications, IoT-nodes communicate with each other through bi-directional wireless technologies, such as Zigbee, Bluetooth, etc. In addition, the IoT nodes can communicate to CA either directly through GPRS/WiFi functionality or via gateway and cloud.

#### 4.2.2 Adversary Model

Following [52], consider a smart living environment where an attacker can have full control of the IoT network, and can modify, alter, drop and replay the wireless messages to mount different attacks. More precisely, an adversary can replay old messages with an intention to get unauthorized access between two smart devices. An attacker can perform the impersonation attack by creating the fake legitimate identity to steal critical information from entities. An attacker can disrupt the operations of the CA/IoT node through DoS and MITM attacks.

#### 4.2.3 Security Goals

The proposed scheme provides following security goals. Note that the security goals are adopted from [14],[53].

TABLE 4.1: Symbols and descriptions

Notations	Descriptions
$r_U, R_U$	Random integer and E.C. point generated by node U
$G, n$	Base point generator and its order
$r_{CA}$	A random integer value generated by CA
$Cert_N$	Implicit certificate of $N^{th}$ node
$e, s$	Hash value of Implicit Certificate and signature
$d_{CA}, d_U, d_V$	Private key of CA, Node U and V
LT	Lifetime of certificate
$U, V, ID_{CA}$	Identity of Node U, V and CA
$Q_{CA}, Q_U, Q_V$	Public key of CA, Node U and V
$K_{UV}$	Shared Secret key between Node U and V
$n_U, n_V$	A random positive integer generated by Node U & V
$H, H_K, E_K$	Hash, keyed Hash and Encryption with $K^{th}$ key
$Key(N), KSn$	Symmetric keys used for encryption and decryption

1. *Mutual authentication and session key establishment*: In IoT networks, each node should perform the mutual authentication and verify the genuineness of the requesting node. After performing the mutual authentication, both the nodes should establish a session key to secure the further communication.
2. *Message integrity and freshness*: Message integrity ensures that no alteration has taken place during transit of messages. The received data should be fresh to avoid misinterpretation due to replaying of old messages.
3. *Lightweightness*: The devices in IoT networks are resource constrained, so overhead must be reduced during authentication and key establishment phase.
4. *Safeguard to popular attacks*: The proposed scheme must be resistant to popular attacks like impersonation, replay, node compromise, man-in-the-middle attack.

### 4.3 Proposed Scheme

Assume a distributed smart environment, for instance a smart home (also known as a home area network (HAN)), which consists of several WSN-IoT nodes. These nodes

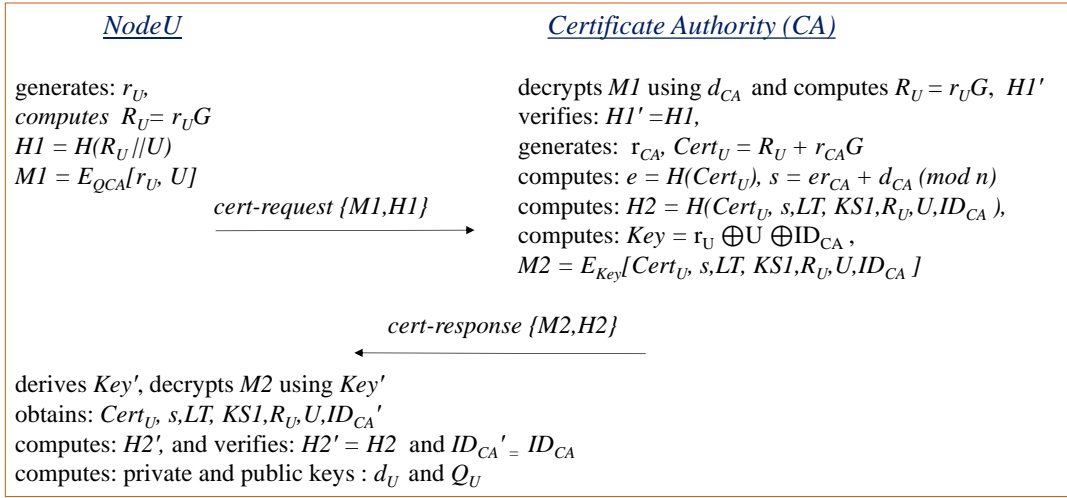


FIGURE 4.3: Generation of Implicit Certificate.

collect data within a smart home and forward it to the IoT cloud and to the user. In order to provide security in such application, this section proposes a robust and lightweight authentication scheme. Note that in order to run the proposed scheme (i) all the entities are assumed to have identical cryptographic systems including encryption and hashing algorithms, (ii) each certificate has its lifetime, e.g., a year. The proposed scheme consists of three phases: system set-up phase, registration phase, and authentication and key exchange phase.

### 4.3.1 System Set-up phase

In this phase, the CA off-line initializes the cryptographic mechanisms (such as, EC,  $n$ , point generator, hash function, symmetric encryption algorithm). Table 4.1 shows the notations and descriptions. Note that the background on ECC is omitted intentionally due to the space limit. However, the interested may refer to [54] for ECC details. The CA generates own public key ( $Q_{CA}$ ) and private key ( $d_{CA}$ ). In addition, it generates a key pool of secret keys (e.g.,  $KS1, KS2, \dots, KS_n$ ) for the HANs ( $HAN1, HAN2, \dots, HAN_n$ ). It then publishes EC,  $n$ , point generator,  $Q_{CA}$ .

### 4.3.2 IoT-node registration phase

In each home area network ( $HAN_i$ ), an IoT node (e.g., node U) needs to be registered to the CA and obtains security credentials including a certificate and a key. The flow of registration phase is depicted in Fig. 4.3 and illustrated as follows:

1. Initially, the node U generates a random number  $r_U$  and computes  $R_U = r_U G$ . It then computes  $H1 = H(R_U || U)$  and  $M1 = E_{Q_{CA}}[r_U, U]$ . Finally, the node U sends a *cert-request* message  $\{M1, H1\}$  to the CA.
2. Upon receiving *cert-request*, the CA decrypts  $M1$  using  $d_{CA}$  and obtains  $r_U, U$ , and computes  $R_U = r_U G$  and  $H1'$  and verifies  $H1' == H1$ . It then generates a random number  $r_{CA}$  and implicit certificate  $Cert_U = R_U + r_{CA} G$ , computes  $e = H(Cert_U)$ ,  $s = er_{CA} + d_{CA} \pmod{n}$ ,  $H2 = H(Cert_U, s, LT, KS1, R_U, U, ID_{CA})$ ,  $Key = (r_U \oplus U \oplus ID_{CA})$  and  $M2 = E_{Key}[Cert_U, s, LT, KS1, R_U, U, ID_{CA}]$ . Here,  $LT$  is the certificate lifetime of node U. Finally, the CA sends *cert-response* message  $\{M2, H2\}$  to the node U.
3. The Node U derives  $Key' = (r_U \oplus U \oplus ID_{CA})$  decrypts  $M2$  using  $Key'$  and obtains  $Cert_U, s, LT, KS1, R_U, U, ID_{CA}$  and stores them. Now it computes  $H2'$  and verifies  $H2' == H2$ . Upon successful verification, the node U computes own public and privacy keys from the received implicit certificate, as follows:

$$\begin{aligned}
d_U &= er_U + s \pmod{n} \\
Q_U &= d_U G \\
&= (er_U + s \pmod{n})G \\
&= (er_U + er_{CA} \pmod{n} + d_{CA} \pmod{n} \pmod{n})G \\
&= (er_U + er_{CA} \pmod{n} + d_{CA} \pmod{n})G \\
&= e(r_U + r_{CA})G + d_{CA}G \\
&= e(r_U G + r_{CA} G) + Q_{CA} \\
&= e(R_U + r_{CA} G) + Q_{CA} \\
Q_U &= eCert_U + Q_{CA}
\end{aligned}$$

### 4.3.3 Mutual authentication and key exchange phase

The flow of mutual authentication and pair-wise key establishment is shown in Fig. 4.4. This phase invokes when two nodes (node U and node V) want to negotiate a secret key within a HAN.

1. In the proposed scheme, the node U initiates the communication and it generates a random number  $n_U$ ,  $Key1 = (n_U \oplus U \oplus ID_{CA})$  and  $Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$ . Here,  $H_{Key1}$  is a keyed-hash. Now, it computes  $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$ , and sends *key-request*  $\{Token1, Z1\}$  to the node V.
2. Upon receiving *key-request* message, the node V decrypts  $Z1$  using  $KS1$ , obtains  $Cert_U, LT, n_U, U, ID_{CA}$ . It first verifies the lifetime  $LT$  of the certificate and  $ID_{CA}$  of the CA. If these conditions are true then it goes to the next step. In order to verify the authenticity of node U, now the node V derives  $Key1'$  and computes

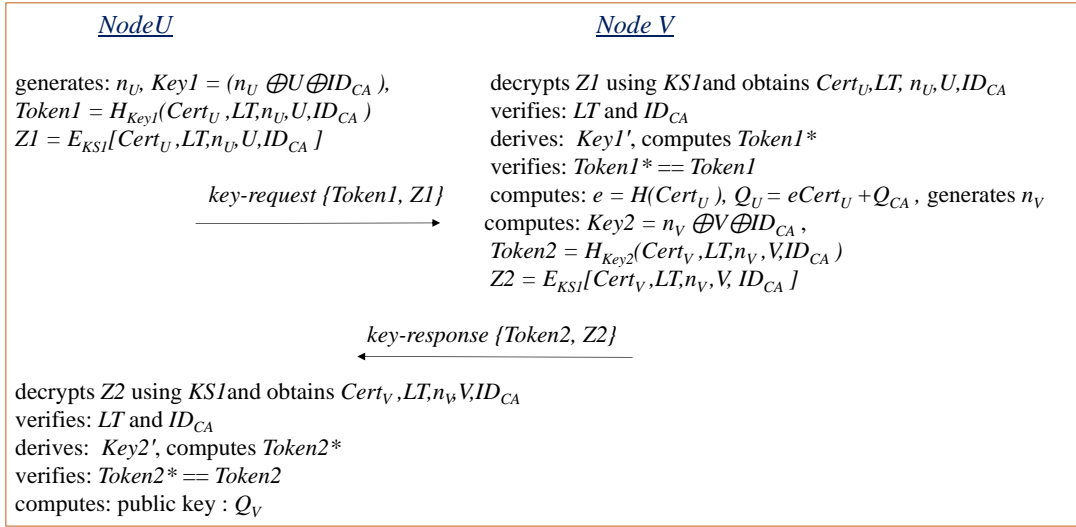


FIGURE 4.4: Mutual Authentication and pair wise key establishment.

$Token1^*$  and verifies  $Token1^* == Token1$ . If this condition fails then it aborts the session. Otherwise, the node V computes node U's public key as follows:  $e = H(Cert_U)$  and  $Q_U = eCert_U + Q_{CA}$ . The proof is as same as shown in the IoT node registration phase (refer to step 3).

3. Now, the node V generates  $n_V, Key2 = (n_V \oplus V \oplus ID_{CA})$  and  $Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$ . Here,  $H_{Key2}$  is a keyed-hash. It computes  $Z2 = E_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$ . It sends *key-response* message  $\{Token2, Z2\}$  to the node U. Finally, the node V computes a pair-wise key ( $K_{UV} = d_V Q_U$ ) using own private key  $d_V$  and U's public key  $Q_U$ .
4. Upon receiving *key-response* message, the node U decrypts Z2 using KS1, obtains  $Cert_V, LT, n_V, V, ID_{CA}$ . It first verifies the lifetime  $LT$  of the certificate and  $ID_{CA}$  of the CA. If these conditions are true then it goes to the next step. In order to verify the authenticity of node V, now the node U derives  $Key2'$  and computes  $Token2^*$  and verifies  $Token2^* == Token2$ . If fails then it aborts the session. Otherwise, the node U computes node V's public key ( $Q_V$ ) and pair-wise key ( $K_{UV}$ ) as follows:

$$\begin{aligned}
 e &= H(Cert_V) \text{ and } Q_V = d_V G \\
 &= (er_V + s(\text{mod } n))G \\
 &= (er_V + er_{CA}(\text{mod } n) + d_{CA}(\text{mod } n)(\text{mod } n))G \\
 &= (er_V + er_{CA}(\text{mod } n) + d_{CA}(\text{mod } n))G \\
 &= er_V + r_{CA}(\text{mod } n)G + d_{CA}G \\
 &= e(r_V G + r_{CA}G) + Q_{CA} \\
 &= e(r_V + r_{CA})G + Q_{CA} \\
 Q_V &= e(Cert_V) + Q_{CA}
 \end{aligned}$$

$$\mathbb{K}_{UV} = d_U Q_V$$

Alternatively,  $\mathbb{K}_{UV} = d_U d_V G$

The pair-wise key is established successfully.

## 4.4 Security & Comparative Analysis

### 4.4.1 Formal analysis

Following [14], [35], we utilize AVISPA (automatic verification of internet protocols and applications) tool to evaluate the security strength of the proposed RLMA against the Dolev-Yao attack model. The AVISPA tool uses High Level Protocol Specification Language (HLPSL). The HLPSL script is further translated to Intermediate format (IF) using a HLPSL2IF translator [55]. The IF is feed to the backend, e.g., on-the fly model-checker (OFMC). For more details on the backends, the reader may refer to [55]. Finally the backend generates the Output file (OF) concluding the protocol as safe or unsafe.

The HLPSL script of a protocol always begins with the basic roles. These roles are played by agents and contain local declarations. It defines the transitions when certain events are met and the corresponding changes in the states of the node. On the other hand, composition role have no transition section and executes various sessions in parallel. The last role i.e. environment role is very significant as it declares global constants and may composed of one or more sessions. The knowledge of the intruder ( $i$ ) is also declared in this role and he may play some roles to camouflage profile of legitimate users. The channel ( $dy$ ) uses the Dolev-Yao (DY) attack model for communication between the nodes.

To assess the strength of RLMA, the mutual authentication and pairwise key establishment phase is scripted in HLPSL and tested on AVISPA. Initially, basic roles of node U and V are defined which comprise of agent details ( $U, V$ ), crypto-operations, local declarations ( $Key1$  etc.), channel ( $dy$ ), initial state and transitions. Due to HLPSL keyword reservations, some of the parameters are represented with different acronyms in AVISPA as compared to acronyms used in algorithm. Those acronyms are  $i$  (intruder),  $ID_U$  (Identity of Node U), and  $ID_V$  (Identity of Node V). Node U acts as an initiator. After initialization at  $State = 0$  [RCV( $start$ )], it transitions to  $State = 1$ , where fresh nonce is prepared,  $N'_u := new()$  followed by generation of  $Token1' = \{Hash(Cert_u.Lt.Nu.Id_u.Idca)\}$ , and  $Z1' = \{Cert_u.Lt.Nu.Id_u.Idca\}_{Ks1}$ . Node U sends  $Token1' \& Z1'$ , ( $SND(Token1', Z1')$ ) to Node V for accomplishing mutual authentication and pair wise key establishment considering same channel ( $dy$ ) properties. The goal predicates set by Node U is privacy of  $Cert_u \& Nu'$  as shown in Fig. 4.5.

```

role nodeU (U,V: agent,
            Hash: hash_func,
            Qca: public_key,
            Key1,Key2,Ks1: symmetric_key,
            SND, RCV: channel (dy))
played_by U def=
local
State                                     :nat,
Idu,Certu,Lt,Idca,Nv,Certv,Nu,Idv       :text,
Token1,Token2,Z1,Z2                     :message

init State:= 0
transition
1. State = 0  /\ RCV(start)  =|>
   State' := 1  /\ Nu' := new()
                /\ Key1' := xor(Nu,xor(Idu,Idca))
                /\ Token1' := Hash(Certu.Lt.Nu.Idu.Idca)
                /\ Z1' := {Certu.Lt.Nu.Idu.Idca}_Ks1
                /\ SND(Token1',Z1')
                /\ secret ({Certu,Nu'},sub1,{U,V})

2. State = 2  /\ RCV(Token2',Z2') =|>
   State' := 3  /\ Key2' := xor(Nv,xor(Idv,Idca))
                /\ Token2' := Hash(Certv.Lt.Nv.Idv.Idca)
                /\ Z2' := {Certv.Lt.Nv.Idv.Idca}_Key2'
                /\ witness(U,V,nodeV_nodeU_lt,Lt)

end role

```

FIGURE 4.5: Specification of the Node U role.

Node V receives the  $Token1'$  and  $Z1'$  in its initial state,  $State = 1$  [RCV( $Token1'$ ,  $Z1'$ )] and extracts information during  $2^{nd}$  State. Similarly, Node V sends  $Token2'$ ,  $Z2'$  to Node U for successful accomplishment of mutual authentication and key establishment as shown in Fig. 4.6. The message confidentiality of  $Z2'$  and authentication of  $Token2'$  is modelled in terms of goals predicate, secrecy  $\{Certv, Nv\}$  and witness  $\{nodeU\_nodeV\_lt\}$  respectively. Witness ensures that the lifetime ( $LT$ ) of the certificate ( $Certu$ ) is verified before use.

Likewise, Node U recovers the information from the received message [RCV( $Token2'$ ,  $Z2'$ )]. Further, Node U at  $State = 3$ , verifies (witness( $U, V, nodeV\_nodeU\_lt, Lt$ )) the validity of  $Certv$  before processing the request of pairwise key establishment.

Fig. 4.7 shows the composition of arguments used by agents,  $nodeU(U, V, Hash, Qca, Key1, Key2, Ks1, SU, RU) / \setminus nodeV(U, V, Hash, Qca, Key1, Key2, Ks1, SV, RV)$ . These arguments are either sent or used by agents during the session. The most important is environment role because it constitutes of global constants declarations, defines intruder knowledge, elucidate composition of sessions and set up goals of interest. As per the Dolev-Yao attack model, intruder is able to eavesdrop, intercept and analyze the information for e.g.,  $nodeU, nodeV, h, key1i, key2i, ks1i, qca$ . The intruder knowledge is specified in environment and is used by security protocol analyzer tool (OFMC, CL-AtSe) during vulnerability evaluation of protocol against attacks. The next part



```

role nodeV (U,V: agent,
            Hash: hash_func,
            Qca: public_key,
            Key1,Key2,Ks1: symmetric_key,
            SND, RCV: channel (dy))
played_by V def=
local
State                                     :nat,
Idu,Certu,Lt,Idca,Nv,Nu,Certv,Idv,E:text,
Token1,Token2,Z1,Z2                       :message

init State:= 1
transition
1. State = 1 /\ RCV(Token1',Z1') =|>
   State' := 2 /\ Z1' := {Certu.Lt.Nu.Idu.Idca}_Ks1
                /\ Key1' := xor(Nu,xor(Idu,Idca))
                /\ Token1' := Hash(Certu.Lt.Nu.Idu.Idca)
                /\ Key2' := xor(Nv,xor(Idv,Idca))
                /\ Token2' := Hash(Certv.Lt.Nv.Idv.Idca)
                /\ Z2' := {Certv.Lt.Nv.Idv.Idca}_Key2'
                /\ SND (Token2',Z2')
                /\ secret ({Certv,Nv},sub2,{U,V})
                /\ witness(V,U,nodeU_nodeV_lt,Lt)

end role

```

FIGURE 4.6: Specification of the Node V role.

```

role session (U,V: agent,
             Hash: hash_func,
             Qca: public_key,
             Key1,Key2,Ks1: symmetric_key)
def=
local SU,RU,SV,RV: channel(dy)
composition
  nodeU(U,V,Hash,Qca,Key1,Key2,Ks1,SU,RU)
  /\ nodeV(U,V,Hash,Qca,Key1,Key2,Ks1,SV,RV)
end role

```

FIGURE 4.7: Specification of the Session role.

of the environment role specifies the various sessions of message exchanges among nodes. Though it is expected to have sessions amongst legitimate agents only ( $nodeU, nodeV, h, qca, key1, key2, ks1$ ), but the possibility of intruder intervening in the session of legitimate nodes also prevails ( $nodeU, i, h, qca, key1i, key2i, ks1i$ ), ( $i, nodeV, h, qca, key1i, key2i, ks1i$ ). A total of four goals are specified out of which two are associated to secrecy and rest two corresponds to authentication as shown in Fig. 4.8. The description of the goals are:

- Secrecy\_of sub1 represents that  $\{Cert_U, N_U\}$  are kept secret between node U and node V.
- Secrecy\_of sub2 represents that  $\{Cert_V, N_V\}$  are kept secret between node V and node U.

- Authentication on nodeU\_nodeV\_lt states that the lifetime (i.e., *LT*) of certificate  $\{Cert_U\}$  will be verified at the Node V.
- Authentication on nodeV\_nodeU\_lt states that the lifetime (i.e., *LT*) of certificate  $\{Cert_V\}$  will be verified at the Node U.

The robustness of proposed protocol against attacks is verified using OFMC backend. Fig. 4.9 illustrates that RLMA can withstand against severe attacks and is reported safe to use in Internet based applications. Likewise OFMC, the CL-AtSe backend also reported safe. Hence, the attacks considered in the DY attack model cannot harm the RLMA security protocol.

```

role environment ()
def=
const nodeU,nodeV: agent,
qca: public_key,
key1,key2,ks1,key1i,key2i,ks1i: symmetric_key,
idu,certu,lt,idca,e,nv,nu,certv,idv: text,
h: hash_func,
nodeU_nodeV_lt,nodeV_nodeU_lt,sub1,sub2: protocol_id

intruder_knowledge={nodeU,nodeV,h,key1i,key2i,ks1i,qca}

composition
session(nodeU,nodeV,h,qca,key1,key2,ks1)
/\session(nodeU,i,h,qca,key1i,key2i,ks1i)
/\session(i,nodeV,h,qca,key1i,key2i,ks1i)
end role

goal
secrecy_of sub1
secrecy_of sub2
authentication_on nodeU_nodeV_lt
authentication_on nodeV_nodeU_lt
end goal

environment ()

```

FIGURE 4.8: Specification of the goal and environment for the proposed RLMA.

<pre> % OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/IoT-HAN.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.04s visitedNodes: 19 nodes depth: 4 plies </pre>	<pre> % ATSE % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/IoT-HAN.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 0 states Reachable : 0 states Translation: 0.02 seconds Computation: 0.00 seconds </pre>
---	---

FIGURE 4.9: RLMA results using OFMC and CL-AtSe backend.

### 4.4.2 Informal Proof

Following the attack model (as shown in section 4.2.2), this section deals with the understanding of how the designed protocol withstand against the attacks, such as modification of messages, known key attack, impersonation attack, replay, node compromise attack, etc.

**Proposition 1:** Secure against message modification.

**Proof.** Consider a communication between the node U and V, where an attacker intercepts *key – request*  $\{Token1, Z1\}$  and tries to fabricates  $Z1$  to  $Z1'$  using own key. Then it sends  $Token1, Z1'$  to the node V. Since,  $Z1'$  is computed via a wrong key (i.e., adversary key), it cannot be decrypted at the legal node V. In addition, as  $Z1'$  cannot be decrypted, resultant  $Token1$  cannot be verified. Note that here  $Token1$  is a keyed-hash ( $H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$ ) and the key is only generated by the legitimate nodes using  $n_U, U, ID_{CA}$ . Hence, message modification cannot work from the node U to V communication. Likewise, the *key – response* message is secure from the node V to U communication.

**Proposition 2:** RLMA is safe against impersonation attack.

**Proof.** Impersonation attacks can be prevented by properly authenticating the nodes e.g., Node U, computes the public key of Node V ( $Q_V$ ), using  $Cert_V$  and public key of CA,  $Q_{CA}$ . Likewise, the node V computes the  $Q_U$ . Both interested entities use their private keys ( $d_U, d_V$ ) and opposite entity public Key ( $Q_U, Q_V$ ) for generating the session key  $\{Node\ U, K_{UV} = d_U Q_V; Node\ V, K_{UV} = d_V Q_U\}$ . The process guarantees the execution of same secret keys at both entities when the certificates are issued by the valid CA. Node U, can trust the received certificate if it is encrypted by secret key ( $Key = r_U \oplus U \oplus ID_{CA}$ ), which is exchanged between CA and node U. Therefore, impersonation attacks are difficult to conduct in RLMA as nodes among themselves use keyed-hash based Token approach for mutual authentication.

**Proposition 3:** RLMA is resistant to node compromise attack.

**Proof.** It is widely accepted that smart devices are difficult to prevent if they are not tamper proof [14]. Assume if the attacker captures the node and tries to collect the information. The information may constitute of a Certificate ( $Cert_N$ ). As, each certificate has its lifetime and unique nonce, the misuse of a compromised node can be prevented. In a smart home (i.e., HAN), as every node is embedded with unique id, certificate and KS1, thus compromising these parameter cannot compromise the security of non-compromised HANs. Therefore, the proposed scheme addresses security against node compromise attack to some level.

**Proposition 4:** Secure against Known key attack.

**Proof.** Known-key attack means that if a session is compromised then it should not compromise other session keys. In our scheme, suppose an attacker tries to generate a pair-wise session key ( $K_{UV}$ ). However, this key does not help to deduce the key of other sessions since the pair-wise key is being computed over a nonce ( $N$ ) and a high entropy random number ( $r$ ). Note that these parameters are independent and different for each session. More precisely, a fresh random number guarantees that the certificate is unique for each node  $\{(R_U = r_U G), (Cert_U = R_U + r_{CA} G)\}$  which further certify that generation of session key,  $\{K_{UV} = d_U(e(Cert_V + Q_{CA}))\}$  is independent and distinct for every session, thereby protecting the protocol against known key and ephemeral secret leakage attacks.

**Proposition 5:** Resilient to MITM attack.

**Proof.** The attacker node may have eavesdropped the messages exchanged between nodes and CA or between nodes. The attacker may have intentions to disrupt the system by retrieving the information as a middle agent and relay it after modifying. The attacker needs  $d_{CA}$  and  $E_{Key}$  to compute  $\{E_{Key}, [Cert_U, s, LT, R_U, U]\}$ , which he would never be able to get as  $d_{CA}$  is the private key of CA and never shared over the medium, thus attacker would not be able to modify the authenticator messages of RLMA. Moreover, the legitimate devices are mutually authenticated,  $K_{UV} = d_V d_U G$  with the secret key ( $K_{UV}$ , never shared over medium), hence it would not be possible for an attacker to launch MITM.

**Proposition 6:** RLMA is resistant to Denial-of-service (DoS) and to replay attack.

**Proof.** Protecting a network from denial-of-service attacks is very hard as it can be mounted at every layer in a smart environment. However, a replay attack is one of them that can degrade the smart environment performance severely [56]. For instance, in the proposed scheme – suppose an adversary (A) eavesdrops and intercepts the valid messages  $\{Token1, Z1\}$  and  $\{Token2, Z2\}$  between the node U and V. Later adversary tries to replay  $\{Token1, Z1\}$  to node V to keep the node V busy. However, this attempt fails as  $Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$  and  $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$  contains a fresh nonce ( $n_U$ ), which is utilized to protect against replay attack. Similarly, the attacker intercepts the valid message  $\{Token2, Z2\}$  and later replay's to node U. This attempt fails as  $Token2 = H_{Key2}(Cert_V, LT, n_V, V, ID_{CA})$  and  $Z2 = E_{KS1}[Cert_V, LT, n_V, V, ID_{CA}]$  utilize nonce ( $n_V$ ). Hence, a replay attempt is detected very easily at node U. Moreover, nonce cannot be modified as it is shielded with  $KS1$  and keyed-hash with SHA-1. Therefore, the proposed scheme is safeguard to a replay attack, and to a DoS attack to some extent (i.e., a partial protection against DoS).

**Proposition 7:** RLMA attained Mutual Authentication.

**Proof.** The main purpose of mutual authentication is to cease the unauthorized access of intruders into the network. In our approach, mutual authentication is carried out between two nodes as follows:

- $U \rightarrow V : \text{Token1}, Z1;$   
 $Z1 = E_{KS1}[\text{Cert}_U, LT, n_U, U, ID_{CA}]$   
 $\text{Key1} = (n_U \oplus U \oplus ID_{CA})$   
 $\text{Token1} = H_{\text{Key1}}(\text{Cert}_U, LT, n_U, U, ID_{CA})$
- $U \leftarrow V : \text{Token2}, Z2;$   
 $Z2 = E_{KS1}[\text{Cert}_V, LT, n_V, V, ID_{CA}]$   
 $\text{Key2} = (n_V \oplus V \oplus ID_{CA})$   
 $\text{Token2} = H_{\text{Key2}}(\text{Cert}_V, LT, n_V, V, ID_{CA})$

Upon receiving *key – request* from node U, the node V decrypts  $Z1'$  and computes  $\text{Key1}'$ ,  $\text{Token1}^*$  and verifies  $\text{Token1}^* == \text{Token1}$ , for mutual authentication. Successful verification clearly indicate the legitimacy of node U. Similarly, node U verifies the authenticity of node V by evaluating  $\text{Token2}$ . As keyed-hash is a one way function, so  $\text{Token2}$  cannot be reversed. Therefore, unauthorized nodes can never read the content of  $H_{\text{Key2}}(\text{Cert}_V, LT, n_V, V, ID_{CA})$ .

**Proposition 8:** Message freshness.

**Proof.** The proposed protocol ensures the presence of freshness component in messages through nonces ( $N$ ) and ephemeral random numbers ( $r$ ). The freshness not only protects against the replay and DoS attacks but also restricts the entities to prevent wastage of the resources in processing the old requests e.g., one or more components ( $n_U, n_V, r_U, R_U$ ) of the freshness is added in every single exchange of message, e.g.,  $U \leftrightarrow CA, U \leftrightarrow V$ ,

- $\{E_{Q_{CA}}[r_U, U]\} || H(R_U || U) (\because R_U = r_U G)$
- $Z1 = E_{KS1}[\text{Cert}_U, LT, n_U, U, ID_{CA}]$
- $\text{Token2} = H_{\text{Key2}}(\text{Cert}_V, LT, n_V, V, ID_{CA})$

The expressions prove the attainment of freshness property.

**Proposition 9:** Secure session key agreement.

**Proof.** The key agreement can be observed in the expression:

- Node U:  $K_{UV} = d_U Q_V; Q_V = e \text{Cert}_V + Q_{CA};$   
 $Z2 = E_{KS1}[\text{Cert}_V, LT, n_V, V, ID_{CA}]$   
 $\text{Cert}_V = R_V + r_{CA} G$
- Node V:  $K_{UV} = d_V Q_U; Q_U = e \text{Cert}_U + Q_{CA};$   
 $Z1 = E_{KS1}[\text{Cert}_U, LT, n_U, U, ID_{CA}]$   
 $\text{Cert}_U = R_U + r_{CA} G$

where  $LT$  ensures the expiry of the certificate and in turn session after a certain time period. Hence, new key will be formed for each session. Moreover it can be observed that the certificates are not sent in plain text, thereby obtaining security of the parameters used for key establishment. In addition  $R_U$  and  $R_V$ , will be different for each session which guarantees a different  $K_{UV}$  for every session. In this way, a secure session key agreement is provided between the node  $U$  and  $V$ .

**Proposition 10:** RLMA procured the property of anonymity and/or untraceability.

**Proof.** Untraceability can be achieved by keeping the identity of the device hidden [57]. Attacker usually tries to track the device by eavesdropping of messages. In the RLMA, the ID's of the devices are not sent in plaintext, thereby it will be hard to trace the communicating parties, e.g.,  $Z1 = E_{KS1}[Cert_U, LT, n_U, U, ID_{CA}]$ , and  $Token1 = H_{Key1}(Cert_U, LT, n_U, U, ID_{CA})$ , are shared by node  $U$  with node  $V$ . The information contains the ID's, which is encrypted to ensure that adversary could not find a way to decode the identities of the devices in communication. Therefore, the proposed protocol ensures the untraceability of entities.

Finally, we summarize the security features of RLMA and compare its security with the state-of-the-art schemes. Table 4.2 shows that protocol proposed in [19], is prone to node compromise attack. More precisely, a single node compromises may lead to several attacks to the whole network. Other protocols (e.g., [20, 21, 22, 23]) are subjected to known key attack. The schemes presented in [21] is vulnerable to impersonation attacks. In addition, it can be noticed that most of the state-of-the-art schemes do not consider the property of anonymity and/or untraceability, which is paramount requirement in many of smart environments use-cases where privacy is equally important, such as smart healthcare monitoring. In summary, it can be observed from Table 4.2 that the proposed scheme can provide more security features than the existing schemes.

## 4.5 Performance Analysis

### 4.5.1 Experimental setting

We have experimented a prototype of RLMA scheme on a TelosB mote/device powered by TinyOS. Here, a TelosB mote equipped with a 16 bit processor (i.e., Texas Instruments MSP430 processor) that runs at a clock frequency of 8 MHz having 48 KB and 10 KB of ROM and RAM respectively [58]. We built a network of two TelosB nodes, i.e., node  $U$  and node  $V$  and a laptop (*Configuration:* Intel core i3-2310M processor with clock frequency and RAM of 2.10 GHz and 4 GB respectively). For the experimental purpose, we utilized a rich set of cryptographic libraries including AES (Advanced Encryption Standard), one-way hash function (i.e., SHA-1) and TinyECC [59]. In our

TABLE 4.2: Analysis and Comparison of Protocols based on protection against attacks and security properties

$\mathcal{A} \ \& \ \mathcal{SF}$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
$\mathcal{A}_1$		✓		✓	✓	✓
$\mathcal{A}_2$	✓	✓		✓	✓	✓
$\mathcal{A}_3$		✓	✓	✓	✓	✓
$\mathcal{A}_4$						✓
$\mathcal{A}_5$	✓	✓	✓		✓	✓
$\mathcal{A}_6$			$\mathcal{P}$		$\mathcal{P}$	$\mathcal{P}$
$\mathcal{A}_7$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_1$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_2$	✓	✓	✓	✓		✓
$\mathcal{SF}_3$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_4$		✓				✓
$\mathcal{SF}_5$			✓			✓

Acronyms:  $\mathcal{A}$ : Attacks,  $\mathcal{SF}$ : Security features,  $\mathcal{A}_1$ : Modification of messages,  $\mathcal{A}_2$ : Impersonation,  $\mathcal{A}_3$ : Node compromise,  $\mathcal{A}_4$ : Known key,  $\mathcal{A}_5$ : MITMA,  $\mathcal{A}_6$ : Denial of service,  $\mathcal{A}_7$ : Replay,  $\mathcal{SF}_1$ : Mutual authentication,  $\mathcal{SF}_2$ : Message freshness,  $\mathcal{SF}_3$ : Session key agreement,  $\mathcal{SF}_4$ : Anonymity and/or Un-traceability,  $\mathcal{SF}_5$ : Confidentiality,  $\mathcal{P}$ : Partially protected,  $S_1$ : [19],  $S_2$ : [20],  $S_3$ : [21],  $S_4$ : [22],  $S_5$ : [23],  $S_6$ : RLMA

experiment, we use the following message sizes, for instance IDs = 1 byte, hashing = 20 bytes, pseudo random number = 4 bytes, lifetime = 4 bytes, certificate = 16 bytes,  $s$  = 20 bytes, nonce = 4 bytes, and symmetric key size = 16 bytes. Therefore, the total length of messages in RLMA, i.e., *key-request* and *key-response* are 46 bytes each.

## 4.5.2 Evaluation of RLMA

We evaluated the performance of RLMA considering computation, communication and energy prices for the authentication and key establishment phase.

### 4.5.2.1 Computational costs

As shown in Fig. 4.4, the Node U initiates the communication and sends a *Key-request* packet to the node V, which is further connected to the server, i.e., the laptop. Moreover, the node U receives (i.e., a *Key-response*) from the node V. Nevertheless, the total execution time taken by Node U is 1177.33 ms, for performing mutual authentication and key establishment with Node V, as shown in Fig. 4.10. We further evaluate the execution time for individual cryptographic operations. As shown in Table 4.3, SHA-1, AES-encryption, AES-decryption, and multiplication take 112.32 ms, 16.38 ms, 178.10 ms, and 870.53 ms, respectively. This computation time can be reduced by using more high

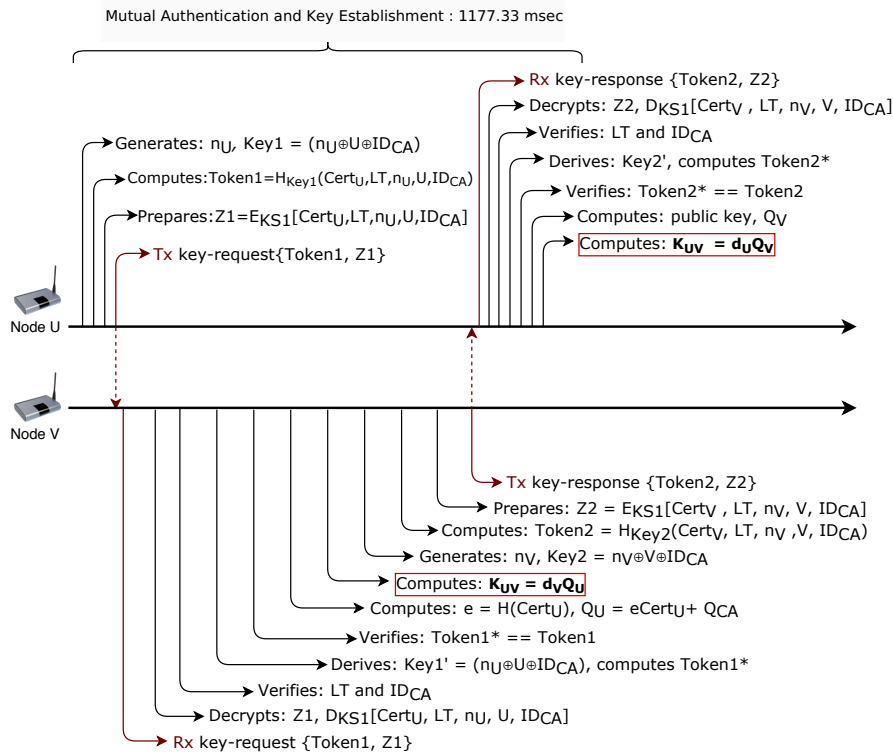


FIGURE 4.10: Time Elapsed for Mutual Authentication and Key Establishment.

TABLE 4.3: Execution time and Energy costs

	$O_1$	$O_2$	$O_3$	$O_4$	$O_T$
Execution Time (ms)	112.32	16.38	178.10	870.53	1177.33
Energy Costs (mJ)	0.606	0.088	0.961	4.701	6.356

Acronyms:  $O_1$ : hash,  $O_2$ : encryption,  $O_3$ : decryption,  $O_4$ : multiplication,  $O_T$ :  $O_1 + O_2 + O_3 + O_4$

class smart devices, e.g., raspberry pi. However, in terms of the key establishment time, it is a well-suited time for the resource-constrained devices in smart environments.

We further evaluated energy-efficiency for the cryptographic operations as the smart objects are battery powered devices in many of use-cases. Following the formula (i.e.,  $\{E = V \times I \times t\}$ ) used in [21], we calculated the energy prices for our cryptographic operations. Here,  $V$ ,  $I$  and  $t$  are the voltage, current and execution time, respectively. We have adopted the values of  $V = 3V$  and  $I = 1.8mA$  from [60]. The value of ' $t$ ' for RLMA is measured from the experiment, as shown in Table 4.3. On a battery-powered smart device, the total energy required for the proposed RLMA is 6.356 mJ. More precisely, Table 4.3 also demonstrates the total energy incurred by RLMA for executing individual cryptographic operations, e.g., hash, encryption, decryption, and multiplication are 0.606 mJ, 0.088 mJ, 0.961 mJ, and 4.701 mJ, respectively.



TABLE 4.4: Computation Cost Comparisons

$T_{op}$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
$O_1$	2H	8H	1H	1H	1H	2H
$O_2$	-	-	2MAC	6MAC	1MAC	-
$O_3$	2HMAC	-	-	-	-	-
$O_4$	-	-	-	-	1E + 2D	1E + 1D
$O_5$	-	-	-	2S	-	-
$O_6$	2M	-	-	1M	2M	1M

Acronyms:  $T_{op}$ : type of operation,  $O_1$ : hash (H),  $O_2$ : message authentication code (MAC),  $O_3$ : hash based message authentication code (HMAC),  $O_4$ : encryption (E) /decryption (D),  $O_5$ : signatures (S),  $O_6$ : multiplication (M),  $S_1$ : [19],  $S_2$ : [20],  $S_3$ : [21],  $S_4$ : [22],  $S_5$ : [23],  $S_6$ : RLMA

In addition, a comparison of the computation cost among state-of-the-art schemes is presented in Table 4.4. Note that we simply chose asymmetric key based schemes for the comparison purposes. For the convenience of evaluation, following notations are being used:

- H: the time for performing a hash operation.
- MAC: the time for performing a MAC operation.
- HMAC: the time for performing a HMAC operation.
- E: the time for performing an encryption operation.
- D: the time for performing an decryption operation.
- S: the time for performing a signature operation.
- M: the time for performing a multiplication operation.

It can be seen from Table 4.4, the proposed RLMA makes use of 2 hash operations, 1 time encryption (E) & decryption (D), and 1 time multiplication operation for executing the mutual authentication and key establishment between the node U & V. Whereas the schemes proposed in [19, 20, 22, 23] makes use of excessive hash, MAC, HMAC, signatures, encryption, decryption and multiplication operations, which may not be efficient for the resource-hungry nodes. In addition, the scheme proposed in [21] incurred less computations than the proposed RLMA but does not provide adequate security services as shown in Table 4.2.

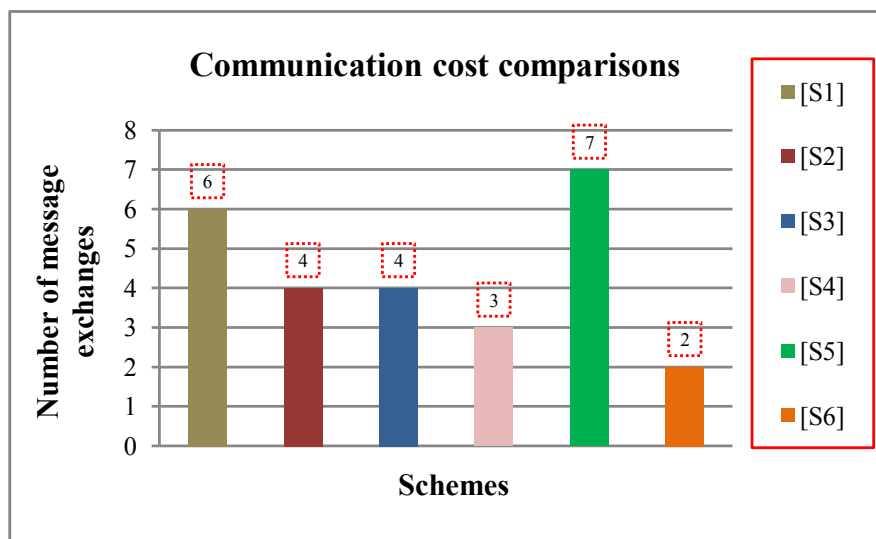


FIGURE 4.11: Communication cost comparisons in terms of the number of message exchanges.

TABLE 4.5: Communication Energy Costs

Cost		$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
Node $U$	$S$	624	1008	-	768	-	368
	$R$	624	368	-		-	368
Node $V$	$T$	1248	1376	-	768	-	736
$E.C. (mJ)$		0.898	1.023	-	0.587	-	0.562

Acronyms: Symbol (-): unspecified,  $S$ : bits sent by node  $U$  to node  $V$ ,  $R$ : bits received by node  $U$  from node  $V$ ,  $T$ : total exchange of bits,  $E.C.$ : energy consumed,  $S_1$ : [19],  $S_2$ : [20],  $S_3$ : [21],  $S_4$ : [22],  $S_5$ : [23],  $S_6$ : RLMA

#### 4.5.2.2 Communication cost

To investigate the communication cost, we have evaluated the energy required to transmit/receive the *key-request* and *key-response* messages between the node  $U$  and  $V$ . Following the scheme proposed in [35], transmitting and receiving a bit on TelosB consumes  $0.72 \times 10^{-3}mJ$  and  $0.81 \times 10^{-3}mJ$  of energy, respectively. Therefore, to send a *key-request* (i.e., 368 bits) to node  $V$ , the node  $U$  requires  $0.264 mJ$ . Likewise, to receive a *key-response* (i.e., 368 bits) from node  $V$ , the node  $U$  needs  $0.298 mJ$  energy. The total energy required for communication by RLMA is  $0.562 mJ$  as shown in Table 4.5 and it can also be noticed that the proposed scheme incurred less communication energy than the other schemes.

Finally, from Fig. 4.11, it is easy to visualize that a practical authentication and key establishment in the proposed scheme ( $S_6$ ) requires 2 message exchanges, whereas the schemes proposed in [20]( $S_2$ ), [21]( $S_3$ ) require 4 message exchanges and the scheme

proposed in [19]( $S_1$ ), [22]( $S_4$ ), [23]( $S_5$ ) needs 6, 3, and 7 message exchanges respectively. It should be noted that in real-world applications the actual number of message exchanges may vary if the packet transmission required multi-hop communications.

Considering computational, communication and node energy costs, it is clear that the proposed RLMA is efficient compared to other related schemes.



## Chapter 5

# Robust and Lightweight Key Exchange (LKE) Protocol for Industry 4.0

### Chapter Overview

5.1	Introduction . . . . .	47
5.2	System Model, Adversary Model, and Security and other goals . . . . .	50
5.3	Proposed Scheme . . . . .	53
5.4	Security Analysis . . . . .	61
5.5	Performance and Comparative Analysis . . . . .	67

### 5.1 Introduction

Industry 4.0 has enabled the connectivity of physical devices such as robots to the internet. This revolution has brought extensive solutions to control the machines remotely along with retrieving of useful data from the remote locations [61]. The industrial revolution has been presented in Figure 5.1. The transformation of industrial processes took ages to reach an autonomous state. It began in the decade of 1780 when machines were first used for performing industrial tasks. The next progress almost took a century which advanced the manufacturing process with the involvement of *massive manpower* and *assembly lines* (driven by electricity) for enhanced production. The year of 1968 brought another milestone in the history of industrial transformation. This generation of Industry experienced automation in production processes through the *integration of electronics* and *computers* into the machines. The present generation of Industry (I4.0) is even more powerful than all the predecessors. The machines of this generation are too

smart; they can *sense*, *monitor*, and *measure* the physical quantities and seamlessly report to the connected devices. Industrial IoT enabled the administrators to monitor the processes in real time thus helping them to make instant decisions and analysis [62].

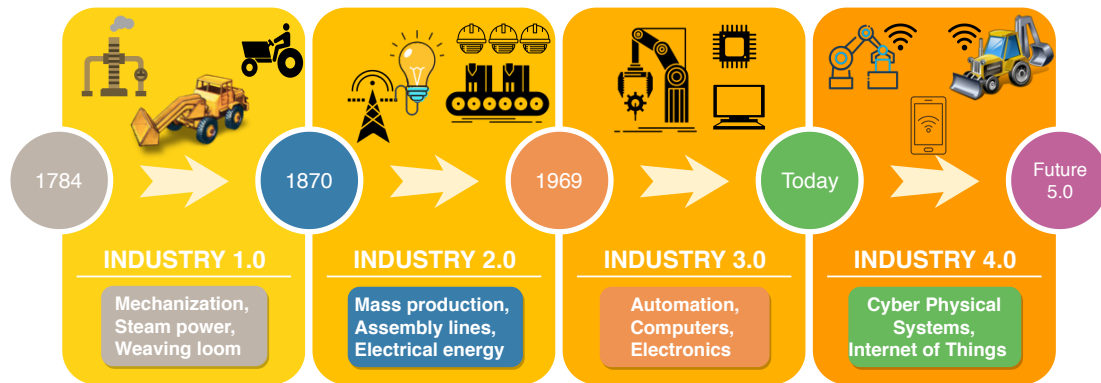


FIGURE 5.1: Industrial revolution.

*IIoT* evolution has not only transformed the manufacturing processes of industry rather has helped the logistics department to locate the movement of goods carriers, predict the arrival timings and help the carriers in finding the path with less traffic and better road conditions [63], [64]. The amalgam of physical and cyber technology in industries are leaving footprints of success [65]. *Bosch Inc.* is using *IIoT* for monitoring the lubricating valves and filters to reduce manual testing and maintenance costs [66]. *Volkswagen Group* manufactures the *lamborghini* in a smart factory where automatic guided vehicles carry the car components from one workstation to another. Apart from the movement, workers of the industry are able to see the progress, control and monitor the workstations processes remotely, thereby eliminating the need of physical presence of workers and managers in the industry [67]. Likewise, enormous features of *IIoT* has led the pharmacy and agriculture industry to incorporate *I4.0* processes for medicine testing [68] and prediction of disease in crops, respectively [2].

Indeed few industries are able to transform but still lot many are struggling. The biggest challenge in realization of Industry 4.0 is interoperability, compatibility, and reliability; Interoperability amongst plethora of machines from various manufacturers, compatibility of machines with existing infrastructure and their operational reliability is still a major concern. The data security is yet another, but extremely critical parameter in the accomplishment of *IIoT* [69]. The authors in [70] stressed on the uncontrollable nature of the Machine to Machine (M2M) communications and highlighted the threats to large volume of critical data in absence of robust security measures. Absentia of security measures in *IIoT* may pave the way towards cyberattacks. Cyberattacks can cause physical damage to infrastructure of industry and may endanger workers lives [71, 72, 73]. In one incident, the incompetence of key exchange and mutual authentication protocol enabled the attackers to get illegal access of the complete BACnet at Sochi Arena [74].

Another incidence was brought into limelight by forbes, where attackers used malicious codes and radio hardware to exploit the industrial network. Attackers got successful when they took over the control of construction cranes, excavators, scrapers and other large machinery from legitimate engineers of the industry. Another instance Zimperium Inc. reported recently about the electric bike made by Chinese company Xiaomi Inc. The electric bike was admitting requests like acceleration, braking, locking and unlocking from even illegitimate users and devices [75]. In addition, a survey paper concludes that IIoT networks may suffer from masquerade and disclosure attacks if the network is not enabled with proper authentication mechanisms [76].

Many potential attacks are conducted in the recent past, e.g., *Mirai* and *IoTroop* botnet where attackers exploited the vulnerabilities of the system i.e., *access control* procedures, *incompatibility* of security protocols due to *heterogeneous* nature of devices etc. [3]. These attacks in *Cyber-physical systems* (CPS) are majorly caused due to use of unsecured medium for signalling actuators [77]. The vulnerability in the IIoT can be more dangerous due to sensitive nature of data, for instance, a little loss of precision in chemical formation could produce a complete different medicine, thus posing disastrous health effects. The key vulnerabilities found behind the aforementioned incidents are inadequate and improper mutual authentication and key exchange procedures. Therefore, security analysts have advised to implement secure key exchange and strong mutual authentication procedures for ensuring security and privacy of the data [78], [79].

The entities involved in the key exchange and mutual authentication are usually heterogeneous and have different resource availability like gateways are resource-rich devices whereas smart IoT nodes are resource-deprived. Therefore, security protocols must be computation and communication inexpensive [34, 80]. Traditional models of authentication are too clumsy (computation and communication overhead) and cannot be applied directly to IIoT environment [15, 81, 82, 83]. Therefore, new security and privacy paradigms must be developed to cater the need of IIoT networks.

### 5.1.1 Our Contribution

- We propose a Robust and Lightweight Key Exchange (LKE) protocol for Industrial IoT networks.
- To achieve the robustness and efficiency, ECQV implicit certificates, asymmetric and symmetric key cryptography, keyed-hash, and nonces are used.
- The proposed scheme assures mutual authentication between industrial node and gateway before secret key generation.
- The proposed protocol provisions the renewal of expired certificates to support long term connectivity between entities and strengthening security measures (e.g., prevention from impersonation and replay attacks etc.).

- The strength of LKE is tested using formal and informal security analysis where it is found that LKE exhibits essential security properties, like authentication, confidentiality etc., and is also resistant against impersonation, replay, and MITM attacks etc.
- The performance of the proposed scheme is compared with the state-of-the-art to show its superiority over them in terms of computational and communicational efficiency.

## 5.2 System Model, Adversary Model, and Security and other goals

### 5.2.1 System Model

Fig. 5.2 depicts an IIoT network controlled and monitored over the internet. The architecture of the IIoT constitutes of IoT sensor nodes deployed at machines which communicates to Certification Authority (CA) and cloud via gateway using wireless bi-directional link. The user gets access to information through cloud.

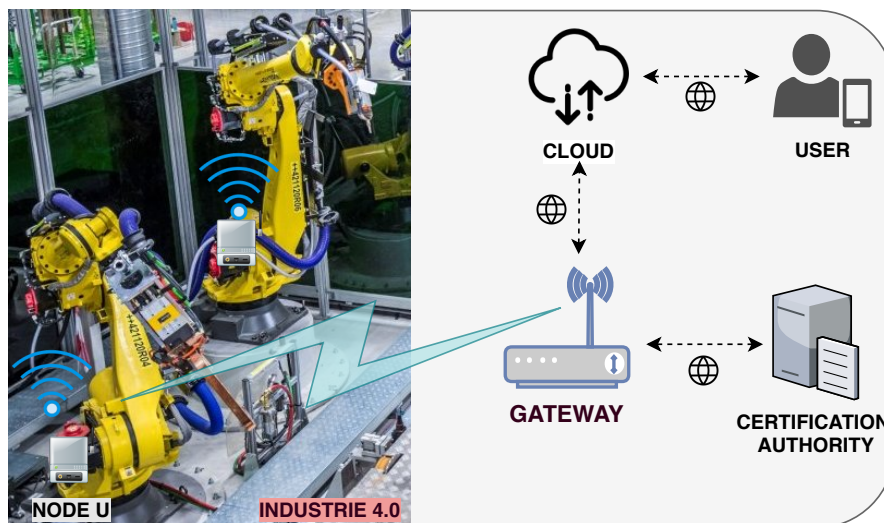


FIGURE 5.2: System Model for mutual authentication and key exchange between IoT devices in Industry 4.0.

#### 5.2.1.1 WSN-IIoT network

The machines in the industry are equipped with the sensor nodes. The sensor nodes receive control signals (e.g., turn on/off the machine etc.) from operator, collect data



from machines (e.g., production count, temperature of machine, pressure, etc.) and relay it wirelessly to the gateway using low powered modules e.g., Zigbee (IEEE 802.15.4) and Z-Wave (e.g., ZW0500).

### 5.2.1.2 Gateway

Gateway is usually stationary and powered with mains. Gateway acts as an intermediary to support the communication between smart IoT sensor node, cloud (e.g., Kinsta and Microsoft Azure) and CA. It supports IEEE 802.3 and IEEE 802.11 standard for transporting the data over the internet. The gateway is responsible for authenticating the nodes deployed in the IIoT network before relaying their information to cloud.

### 5.2.1.3 Certification Authority

The certification authority (e.g., *Symantec*, *GeoTrust*, etc.) creates a database of the nodes deployed in the network and utilizes it later to conduct mutual authentication before issuing certificates to nodes. CA issues unique implicit certificates to each sensor node which is required by them to construct their public and private keys.

## 5.2.2 Adversary Model

The proposed scheme has adopted the Dolev-Yao adversary model advised in [24, 52, 84]. As per the threat model, adversary has the capabilities to discover the vulnerabilities of the industrial network; these vulnerabilities can be used to exploit the potential resources of the industries. Consider an IIoT enabled smart car manufacturing industry [85] where sensor nodes are deployed to monitor and control the activities of robotic arms, manage the logistics, and identify the raw material requirements at warehouse, etc. Following the Dolev-Yao adversary model, the robotic industrial machines (nodes), logistics and warehouse network devices (gateway), etc. are prone to attacks. An adversary in IIoT can snoop on all the conversations that occur between industrial nodes, gateway and CA. More specifically, an adversary can capture, modify and replay the messages exchanged between network entities to get privileged access of industrial robotic arms (e.g., welding, painting, transportation, and assembling), etc. Additionally, adversary can impersonate as legitimate industrial node to steal precious RFID tag information. Physical capturing of the devices (nodes and gateway) inside the smart industries is not possible as they are secured using physical locks along with monitoring through surveillance cameras. The attacker can attempt to alter the lifetime of the expired authenticator to intrude illegally into the industrial network for introducing malware in the computerised production units of industry. Moreover, the adversary can intercept the messages exchanged between the network entities to retrieve security

parameters to generate future secret keys, actuate driverless cars, etc. The adversary can construct and inject new messages into the network to launch DoS attack to cause obstruction in sending control commands to the industrial machines (e.g., warehouse storage sequencing error). Conclusively, the adversary has adequate capabilities to hinder the smooth and secure functioning of the manufacturing units, warehouses, and logistics etc. The adversarial attacks can result in financial and reputation loss, business disruption, and decreased efficiency etc.

### 5.2.3 Security and other goals

*Security goals* subsection discusses the desirable security properties that a security protocol must exhibit to be declared as *robust*, whereas, *other goals* subsection discusses those preferable properties that prove the protocol as *efficient*.

#### 5.2.3.1 Security Goals

LKE complies with the significant security properties. Note that the security properties are adopted from [14, 53].

1. **Mutual authentication and secret key establishment:** Industrial IoT networks are sensitive as industrial machines are involved. The nodes must perform mutual authentication followed by the secret key exchange to protect their communications from illegitimate nodes.
2. **Message integrity and freshness:** Alterations devastate the real content of the message and stale messages may trigger non-permissible actions. Therefore, security protocols must incorporate certain procedures to let the entities verify the message integrity and freshness.
3. **Defense against prominent attacks:** Impact of attacks can be mild or severe and may lead to temporary or permanent suspension of the industry operations. Therefore, security protocols must be resistant to prominent attacks like impersonation, replay, alteration of information, DoS, MITM, and known key.
4. **Data Privacy:** Industrial IoT network carries very sensitive information (e.g., control commands, confidential manufacturing process, secret keys or credentials, and authentication passcode, etc.). Any disclosure of such information may disrupt the business operations as well as can tarnish the reputation of the industry. The consequences of the disclosure of information to unauthorized entities may vary from benign to severe. Therefore, security protocols must ensure that information exchanged must remain confidential even if the adversary captures the messages.

5. **Identity Anonymity:** Adversaries are always seeking crucial details like identities of the industrial nodes and other network devices. These identities can be used by the adversaries as a useful element to conduct a MITM attack, etc. Therefore, it is desirable that the identity of the nodes and other network devices (e.g., gateway) should remain anonymous. Identity anonymity not only prevents attacks (e.g., MITM and impersonation attacks, etc). rather keeps the overall communication anonymous.

### 5.2.3.2 Other Goals

LKE exhibits the prominent properties like *lightweightness* and *certificate renewal* to ensure *efficiency* and *long term connectivity*, respectively.

1. **Lightweightness:** As the IoT nodes are usually resource-deprived; therefore the nodes must perform limited computations and communications while still achieving the highest possible degree of security.
2. **Certificate Renewal:** The implicit certificates generated by CA for industrial nodes have time-bound validity. The expiration of the certificates terminates the communication session between industrial nodes and gateway. It is highly desirable to provide a certificate renewal process to allow the interested industrial nodes to re-establish a new secure session with the gateway. The certificates of the legitimate industrial nodes should be renewed with bare minimum computation and communication complexities.

## 5.3 Proposed Scheme

This section describes the working of robust and lightweight key exchange protocol for distributed IIoT applications. Fig. 5.2 portrays a smart car manufacturing industry in German [85] where FANUC 2000 IC robots are being used. A system model considering this scenario is presented in Fig. 5.2. The FANUC 2000 IC robots are equipped with the IoT sensor nodes. The sensor nodes collect the data and forward it to the cloud via gateway. In order to ensure the security of this communication, a mutual authentication and key exchange protocol is presented in this section.

The notations with their description, sizes and methodologies are provided in Table 5.1. The proposed scheme consists of 3 phases: (A) *System set-up and registration phase*, (B) *Certificate and Node Key generation phase*, and (C) *Light-weight Key establishment phase*. Furthermore, section (D) demonstrates the renewal process of revoked certificates.

TABLE 5.1: Symbols, Abbreviations and Operators description

Notations	Descriptions	Size (bytes)	Method
$r_U, r_{CA}$	A random integer generated by node U and Certification Authority (CA)	2, 2	LCC
$G_{(x,y)}, n$	Base point generator and its order	16, 2	ECC
$R_U$	Elliptic curve point for certificate request sent by node U	16	ECC
$Cert_N$ & e	Implicit certificate of $N^{th}$ node and its Hash value	16, 16	Implicit, MD5
$d_{CA}, Q_{CA}$	Private and public key of CA	32, 32	ECC-256
$d_U, Q_U$	Private and public key of node U	32, 32	ECC-256
$d_G, Q_G$	Private and public key of Gateway ( $G_w$ )	32, 32	ECC-256
$s_{(U/G)}, CR_{Req}$	Implicit signature, Certificate renewal request	32, 1	-
TS and LT	Time Stamp and Lifetime	8, 8	Date, time & time zone
$id_U, id_G, id_{CA}$	Identity of node U, Gateway and CA	1, 1, 1	-
$K_{UG}$	Shared Secret key between node U and Gateway	32	AES-256
$N_X$	It is a random positive integer called as nonce. X indicates the order of nonce.	8	LCC
E and D	Encryption and Decryption	-	AES-128
$\parallel, \oplus, +, mod$	Concatenation, XOR, Addition, modulus	-	-
$K_T$	Temporary key	16	AES-128
HMAC, Hash	Hash-based message authentication code, Hash (message digest)	16, 16	HMAC-MD5, MD5
$A_N$	Authenticator generated by CA for $N^{th}$ node	65	ECC-256

Acronyms: **LCC**: Linear Congruential Generator, **ECC**: Elliptic Curve Cryptography, **MD5**: Message Digest, **AES**: Advanced Encryption Standard

To demonstrate the working of protocol, some assumptions are considered; (a) CA is a trusted and tamper-proof entity and has no constraints concerning computational strength and storage space, (b) The CA, gateway, and IIoT sensor nodes are considered to have the equivalent capability of executing the cryptography operations (e.g., ciphering, hashing functions), (c) Gateway has no limitations concerning computational strength, storage (tamper-proof), and broadcasts its ID ( $id_G$ ) at regular intervals, (d) Gateway has finished the registration at CA and formed the pair of keys (public,  $Q_G$  and private,  $d_G$ ) through Authenticator  $A_G$ .

### 5.3.1 System Set-up and Registration Phase

Prior to the network deployment, all the IoT sensor nodes get registered offline to the CA and obtain security credentials such as *Generator point* and order of Elliptic Curve Cryptography (ECC). Note that we intentionally omitted the ECQV background, interested readers may refer to [86]. During registration, CA assigns unique identity (e.g.,  $id_U$ ) to each node and stores it in the node memory. In addition, CA provides its public key,  $Q_{CA}(= d_{CA}G)$  to registered nodes. Finally, CA prepares a database of all registered nodes ( $id_U, id_G...$ ) and stores them in memory.

### 5.3.2 Certificate and Node Key Generation Phase

It is an initial phase where the deployed nodes configure themselves automatically. Let us consider a node U ( $id_U$ ) as one of nodes deployed in the network. The node U requests CA for generating and provisioning its implicit certificate. This certificate is required by the node U to prove its legitimacy among other entities and also to generate its public ( $Q_U$ ) and private key ( $d_U$ ). This phase is invoked only during *first time network setup*. The complete process of certificate and node key generation is illustrated in Fig. 5.3 along with demonstration in this section.

#### Dialogue Exchange between Node U, Gateway & CA

{Note:  $O_N$  and  $M_N$  represents Operation and Message number, respectively (Here N comprises of positive integer values e.g., 1, 2, 3 etc.)}

At first, the Node U ( $N_U$ ) generates a random integer,  $r_U$  ( $O_1$ ) and elliptic curve (EC) point,  $R_U$  ( $O_2$ ). Upon generation,  $N_U$  prepares a message comprising of its identity ( $id_U$ ), gateway identity ( $id_G$ ), EC point ( $R_U$ ), and nonce ( $N_1$ ). The message is hashed ( $O_3$ ) and encrypted with  $Q_{CA}$  to ensure integrity and confidentiality, respectively.  $N_U$  sends the message  $M_1$  to Gateway ( $G_w$ ).

$O_1: r_U \in_R [1, \dots, n - 1]$

$O_2: R_U = r_U G$

$O_3: H_1 = Hash(id_U || R_U || id_G || N_1)$

$M_1: E_{Q_{CA}}[id_U || R_U || id_G || N_1 || H_1]$  {Node U  $\rightarrow$   $G_w$ }

$G_w$  appends the credentials of CA ( $id_{CA}$ ) and itself ( $id_G$ ) together with the fresh nonce

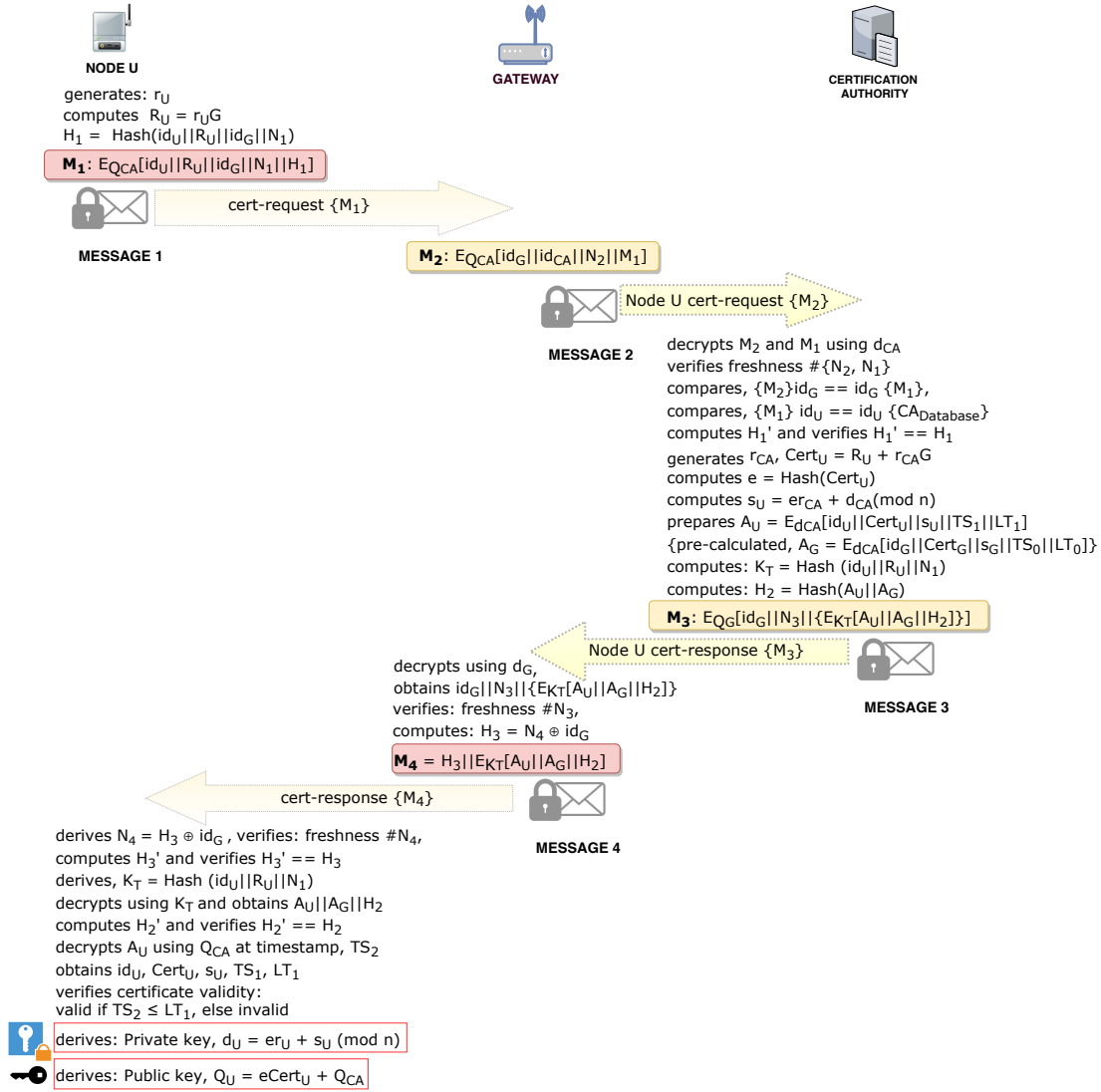


FIGURE 5.3: Certificate and Node Key Generation Phase.

( $N_2$ ) to the received message  $M_1$ , encrypts it with  $Q_{CA}$  and sends it ( $M_2$ ) to CA.

$$M_2: E_{Q_{CA}}[id_G || id_{CA} || N_2 || M_1] \{G_w \rightarrow CA\}$$

CA receives  $M_2$  and decrypts it using  $d_{CA}$  ( $O_4$ ) to extract  $M_1$ . Furthermore, CA decrypts  $M_1$  using its private key,  $d_{CA}$  to fetch credentials of  $N_U$  ( $O_5$ ). Post decryption, CA examine the nonces  $N_2$  and  $N_1$  for verifying the freshness of the received message. CA retrieves the identity of the gateway ( $id_G$ ), and the identity of node U ( $id_U$ ) from the messages  $M_2$  and  $M_1$ , respectively to compare  $\{(M_2)id_G == id_G(M_1)\}$  and  $\{(M_1)id_U == id_U(CA_{Database})\}$ , and prove that messages have arrived from trustworthy nodes only. After validating the freshness and faithfulness of the messages, CA computes and verifies the hash,  $H_1' == H_1$  in  $O_6$  to inspect the integrity of node credentials and request.

Afterwards, CA generates random integer,  $r_{CA}$  ( $O_7$ ), implicit certificate,  $Cert_U$  ( $O_8, O_9$ ), signatures,  $s_U$  ( $O_{10}$ ) followed by Authenticator ( $A_U$ ) in  $O_{11}$ . As aforementioned  $A_G$  is

stored in CA database because the gateway has finished the registration with CA prior to nodes.  $A_U$  is encrypted by  $d_{CA}$  and constitutes of node identity ( $id_U$ ), certificate ( $Cert_U$ ), signature ( $s_U$ ), timestamp ( $TS_1$ ) and lifetime ( $LT_1$ ).  $TS_1$  is the current timestamp of CA whereas  $TS_0$  represents the timestamp of CA used during preparation of  $G_w$  authenticator ( $A_G$ ).  $LT_0$  and  $LT_1$  defines the validity (lifetime) of the  $G_w$  and  $N_U$  certificate, respectively and their validity depends upon sensitivity of the application (may range from 3 ~ 12 months). Timestamp and Lifetime parameters allows the message recipients to verify the legitimacy of the request which in turn prevents the network from replay and other similar attacks.  $A_U$  is resistant against modifications because the attacker does not have the secret key of CA ( $d_{CA}$ ) required to perform the alterations.

$$O_4: D_{d_{CA}}[id_G || id_{CA} || N_2 || M_1]$$

$$O_5: D_{d_{CA}}[id_U || R_U || id_G || N_1 || H_1]$$

$$O_6: H'_1 = Hash(id_U || R_U || id_G || N_1) \{H'_1 == H_1\}$$

$$O_7: r_{CA} \in_R [1, \dots, n-1]$$

$$O_8: Cert_U = R_U + r_{CA}G$$

$$O_9: e = Hash(Cert_U)$$

$$O_{10}: s_U = er_{CA} + d_{CA}(\text{mod } n)$$

$$O_{11}: A_U = E_{d_{CA}}[id_U || Cert_U || s_U || TS_1 || LT_1]$$

$$\{\text{pre-calculated } A_G = E_{d_{CA}}[id_G || Cert_G || s_G || TS_0 || LT_0]\}$$

CA computes  $K_T$  in  $O_{12}$  using identity of  $N_U$  ( $id_U$ ), EC point of  $N_U$  ( $R_U$ ) and nonce ( $N_1$ ). CA concatenates gateway identity ( $id_G$ ), fresh nonce ( $N_3$ ) along with  $K_T$  encrypted message ( $A_U, A_G, H_2$ ) and encrypts it with  $Q_G$  ( $O_{13}, M_3$ ). The  $A_U, A_G$  cannot be decrypted and forged by any unauthorized entity as EC point knowledge is only available with  $N_U$ . Nonce is also used to prevent from replay attacks. The prepared message  $M_3$  is sent to the gateway.

$$O_{12}: K_T = Hash(id_U || R_U || N_1)$$

$$O_{13}: H_2 = Hash(A_U || A_G)$$

$$\boxed{M_3: E_{Q_G}\{id_G || N_3 || E_{K_T}[A_U || A_G || H_2]\}} \{CA \rightarrow G_w\}$$

Gateway decrypts the received message,  $M_3$  using his private key,  $d_G$  and verifies the nonce,  $N_3$  ( $O_{14}$ ). Post successful verification, gateway forwards the message  $M_4$  to  $N_U$ . Message  $M_4$  comprises of  $H_3$  ( $N_4 \oplus id_G$ ;  $O_{15}$ ),  $K_T$  encrypted authenticators ( $A_U, A_G$ ) and hash ( $H_2$ ).

$$O_{14}: D_{d_G}\{id_G || N_3 || [E_{K_T}[A_U || A_G || H_2]]\}$$

$$O_{15}: H_3 = N_4 \oplus id_G$$

$$\boxed{M_4: H_3 || E_{K_T}[A_U || A_G || H_2]} \{G_w \rightarrow \text{Node } U\}$$

Node  $U$  derives  $N_4$  and verifies its freshness. In addition, Node  $U$  computes  $H'_3$  and check for integrity  $O_{16}$ . Node  $U$  accepts the message if the nonce is fresh ( $\#N_4$ ) and integrity is preserved ( $H'_3 == H_3$ ). Node  $U$  computes  $K_T$  and recovers  $A_U$  and  $A_G$  in  $O_{17}$  and  $O_{18}$ , respectively. Further  $N_U$  computes  $H'_2$  for verifying message integrity ( $O_{19}$ ) followed by decryption of  $A_U$  using  $Q_{CA}$  in  $O_{20}$ .  $N_U$  verifies the validity of the  $Cert_U$  before processing further.  $N_U$  computes private key,  $d_U$  and public key,  $Q_U$  using

$Cert_U, s_U, r_U$  and  $Q_{CA}$  ( $O_{21} - O_{22}$ ).

$O_{16}: H'_3 = N_4 \oplus id_G \{H'_3 == H_3\}$

$O_{17}: K_T = Hash(id_U || R_U || N_1)$

$O_{18}: D_{K_T}[A_U || A_G || H_2]$

$O_{19}: H'_2 = Hash(A_U || A_G) \{H'_2 == H_2\}$

$O_{20}: D_{Q_{CA}}[id_U || Cert_U || s_U || TS_1 || LT_1]$  (Decrypted at  $TS_2$ )

**Note:**  $Cert_U$  is valid if this condition is true:  $TS_2 \leq LT_1$ , else invalid.

$O_{21}: d_U = er_U + s_U \pmod n$  {private key}

$O_{22}: Q_U = d_U G$

$= (er_U + s_U \pmod n) G$

$= (er_U + er_{CA} \pmod n + d_{CA} \pmod n) \pmod n G$

$= (er_U + er_{CA} \pmod n + d_{CA} \pmod n) G$

$= e(r_U + r_{CA})G + d_{CA}G$

$= e(r_U G + r_{CA} G) + Q_{CA}$

$= e(R_U + r_{CA} G) + Q_{CA}$

$Q_U = eCert_U + Q_{CA}$  {public key}

Node U has successfully constructed  $Q_U$  and  $d_U$ .

### 5.3.3 Light-weight Key Establishment Phase

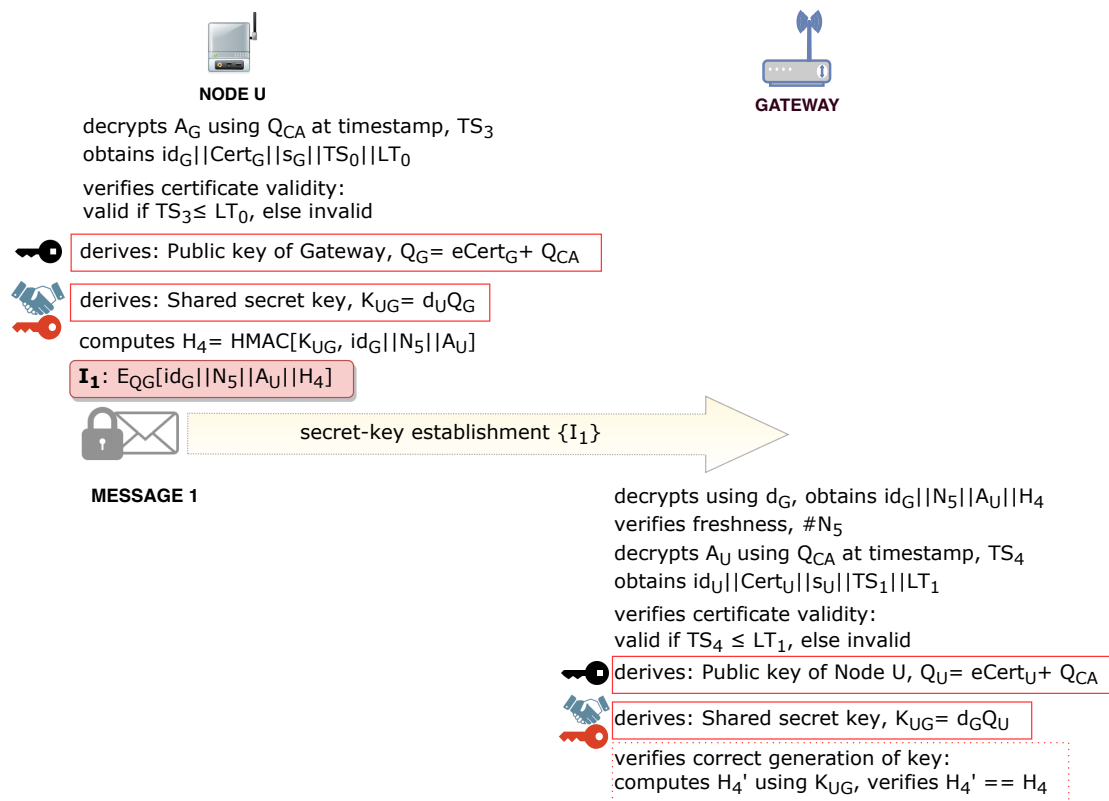


FIGURE 5.4: Key Establishment between Node U and Gateway.



Key establishment process is initiated by  $N_U$ . Fig. 5.4 illustrates the whole process. Note that  $S_N$  and  $I_N$  represents Operation and Message number, respectively (Here  $N$  comprises of positive integer values e.g., 1, 2, 3 etc.).  $N_U$  derives the credentials of  $G_w$  by decrypting the  $A_G$  with  $Q_{CA}$  ( $S_1$ ). Subsequently,  $N_U$  verifies the lifetime of the  $Cert_G$  and if found unexpired then it retrieves  $Q_G$  ( $S_2, S_3$ ). Finally shared secret key is produced i.e.,  $K_{UG} = d_U Q_G$  ( $S_4$ ). Post generation of  $K_{UG}$ ,  $N_U$  computes HMAC of  $id_G, N_5, A_U$  with secret key,  $K_{UG}$  ( $S_5$ ). Following that  $N_U$  prepares  $I_1$  and sends it to  $G_w$ .

$S_1: D_{Q_{CA}}[id_G || Cert_G || s_G || TS_0 || LT_0]$  (Decrypted at  $TS_3$ )

Note:  $Cert_G$  is valid if this condition is true:  $TS_3 \leq LT_0$ , else invalid.

$S_2: e = Hash(Cert_G)$

$S_3: Q_G = eCert_G + Q_{CA}$

$S_4: K_{UG} = d_U Q_G$  {shared secret key}

$S_5: H_4 = HMAC[K_{UG}, id_G || N_5 || A_U]$

$\mathbb{I}_1: E_{Q_G}[id_G || N_5 || A_U || H_4]$  {Node  $U \rightarrow G_w$ }

$S_6: D_{d_G}[id_G || N_5 || A_U || H_4]$

$S_7: D_{Q_{CA}}[id_U || Cert_U || s_U || TS_1 || LT_1]$  (Decrypted at  $TS_4$ )

Note:  $Cert_U$  is valid if this condition is true:  $TS_4 \leq LT_1$ , else invalid.

$S_8: Q_U = eCert_U + Q_{CA}$

$S_9: K_{UG} = d_G Q_U$  {shared secret key}

$S_{10}: H'_4 = HMAC[K_{UG}, id_G || N_5 || A_U]$  { $H'_4 == H_4$ }

$G_w$  decrypts the received message,  $I_1$  using  $d_G$  and  $Q_{CA}$  ( $S_6, S_7$ ) and produces  $Q_U$  ( $S_8$ ). Note that gateway evaluates the validity of the certificate before producing  $Q_U$ . Later, Gateway utilizes  $Q_U$  to produce shared secret key i.e.,  $K_{UG} = d_G Q_U$ . As a result, both the entities generate the same secret keys securely ( $S_{10}$ ). Note that lifetime of the keys {public ( $Q_{CA}, Q_G, Q_U$ ), private ( $d_{CA}, d_G, d_U$ ), secret key ( $K_{UG}$ )} depend upon sensitivity of data and is application dependent.

### 5.3.4 Certificate Renewal Phase

In real time scenarios, each certificate is integrated with validity. After the lapse of certificate validity, the secret key (e.g.,  $K_{UG}$ ) becomes invalid and results in termination of communication session between an industrial node (e.g.,  $N_U$ ) and gateway (e.g.,  $G_w$ ). Consequently, the industrial nodes that seek to continue the communication with the gateway initiates a certificate renewal process with the CA. Upon the accomplishment of certificate renewal, the new secret key is negotiated between the industrial node and gateway. The process of renewal is depicted in Fig. 5.5 as well as justified through dialogue exchange in this section.

{Note:  $C_N$  and  $D_N$  represents operation and message number, respectively (Here  $N$  comprises of positive integer values e.g., 1, 2, 3 etc.)}

$C_1: r_{2U} \varepsilon_R [1, \dots, n - 1]$

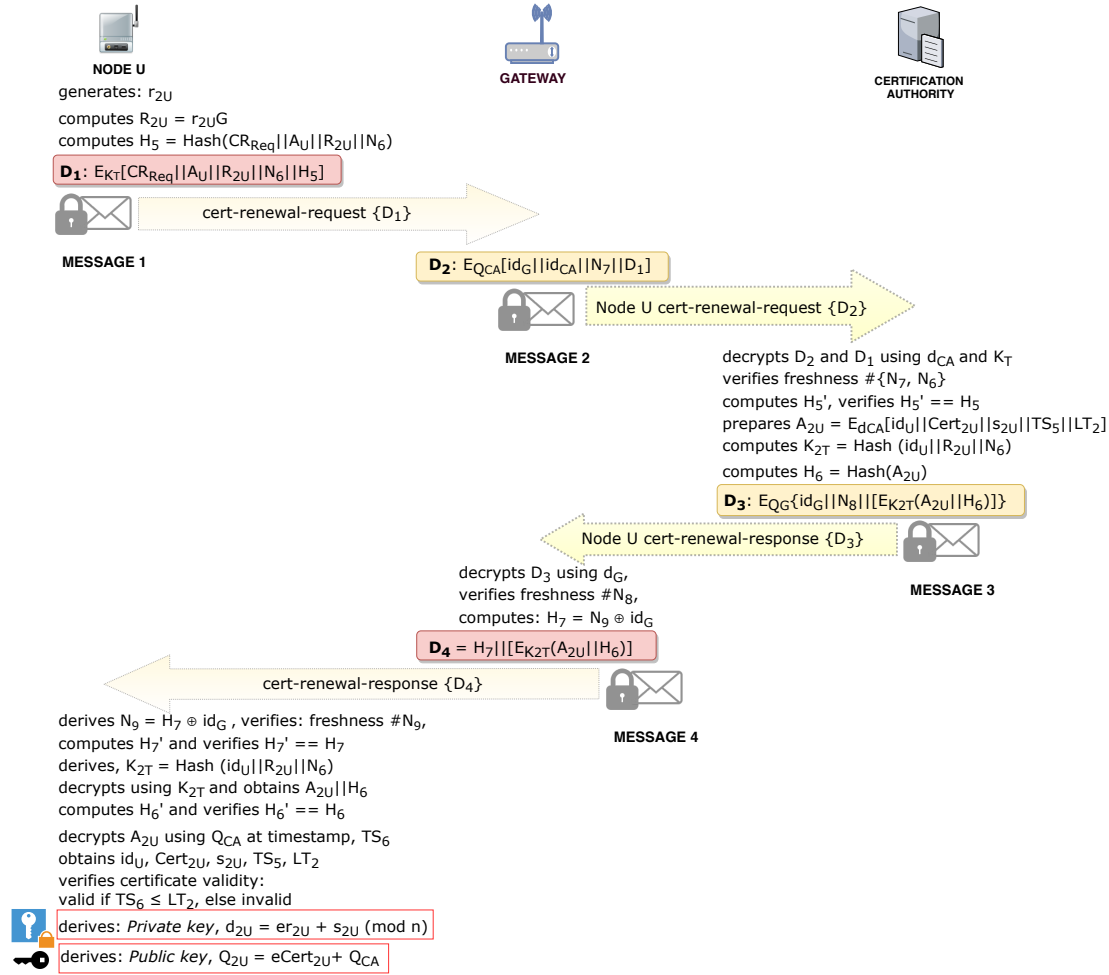


FIGURE 5.5: Certificate Renewal Phase.

$$C_2: R_{2U} = r_{2U}G$$

$$C_3: H_5 = \text{Hash}(CR_{Req}||A_U||R_{2U}||N_6)$$

$$D_1: E_{K_T}[CR_{Req}||A_U||R_{2U}||N_6||H_5] \{Node\ U \rightarrow G_w\}$$

$$D_2: E_{Q_{CA}}[id_G||id_{CA}||N_7||D_1] \{G_w \rightarrow CA\}$$

$$C_4: D_{d_{CA}}[id_G||id_{CA}||N_7||D_1]$$

$$C_5: D_{K_T}[CR_{Req}||A_U||R_{2U}||N_6||H_5]$$

$$C_6: A_{2U} = E_{d_{CA}}[id_U||Cert_{2U}||s_{2U}||TS_5||LT_2]$$

$$C_7: K_{2T} = \text{Hash}(id_U||R_{2U}||N_6)$$

$$C_8: H_6 = \text{Hash}(A_{2U})$$

$$D_3: E_{Q_G}[id_G||N_8||\{E_{K_{2T}}[A_{2U}||H_6]\}] \{CA \rightarrow G_w\}$$

$$C_9: D_{d_G}[id_G||N_8||\{E_{K_{2T}}[A_{2U}||H_6]\}]$$

$$C_{10}: H_7 = N_9 \oplus id_G$$

$$D_4: H_7||\{E_{K_{2T}}[A_{2U}||H_6]\} \{G_w \rightarrow Node\ U\}$$

$$C_{11}: K_{2T} = \text{Hash}(id_U||R_{2U}||N_6)$$

$$C_{12}: D_{K_{2T}}[A_{2U}||H_6]$$

$$C_{13}: D_{Q_{CA}}[id_U||Cert_{2U}||s_{2U}||TS_5||LT_2] \text{ (Decrypted at } TS_6)$$

Note:  $Cert_{2U}$  is valid if this condition is true:  $TS_6 \leq LT_2$ , else invalid.

$C_{14}$ :  $d_{2U} = er_{2U} + s_{2U}(\text{mod } n)$  {private key}

$C_{15}$ :  $Q_{2U} = eCert_{2U} + Q_{CA}$  {public key}

## 5.4 Security Analysis

The strength of the proposed protocol, LKE has been analyzed through formal and informal analysis. The inferences obtained from analysis are presented in this section.

### 5.4.1 Formal Analysis

Following [14, 27, 34, 81, 83], and [87], we have used AVISPA (Automated Validation of Internet Security Protocols and Applications) tool to examine the robustness of the proposed protocol under the influence of the Dolev-Yao adversary model. The examination using the AVISPA requires conversion of security protocol algorithm to High Level Protocol Specification Language (HLPSL). AVISPA transforms the HLPSL script file to an Intermediate Format (IF) using a HLPSL2IF translator. The Intermediate Format is then provided to the backend (e.g., on-the-fly model-checker (OFMC)) of the AVISPA for compilation of results. The discussion on various backends of AVISPA is intentionally omitted, interested readers may refer to [55]. Conclusively, the backend produces the Output file (OF) inferring the protocol as safe or unsafe.

The HLPSL script initially discusses the basic roles to be played by the agents and define local declarations. Basic role represents the change in the states of the node when certain events are met. Contrarily, composition role does not make any transitions rather administer numerous sessions concurrently. The environment role is the last section of the script which constitutes of one or more sessions and global constants. In addition, the behaviour of the intruder ( $i$ ) is also defined in the environment role. It is also mentioned in this role that communication between the entities happen over the compromised channel ( $dy$ ), i.e., the channel is vulnerable to all types of attacks mentioned in the Dolev-Yao (DY) adversary model.

To evaluate the robustness of LKE, the mutual authentication and secret key establishment phase is scripted in HLPSL and examined on AVISPA. At first, basic roles of node  $U$  and  $Gw$  are described which includes agent attributes ( $U, Gw$ ), crypto operations, local declarations ( $Qu, Du$ , etc.), channel ( $dy$ ), initial state and transitions. Node  $U$  initiates the communication. Post initialization at  $State = 0$  [RCV( $start$ )], it succeeds to  $State = 1$ , where fresh nonce is constructed,  $N5' := new()$  and appended with  $Au' = \{Id_u.Cert_u.Su.Ts1.Lt1\}.Qca$ , and  $H4' = Hmac(Kug.Idg.N5.Au)$ . Node  $U$  transmits  $I1'$

<pre> role nodeU (U,Gw: agent,             Hmac: hash_func,             Qca,Qg,Qu: public_key,             Dg,Du,Kug: symmetric_key,             SND,RCV: channel (dy)) played_by U def= local State: nat, Idu,Idg,Certu,Certg,Su,Sg,Ts0, Ts1,Lt0,Lt1,N5,Au,Ag,H4:text, I1: message init State:= 0 transition 1. State = 0 /\ RCV(start) = &gt;    State' := 1 /\ N5' := new() /\ Ag' := {Idg.Certg.Sg.Ts0.Lt0}_Qca /\ Au' := {Idu.Certu.Su.Ts1.Lt1}_Qca /\ H4' := Hmac(Kug.Idg.N5.Au) /\ I1' := {Idg.N5.Au.H4}_Qg /\ SND(I1') /\ secret({Idg,Au'},sub1,{U,Gw}) end role </pre>	<pre> role gateway (U,Gw: agent,              Hmac: hash_func,              Qca,Qg,Qu: public_key,              Dg,Du,Kug: symmetric_key,              SND,RCV: channel (dy)) played_by Gw def= local State :nat, Idu,Idg,Certu,Su, Ts1,Lt1,N5,Au,H4:text, I1: message init State:= 1 transition 1. State = 1 /\ RCV(I1') = &gt;    State' := 2 /\ I1' := {Idg.N5.Au.H4}_Dg /\ Au' := {Idu.Certu.Su.Ts1.Lt1}_Qca /\ H4' := Hmac(Kug.Idg.N5.Au) /\ witness(Gw,U,nodeU_gateway_n5,N5) /\ witness(Gw,U,nodeU_gateway_lt1,Lt1) end role </pre>
(a)	(b)

FIGURE 5.6: Role Specification of the Node U and Gateway.

<pre> role session (U,Gw: agent,             Hmac: hash_func,             Qca,Qg,Qu: public_key,             Dg,Du,Kug: symmetric_key) def= local SU,RU,SGw,RGw: channel(dy) composition nodeU(U,Gw,Hmac,Qca,Qg,Qu,Dg,Du,Kug,SU,RU) /\gateway(U,Gw,Hmac,Qca,Qg,Qu,Dg,Du,Kug,SGw,RGw) end role </pre>	<pre> role environment () def= const nodeU,gateway: agent,       qca,qg,qu: public_key,       dg,du,kug,dgi,dui,kugi: symmetric_key,       idu,idg,certu,certg,su,sg,       ts0,ts1,lt0,lt1,n5,au,ag,h4: text,       hmac: hash_func,       nodeU_gateway_n5,nodeU_gateway_lt1,       sub1: protocol_id intruder_knowledge = {nodeU,gateway,hmac,dgi,dui,qca,qg,qu} composition session(nodeU,gateway,hmac,qca,qg,qu,dg,du,kug) /\session(nodeU,i,hmac,qca,qg,qu,dgi,dui,kugi) /\session(i,gateway,hmac,qca,qg,qu,dgi,dui,kugi) end role </pre>
(a)	(b)
<pre> goal secrecy_of sub1 authentication_on nodeU_gateway_n5 authentication_on nodeU_gateway_lt1 end goal environment () </pre>	
(c)	

FIGURE 5.7: Specification of the session, environment and goal for the proposed LKE.

to the gateway for achieving mutual authentication and secret key establishment assuming dolev-yao (*dy*) channel characteristics. The goal predicates set by the Node *U* is the privacy of the authenticator, i.e.,  $Au'$  & anonymity of the gateway identity i.e.,  $Idg'$  as depicted in Fig. 5.6 (a).

Gateway receives the  $I1'$  in its initial state,  $State = 1$  [RCV( $I1'$ )] and retrieves the data during 2<sup>nd</sup> State. Gateway executes specific operations for the strong realization of

mutual authentication and key establishment as presented in Fig. 5.6 (b).  $Gw$  decrypts  $I1'$  using  $Dg$  and extracts  $Idg, N5, Au, H4$ . Similarly,  $Gw$  decrypts  $Au'$  using  $Qca$  and recovers node  $U$  credentials ( $Idu, Certu, Su$ ) with timestamp ( $Ts1$ ) and lifetime ( $Lt1$ ). The verification of  $N5'$  and  $Lt1'$  is formed in terms of goals predicate *witness*  $\{nodeU\_gateway\_lt1\}$  and  $\{nodeU\_gateway\_n5\}$ . Witness makes sure that the lifetime ( $LT$ ) of the certificate ( $Certu$ ) and freshness ( $N5$ ) of the message ( $I1'$ ) is validated before use. Gateway at  $State = 2$ , examines (*witness*( $Gw, U, nodeU\_gateway\_lt1, Lt$ )) the validity of  $Certu$  along with freshness (*witness*( $Gw, U, nodeU\_gateway\_n5, N5$ )) before initiating the process of secret key establishment.

Fig. 5.7 (a) demonstrates the structure of agents arguments.

$(U, Gw, Hmac, Qca, Qg, Qu, Dg, Du, Kug, SU, RU)$

$(U, Gw, Hmac, Qca, Qg, Qu, Dg, Du, Kug, SGw, RGw)$

Aforementioned arguments are either transmitted or applied by the agents during the session. The most significant is the environment role because it declares global constants, describes intruder behaviour, elucidates organization of sessions, and establishes goals of interest. Following DY adversary model, an attacker can eavesdrop, obstruct, and examine the information e.g.,  $\{nodeU, gateway, hmac, dgi, dui, qca, qg, qu\}$  etc. The intruder information is declared in the environment role and is utilized by the AVISPA (OFMC, Constraint-Logic-based Attack Searcher (CL-AtSe)) during the vulnerability assessment of the LKE against attacks. The subsequent segment of the environment role (Fig. 5.7 (b)) defines the numerous sessions of dialogue exchanges between entities.

Although it is anticipated to have sessions between legitimate agents only ( $nodeU, gateway, hmac, qca, qg, qu, dg, du, kug$ ), but the likelihood of intruder intruding in the session of authentic nodes also exists ( $nodeU, i, hmac, qca, qg, qu, dgi, dui, kugi$ ), ( $i, gateway, hmac, qca, qg, qu, dgi, dui, kugi$ ).

Overall, 3 goals are defined out of which one is linked with secrecy, and the other 2 corresponds to authentication as exhibited in Fig. 5.7 (c). The summary of the goals are:

- *Secrecy\_of sub1* represents that  $\{Au, Idg\}$  are kept secret between node  $U$  and gateway.
- *Authentication\_on nodeU\_gateway\_lt1* states that the lifetime (i.e.,  $Lt1$ ) of certificate  $\{Certu\}$  will be validated at the gateway.
- *Authentication\_on nodeU\_gateway\_n5* states that the freshness (i.e.,  $N5$ ) of message  $\{I1'\}$  will be confirmed at the gateway.

The strength of the LKE against attacks is tested using the OFMC backend. Fig. 5.8 (a) and Fig. 5.8 (b) demonstrate that LKE can resist critical attacks and is declared safe

<pre> % OFMC SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/ results/IIOT.if GOAL as_specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.04s visitedNodes: 18 nodes depth: 4 plies </pre>	<pre> % CL-AtSe SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/ results/IIOT.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 0 states Reachable : 0 states Translation: 0.01 seconds Computation: 0.00 seconds </pre>
(a)	(b)

FIGURE 5.8: LKE results using OFMC and CL-AtSe backend.

to use in Industrial IoT applications. Similarly, the CL-AtSe backend also declared the protocol as safe. Consequently, the attacks studied in the Dolev-Yao adversary model cannot damage the LKE security protocol.

## 5.4.2 Informal Analysis

The informal analysis proves the robustness of the proposed protocol against many known attacks.

1. **Prevention against Replay:** LKE can resist against replay attack. Suppose adversary (i.e., Eve) eavesdrops the message exchanged between Node U and Gateway and captures either,

$$M_1: E_{Q_{CA}}[id_U || R_U || id_G || N_1 || H_1],$$

$$M_4: H_3 || E_{K_T}[A_U || A_G || H_2],$$

$$I_1: E_{Q_G}[id_G || N_5 || A_U || H_4] \text{ or all.}$$

An adversary may launch the replay attack by resending the message  $M'_1$  or  $I'_1$  at different time intervals to the gateway to perform unauthorized operations. Replayed  $M'_1$  is received and processed by the Gateway,  $M_2: E_{Q_{CA}}[id_G || id_{CA} || N_1 || M'_1]$  and sent to CA. Since  $M'_1$  contains the old nonce ( $N_1$ ), therefore verification fails at CA. Similarly, adversary replay's  $I'_1$  to gateway but it is also perceived as dishonest as it contains the old nonce ( $N_5$ ). In the same way, adversary may eavesdrop messages exchanged between CA and  $G_w$  e.g.,  $M'_2$  ( $M_2: E_{Q_{CA}}[id_G || id_{CA} || N_2 || M_1]$ ) and  $M'_3$  ( $M_3: E_{Q_G}\{id_G || N_3 || [E_{K_T}[A_U || A_G || H_2]]\}$ ) and replay it to obtain authorizations. However, it will be identified as fraudulent due to presence of old nonces

$(N_2, N_3)$  in the replayed messages. Furthermore, adversary cannot read and alter the nonces  $(N_1, N_2, N_3, N_5)$  as messages  $M_1, M_2$  are ciphered with the public key of CA ( $Q_{CA}$ ) and  $M_3$  is encrypted with public key of the Gateway ( $Q_G$ ) and hence any alteration requires either the private key of CA or gateway which is unknown to Eve. Thus, proposed scheme is resilient to replay attacks.

2. **Prevention against Impersonation:** Impersonation is an identity theft which may lead to disclosure of information to non legitimate entities. In this attack, eve creates the fraudulent message  $E_{Q_{CA}}[id_{eve}||R_{eve}||id_G||N'_1 ||Hash(id_{eve}||R_{eve}||id_G||N_1)]$  to initiate new session by being Node U. Eve could not obtained real identity of Node U while intercepting the information, therefore eve constructed  $id_{eve}$  for impersonating node U. Nonetheless, CA could not verify the fake identity of eve in the database ( $id_{eve} \neq id_U$ ) and aborts the request. Even impersonating Node U during key establishment phase  $E_{Q_G}[id_G||N_5||A_U||HMAC_1[K_{UG}, id_G||N_5||A_U]]$  would not be possible for eve as he does not possess  $K_{UG}$  which is required for generating  $HMAC_1$ . Thus, it is not feasible to launch impersonation attacks in LKE.
3. **Prevention against Modification of Messages:** Assuming that eve captured the messages e.g.,  $M_1, M_2$  etc. Eve intentionally try to forge the messages such as  $E_{Q_G}[id_G ||N_5 ||A_U ||HMAC_{eve}[K_{Ueve}, id_G||N_5||A_U]]$ . It can be detected easily at the gateway since  $HMAC_{eve}$  is not computed using the correct key,  $K_{UG}$ . Similarly, any alterations in message,  $M_4$  requires the knowledge of symmetric key  $K_T$ , which is available either with CA or Node U. It can be witnessed that all the messages exchanged ( $M_1-M_4, I_1$ ) are sent after ciphering (using any of these keys  $E_{Q_{CA}}, E_{Q_G}, E_{K_T}, K_{UG}$ ) and hashing ( $H, HMAC_1$ ), thus leaving no scope for adversary to conduct modifications. Therefore, LKE is free from message forgery attack.
4. **Prevention against Denial of Service (DoS):** Assume an attacker can make use of old captured messages, and can send them to keep the system busy that would lead to the DoS attack [88, 89]. The DoS attacks does not only disrupt the services to be offered to the legitimate entity rather it leads to wastage of node resources like bandwidth, and power etc. LKE mitigates the DoS attacks to some extent. The Eve may intercept and replay  $M_1 \{E_{Q_{CA}}[id_U||R_U||id_G||N_1||Hash(id_U||R_U||id_G||N_1)]\}$  to initiate DoS. As the replayed message contains the old nonce ( $N_1$ ), therefore CA identifies it as a replay attack. Thus, irrespective of initiating new session with the illegitimate node, CA aborts the request and preserves its resources for legitimate nodes. Moreover, the Eve could not alter the nonce of  $M_1$  as it is encrypted. Similarly,  $M_2 - M_4$  and  $I_1$  are prevented from DoS attacks as they all constitutes of fresh nonces,  $N_2 - N_5$ , respectively. Thus, the proposed scheme can resist such DoS attacks.



5. **Prevention against MITM:** The intruder in this attack intercepts the information exchanged between the two legitimate parties and breaks their connection virtually. The intruder process is so transparent and smooth that the legitimate communicating parties never become aware of this virtual breakage. Let's suppose the intruder eavesdrop  $\{E_{Q_{CA}}[id_U || R_U || id_G || N_1 || Hash(id_U || R_U || id_G || N_1)]\}$  the information, and tries to modify it for playing MITM. This attempt would be unsuccessful because any modifications are permitted with use of  $d_{CA}$  which is not available with Eve. Nevertheless, eve may still try a vague attempt to modify with a forged key and replaces the information with  $E_{Q_{CA}}[id_U || R_U || id_G || N_1 || Hash(id_U || R_U || id_G || N_1)]_{eve}$ . However, this attempt can be detected at CA as decrypted information would not produce the correct message,  $id_U || R_U || id_G || N_1 \neq Hash(id_U || R_U || id_G || N_1)$ . Similarly, the eve would not be able to perform MITM using the remaining messages as they are also encrypted with secret keys,  $E_{Q_{CA}}[id_G || id_{CA} || N_2 || M_1]; E_{Q_G}\{id_G || N_3 || E_{K_T}[A_U || A_G || H_2]\}; H_3 || E_{K_T}[A_U || A_G || H_2]; E_{Q_G}[id_G || N_5 || A_U || H_4]$ . Thus, adversary cannot play MITM in LKE protocol.
6. **Prevention against Known Key:** Consider that eve has intercepted previous message exchanges and is trying to retrieve secret key related information from intercepted messages for producing new secret keys. As aforesaid that secret keys (e.g.,  $K_{UG}$ ) in LKE uses a secret random integer (e.g.,  $r_U$ ) which is fresh and independent for each certificate, thereby making the future secret key ( $K_{UG1} = d_U Q_V$ ) different and independent. So even if Eve obtains the old secret key ( $K_{UG}$ ) somehow, he would not be able to construct new secret key ( $K_{UG1}$ ) as it requires the knowledge of  $r_{U1}$  which is not available with eve. Therefore, having the knowledge of past secret keys does not help the eve to initiate new sessions.
- LKE adheres to all essential properties required to provision security in networks.
7. **LKE attains Data Privacy:** Disclosure of information or either key poses a threat to misuse of information. To avoid misuse, LKE encrypts all the messages to prevent unauthorized access. For instance, Node U encrypts  $I_1 : E_{Q_G}[id_G || N_5 || A_U || H_4]$ , thus allowing only the Gateway to decrypt and interpret. Even other messages such as  $M_1 - M_4$  are secured. Thus, even if the adversary intercepts the message  $M_1 - M_4$  and  $I_1$ , he would not be able to access the content without the key, therefore preserving the data confidentiality.
8. **LKE promises Message Integrity:** Alteration ruins the real identity of the message. Forged messages must be detected to prevent the processing of counterfeited requests. LKE makes use of Hash in  $M_1(Hash(id_U || R_U || id_G || N_1))$ ,  $M_2(Hash(id_U || R_U || id_G || N_1))$ ,  $M_3(Hash(A_U || A_G))$ , and  $M_4(Hash(A_U || A_G))$  whereas  $I_1$  uses  $HMAC([K_{UG}, id_G || N_5 || A_U])$ . Hash and HMAC are the one way functions used to preserve integrity in all messages exchanged in the scheme. Thus, proposed scheme (LKE) exhibits the property of message integrity.



9. **LKE ensures Message Freshness:** Replicated authorization messages may provide adversary the access to high privileged and non authorized resources. LKE therefore possess freshness component i.e., timestamp ( $TS_0 - TS_4$ ) and nonce ( $N_1 - N_5$ ), in all messages exchanged between CA, gateway and nodes. The nonce and timestamp ensures the abortion of process at receiving entity when stale requests are received. Thus, LKE exhibits the property of message freshness.
10. **LKE procured the property of Identity Anonymity:** In LKE, the identity of the industrial node ( $id_U$ ), gateway ( $id_G$ ), and CA ( $id_{CA}$ ) are exchanged as ciphertext to ensure attainment of identity anonymity [15, 24, 25, 26, 27]. Assume if an adversary captures the message  $I_1 (E_{Q_G}[id_G || N_5 || A_U || H_4])$  containing the identity details of the gateway ( $id_G$ ), still the adversary would not be able to extract the identity information as it is secured using strong encryption algorithm. Therefore, the communication remains anonymous to others. Similarly, other messages  $M_1$ ,  $M_2$ , and  $M_3$  carrying identity details ( $id_U$ ,  $id_G$ , and  $id_{CA}$ ) preserve the anonymity. Thus, LKE exhibits the property of identity anonymity to some extent.

## 5.5 Performance and Comparative Analysis

The employability of any scheme in practical environment depends upon its performance. The performance attributes of the proposed scheme (considering *Telos B mote* as the node) is observed and presented in this section. Table 5.2 presents the storage cost requirements for various entities involved in the proposed scheme. The storage cost requirements (*all phases*) for node, gateway and CA are 368, 252 and 305 bytes, respectively. The proposed scheme uses only 0.03 % of the total memory space (1 MB) available in CM5000 *Telos B mote* [90] for achieving the authentication. Thus, LKE establishes the secret key post mutual authentication with a less storage space requirement.

Table 5.3 points out the various security features that LKE exhibit along with the various attacks that LKE can resist. From the table it is witnessed that LKE provides robustness against all the potential attacks mentioned in the Dolev-Yao attack model [52]. Table 5.3 signifies the superiority of LKE over existing techniques [1, 15, 24, 25, 26, 27, 28] in terms of resistance against attacks and security features. The various cryptography operations used by node, gateway and CA during network set-up and key establishment phase are given in Table 5.4. It can be well observed from the table 5.4 that resource constrained node executes only a few operations whilst performing registration and key establishment process. The cryptography operations used by the entities (node,  $G_W$ , CA) are asymmetric and symmetric ciphering, hash and hash based message authentication code (HMAC).

TABLE 5.2: Storage Cost of Proposed Algorithm

Parameters	Node	Gateway	CA
$n, G$	✓		✓
$r_U$	✓		
$r_{CA}$			✓
$R_U$	✓		✓
$id_U, id_G, Q_G, Q_{CA}$	✓	✓	✓
$Id_{CA}$		✓	✓
$d_U$	✓		
$Q_U$	✓	✓	
$d_G$		✓	
$d_{CA}$			✓
$K_T$	✓		✓
$K_{UG}$	✓	✓	
$A_U$	✓	✓	✓
$A_G$	✓		✓
$N_1$	✓		✓
$N_2$		✓	✓
$N_3$		✓	✓
$N_4$	✓		
$N_5$	✓	✓	
Total Cost (bytes)	368	252	305

Table 5.5 provides a comparison of proposed scheme with state-of-the-work over computation cost between smart node and gateway. The comparison is carried out for *key establishment phase* only as the *registration phase* occurs once during network initialization. The parameters considered for comparison are asymmetric and symmetric ciphering, hash, HMAC, random number generation, exclusive-OR, and scalar multiplication in ECC. Results disclosed the efficiency of the scheme. LKE executes hash only twice whereas other schemes such as [1, 15, 24, 25, 26, 27, 28] executes hash 14, 16, 17, 12, 14, 21, and 17 times, respectively in key establishment phase. In addition, LKE computes XOR operation 2 times in contrast to 10, 8, 11, 5, 13, and 4 times by the schemes [1, 15, 24, 26, 27, 28], respectively. Similarly, other operations (ciphering, scalar multiplication, etc.) as shown in Table 5.5 are being executed many times by the traditional techniques to perform key establishment, resulting in over-exhaustion of the node resources. Consequently, LKE attains all necessary features like data privacy, authentication, integrity and availability etc. with limited computations.

Communication energy cost of the LKE and existing schemes are mentioned in the Table 5.6. As per the specifications of the Telos B mote [90], transmission and reception

TABLE 5.3: Analysis and Comparison of Protocols based on protection against attacks and security goals

$ASF$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$
$A_1$	✓	✓	✓	✓	✓	✓	✓	✓
$A_2$	✓	✓	✓	✓	✓	✓	✓	✓
$A_3$	✓	✓	✓	✓	✓	✓	✓	✓
$A_4$	×	✓	×	×	×	✓	×	✓
$A_5$	✓	✓	✓	×	×	✓	×	✓
$A_6$	✓	×	✓	×	✓	✓	✓	✓
$SF_1$	✓	✓	✓	✓	✓	✓	✓	✓
$SF_2$	×	✓	×	×	×	×	×	✓
$SF_3$	×	✓	✓	×	✓	✓	✓	✓
$SF_4$	✓	✓	✓	✓	✓	✓	✓	✓
$SF_5$	✓	✓	✓	✓	×	✓	✓	✓
$SF_6$	$\mathcal{P}$	✓	✓	✓	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	✓

Acronyms: ✓: Protected against attacks/Compliance to security and other goals, ×: Vulnerable against attacks/non compliance to security and other goals,  $ASF$ : Attacks and Security features,  $A_1$ : Replay,  $A_2$ : Impersonation,  $A_3$ : Modification of messages,  $A_4$ : DoS,  $A_5$ : MITM,  $A_6$ : Known key,  $SF_1$ : Mutual authentication,  $SF_2$ : Data privacy,  $SF_3$ : Session key security,  $SF_4$ : Message integrity,  $SF_5$ : Message freshness,  $SF_6$ : Identity anonymity,  $\mathcal{P}$ : Partially achieved,  $S_1$ : [1],  $S_2$ : [15],  $S_3$ : [24],  $S_4$ : [25],  $S_5$ : [26],  $S_6$ : [27],  $S_7$ : [28],  $S_8$ : LKE

TABLE 5.4: Computation Cost of Scheme LKE for various phases of Operation

Phase	Node	Gateway	CA	Total Cost
$P_1$	$1AS_E + 1AS_D + 1S_D + 3H_G + 1H_V$	$1AS_E + 1AS_D$	$2AS_E + 2AS_D + 1S_E + 3H_G + 1H_V$	$4AS_E + 4AS_D + 1S_E + 1S_D + 6H_G + 2H_V$
$P_2$	$1AS_E + 1AS_D + 1H_G + 1HMAC_G$	$2AS_D + 1H_G + 1HMAC_V$	-	$1AS_E + 3AS_D + 2H_G + 1HMAC_G + 1HMAC_V$
Total	$2AS_E + 2AS_D + 1S_D + 4H_G + 1H_V + 1HMAC_G$	$1AS_E + 3AS_D + 1H_G + 1HMAC_V$	$2AS_E + 2AS_D + 1S_E + 3H_G + 1H_V$	$5AS_E + 7AS_D + 1S_E + 1S_D + 8H_G + 2H_V + 1HMAC_G + 1HMAC_V$

Acronyms:  $AS_E$  - Asymmetric encryption,  $AS_D$  - Asymmetric decryption,  $H_G$  - Hash generation,  $H_V$  - Hash verification,  $S_E$  - Symmetric encryption,  $S_D$  - Symmetric decryption,  $HMAC_G$  - Hash based MAC generation,  $HMAC_V$  - Hash based MAC verification, *Numerical values* - It indicates the number of times the cryptography operation is being executed,  $P_1$ : Network Set-up Phase,  $P_2$ : Key Establishment Phase

TABLE 5.5: Computation Cost Comparison for Key Establishment Phase: Between Smart Node and Gateway

$S_C$	Node	Gateway	Total Cost
$S_1$	$7H_G + 1R_G + 4L_{XOR}$	$7H_G + 1R_G + 6L_{XOR}$	$14H_G + 2R_G + 10L_{XOR}$
$S_2$	$7H_G + 1R_G + 3L_{XOR}$	$9H_G + 5L_{XOR}$	$16H_G + 1R_G + 8L_{XOR}$
$S_3$	$5H_G + 3L_{XOR}$	$12H_G + 2R_G + 8L_{XOR}$	$17H_G + 2R_G + 11L_{XOR}$
$S_4$	$4H_G + 2S_E$	$8H_G + 4S_E + 1M_{EC}$	$12H_G + 6S_E + 1M_{EC}$
$S_5$	$5H_G + 1L_{XOR} + 2M_{EC}$	$9H_G + 1R_G + 4L_{XOR} + 1M_{EC}$	$14H_G + 1R_G + 5L_{XOR} + 3M_{EC}$
$S_6$	$6H_G + 3L_{XOR} + 1M_{EC}$	$15H_G + 10L_{XOR} + 2M_{EC}$	$21H_G + 13L_{XOR} + 3M_{EC}$
$S_7$	$7H_G + 3L_{XOR} + 2M_{EC}$	$10H_G + 1L_{XOR}$	$17H_G + 4L_{XOR} + 2M_{EC}$
$S_8$	$1AS_E + 1AS_D + 1H_G + 1HMAC_G + 1L_{XOR}$	$2AS_D + 1H_G + 1HMAC_V + 1L_{XOR}$	$1AS_E + 3AS_D + 2H_G + 1HMAC_G + 1HMAC_V + 2L_{XOR}$

Acronyms:  $AS_E$  - Asymmetric encryption,  $AS_D$  - Asymmetric decryption,  $H_G$  - Hash generation,  $HMAC_G$  - Hash based MAC generation,  $HMAC_V$  - Hash based MAC verification,  $R_G$  - Random Number Generation,  $L_{XOR}$  - Logical Operation XOR,  $S_E$  - Symmetric encryption,  $M_{EC}$  - Scalar Multiplication ECC, *Numerical values* - It indicates the number of times the cryptography operation is being executed,  $S_C$ : Schemes,  $S_1$ : [1],  $S_2$ : [15],  $S_3$ : [24],  $S_4$ : [25],  $S_5$ : [26],  $S_6$ : [27],  $S_7$ : [28],  $S_8$ : LKE.

of each bit cost  $0.72 \times 10^{-3} mJ$  and  $0.81 \times 10^{-3} mJ$  of energy, respectively. The total number of bits communicated by the resource constrained smart device during key establishment phase is 1024 bits in [1], 864 bits in [15], 1792 bits in [24], 960 bits in [25], 960 bits in [26], 912 bits in [27], 912 bits in [28], and 720 bits in LKE. Due to small overheads, the energy consumed by LKE is  $0.519 mJ$  which is much lesser than the energy consumed by other schemes. Excessive energy consumption can deplete the energy reserves of the node, i.e., reducing effective lifetime of the node [1, 15, 24, 25, 26, 27, 28]. Therefore, table 5.5 and table 5.6 proves that the LKE is considerably lightweight and energy efficient in contrast to other schemes.

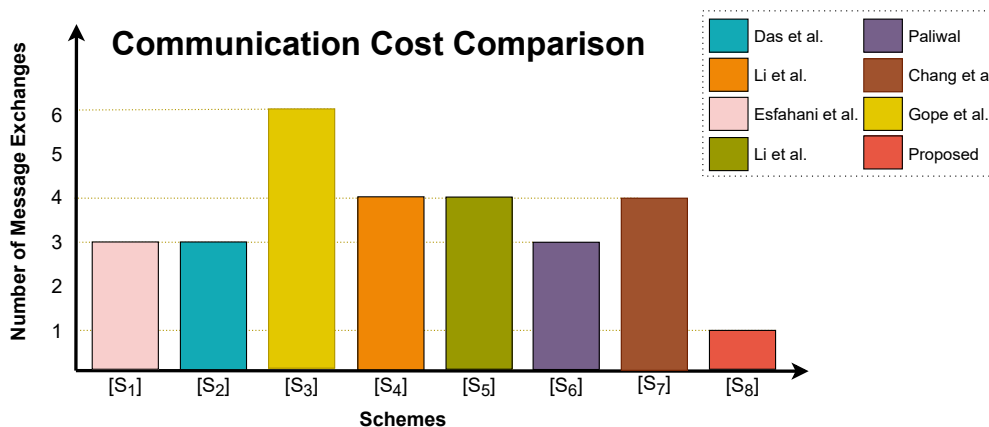


FIGURE 5.9: Communication Cost Comparison.

TABLE 5.6: Energy Cost for communication: Considering Resource Constrained Smart device (Key Establishment Phase)

	$T_X(mJ)$	$R_X(mJ)$	$T_{EC}(mJ)$
[1]	0.461	0.311	0.772
[15]	0.368	0.285	0.653
[24]	0.369	1.036	1.405
[25]	0.230	0.518	0.748
[26]	0.345	0.388	0.733
[27]	$U_D$	0.738	0.738
[28]	0.282	0.421	0.703
LKE	0.519	-	0.519

Acronyms:  $T_X$  - Transmission,  $R_X$  - Reception,  $T_{EC}$  - Total Energy Cost,  $U_D$  - Undisclosed, Hyphen (-) No consumption

Fig. 5.9 shows the total no. of messages transferred between the communicating entities throughout the bilateral authentication and key establishment period. Herein Fig. 5.9,  $S_1:S_8$  represents the schemes  $\{S_1: [1], S_2: [15], S_3: [24], S_4: [25], S_5: [26], S_6: [27], S_7: [28], S_8: \text{LKE}\}$ . It can be noticed that LKE achieves the goal in just 1 message while other schemes [1, 15, 24, 25, 26, 27, 28] exchanged minimum 3 messages to carry out the same piece of work. Excessive exchange of messages indicate more delay, overhead, and energy exhaustion [1, 15, 24, 25, 26, 27, 28]. Therefore, LKE again proves the superiority of being energy and time efficient over existing traditional techniques.



## Chapter 6

# Secure Device-to-Device Communications for 5G enabled Internet of Things applications

### Chapter Overview

6.1	Introduction . . . . .	73
6.2	Background . . . . .	75
6.3	Notations and Security Goals . . . . .	83
6.4	Proposed Protocol . . . . .	84
6.5	Performance, Security and Comparative Analysis . . . . .	90

### 6.1 Introduction

A city is considered smart when its existing infrastructure is combined with the technology to provide a better quality of life [91]. Figure 6.1 illustrates the various smart city solutions. These solutions can be used for checking the parking space availability [92], weather adaptive operation of street lights [93], detection of trash levels in bins for optimized collection routes [94], making intelligent highways with warning messages according to the weather [95, 96], etc. The physical devices used in these systems can be integrated with electronic sensors, mobile devices, and information technology in order to provide real time decisive power [97]. IoT transforms the standalone objects to cyber-physical objects that lead to smarter cities [17].

IoT provides real time access [98] and monitoring of information like warehouse stock details [99], patient health condition in hospital [100], fault intimation of machines [18], etc. Statistics indicate that by 2025, 75 billion IoT devices [101] will produce 79.4

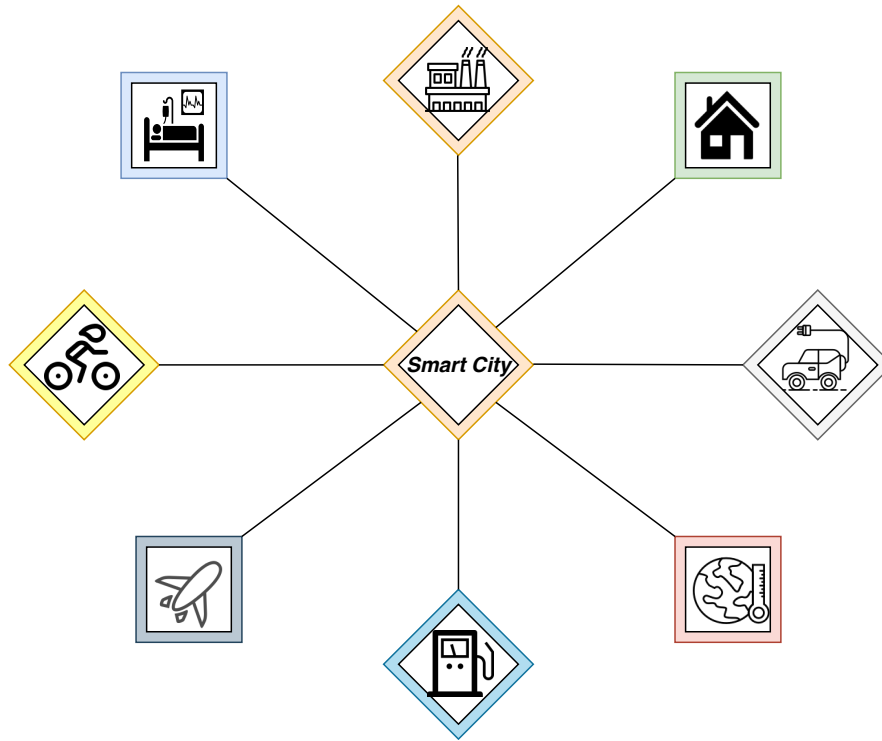


FIGURE 6.1: Smart Cities: Connecting everything, comforting lives

zettabytes (ZB) of data [102], which is gigantic. Therefore, larger bandwidth for communication is required for catering to the requirements of IoT networks [103]. Moreover, any latency in the transfer of real time data can be catastrophic [104]. Given the above needs, the fifth-generation mobile networks (5G) are the most suitable technology for IoT enabled applications of smart cities [105] as they would enable higher transmission capabilities like higher bandwidths, ultra low latency, etc. [106].

5G offers cellular as well as Device to Device (D2D) communications to provide greater bandwidth and ultra low latency required for IoT enabled applications of smart cities [107]. Specifically, the D2D communications that occur in 5G is possible through cellular services and through WiFi (wireless fidelity) Direct [108]. Although cellular coverage is widely useful but it fails to facilitate smart city applications in areas with no or partial coverage [109]. On the other hand, WiFi Direct operates without intermediate infrastructure and offers increased data rate and even lesser latency [110]. WiFi Direct can easily facilitate the close proximity communications for various smart city applications [111]. The review of the requirements of smart cities discloses that D2D services provided via WiFi Direct is the most appropriate technology to implement smart city applications [64, 112].

Although the technology is best fit for the IoT enabled smart city applications [113, 114, 115] but has some security threats [116], which if not taken care can prove fatal for the whole communication system [117]. There are many risk elements that jeopardize the D2D communications. Node impersonation, eavesdropping and message modification



are some of the major attacks [118]. Attackers can easily sneak up on the information being shared on the wireless medium [119]. Consequently, sensitive data is exposed to misuse [120]. An attacker can also act as a legitimate node when it is not one; the receiver cannot decipher the real identity of the sender [121]. The message payload can be modified by an attacker, resulting in misinterpretation and uncertain outcomes [122].

Illegitimate entities can disrupt the private communication between devices, otherwise called as Man-in-the-middle (MITM) attack [123]. Hence, it becomes very crucial to verify the legitimacy of the sender and receiver which is why mutual authentication is required [124, 125, 126]. The communication of the legitimate entities can also be interrupted by initiating a flood attack wherein the target device is bombarded with discovery requests, making it difficult to communicate with the right device on time [127, 128]. Strong discovery protocols are the need of the hour when it comes to D2D communications even though it is a very promising technology.

Clearly, the D2D communications needs protection from these security hazards (e.g., MITM, DoS, etc.) in order to be useful for the data transmission. The security measures must prevent any unauthorized identity to communicate with the authorized identity. There must be some mechanism through which the two genuine parties should exchange keys with each other securely without being captured by an attacker. As a solution to the aforementioned issues, we propose a mutual authentication and pairwise key establishment method that can certify the devices before any actual exchange of information occurs.

## 6.2 Background

The direct transmission of data between two devices through radio signals without having to go through a base station or cellular network in D2D communications brings unprecedented benefits [129]. Bypassing the infrastructure to enable communication reduces cost, improves efficiency and offers low latency [130]. D2D allows spontaneous sharing and exchange of information that speeds up the utility of smart devices enabled by the IoT [131]. The close proximity of devices that enable D2D is making it one of the most essential technologies in applications spread across smart cities like smart parking, area surveillance, waste management, etc. [109]. The assistance provided by the cloud servers in various real time applications [127] is enhanced by the presence of the communicating devices in each other's vicinity as it enables faster data offloading [132].

In small spaces, D2D communications using WiFi Direct dispenses a faster data rate that saves data exchange and computation costs [133]. D2D using WiFi Direct, which is an extension of the traditional WiFi, makes the provision of a scalable framework by

omitting the need of Access Points (APs) [134]. The devices do not require any AP for communication rather the devices themselves act as AP to form peer-to-peer groups [135]. The device serving as AP is also recognised as group owner (GO) [136]. GO device can send the GO signal as a relay for communication amongst all the devices in the group [137]. There is no loss of quality or energy in this method of communication as compared to the traditional data exchange through WiFi adhoc and IEEE 802.11z [138].

An open network is most susceptible to blockage and this is the biggest challenge in D2D communications [139]. Most D2D applications consist of sensitive data and any tampering with it can cause irreversible damage [140]. The security loopholes of open access channels allow the cyber attackers to easily create fake setup links and cause network disruptions [141]. Since the open medium is the leverage of the D2D communication, it becomes more obvious for the devices to protect themselves [142, 143]. Security offered by the WiFi Protection Setup (WPS) is not robust enough for D2D communications [144, 145]. This is why smart security protocols are the need of the hour considering the rising deployment of D2D using 5G in smart city applications [146, 147] directly concerning human safety, healthwise and livingwise.

Security challenges are a rising area of research in D2D technologies but there is still a long way to go considering the available methods of protection from attacks like MITM and DoS [128]. When the responsibility of security is deemed to lie with the devices themselves, a trusted authentication mechanism between two devices can prevent the security risks [148]. This trusted mechanism will allow the devices to authenticate each other mutually followed by pairwise key establishment to prevent from prior mentioned attacks (MITM, DoS, etc.) [29, 149, 150, 151]. This research introduces a strong mutual authentication and key agreement protocol for D2D communications over WiFi direct powered by IoT and driven by 5G.

### **6.2.1 WiFi Direct Overview**

The WiFi Direct protocol enables the devices to establish a wireless connection without any mediator [152]. The peer-to-peer (P2P) communication without the access point (or a router) is initiated by two or more peer devices which first discover each other and then form a P2P group [153] as shown in figure 6.2. The devices set up a communication link before the full information exchange begins. Next step is the negotiation for the selection of a P2P GO, called the handshake process by sending out an intent value [154]. Each device in the P2P group sends out an intent value. The device possessing a greater intent value wins the negotiation and becomes the P2P GO which then acts as the AP as indicated in figure 6.2. This is followed by the beginning of a security process using WPS by the P2P GO [155]. Subsequently, IP addresses for both the devices are generated by the GO using Dynamic Host Configuration Protocol (DHCP) [156]. Thus,

the D2D connection is established successfully between the devices enabled with WiFi Direct.

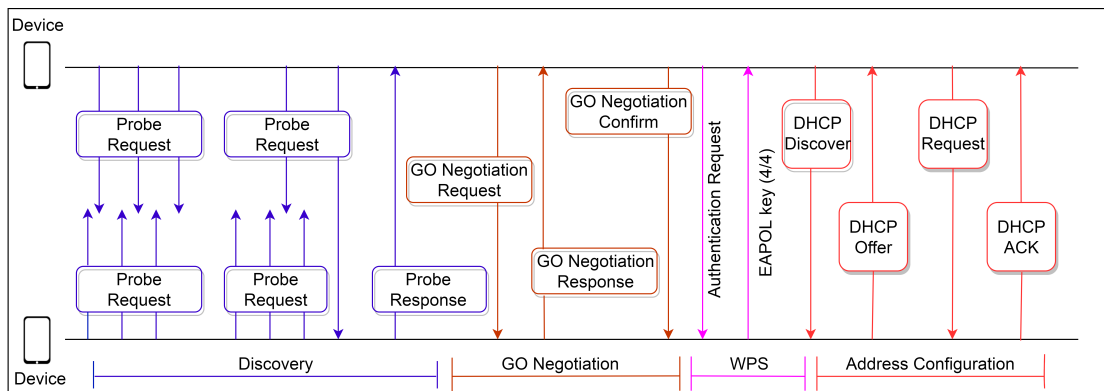


FIGURE 6.2: WiFi Direct Protocol

There are several applications of D2D communications, but two majorly witnessed applications are discussed in this section. Through WiFi Direct, P2P GO can share its internet services obtained from Base station (Cellular) with other devices (clients) of the P2P group [157]. The scenario has been illustrated in figure 6.3. Another scenario is presented in figure 6.4 where numerous devices form a localized adhoc network [158]. Devices are performing various tasks (e.g., image uploading, printing, file sharing, etc.) using WiFi Direct services at no extra cost. This feature of WiFi Direct is particularly helpful during the absence of cellular services or an AP.

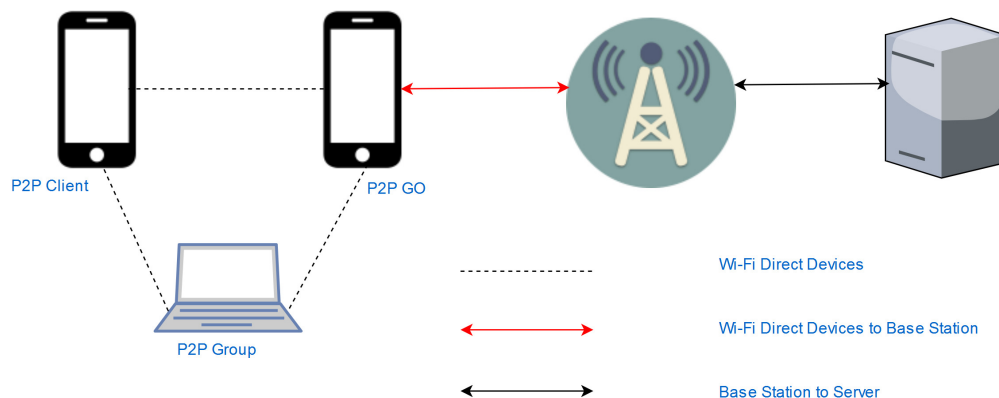


FIGURE 6.3: WiFi Direct application scenario: one device shares its cellular connection.

### 6.2.2 Short-Authentication-String (SAS) based Key Agreement Protocol

The authors of SAS based key agreement protocol claims to introduce a novel mutual authentication and key establishment scheme for D2D communications; the scheme operates with minimum human interaction and limited cryptography operations. SAS protocol employs a commitment scheme where the secret value is enclosed behind the

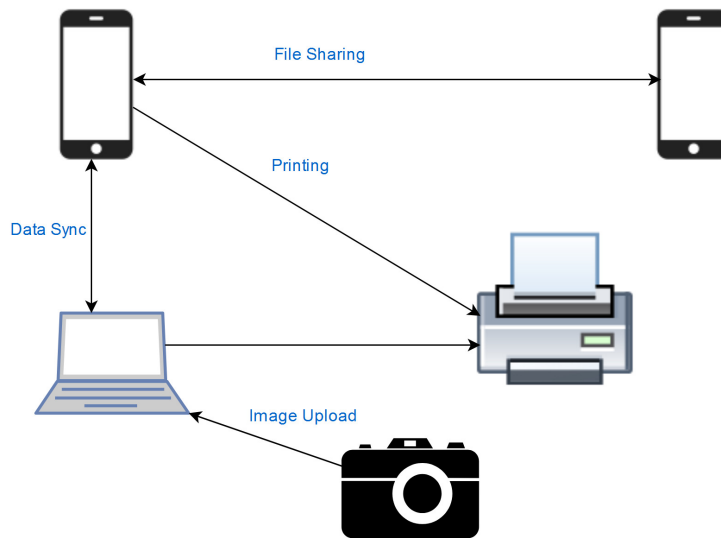


FIGURE 6.4: WiFi Direct application scenario: multiple devices form an ad hoc network

commit/open pair. The commit/open pair is designed in such a way that a commitment value alone does not disclose any information about the secret value. As demonstrated in figure 6.5, the commit/open pair intends to provide secure mutual authentication. SAS makes use of traditional diffie hellman key exchange protocol as the basic underlying mechanism to establish the secret key. Apart from diffie hellman parameters  $g^x$  and  $g^y$ , SAS protocol generates device identifiers  $ID_X$  and  $ID_Y$  along with nonces,  $N_X$  and  $N_Y$ . Device employing SAS protocol generates the secret value,  $m_X = ID_X \parallel g^x \parallel N_X$ , and afterwards compute a commit/open pair,  $m_X = (c, d)$  wherein 'c' and 'd' represents commitment and disclose value, respectively [29].

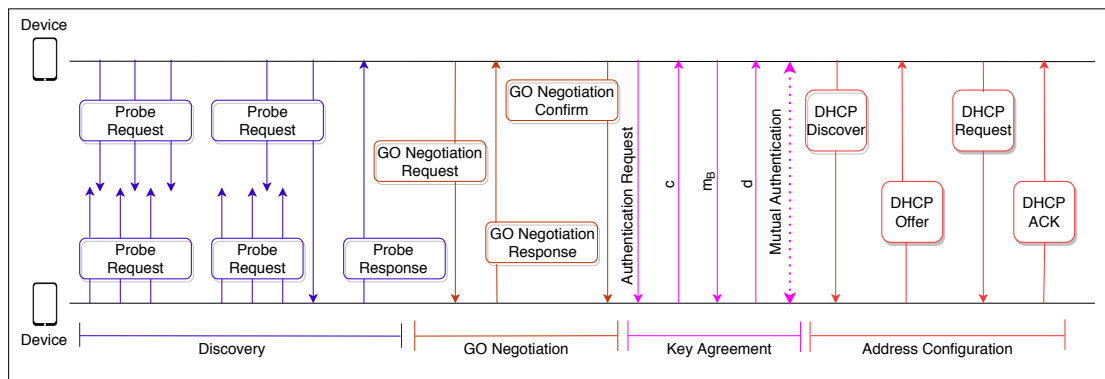


FIGURE 6.5: SAS based key agreement protocol

Let us consider the following commitment scheme wherein  $F$  is a cryptographic hash function:

- **Commit:** Given  $w$ , randomly choose  $p \leftarrow \{0, 1\}^n$ , compute  $c = F(w, p)$ .
- **Open:** Let  $d = (w, p)$ . Output  $w$  if  $c = F(w, p)$ .

In simple terms, the receiving device accepts the secret value  $w$  if  $F(d) \equiv c$  i.e., earlier received commit value.

Consider a scenario, where Alice and Bob want to communicate using D2D communications via WiFi Direct. A couple of messages are exchanged between them to verify the identity of each other followed by common secret key generation. Initially Alice shares a commitment value  $c$  with Bob as shown in figure 6.5. In response, Bob exchanges  $m_B$  with Alice. Upon reception of  $m_B$ , Alice shares the other value of the pair (open value,  $d$ ) to allow Bob to extract the secret value,  $m_A$ . Subsequently, the devices verify the identity of each other by computing the SAS,  $S_A = N_A \oplus N_B$  and  $S_B = N_A \oplus N_B$ . Post successful verification, both the devices compute  $K = g^{ab} \text{ mod } p$  to generate the secret key. In another instance, if the authentication string does not match ( $S_A \neq S_B$ ), the key establishment process is terminated by the devices assuming the presence of MITM attack.

### 6.2.3 Limitations of WiFi WPS

The WiFi Direct is protected through the WiFi Protected Setup (WPS) that uses a secure PIN or push button configuration. WPS attempts to provide security by letting the WiFi Direct GO act as the registrar to send network security credentials to the WiFi Direct clients. Afterward, a set of security keys are generated by the WiFi Protected Access (WPA) protocol to secure the communications. However, during the investigation by the authors in [145], it is found that PIN can be easily discovered by the attacker within a small fraction of time, resulting in the revealing of secret keys and comprising overall communications. Besides, the attacker exploits the vulnerabilities of the push button configuration to get unauthorized access to the registrar or client, hence putting the entire ad-hoc network at high risk [159]. Since the attacker obtained access, messages can be intercepted and modified that can lead to irreversible damage to crucial real-life assets, etc.

### 6.2.4 Cryptanalysis of SAS based Key Agreement Protocol

We have critically analyzed the working of the SAS protocol and observed a few vulnerabilities. The attacker can compromise the whole network by exploiting these vulnerabilities. The vulnerabilities and its possible consequences are illustrated in figure 6.6 and summarised as follows:

1. Short authentication strings,  $S_A$  and  $S_B$  are said to have been compared over a trusted channel. Ideally, wireless channels are always considered vulnerable and subjected to attacks, primarily *eavesdropping*. Lack of trusted wireless channels

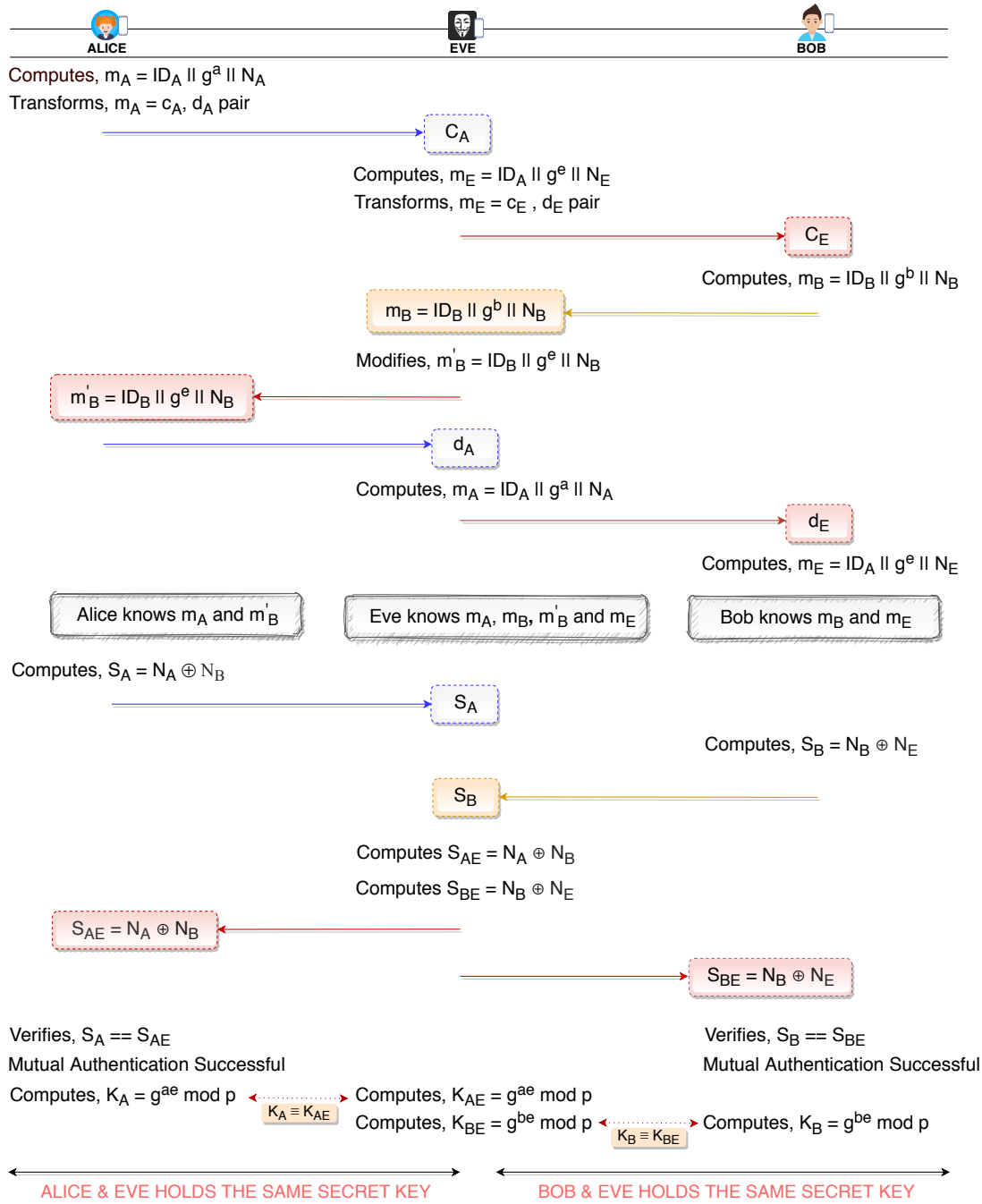


FIGURE 6.6: SAS based key agreement scheme is subjected to MITM attack

is the biggest motivation behind the discovery of well known diffie hellman and ECC diffie hellman key exchange protocols. The security strength of the SAS based key agreement scheme lies in the authentication string ( $S_A, S_B$ ) which is shared in plain text over the vulnerable wireless channel. Eavesdropping of the authentication string can allow the attacker to successfully conduct the MITM attack, thus compromising the communications.

2. The SAS based key agreement protocol in itself does not mention any remark on the requirement of time synchronization between devices. There is no lifetime

(*LT*) or timestamp (*TS*) associated with the secret values ( $m_A, m_B$ ) and SAS ( $S_A, S_B$ ), thus enabling the attacker to misuse the information. Upon reception of the SAS in plain text, the attacker device can read and counterfeit the SAS. Further, the attacker can share this counterfeited SAS with a legitimate node to gain trust and enable the legitimate node to begin computations for secret key establishment.

3. Consider a scenario where Alice begins the communication with Bob but somehow Eve (attacker) intervenes. Alice initially computes,  $m_A = ID_A \parallel g^a \parallel N_A$  and transforms it into  $c_A, d_A$  pair. Eve pretends to Alice as Bob and receives  $c_A$ . But Eve cannot determine  $m_A$  unless provided with  $d_A$ . Similarly, Eve pretends to Bob as Alice, computes  $m_E = ID_A \parallel g^e \parallel N_E$ , and sends the commit value  $c_E$  to Bob instead of sending  $c_A$ . Bob in reply, returns  $m_B = ID_B \parallel g^b \parallel N_B$  to Eve (considering her as Alice). Eve modifies the  $m_B$  to  $m_B' (= ID_B \parallel g^e \parallel N_B)$  and sends it further to Alice. Afterward, Alice shares the open value,  $d_A$  with Eve which enables Eve to retrieve  $m_A (= ID_A \parallel g^a \parallel N_A)$ . Instead of sending  $d_A$ , Eve send  $d_E$  to Bob, thus allowing Bob to retrieve  $m_E (= ID_A \parallel g^e \parallel N_E)$  from  $c_E, d_E$  pair. Currently, Alice knows  $m_A, m_B'$ , Bob knows  $m_B, m_E$  whereas Eve knows  $m_A, m_B, m_B'$ , and  $m_E$ .

Since the secret values,  $m_E$  and  $m_B'$  are exchanged, Alice and Bob begin the computation of a short authentication string,  $S_A = N_A \oplus N_B$  and  $S_B = N_B \oplus N_E$ . The computed authentication strings ( $S_A, S_B$ ) are supposed to be shared by Alice and Bob with each other for mutual authentication. But the Eve exploits the vulnerability of wireless channels and intercepts the short authentication strings shared by Alice,  $S_A = N_A \oplus N_B$ , and Bob,  $S_B = N_B \oplus N_E$ . Since the authentication strings are in readable format and not tied to any timestamp/lifetime, therefore Eve gets enough time to manipulate the information and misuse it. Eve counterfeits the SAS of Alice and Bob and then share the counterfeited value,  $S_{AE} (= N_A \oplus N_B)$  and  $S_{BE} (= N_B \oplus N_E)$  with Alice and Bob, respectively.

In other instance, Eve instead of counterfeiting can also produce  $S_A$  and  $S_B$  as Eve already has the knowledge of  $m_A, m_B$ , and  $m_E$ . Alice and Bob verify  $S_A \stackrel{?}{=} S_{AE}$  and  $S_B \stackrel{?}{=} S_{BE}$ , respectively, and concludes successful mutual authentication without being aware of fraud by Eve. Post successful fraudulent mutual authentication, Alice produces a secret key,  $K_A = g^{ae} \text{ mod } p$ , Bob produces a secret key,  $K_B = g^{be} \text{ mod } p$ , and Eve produces two secret keys,  $K_{AE} = g^{ae} \text{ mod } p$  and  $K_{BE} = g^{be} \text{ mod } p$ . As a result, Alice and Eve hold a common key, and Bob and Eve hold a common key. As Alice and Bob are not aware about the presence of Eve, therefore they continue to use the fraudulent secret key with the belief of securing the communications, however in reality they were getting cheated by Eve. Investigations and analysis revealed that SAS based key agreement protocol is subjected to MITM attack and also not secure enough to be used for D2D communications.

4. SAS based key agreement protocol is partially protected against the DoS attack. There is no process to identify and prevent the DoS attack in the discovery phase. As a result, the devices waste their precious resources in accepting and processing bogus requests (e.g., wasting bandwidth and computing power for processing SAS of fraudulent requests, etc.).
5. Though SAS based key agreement protocol claims to verify the authentication string ( $S_A, S_B$ ) for mutual authentication purposes but lacks to verify the establishment of identical secret keys ( $K_A, K_B$ ) at both entities.

### 6.2.5 Research Contribution

This research work proposes a security protocol for the D2D communications to protect the sensitive applications of smart cities. The important contributions of the research work are summarised as follows:

- We propose a lightweight and strong mutual authentication and key agreement protocol for D2D communications.
- To achieve the robustness and lightwightness, the protocol applies commit/open pair, symmetric-key cryptography, message authentication code (MAC), diffie hellman key exchange (DHKE) algorithm, nonce, and other lightweight cryptography primitives such as bit-wise XOR, etc.
- The proposed protocol enables the WiFi Direct devices to detect the occurrence of DoS attacks through intelligent filtering in the discovery phase; thus preserving precious resources of the devices e.g., computation power.
- The strength of the proposed protocol is verified through formal (*Burrows-Abadi-Needham logic*) security analysis. Analysis revealed that the proposed protocol exhibits essential security properties, like mutual authentication, message freshness, secret key confidentiality, etc. In addition, the protocol is also protected from MITM, DoS, and replay attacks, and so forth.
- The proposed protocol provides early detection to DoS attack and more security against MITM attack in comparison to state-of-the-art.
- The protocol verifies the establishment of identical key at both devices to avoid erroneous use of secret keys for applications like ciphering, etc.



TABLE 6.1: Notations and Denotations

Notation	Denotation
$ID_A, ID_B$	Identity of device A and B
$g, p$	Generator value, large prime number
$a, b$	Private value of device A and B
$N_A, N_B$	Random nonce generated by device A and B
$m_A, m_B$	Message formed by device A and B
$S_A, S_B$	Short authentication string of device A and B
$K_{AB}, K_{BA}$	Common secret key established between device A and B
TS, LT	Timestamp and Lifetime
$Auth_A, Auth_B$	Mutual authenticator of device A and B
$c, d, MAC$	Commit value, open value, and message authentication code

## 6.3 Notations and Security Goals

### 6.3.1 Notations

The notations of the parameters used in the proposed protocol are presented in Table 6.1.

### 6.3.2 Security Goals

The proposed protocol attains the following significant security goals. Note that the security goals are adopted from [14, 33, 53].

1. **Mutual Authentication:** D2D communications are vulnerable due to the open access nature of communication. The IoT enabled smart city applications produce enormous amounts of sensitive data. The access to this data must be restricted to legitimate nodes only to avoid misuse of precious information. The D2D communications can be protected by performing mutual authentication to verify the legitimacy of nodes before initializing the session.
2. **Secret Key Establishment:** IoT devices deployed in smart cities generate pivotal information. The information is exchanged with the cloud and other end users/devices through untrusted wireless channels. Nodes with malicious intentions can easily eavesdrop the information and use it for exploiting the services of legitimate nodes. Therefore, D2D devices must exchange a secret key through a secure process to enable ciphering, etc.

3. **Message Integrity:** Modifications in the message can destroy the essence of the information, thus leaving behind the bogus content of no use. Therefore, D2D devices must verify the integrity of the messages before using the information for further processing.
4. **Message Freshness:** Adversary can capture the messages due to the inherently vulnerable nature of wireless medium. The captured stale messages can be replayed by the adversary to get privileged access to information. Thus, D2D communications must ensure the use of freshness components in the message exchanges to protect from adversarial threats (e.g., replay).
5. **Defense against prominent attacks:** IoT networks are susceptible to threats due to vulnerabilities in security protocols. The impact of attack can be mild or severe and may lead to temporary or permanent suspension of smart city applications. Therefore, security protocols must be robust enough to combat against significant attacks like DoS, MITM, etc.

## 6.4 Proposed Protocol

D2D communications via WiFi Direct is accomplished through four different phases, i.e., *discovery*, *GO negotiation*, *mutual authentication* and *key agreement*, followed by *address configuration*. The most sensitive phases where the adversary prefers to attack are discovery and key agreement phases. The devices operating in the WiFi Direct (WFD) environment always remain partially active to respond to the probe requests of the nearby devices. However, the adversaries exploit the discovery mechanism by sending bogus probe requests to the legitimate devices, thus resulting in draining of precious resources such as battery, computing power, storage space, and so forth. Consequently, the victimised device fails to deliver the services to the legitimate devices i.e., Denial of Service. Besides discovery phase, key agreement phase is also victimised by the adversaries because interception of key exchange messages through MITM attack could compromise the future correspondence between devices in a D2D network.

To counter the threat of DoS attack, the proposed scheme employs an intelligent filtering mechanism at discovery phase. In addition, a robust and lightweight mutual authentication and key establishment (MAKE) scheme at key agreement phase is invoked to protect the WFD enabled devices from MITM attack. The intelligent filtering mechanism sweeps out the malicious requests whereas the MAKE scheme does not allow the impersonated devices to initiate a session with the legitimate devices. Therefore, the proposed protocol presented in figure 6.7 can protect the D2D communications from DoS and MITM attacks and it can be considered as a potential solution for securing various smart city applications.

TABLE 6.2: Filtering of malicious requests at discovery phase to protect from DoS attack

$S_R$	Filtering of malicious requests		
	MAC	RSSI	Action Taken
$S_1$	$MAC^1$	$RSSI^1$	Resources granted
$S_2$	$MAC^2$	$RSSI^1$	Resources denied, forged (MAC) credentials found
$S_3$	$MAC^1$	$RSSI^1$	Resources denied, duplicate (MAC) credentials found
$S_4$	$MAC^2$	$RSSI^2$	Resources granted. RSSI helps to identify the request from the new terminal.

**Acronyms:** MAC: Media Access Control address, RSSI: Received Signal Strength Indicator,  $S_R$ : Scenarios,  $S_1$ : Device registration,  $S_2$ : Forgery of MAC address,  $S_3$ : Bogus request from same device,  $S_4$ : Request from new neighboring device

Table 6.2 discusses the various instances where the protocol applies its intelligent algorithm to differentiate between malicious and legitimate probe requests. Note that two parameters namely Media Access Control (MAC) address and Received Signal Strength Indicator (RSSI) have been used to determine the authenticity of the probe request. MAC address alias physical address of the device is considered unique globally, therefore it has been used to discriminate between WFD enabled devices [38, 160, 161] whereas RSSI has been used to determine the estimated location of the WFD enabled devices [162, 163, 164]. The mechanism used in the discovery phase is more effective in identifying malicious requests originated from stationary devices. Table 6.2 and Algorithm 1 demonstrates the set of rules invoked to effectively discriminate between legitimate and malicious probe requests at discovery phase.

Table 6.2 discusses the set of actions performed by the device upon reception of the probe request. Assume two WFD enabled devices want to communicate with each other. The devices reveal their interest by sending the probe request messages (consisting of MAC address and other details) to each other. Note that the terms *node* and *device* are interchangeably used and represent the same meaning. Let us review the set of actions performed by a node for filtering malicious requests. Device 1 sends a probe request to Device 2 and in turn seeks its probe response to lead further communication. However, unlike traditional security protocols, Device 2 examines the credentials before issuing the probe response. Device 2 particularly verifies two factors i.e., MAC address and RSSI before issuing probe response to the Device 1.

Few Assumptions: The nodes are installed distributively for performing various tasks in smart cities (e.g., smart bins, smart traffic lights, rainwater harvesting, etc.). The devices are installed permanently (stationary). MAC address and RSSI can be used to differentiate between D2D devices. The factors which affect the RSSI (e.g., weather conditions, signal power to antenna unit, and so forth) are considered constant.

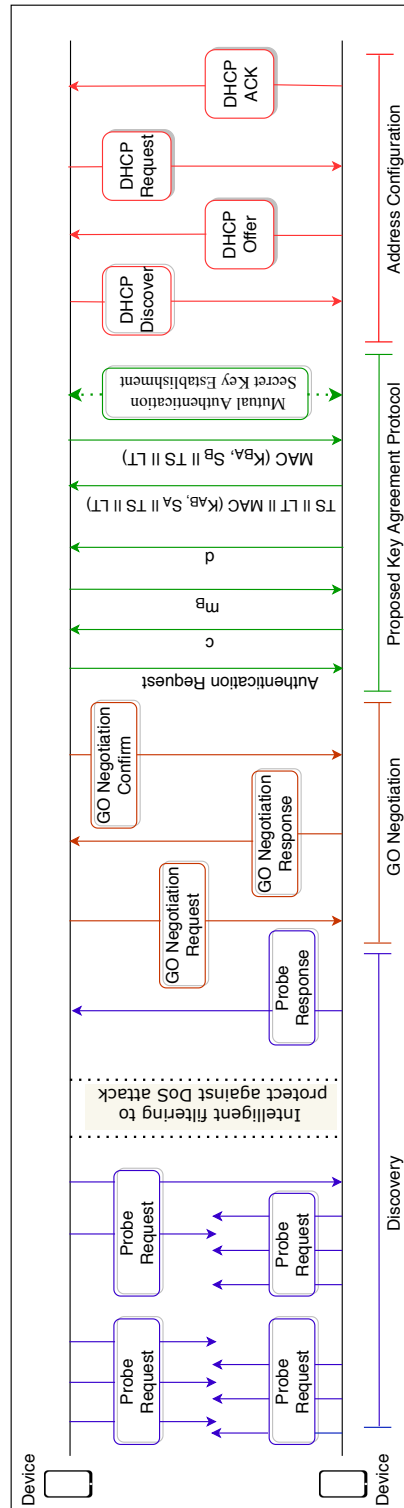


FIGURE 6.7: Proposed security protocol for WiFi Direct based smart city applications

Device 2 retrieves the MAC address and RSSI from the received probe request message that arrived from device 1. Afterward, the retrieved information, i.e., MAC address and RSSI is compared with the information stored in the memory of device 2. Note that  $MAC^1$  indicates the MAC address of device 1 and  $RSSI^1$  indicates the strength

of the device 1 signal at the receiving device,  $D_2$ . Since no similar request has been facilitated by the  $D_2$  earlier, therefore the device 2 considers the request as fresh and grants the response to continue with GO negotiation phase. It is evident from Table 6.2 and Algorithm 1 that in the second instance, the attacker device sends the probe request with a forged MAC address ( $MAC^2$ ) to the legitimate node,  $D_2$ .  $D_2$  compares the  $MAC^2$  and  $RSSI^1$  with its database entries. Upon verification,  $D_2$  finds that a request has already been received earlier with the similar RSSI value, therefore  $D_2$  aborts the communication considering it as a malicious request. In another instance  $S_3$ ,  $D_2$  receives the request with a counterfeited MAC address and similar RSSI value, hence concludes it as bogus request with malicious intentions. Consequently,  $D_2$  prohibits the processing of malicious requests, thus resulting in preservation of resources for legitimate nodes. It is evident from the  $S_4$  that the proposed protocol easily determines the reception of a new probe request from a legitimate WFD enabled device by examining the MAC address and RSSI value.

It can be observed from algorithm 1 that the alterations proposed in the discovery phase of the conventional WiFi Direct protocol can partially protect the D2D communications from DoS attacks.

---

**Algorithm 1** Filtration of malicious requests at discovery phase
 

---

**Input:** MAC address, RSSI

**Output:** Resources, DoS

```

1:  $D_1$  send_request  $D_2$                                 ▷  $D_1$  send probe request to  $D_2$ 
2:  $D_2$  receive_request  $D_1$                                ▷  $D_2$  stores  $MAC^1, RSSI^1$  of  $D_1$ 
3: if  $MAC^2 == MAC^1$  then                                ▷  $MAC^2, RSSI^2 \in$  new probe request
4:   if  $RSSI^2 == RSSI^1$  then                             ▷  $D_2$  compares with database
5:     DoS()                                               ▷ Denial of service
6:   end if
7: else
8:   if  $MAC^2 != MAC^1$  then
9:     if  $RSSI^2 != RSSI^1$  then
10:      service_provide()                                  ▷ Resources granted
11:    end if
12:  else
13:    DoS()                                               ▷ Denial of service
14:  end if
15: end if

```

---

The proposed protocol intends to fill the shortcomings of the SAS based key agreement protocol. The proposed protocol integrates timestamp (TS) and lifetime (LT) to prevent the misuse of  $S_A$ ,  $S_B$ ,  $m_A$ , and  $m_B$  by attacker for gaining trust of legitimate nodes. Besides, the proposed protocol also protects the D2D communications from MITM attack by prohibiting the transmission of short authentication strings in plain text. Moreover, the protocol verifies the generation of identical secret keys on both devices to avoid

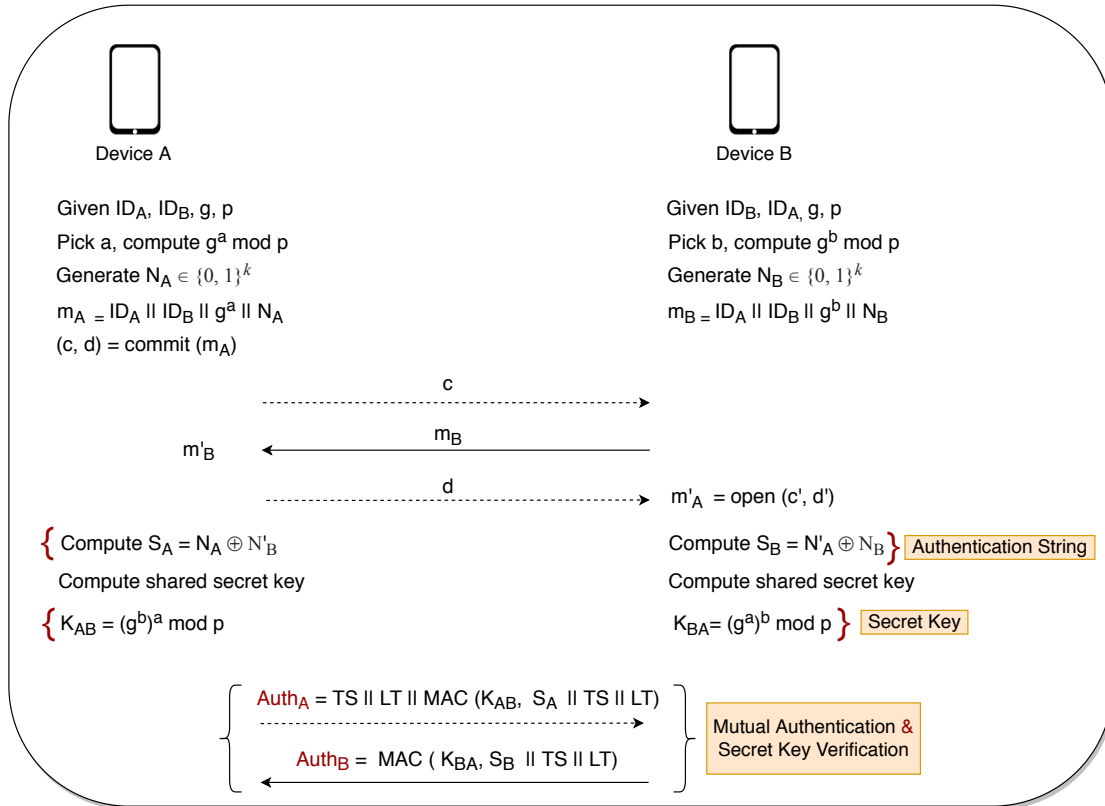


FIGURE 6.8: Mutual Authentication and Secret Key Establishment between Device A and B

use of erroneous keys. Note that errors can be caused due to many reasons: analog to digital conversion process, wireless channel fading, modifications by attackers, etc.

The working of the proposed mutual authentication and key establishment scheme is illustrated in figure 6.8. Consider a scenario where Alice (Device A) wants to communicate with Bob (Device B). Besides, diffie hellman parameters ( $g$  and  $p$ ), Alice and Bob generates a random nonce,  $N_A$  and  $N_B$ , respectively, to add a freshness component in the message. A private value is also chosen by Alice ' $a$ ' and Bob ' $b$ ' to be applied to diffie hellman parameters,  $g^a$  and  $g^b$ . After the computations, Alice and Bob concatenates the device identifiers ( $ID_A, ID_B$ ), diffie hellman parameter ( $g^a, g^b$ ), and the random nonce ( $N_A, N_B$ ) to form the secret value,  $m_A (= ID_A \parallel ID_B \parallel g^a \parallel N_A)$  and  $m_B (= ID_A \parallel ID_B \parallel g^b \parallel N_B)$  respectively.

But instead of sending  $m_A$  to Bob, Alice breaks the information into a commit/open pair. The secret value ( $m_A$ ) is divided into two separate components in such a way that commitment value ( $c$ ) does not disclose anything about hidden secret value unless open value ( $d$ ) is not applied. Therefore, Alice shares commit value  $c$  with Bob; Bob in turn shares the secret value  $m_B$  with Alice. Upon reception of  $m_B$ , Alice sends the open value  $d$  to allow Bob retrieve  $m_A$ .

After the exchange of messages, both Alice and Bob begin the computation of short authentication strings for mutual authentication. Alice computes  $S_A = N_A \oplus N_B$  whereas Bob computes  $S_B = N_A \oplus N_B$ . After generating the authentication strings, Alice and Bob computes the secret key; Alice,  $K_{AB} = (g^b)^a \bmod p$  and Bob,  $K_{BA} = (g^a)^b \bmod p$ .

Alice and Bob have formed the secret keys but it remains unused unless the devices verify the authenticity of each other. Alice generates the TS with LT to avoid replay attacks. It is recommended to keep the lifetime (LT) value equivalent to the propagation time of message from source to destination, to prohibit the attacker from capturing, modifying and replaying [165]. Alice computes the message authentication code of  $S_A$ ,  $TS$ , and  $LT$  with the secret key,  $K_{AB}$ . The message authentication code transforms the information to a non-interpretable form, thereby not permitting the attacker to interpret, alter, and misuse. Similarly, Bob also concatenates the  $S_B$  with received  $TS$  and  $LT$  to compute the message authentication code with secret key,  $K_{BA}$ . Lastly, both the devices exchange and compare the message authentication code, if it renders the same value, it indicates successful mutual authentication and key establishment.

---

**Algorithm 2** Mutual Authentication and Secret Key Establishment
 

---

**Input:** Device identities captured during discovery phase:  $ID_A, ID_B$ ; Global Public Element:  $g, p$ ; Every device choose a secret value ( $s$ ) and compute  $g^s \bmod p$   $\{g^a \bmod p, g^b \bmod p\}$ ; Computation of Nonce  $N_A$  &  $N_B$  at devices A & B respectively; Message Authentication Code (MAC); Timestamp (TS); Lifetime (LT)

**Output:** Successful Mutual Authentication and Key Establishment (MAKE), Unsuccessful Mutual Authentication and Key Establishment (MAKE)

```

1:  $D_B \leftarrow c$                                 ▷ Device A send a commit value to Device B
2:  $D_A \leftarrow m_B$                                 ▷ Device B sends the  $m_B$  to device A
3:  $D_B \leftarrow d$                                 ▷ Device A sends a open value to Device B
4:  $S_A = N_A \oplus N'_B$                             ▷ Device A compute authentication string
5:  $S_B = N'_A \oplus N_B$                             ▷ Device B compute authentication string
6:  $K_{AB} = (g^b)^a \bmod p$                         ▷ Device A compute secret key
7:  $K_{BA} = (g^a)^b \bmod p$                         ▷ Device B compute secret key
8:  $D_B \leftarrow Auth_A = TS || LT || MAC_A(K_{AB}, S_A || TS || LT)$ 
9:  $D_A \leftarrow Auth_B = MAC_B(K_{BA}, S_B || TS || LT)$ 
10: if  $MAC_A == MAC_B$  then                                ▷  $D_A$  and  $D_B$  verifies
11:   Successful MAKE
12: else
13:   Unsuccessful MAKE
14: end if

```

---

The protocol only makes use of lightweight cryptography operations like message authentication code and bit-wise XOR, therefore it can be considered as the best fit for resource constrained devices and applications of smart cities.

## 6.5 Performance, Security and Comparative Analysis

### 6.5.1 Performance Analysis

The proposed scheme aims to protect the D2D communications from adversarial threats, especially DoS and MITM attacks. The performance of the scheme has been analyzed through simulations executed on a platform with specifications as: intel core *i3-2310M* processor with operating frequency of *2.10 GHz*, 64-bit operating system, *2 GB DDR3* RAM, cache memory *3 MB*, memory speed *1333 MT/s*, and supported wireless LAN standards *IEEE 802.11 b/g/n*. As mentioned earlier, the presented scheme is more effective in protecting against DoS attacks triggered by malicious stationary nodes, therefore the current performance investigation has been confined to stationary nodes only, whereas evaluation of the scheme under the influence of malicious mobile nodes is the future scope of the research.

The simulation results of the proposed scheme have been demonstrated in this section. Initially the interested device sends a probe request to the target device during the discovery phase as shown in figure 6.9. Since the probe request is fresh, the device 2 allocates the resources to the source node for continuing further.

<pre>Device 1 Credential MAC&lt;1&gt;: DE-56-0A-DC-E6-88 Sending request to Device2 for allocation of resources</pre>	<pre>Received Request from nearby Device Credentials extracted from received request MAC&lt;1&gt;: DE-56-0A-DC-E6-88 RSSI&lt;1&gt; : -80dBm Device Credentials compared with Database &lt;Database Empty&gt; &lt;Resource Allocated&gt;</pre>
---	---

FIGURE 6.9: Initiation of communication between Device 1 and 2

The attacker can deploy its malicious node in the network to trigger DoS attacks. Malicious node executes the DoS attack by flooding the victim node with bogus requests. Consequently, the legitimate node exhausts all its precious resources (e.g., system crash) in processing the fake requests, thus preventing the node from processing the valid requests [166, 167, 168, 169]. Figure 6.10 illustrates how the proposed scheme empowers the legitimate node to identify the probe requests with forged credentials. Upon verification of the probe request, it is found that the request received has arrived from the same node (Device 1) with intentions to trigger DoS attack in the network. Therefore, the legitimate node (Device 2) refrains from processing the request, thus preserving resources to serve genuine requests in future.

As illustrated in figure 6.11, the attacker node tries to re-attempt the DoS attack by sending the bogus request, however the device 2 examines the request and finds out that a request with similar credentials has already been processed, thereby discarding the request.



<pre>Device 1 Credential with forged MAC MAC&lt;2&gt;: AG-48-IG-L9-46-78 Sending request to Device2 for allocation of resources</pre>	<pre>Received Request from nearby Device Credentials extracted from received request MAC&lt;2&gt; : AG-48-IG-L9-46-78 RSSI&lt;1&gt; : -80dBm Device Credentials compared with Database &lt;Similar Credential exist in Database&gt; &lt;Denial of Service&gt;</pre>
---	---

FIGURE 6.10: Case of forged MAC address

<pre>Device 1 MAC&lt;1&gt;: DE-56-0A-DC-E6-88 Sending Bogus request to Device2 for allocation of resources</pre>	<pre>Device 2 Received Request from nearby Device Credentials extracted from received request MAC&lt;1&gt; : DE-56-0A-DC-E6-88 RSSI&lt;1&gt; : -80dBm Device Credentials compared with Database &lt;Similar Credentials exist in Database&gt; &lt;Denial of Service&gt;</pre>
--	---

FIGURE 6.11: Bogus request with same credentials

When another legitimate device in the vicinity reveals its interest through a probe request, the device 2 follows the same procedure and verifies the credentials. It is evident from the figure 6.12 that the new device is allocated with the resources because its credentials (MAC address and RSSI) are different from the stored credentials in the database. Since the scheme is tested for stationary nodes without considering the factors affecting RSSI, it can be inferred that the proposed scheme can partially protect the D2D communications from DoS attacks.

<pre>Device 3 Credential MAC&lt;3&gt;: CD-26-AE-29-BF-18 Sending request to Device2 for allocation of resources</pre>	<pre>Device 2 Received Request from nearby Device Credentials extracted from received request MAC&lt;3&gt;: CD-26-AE-29-BF-18 RSSI&lt;3&gt; : -78dBm Device Credentials compared with Database &lt;Dissimilar Credentials&gt;&lt;new neighbour Device&gt; &lt;Resource Allocated&gt;</pre>
---	--

FIGURE 6.12: Case of new neighbouring device

Since the smart city applications contain precious information, therefore devices must verify the authenticity of each other followed by secret key establishment to secure the communication from adversarial threats. Algorithm 2 and figure 6.8 demonstrates the steps of execution while figure 6.13 and figure 6.14 reveals the outcome of execution. It is apparent from the simulations that Device A and Device B are able to exchange the secret keys securely after successful mutual authentication.

## 6.5.2 Security Investigation

Security investigation using BAN logic is performed to examine the robustness of the scheme in diverse compromised conditions.

```

<<Device A>>
Given ID(A) = 1, ID(B) = 2, g = 7, p = 2496232377
Pick a; a = 4
Compute g^a mod p = 2401
Generate NA = b1ed7

Formation of mA = ID(A) || ID(B) || g^a || NA = 122401b1ed7

<<Generate (c,d) pair>>

Sending commit value to Device B

Sending open value to Device B
Compute SA=NA XOR NB = 54706
Compute Shared Secret key
KAB = (g^b)^a mod p = 1164449217
Mutual Authentication process begins:
AuthA= <TS || LT || MAC1(KAB, SA || TS || LT)>
AuthA = 0705181120||0805181120|| cdb59aed872e4a36f89b69f910830ebf72210dbe2367dfde267227168d3ff4d7
Sending AuthA to DeviceB
Receiving AuthB from device B
<<AuthB = MAC2(KBA, SB || TS || LT) = cdb59aed872e4a36f89b69f910830ebf72210dbe2367dfde267227168d3ff4d7>>
Compare MAC1==MAC2
Mutual Authentication and Key Establishment is successful

```

FIGURE 6.13: Computation and communication by Device A for performing MAKE

```

<<Device B>>
Given ID(A) = 1, ID(B) = 2, g = 7, p = 2496232377
Pick b; b = 5
Compute g^b mod p = 16807
Generate NB = e59d1
Formation of mB = ID(A) || ID(B) || g^b || N = 1216807e59d1

Sending mB to Device A

Compute SB=NA XOR NB = 54706
Compute Shared Secret key
KBA = (g^a)^b mod p = 1164449217
Mutual Authentication process begins:
Receiving AuthA from device A
AuthA = 0705181120||0805181120|| cdb59aed872e4a36f89b69f910830ebf72210dbe2367dfde267227168d3ff4d7

AuthB= <MAC2(KBA, SB || TS || LT)>
AuthB = cdb59aed872e4a36f89b69f910830ebf72210dbe2367dfde267227168d3ff4d7

Sending AuthB to DeviceA
Compare MAC1==MAC2
Mutual Authentication and Key Establishment is successful

```

FIGURE 6.14: Computation and communication by Device B for performing MAKE

### Formal Proof using BAN Logic

The formal proof using BAN logic [14] evaluates the strength of the protocol using logical rules to verify the accomplishment of secret key between both devices in the insecure environment.

1. BAN Logic Notations: The notations and symbols used are directly adopted from [14] to evaluate the robustness of the scheme as follows:

- $X \equiv Y$ : Let X and Y be two principal entities where 'X believes Y'.
- $X \triangleleft Y$ : Only 'X sees Y', i.e., only X can interpret and replicate Y.
- $X \sim Y$ : 'X once said Y', principal X sent a message comprising Y.
- $X \Rightarrow Y$ : 'X has control over Y', the principal X is an authority on Y and should be trusted.
- $X \xleftrightarrow{M} Y$ : The principal X and Y have message (M) that contains the secret parameters.
- $\#(Y)$ : *Fresh*(Y), i.e., Y is not used in earlier message exchanges.
- $X \xleftrightarrow{K} Y$ : The X and Y used a secret key K for securing the communication. It is believed that key K is disclosed only to the designated legitimate principals.
- $\{M\}_K$ : Message M is encrypted using the secret key K.
- $\langle M \rangle_N$ : M is amalgamated with the secret parameter N.
- $\xrightarrow{K^{-1}} X$ : X has the private key, K.
- $\xrightarrow{K} X$ : X has the public key, K.

2. BAN Logical Rules: The logical rules referred from [14] are invoked to examine the protocol as follows:

(a) *Message-meaning rule*

$$\frac{X \equiv Y \xleftrightarrow{K} X, X \triangleleft \{M\}_K}{X \equiv Y \sim M}$$

(b) *Nonce-verification rule*

$$\frac{X \equiv \#(M), X \equiv Y \sim M}{X \equiv Y \equiv M}$$

(c) *Control rule*

$$\frac{X \equiv Y \Rightarrow M, X \equiv Y \equiv M}{X \equiv M}$$

- (d) If a principal sees a formula, then it also sees its components, provided he knows the necessary keys

$$\frac{X \triangleleft \langle M \rangle_N}{X \triangleleft M}, \frac{X \triangleleft (M, N)}{X \triangleleft M}$$

- (e) *Fresh rule*

$$\frac{X \models \#(M)}{X \models \#(M, N)}$$

If one part of a formula is fresh, then the entire formula must also be fresh [14].

3. Formal Verification of the proposed protocol: The evaluation process is carried out in four steps: (i) *message idealization*, (ii) *assumptions*, (iii) *expected goals*, and (iv) *logic verification*.

(i) **Message idealization**: Message idealization specify the messages exchanged between Device A and B. The idealized messages for the proposed protocol are summarized as follows:

Between Device A and B:

$$\boxed{M_1} \quad D_B \triangleleft c \quad \{i.e., \text{commit value}\}$$

$$\boxed{M_2} \quad D_A \triangleleft m_B \quad (= ID_A, ID_B, g^b, N_B) \quad \{i.e., \text{secret value of Device B}\}$$

$$\boxed{M_3} \quad D_B \triangleleft d \quad \{i.e., \text{open value to compute secret value } m_A\}$$

$$\boxed{M_4} \quad D_B \triangleleft Auth_A \quad (= TS, LT, MAC_{K_{AB}}(S_A, TS, LT) \{i.e., \text{MAKE}\})$$

$$\boxed{M_5} \quad D_A \triangleleft Auth_B \quad (= MAC_{K_{BA}}(S_B, TS, LT) \{i.e., \text{MAKE}\})$$

- (ii) **Assumptions**: Following are the inherent assumptions:

- For the Device A:

$$\boxed{A_1} \quad D_A \models D_B \xleftrightarrow{m_B} D_A$$

$$\boxed{A_2} \quad D_A \models D_B \xleftrightarrow{Auth_B} D_A$$

$$\boxed{A_3} \quad D_A \models \#(N_A)$$

$$\boxed{A_4} \quad D_A \models \#(TS)$$

$$\boxed{A_5} \quad D_A \models \#(LT)$$

$$\boxed{A_6} \quad D_A \models D_B \xleftrightarrow{K_{BA}} D_A$$

$$\boxed{A_7} \quad D_A \models (D_B \Rightarrow D_B \xleftrightarrow{K_{BA}} D_A)$$

- For the Device B:

$$\boxed{A_8} \quad D_B \models D_A \xleftrightarrow{c} D_B$$

$$\boxed{A_9} \quad D_B \models D_A \xleftrightarrow{d} D_B$$

$A_{10}$	$D_B \mid\equiv D_A \xleftrightarrow{Auth_A} D_B$
$A_{11}$	$D_B \mid\equiv \#(N_B)$
$A_{12}$	$D_B \mid\equiv \#(TS)$
$A_{13}$	$D_B \mid\equiv \#(LT)$
$A_{14}$	$D_B \mid\equiv D_A \xleftrightarrow{K_{AB}} D_B$
$A_{15}$	$D_B \mid\equiv (D_A \Rightarrow D_A \xleftrightarrow{K_{AB}} D_B)$

(iii) **Expected goals:** The security goals define the necessary security properties that a security protocol should exhibit. The significant security goals that the proposed protocol aims to achieve is summarized as follows:

$G_1$	$D_B \mid\equiv D_A \mid\equiv D_B \xleftrightarrow{K_{BA}} D_A$ i.e., session key ( $K_{BA}$ )
$G_2$	$D_A \mid\equiv D_B \mid\equiv D_A \xleftrightarrow{K_{AB}} D_B$ i.e., session key ( $K_{AB}$ )
$G_3$	$D_B \mid\equiv \#(N_A, TS, LT)$ i.e., freshness
$G_4$	$D_A \mid\equiv \#(N_B)$ i.e., freshness

(iv) **Logic verification:** Based on the assumptions, message idealisation, and rules, the logic verification has been performed and proven as follows:

$$\boxed{\text{Goal 1}} \quad D_B \mid\equiv D_A \mid\equiv D_B \xleftrightarrow{K_{BA}} D_A$$

**Proof** According to  $A_8$  ( $D_B \mid\equiv D_A \xleftrightarrow{c} D_B$ ) and  $A_9$  ( $D_B \mid\equiv D_A \xleftrightarrow{d} D_B$ ),  $D_B$  believes  $D_A$  and the message ( $M_1, M_3$ ) that contain secret parameters which will be used to derive the secret session key ( $K_{BA}$ ).  $D_B$  computes  $m_A$  from ( $c, d$ ) pair and obtain:

$$D_B \triangleleft \#N_A$$

By applying *message meaning rule* and  $A_3$ , we obtain

$$D_B \mid\equiv D_A \mid\sim \langle m_A \rangle_{\#N_A}$$

$$D_B \mid\equiv D_A \mid\sim \langle m_A \rangle_{g^a}$$

If the device  $D_B$  believes the device  $D_A$  and so the  $\#N_A$ , then it also believes in the diffie hellman parameter ( $g^a$ ) shared by device  $D_A$ . By applying  $A_6$  and  $A_7$ , we obtain that:

$$D_B \mid\equiv D_A \mid\equiv D_B \xleftrightarrow{K_{BA}(=(g^a)^b \text{ mod } p)} D_A$$

$$D_B \mid\equiv (D_A \Rightarrow D_A \xleftrightarrow{K_{BA}} D_B)$$

Hence, the goal  $G_1: D_B \mid\equiv D_A \mid\equiv D_B \xleftarrow{K_{BA}} D_A$  has been achieved, and similarly, the goal  $G_2$  can be accomplished.

**Goal 3**  $D_B \mid\equiv \#(N_A, TS, LT)$ , where  $N_A$ ,  $TS$ , and  $LT$  are the components of different messages sent by  $D_A \rightarrow D_B$ .

**Proof**

According to  $M_3: D_B \triangleleft d$ ,  $D_B$  receives the open value ( $d$ ) from  $D_A$ .  $D_B$  computes the secret value  $m_A$  using the commit/open pair ( $c, d$ ) and deduce nonce,  $N_A$ .  $N_A$  is a random positive integer generated by  $D_A$  for freshness. By applying rule 'b' (Nonce verification), rule 'e' (Freshness), and  $A_3$ , we obtain:

$$D_B \mid\equiv \# N_A$$

Considering  $A_{10}$ ,  $D_B$  receives  $Auth_A$  in the message  $M_4$  from  $D_A$ .  $D_B$  retrieves the  $TS$  and  $LT$  from the  $Auth_A$ .  $TS$  and  $LT$  are incorporated by  $D_A$  for freshness and prevention against replay attacks. By applying rule 'd', rule 'e',  $A_4$ , and  $A_5$  we obtain:

$$D_B \mid\equiv \#(TS, LT)$$

Thus, the goal  $G_3$  has been realized successfully, i.e.,  $D_B$  believes that  $\# N_A, TS, LT$  are fresh. Likewise, goal  $G_4$  (i.e.,  $D_A \mid\equiv \# N_B$ ) can be attained.

### 6.5.3 Comparative Analysis

It is worth noting that the proposed scheme is a subset of traditional WiFi Direct and SAS based key agreement protocol. The proposed scheme is an initiative to strengthen the security and eliminate the vulnerabilities of conventional protocols disclosed by the authors in Section 2.3 and 2.4.

Figure 6.15 illustrates the difference between the WiFi Direct, SAS based key agreement, and proposed protocol. In the past, most approaches have not focused on safeguarding the discovery phase, despite knowing the fact that adversaries prefer to accomplish DoS attacks in this phase. On the contrary, the proposed scheme (refer algorithm 1 and table 6.2) not only prevents the legitimate devices from processing the fake requests but also enhances the active lifetime of the devices and ad-hoc networks.

The authentication set-up of conventional protocols are insufficient to provide complete security solutions. Figure 6.16 highlights the difference in the mutual authentication and key exchange process of proposed and conventional protocols. Despite the fact that SAS based key agreement protocol made a decent attempt to prevent MITM attack, the cryptanalysis revealed the vulnerabilities that can be easily exploited by the

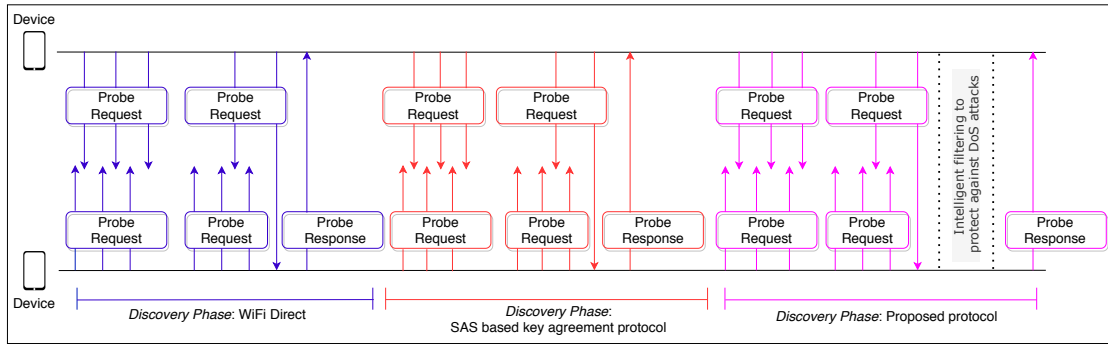


FIGURE 6.15: Discovery phase comparison of proposed protocol with conventional protocols

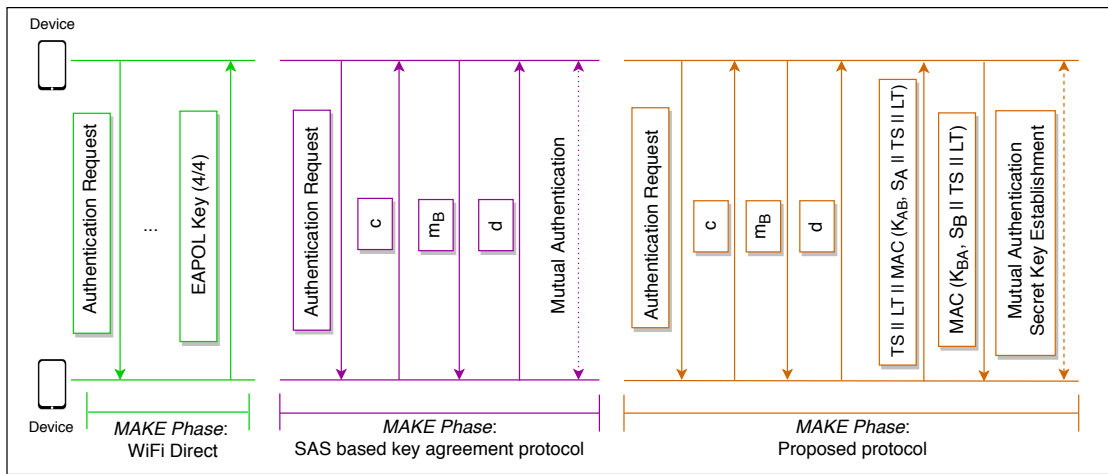


FIGURE 6.16: MAKE phase comparison of proposed protocol with conventional protocols

attackers to accomplish the MITM attack. The proposed protocol extends the SAS based key agreement approach to overcome the limitations, and extend the robustness of the approach.

Some significant observations can be drawn from the figure 6.16 which are summarized as follows: (i) the authentication strings are not sent in plain text, (ii) timestamp and lifetime are incorporated to prevent replay attacks, and (iii) verifies the establishment of identical key at both devices to avoid erroneous use, and so forth. The successful execution, robustness verification, and comparative analysis proves the supremacy of the proposed scheme in contrast to the traditional approaches.





## Chapter 7

# Conclusions

### Chapter Overview

7.1	Summary . . . . .	99
7.2	Immediate Impact . . . . .	100
7.3	Future Directions . . . . .	100
7.4	Ethical Considerations . . . . .	101

### 7.1 Summary

The IoT is transforming the conventional practices and evolving new era of smart computing. The IoT has many potential applications including smart home area network, Industry 4.0, smart cities, etc. Industry 4.0 strives to achieve faster production, resource efficiency, reduced costs, better product quality, automation, and quicker fault detection whereas cities have revolutionized the ecosystem by automating the majority of the processes, including billing of energy meters, emergency services during accidents and disasters, and so on. However, due to insecure wireless channels, the IoT applications become an easy target for the attacker. Considering the above risk exposure, three strong mutual authentication and key establishment protocols for diverse applications IoT have been developed to conquer the adversarial threats. The proposed protocols are robust yet lightweight and protect the network and systems from all prominent attacks likes modification, impersonation, MITM, DoS, replay, and known key etc. The accomplishment of security properties such as integrity, privacy, and freshness, etc. has been verified through security analysis. The proposed protocols have been compared with conventional protocols on the parameter of communication and computation cost; it has been observed that the proposed protocols are more effective and efficient. The accomplishment of all essential security requirements make the proposed protocols well deserved for the IoT environment.

## 7.2 Immediate Impact

The research contribution of the thesis has scientific as well as social impact. The applicability of IoT devices suffers due to absence of essential security measures because of the resource-constrained nature of these devices. The novel research work presented in this thesis intends to accomplish essential security measures for IoT networks with minimum computation and communication requirements, etc. The research outcomes would serve as a reference to the forthcoming research projects on lightweight mutual authentication and key establishment protocols, thus pushing the research frontier towards more efficient and robust security protocols. Further, the research work provides a foundation for future research as well. The IoT applications are used in daily lives, for example, home to industrial automation, traffic to patient management, and so forth. The security requirements of these applications are message integrity, privacy, anonymity, and authorized access, etc. The proposed protocols exhibit the aforementioned security properties, thus enhancing user trust and faith in IoT applications. The research work can be accompanied by WSN, cloud, and fog computing, etc. to benefit the other closely related sectors.

## 7.3 Future Directions

The proposed protocols can be extended in future to address the needs of social and scientific community. Few long term goals are summarized as follows.

1. **Quantum computing:** Existing security schemes will no longer be usable since quantum computers can easily crack the conventional algorithms that seem impossible with existing computer systems. Moreover, quantum security has following benefits: reduced handshaking, easy exchange of longer keys, decreased battery consumption, and minimum network overhead. Hence, new quantum-safe security protocols need to be developed to protect the networks and devices from attackers with quantum computing capability.
2. **Blockchain:** Blockchain provides decentralized and private security solutions in the networking, however, delays, excessive energy consumption, and overheads make it unfit for IoT applications. Therefore, research on lightweight blockchain is required to address the security issues of IoT networks with minimum resource utilization.
3. **Physically Unclonable Function (PUF):** Most of the IoT applications demand the deployment of IoT devices in a hostile environment (e.g., agriculture farm, industrial furnaces, etc.). The unfriendly environment makes the IoT devices vulnerable to physical, cloning, and side-channel attacks. Thus, the existing protocols

need amendments to protect the IoT devices from these attacks. The PUF can be explored as a possible option for achieving the physical security.

4. **Fog Computing:** Majority of the nodes in the IoT environment are resource-deprived, therefore more lightweight security protocols should be developed by outsourcing the complex computations to the fog nodes.
5. **Secure Mobility:** The devices used in the IoT networks are either stationary or non-stationary. The stationary devices in the IoT network are comparatively more secure in contrast to the mobile devices. The mobility brings forth new challenges; for instance, a mobile IoT device might require multiple intermediate networks (trusted/un-trusted) during the movement to make a connection with the recipient device, hence non trusted gateways (networks) in the pathway increase the likelihood of cyber-attacks. Moreover, in most of the existing protocols, the end gateway is considered as a trustworthy entity whereas mobility raises the concern on the integrity of all those gateways other than the trusted. Besides, mobile IoT devices cannot be monitored physically every time to verify the ownership and legitimate use. Although rare but holds a probability, where a legitimate mobile IoT device is stolen with an active authenticated session. The consequences of all aforementioned scenarios can result in loss of confidentiality, integrity, availability, and so forth. Therefore, new protocols that can cater to the demands of secure mobility should be developed using homomorphic encryption, and Zero-Knowledge Proofs (ZKP), etc.

## 7.4 Ethical Considerations

The research has been carried out with utmost honesty while ensuring scientific integrity. The thesis contains only real and unbiased findings of experimentation. The protocols and their elaboration are free from all types of plagiarism; moreover, the works referred to conduct the research have been duly credited. We have used only public available cryptography primitives, schemes, and scientific tools to avoid ethical shenanigans. The literature has been ethically downloaded from the portal provided by the University. Furthermore, only authentic scientific tools have been used to conduct the research.



## References

- [1] A. Esfahani, G. Mantas, R. Maticsek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. Tauber, C. Schmittner, and J. Bastos, "A lightweight authentication mechanism for m2m communications in industrial iot environment," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 288–296, 2019.
- [2] S. Kim, M. Lee, and C. Shin, "Iot-based strawberry disease prediction system for smart farming," *Sensors*, vol. 18, no. 11, p. 4051, 2018.
- [3] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2018.
- [4] Statista, "Forecast end-user spending on iot solutions worldwide from 2017 to 2025 (in billion u.s. dollars)," <https://www.statista.com/statistics/976313/global-iot-market-size/#:~:text=The%20global%20market%20for%20Internet,around%201.6%20trillion%20by%202025.>, 2019, accessed: 2021-03-01.
- [5] L. Columbus, "roundup of internet of things forecasts," *Forbes, Dec*, vol. 10, p. 2017, 2017.
- [6] T. Alam, "A reliable communication framework and its use in internet of things (iot)," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 3, no. 5, pp. 450–456, 2018.
- [7] A. C. Djedouboum, A. A. Abba Ari, A. M. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4474, 2018.
- [8] N. Kaushik and T. Bagga, "Internet of things (iot): Implications in society," *Available at SSRN 3563104*, 2020.
- [9] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4946–4967, 2019.

- [10] A. Khanna and S. Kaur, "Internet of things (iot), applications and challenges: A comprehensive review," *Wireless Personal Communications*, 2020.
- [11] R. Borgaonkar, L. Hirschi, S. Park, and A. Shaik, "New privacy threat on 3g, 4g, and upcoming 5g aka protocols," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 108–127, 2019.
- [12] D. Kim and H. K. Kim, "Automated dataset generation system for collaborative research of cyber threat analysis," *Security and Communication Networks*, vol. 2019, 2019.
- [13] "PHI of 33,420 BJC Healthcare Patients Exposed on Internet for 8 Months," <https://www.hipaajournal.com/phi-of-33420-bjc-healthcare-patients-exposed-on-internet-for-8-months>, 2018, online; accessed October 5, 2020.
- [14] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous secure framework in connected smart home environments," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 968–979, 2017.
- [15] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.
- [16] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019.
- [17] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, and P. Kumar, "Robust and lightweight mutual authentication scheme in distributed smart environments," *IEEE Access*, vol. 8, pp. 69 722–69 733, 2020.
- [18] G. S. Gaba, G. Kumar, H. Monga, T.-H. Kim, M. Liyanage, and P. Kumar, "Robust and lightweight key exchange (lke) protocol for industry 4.0," *IEEE Access*, vol. 8, pp. 132 808–132 824, 2020.
- [19] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in iot devices with minimal airtime consumption," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, 2017.
- [20] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for iot smart home using cumulative keyed-hash chain," *Journal of information security and applications*, vol. 45, pp. 156–175, 2019.
- [21] S. Patel, D. R. Patel, and A. P. Navik, "Energy efficient integrated authentication and access control mechanisms for internet of things," in *Internet of Things and Applications (IOTA), International Conference on*. IEEE, 2016, pp. 304–309.

- [22] M. Hossain, S. Noor, and R. Hasan, "Hsc-iot: A hardware and software co-verification based authentication scheme for internet of things," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2017 5th IEEE International Conference on*. IEEE, 2017, pp. 109–116.
- [23] S. Dey and A. Hossain, "Session-key establishment and authentication in a smart home network using public key cryptography," *IEEE Sensors Letters*, vol. 3, no. 4, pp. 1–4, 2019.
- [24] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.
- [25] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K.-K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606–1615, 2018.
- [26] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.
- [27] S. Paliwal, "Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things," *IEEE Access*, vol. 7, pp. 136 073–136 093, 2019.
- [28] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on wireless communications*, vol. 15, no. 1, pp. 357–366, 2015.
- [29] W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, "Secure device-to-device communications over wifi direct," *IEEE Network*, vol. 30, no. 5, pp. 4–9, 2016.
- [30] G. S. Gaba, G. Kumar, T.-H. Kim, H. Monga, and P. Kumar, "Secure device-to-device communications for 5g enabled internet of things applications," *Computer Communications*, 2021.
- [31] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in *2012 international conference on computer science and electronics engineering*, vol. 3. IEEE, 2012, pp. 648–651.
- [32] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2019.

- [33] K. Choudhary, G. S. Gaba, I. Butun, and P. Kumar, "MAKE-IT: A lightweight mutual authentication and key exchange protocol for industrial internet of things," *Sensors*, vol. 20, no. 18, p. 5166, 2020.
- [34] S. Khan, A. I. Alzahrani, O. Alfarraj, N. Alalwan, and A. H. Al-Bayatti, "Resource efficient authentication and session key establishment procedure for low-resource iot devices," *IEEE Access*, vol. 7, pp. 170 615–170 628, 2019.
- [35] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.
- [36] S. Sciancalepore, A. Caposelle, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for iot systems," in *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*. ACM, 2015, pp. 37–42.
- [37] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate-based pairwise key establishment protocol for wireless sensor networks," in *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*. IEEE, 2013, pp. 667–674.
- [38] A. B. Forouzan, *Data Communications & Networking (5E)*. Tata McGraw-Hill Education, 2013.
- [39] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual authentication in iot systems using physical unclonable functions," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327–1340, 2017.
- [40] N. A. Gunathilake, W. J. Buchanan, and R. Asif, "Next generation lightweight cryptography for smart iot devices:: implementation, challenges and applications," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 707–710.
- [41] C. T. Hager, S. F. Midkiff, J.-M. Park, and T. L. Martin, "Performance and energy efficiency of block ciphers in personal digital assistants," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE, 2005, pp. 127–136.
- [42] W. M. Khan and I. A. Zualkernan, "Sensepods: A zigbee-based tangible smart home interface," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 145–152, May 2018.
- [43] P. Sundaravadivel, K. Kesavan, L. Kesavan, S. P. Mohanty, and E. Kougiannos, "Smart-log: A deep-learning based automated nutrition monitoring system in the iot," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, pp. 390–398, Aug 2018.



- [44] D. Díaz-Sánchez, R. S. Sherratt, F. Almenarez, P. Arias, and A. Marín, "Secure store and forward proxy for dynamic iot applications over m2m networks," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 4, pp. 389–397, November 2016.
- [45] Y.-T. Lee, W.-H. Hsiao, Y.-S. Lin, and S.-C. T. Chou, "Privacy-preserving data analytics in cloud-based smart home with community hierarchy," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 2, pp. 200–207, 2017.
- [46] J. G. An, F. Le Gall, J. Kim, J. Yun, J. Hwang, M. Bauer, M. Zhao, and J. S. Song, "Towards global iot-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet of Things Journal*, 2019.
- [47] Alfred, "Smart cities around the world were exposed to simple hacks)," <https://www.cnet.com/news/smart-cities-around-the-world-were-exposed-to-simple-hacks/>, 2018, online; accessed 13 December 2018.
- [48] B. Ali and A. I. Awad, "Cyber and physical security vulnerability assessment for iot-based smart homes," *Sensors*, vol. 18, no. 3, p. 817, 2018.
- [49] D. Goodin, "Leak of 1,700 valid passwords could make the iot mess much worse," <https://arstechnica.com/information-technology/2017/08/leak-of-1700-valid-passwords-could-make-the-iot-mess-much-worse>, 2018, online; accessed 28 October 2018.
- [50] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Communications Surveys & Tutorials*, 2018.
- [51] T. Dunlap, "The 5 Worst Examples of IoT Hacking and Vulnerabilities in Recorded History," <https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities>, 2018, online; accessed 04 August 2018.
- [52] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.
- [53] J. Kim, J. Baek, and T. Shon, "An efficient and scalable re-authentication protocol over wireless sensor network," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 516–522, 2011.
- [54] Matthew Campagna, "SEC4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)," <http://www.sec.gov/sec4-1.0.pdf>, 2013, online; accessed 04 August 2018.
- [55] L. Viganò, "Automated security protocol analysis with the avispa tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006.

- [56] N. Enneya, A. Baayer, and M. ElKoutbi, "A dynamic timestamp discrepancy against replay attacks in manet," in *Informatics Engineering and Information Science*, A. Abd Manaf, S. Sahibuddin, R. Ahmad, S. Mohd Daud, and E. El-Qawasmeh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 479–489.
- [57] D. Samfat, R. Molva, and N. Asokan, "Untraceability in mobile networks," in *Proceedings of the 1st annual international conference on Mobile computing and networking*. ACM, 1995, pp. 26–36.
- [58] M. Corporation, "Telos Ultra low power IEEE 802.15.4 compliant wireless sensor module)," <http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf>, 2018, online; accessed 04 August 2018.
- [59] A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 245–256.
- [60] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005, pp. 364–369.
- [61] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial internet of things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78 238–78 259, 2018.
- [62] Infosys, "Industry 4.0 as an evolution, not a revolution," <https://www.infosys.com/about/knowledge-institute/insights/Documents/industry-4.0-evolution.pdf>, 2019, online; accessed January 2, 2020.
- [63] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [64] Deloitte, "Industry 4.0: An introduction)," 2015, online; accessed January 2, 2020.
- [65] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017.
- [66] M. Kohler, "Industry 4.0: 10 use cases for software in connected manufacturing," <https://blog.bosch-si.com/industry40/industry-4-0-10-use-cases-for-software-in-connected-manufacturing/>, 2018, online; accessed February 7, 2020.
- [67] S. Souchet, "Industry 4.0 case studies," <https://home.kpmg/xx/en/home/insights/2018/11/industry-4-0-case-studies.html>, 2019, online; accessed February 7, 2020.

- [68] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
- [69] H. Mouratidis and V. Diamantopoulou, "A security analysis method for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4093–4100, 2018.
- [70] T. P. Raptis, A. Passarella, and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *IEEE Access*, vol. 7, pp. 97 052–97 093, 2019.
- [71] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467–3501, 2019.
- [72] E. T. Nakamura and S. L. Ribeiro, "A privacy, security, safety, resilience and reliability focused risk assessment methodology for iiot systems steps to build and use secure iiot systems," in *2018 Global Internet of Things Summit (GIoTS)*. IEEE, 2018, pp. 1–6.
- [73] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, "Security issues in iiot: A comprehensive survey of attacks on iiot and its countermeasures," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*. IEEE, 2018, pp. 124–130.
- [74] S. L. Keoh, "Cyber-Physical Systems Are at Risk," <https://www.infosecurity-magazine.com/next-gen-infosec/cyberphysical-systems-risk-1/>, 2019, online; accessed March 4, 2020.
- [75] T. Armerding, "Cyber-physical attacks are growing alongside the IoT," <https://www.synopsys.com/blogs/software-security/cyber-physicalattacks/>, 2019, online; accessed March 4, 2020.
- [76] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the internet of things," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1636–1675, 2018.
- [77] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.
- [78] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities," *IEEE Access*, vol. 5, pp. 26 521–26 544, 2017.
- [79] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

- [80] H. Kim and E. A. Lee, "Authentication and authorization for the internet of things," *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.
- [81] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, vol. 2017, 2017.
- [82] K. Chen, S. Zhang, Z. Li, Y. Zhang, Q. Deng, S. Ray, and Y. Jin, "Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice," *Journal of Hardware and Systems Security*, vol. 2, no. 2, pp. 97–110, 2018.
- [83] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.
- [84] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [85] B. B. Journal, "Hungary's ratio of robot use among lowest in eu," [https://bbj.hu/economy/hungarys-ratio-of-robot-use-among-lowest-in-eu\\_160380](https://bbj.hu/economy/hungarys-ratio-of-robot-use-among-lowest-in-eu_160380), 2018, accessed: 2019-03-29.
- [86] M. Campagna, "Sec 4: Elliptic curve qu-vanstone implicit certificate scheme (ecqv)," *Technical Report*, 2013.
- [87] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, 2018.
- [88] Y. Zhang, C. Chen, and J. He, "Dos attack on networked control system: From the viewpoint on communication-control cost," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 5695–5700.
- [89] N. Enneya, A. Baayer, and M. ElKoutbi, "A dynamic timestamp discrepancy against replay attacks in manet," in *International Conference on Informatics Engineering and Information Science*. Springer, 2011, pp. 479–489.
- [90] S. Fajarado, "CM5000 Datasheet," <http://www.epssilon.cl/files/EPS5000.pdf>, 2010, online; accessed February 15, 2020.
- [91] A. S. Alfa, B. T. Maharaj, H. A. Ghazaleh, and B. Awoyemi, "The role of 5g and iot in smart cities," in *Handbook of Smart Cities*. Springer, 2018, pp. 31–54.
- [92] S. C. K. Tekouabou, W. Cherif, H. Silkan *et al.*, "Improving parking availability prediction in smart cities with iot and ensemble-based model," *Journal of King Saud University-Computer and Information Sciences*, 2020.

- [93] F. Marino, F. Leccese, and S. Pizzuti, "Adaptive street lighting predictive control," *Energy Procedia*, vol. 111, pp. 790–799, 2017.
- [94] T. Ali, M. Irfan, A. S. Alwadie, and A. Glowacz, "Iot-based smart waste bin monitoring and municipal solid waste management system for smart cities," *Arabian Journal for Science and Engineering*, 2020.
- [95] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.
- [96] C. K. Toh, J. A. Sanguesa, J. C. Cano, and F. J. Martinez, "Advances in smart roads for future smart cities," *Proceedings of the Royal Society A*, vol. 476, no. 2233, p. 20190439, 2020.
- [97] P. A. Networks, "Building Secure Smart Cities in the Age of 5G and IoT," <https://www.paloaltonetworks.com/cyberpedia/smart-cities-in-the-age-of-5g-and-iot>, 2020, online; accessed September 4, 2020.
- [98] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. Khoukhi, "Iot technologies for smart cities," *IET Networks*, vol. 7, no. 1, pp. 1–13, 2017.
- [99] E. Park, A. P. Del Pobil, and S. J. Kwon, "The role of internet of things (iot) in smart cities: Technology roadmap-oriented approaches," *Sustainability*, vol. 10, no. 5, p. 1388, 2018.
- [100] K. T. Kadhim, A. M. Alsahlany, S. M. Wadi, and H. T. Kadhum, "An overview of patient's health status monitoring system based on internet of things (iot)," *Wireless Personal Communications*, pp. 1–28, 2020.
- [101] L. Horwitz, "The future of IoT miniguide: The burgeoning IoT market continues," <https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html>, 2019, online; accessed September 6, 2020.
- [102] J. H. Park, "Symmetry-adapted machine learning for information security," 2020.
- [103] S. Poslad, A. Ma, Z. Wang, and H. Mei, "Using a smart city iot to incentivise and target shifts in mobility behaviour—is it a piece of pie?" *Sensors*, vol. 15, no. 6, pp. 13 069–13 096, 2015.
- [104] S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, "An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment," *PloS one*, vol. 14, no. 11, p. e0224934, 2019.
- [105] D. Minoli and B. Occhiogrosso, "Practical aspects for the integration of 5g networks and iot applications in smart cities environments," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

- [106] K. Katzis and H. Ahmadi, "Challenges implementing internet of things (iot) using cognitive radio capabilities in 5g mobile networks," in *Internet of Things (IoT) in 5G Mobile Technologies*. Springer, 2016, pp. 55–76.
- [107] Z. Li, F. S. Moya, F. Gabor, J. M. B. Da Silva Jr., and K. Koufos, "Device-to-device (d2d) communications," in *5G Mobile and Wireless Communications Technology*. Cambridge University Press, 2016, pp. 107–136.
- [108] U. N. Kar and D. K. Sanyal, "An overview of device-to-device communication in cellular networks," *ICT express*, vol. 4, no. 4, pp. 203–208, 2018.
- [109] P. Phunchongharn, E. Hossain, and D. I. Kim, "Resource allocation for device-to-device communications underlying lte-advanced networks," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 91–100, 2013.
- [110] N. Instruments, "Applications of Device-to-Device Communication in 5G Networks," <https://spectrum.ieee.org/computing/networks/applications-of-devicetodevice-communication-in-5g-networks>, 2020, online; accessed September 4, 2020.
- [111] M. Usman, F. Granelli, and M. R. Asghar, *5G and D2D Communications at the Service of Smart Cities*. John Wiley & Sons Ltd, 2019.
- [112] C. Kai, H. Li, L. Xu, Y. Li, and T. Jiang, "Energy-efficient device-to-device communications for green smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1542–1551, 2018.
- [113] M. M. Elmesalawy, "D2d communications for enabling internet of things underlying lte cellular networks," *Journal of Wireless Networking and Communications*, vol. 6, no. 1, pp. 1–9, 2016.
- [114] B. M. ElHalawany, R. Ruby, and K. Wu, "D2d communication for enabling internet-of-things: outage probability analysis," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2332–2345, 2019.
- [115] P. Pawar and A. Trivedi, "Device-to-device communication based iot system: benefits and challenges," *IETE Technical Review*, vol. 36, no. 4, pp. 362–374, 2019.
- [116] M. Wang and Z. Yan, "A survey on security in d2d communications," *Mobile Networks and Applications*, vol. 22, no. 2, pp. 195–208, 2017.
- [117] P. Gandotra, R. K. Jha, and S. Jain, "A survey on device-to-device (d2d) communication: Architecture and security issues," *Journal of Network and Computer Applications*, vol. 78, pp. 9–29, 2017.
- [118] V. Siddhartha, G. S. Gaba, and L. Kansal, "A lightweight authentication protocol using implicit certificates for securing iot systems," *Procedia Computer Science*, vol. 167, pp. 85–96, 2020.

- [119] M. Park, H. Oh, and K. Lee, "Security risk measurement for information leakage in iot-based smart homes from a situational awareness perspective," *Sensors*, vol. 19, no. 9, p. 2148, 2019.
- [120] L. Tawalbeh, F. Muheidat, M. Tawalbeh, M. Quwaider *et al.*, "Iot privacy and security: Challenges and solutions," *Applied Sciences*, vol. 10, no. 12, p. 4102, 2020.
- [121] H. Lee, D. Kang, J. Ryu, D. Won, H. Kim, and Y. Lee, "A three-factor anonymous user authentication scheme for internet of things environments," *Journal of Information Security and Applications*, vol. 52, p. 102494, 2020.
- [122] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2019.
- [123] P. Kumar and G. S. Gaba, "Biometric-based robust access control model for industrial internet of things applications," *IoT Security: Advances in Authentication*, pp. 133–142, 2020.
- [124] M. Wang, Z. Yan, B. Song, and M. Atiquzzaman, "Aaka-d2d: Anonymous authentication and key agreement protocol in d2d communications," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBD-Com/IOP/SCI)*. IEEE, 2019, pp. 1356–1362.
- [125] M. S. Raniyal, I. Woungang, S. K. Dhurandher, and S. S. Ahmed, "Passphrase protected device-to-device mutual authentication schemes for smart homes," *Security and Privacy*, vol. 1, no. 3, p. e42, 2018.
- [126] A. Celik, J. Tetzner, K. Sinha, and J. Matta, "5g device-to-device communication security and multipath routing solutions," *Applied Network Science*, vol. 4, no. 1, pp. 1–24, 2019.
- [127] A. P. G Lopes and P. R. Gondim, "Mutual authentication protocol for d2d communications in a cloud-based e-health system," *Sensors*, vol. 20, no. 7, p. 2072, 2020.
- [128] A. Hadiks, Y. Chen, F. Li, and B. Liu, "A study of stealthy denial-of-service attacks in wi-fi direct device-to-device networks," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2014, pp. 507–508.
- [129] L. Wang, L. Liu, X. Cao, X. Tian, and Y. Cheng, "Sociality-aware resource allocation for device-to-device communications in cellular networks," *IET Communications*, vol. 9, no. 3, pp. 342–349, 2015.



- [130] H. H. Hussein, H. A. Elsayed, and S. M. Abd El-kader, "Intensive benchmarking of d2d communication over 5g cellular networks: prototype, integrated features, challenges, and main applications," *Wireless Networks*, pp. 1–20, 2019.
- [131] O. ElGarhy and L. Reggiani, "Increasing efficiency of resource allocation for d2d communication in nb-iot context," *Procedia computer science*, vol. 130, pp. 1084–1089, 2018.
- [132] J. Liu, Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, "Device-to-device communications achieve efficient load balancing in lte-advanced networks," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 57–65, 2014.
- [133] Z. Belghazi, N. Benamar, A. Addaim, and C. A. Kerrache, "Secure wifi-direct using key exchange for iot device-to-device communications in a smart environment," *Future Internet*, vol. 11, no. 12, p. 251, 2019.
- [134] Z. Yan, H. Xie, P. Zhang, and B. B. Gupta, "Flexible data access control in d2d communications," *Future generation computer systems*, vol. 82, pp. 738–751, 2018.
- [135] M. A. Khan, R. Hamila, and M. O. Hasna, "Optimal group formation in dense wi-fi direct networks for content distribution," *IEEE Access*, vol. 7, pp. 161 231–161 245, 2019.
- [136] W. Cherif, M. A. Khan, F. Filali, S. Sharafeddine, and Z. Dawy, "P2p group formation enhancement for opportunistic networks with wi-fi direct," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.
- [137] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: overview and experimentation," *IEEE wireless communications*, vol. 20, no. 3, pp. 96–104, 2013.
- [138] W.-F. Alliance, "Discover Wi-Fi | Wi-Fi Direct," <https://www.wi-fi.org/discover-wi-fi/wi-fi-direct>, 2020, online; accessed September 13, 2020.
- [139] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, "Secure key establishment for device-to-device communications," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 336–340.
- [140] J. Wang, H. Li, F. Guo, W. Zhang, and Y. Cui, "D2d big data privacy-preserving framework based on (a, k)-anonymity model," *Mathematical Problems in Engineering*, vol. 2019, 2019.
- [141] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, "Security and privacy in device-to-device (d2d) communication: A review," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1054–1079, 2017.



- [142] I. Abualhaol and S. Muegge, "Securing d2d wireless links by continuous authenticity with legitimacy patterns," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 5763–5771.
- [143] P. Gandotra and R. K. Jha, "Device-to-device communication in cellular networks: A survey," *Journal of Network and Computer Applications*, vol. 71, pp. 99–117, 2016.
- [144] D. Costantin, K. Sansurooah, and P. A. Williams, "Vulnerabilities associated with wi-fi protected setup in a medical environment," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2017, pp. 1–12.
- [145] S. Viehböck, "Brute forcing wi-fi protected setup," *Wi-Fi Protected Setup*, vol. 9, 2011.
- [146] D. Fang, Y. Qian, and R. Q. Hu, "Security for 5g mobile wireless networks," *IEEE Access*, vol. 6, pp. 4850–4874, 2017.
- [147] M. Usman, M. R. Asghar, F. Granelli, and K. Qaraqe, "Integrating smart city applications in 5g networks," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, 2018, pp. 1–5.
- [148] M. Wang, Z. Yan, and V. Niemi, "Uaka-d2d: Universal authentication and key agreement protocol in d2d communications," *Mobile Networks and Applications*, vol. 22, no. 3, pp. 510–525, 2017.
- [149] R. Sedidi and A. Kumar, "Key exchange protocols for secure device-to-device (d2d) communication in 5g," in *2016 Wireless Days (WD)*. IEEE, 2016, pp. 1–6.
- [150] M. Wang and Z. Yan, "Privacy-preserving authentication and key agreement protocols for d2d group communications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3637–3647, 2017.
- [151] L. Wang, Y. Tian, D. Zhang, and Y. Lu, "Constant-round authenticated and dynamic group key agreement protocol for d2d group communications," *Information Sciences*, vol. 503, pp. 61–71, 2019.
- [152] J. H. Lee, M.-S. Park, and S. C. Shah, "Wi-fi direct based mobile ad hoc network," in *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2017, pp. 116–120.
- [153] C. Casetti, C. F. Chiasserini, L. C. Pelle, C. Del Valle, Y. Duan, and P. Giaccone, "Content-centric routing in wi-fi direct multi-group networks," in *2015 IEEE 16th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*. IEEE, 2015, pp. 1–9.

- [154] M. A. Khan, W. Cherif, and F. Filali, "Group owner election in wi-fi direct," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2016, pp. 1–9.
- [155] H. Systique, "White paper on wi-fi direct," *Hughes Systique Corporation*, 2020.
- [156] T. Oide, T. Abe, and T. Suganuma, "Infrastructure-less communication platform for off-the-shelf android smartphones," *Sensors*, vol. 18, no. 3, p. 776, 2018.
- [157] M. El Alami, N. Benamar, M. Younis, and A. A. Shahin, "A framework for hotspot support using wi-fi direct based device-to-device links," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 552–557.
- [158] H. Yao, Z. Yang, H. Jiang, and L. Ma, "A scheme of ad-hoc-based d2d communication in cellular networks." *Adhoc & Sensor Wireless Networks*, vol. 32, 2016.
- [159] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi, "Secure in-band wireless pairing." in *USENIX security symposium*. San Francisco, CA, USA, 2011, pp. 1–16.
- [160] C. Wu and R. Buyya, *Cloud Data Centers and Cost Modeling: A complete guide to planning, designing and building a cloud data center*. Morgan Kaufmann, 2015.
- [161] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of mac address randomization in mobile devices and when it fails," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383, 2017.
- [162] M. M. Abdellatif, J. M. Oliveira, and M. Ricardo, "Neighbors and relative location identification using rssi in a dense wireless sensor network," in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*. IEEE, 2014, pp. 140–145.
- [163] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A location estimation algorithm based on rssi vector similarity degree," *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, 2014.
- [164] D. Maduskar and S. Tapaswi, "Rssi based adaptive indoor location tracker," *Scientific Phone Apps and Mobile Devices*, vol. 3, no. 1, p. 3, 2017.
- [165] R. M. Hashmi, A. M. Siddiqui, M. Jabeen, K. Shehzad, A. Zubair, and K. Alimgeer, "Improved secure network authentication protocol (isnap) for ieee 802.16," in *2009 International Conference on Information and Communication Technologies*. IEEE, 2009, pp. 101–105.
- [166] D. Rountree, *Security for Microsoft Windows system administrators: introduction to key information security concepts*. Elsevier, 2011.

- 
- [167] D. York, *Seven deadliest unified communications attacks*. Syngress, 2010.
- [168] K. Verma, H. Hasbullah, and A. Kumar, "Prevention of dos attacks in vanet," *Wireless personal communications*, vol. 73, no. 1, pp. 95–126, 2013.
- [169] G. Kumar, "Denial of service attacks—an updated perspective," *Systems science & control engineering*, vol. 4, no. 1, pp. 285–294, 2016.



## Appendix A

# AVISPA Installation

In order to install AVISPA *v1.1*, you need to extract the archive `avispa-package-1.1.Linux-i686.tgz` in the desired directory, which will create a new sub-directory named `avispa-1.0` populated by a number of files and sub-directories. Then you need to set the environment variable `AVISPA_PACKAGE` to refer to the absolute path ending in `avispa-1.1`, and to put the script called `avispa` in the execution path of your shell. The commands to install the AVISPA in bash shell environment are:

- `tar -xzf /home/xyz/avispa-package-1.1.Linux-i686.tgz`
- `export AVISPA_PACKAGE=/opt/avispa-1.1`
- `export PATH=PATH :AVISPA_PACKAGE`

Now you should be able to execute AVISPA, using the command `avispa`. Please see the README file for information about the command line options of AVISPA. The AVISPA package provides a user-friendly mode for XEmacs to allow a simple interaction between the user and the modules of the AVISPA package. To set-up the XEmacs mode follows the instructions below:

```
cd $AVISPA_PACKAGE/contrib tar -xzf avispa-mode.tgz
```

This command will create a directory `temporary-avispa` containing a makefile for installing the XEmacs mode. Follow the instruction in `temporary-avispa/help.txt`; when done, delete the temporary directory `temporary-avispa`.

The AVISPA package further provides the `hlpdoc` tools for documenting HLPSL specifications in LATEX and HTML format. To set them up, follow the instructions below:

```
cd $AVISPA_PACKAGE/contrib/hlpdoc tar xzf hlpdoc.tgz
```

Then follow the instructions in the local `INSTALL` file. Usage of the `hlpdoc` tools is explained in the local `README` file.



## Appendix B

# Burrows–Abadi–Needham (BAN) logic

### Notations and Symbols:

- $X \mid\equiv Y$ : Let  $X$  and  $Y$  be two principal entities where ' $X$  believes  $Y$ '.
- $X \triangleleft Y$ : Only ' $X$  sees  $Y$ ', i.e., only  $X$  can interpret and replicate  $Y$ .
- $X \mid\sim Y$ : ' $X$  once said  $Y$ ', principal  $X$  sent a message comprising  $Y$ .
- $X \mid\Rightarrow Y$ : ' $X$  has control over  $Y$ ', the principal  $X$  is an authority on  $Y$  and should be trusted.
- $X \xleftrightarrow{M} Y$ : The principal  $X$  and  $Y$  have message ( $M$ ) that contains the secret parameters.
- $\#(Y)$ : *Fresh*( $Y$ ), i.e.,  $Y$  is not used in earlier message exchanges.
- $X \xleftrightarrow{K} Y$ : The  $X$  and  $Y$  used a secret key  $K$  for securing the communication. It is believed that key  $K$  is disclosed only to the designated legitimate principals.
- $\{M\}_K$ : Message  $M$  is encrypted using the secret key  $K$ .
- $\langle M \rangle_N$ :  $M$  is amalgamated with the secret parameter  $N$ .
- $\xrightarrow{K^{-1}} X$ :  $X$  has the private key,  $K$ .
- $\xrightarrow{K} X$ :  $X$  has the public key,  $K$ .

BAN Logical Rules:1. *Message-meaning rule*

$$\frac{X \models Y \stackrel{K}{\leftrightarrow} X, X \triangleleft \{M\}_K}{X \models Y \mid \sim M}$$

2. *Nonce-verification rule*

$$\frac{X \models \#(M), X \models Y \mid \sim M}{X \models Y \models M}$$

3. *Control rule*

$$\frac{X \models Y \mid \Rightarrow M, X \models Y \models M}{X \models M}$$

## 4. If a principal sees a formula, then it also sees its components, provided he knows the necessary keys

$$\frac{X \triangleleft \langle M \rangle_N}{X \triangleleft M}, \frac{X \triangleleft (M, N)}{X \triangleleft M}$$

5. *Fresh rule*

$$\frac{X \models \#(M)}{X \models \#(M, N)}$$

If one part of a formula is fresh, then the entire formula must also be fresh.



## Appendix C

# Elliptic Curve Cryptography (ECC)

### C.1 Elliptic Curve Arithmetic (ECA)

The security protocols earlier used the RSA algorithm for public-key encryption and digital signature applications. However, with time the key size of RSA got increased that intensified the burden on computing systems. These complications gave rise to ECC. Fig. C.1 illustrates the difference in key length between RSA and ECC. The ECC provides equal security with very little key size, thus reducing the burden on the computing system.

In Elliptic Curve Arithmetic (ECA), exponentiation indicates repeated multiplication, (for example,  $a^2 \bmod q = (a \times a) \bmod q$ ) whereas multiplication indicates repeated addition (for example,  $a \times 2 = a + a$ ).

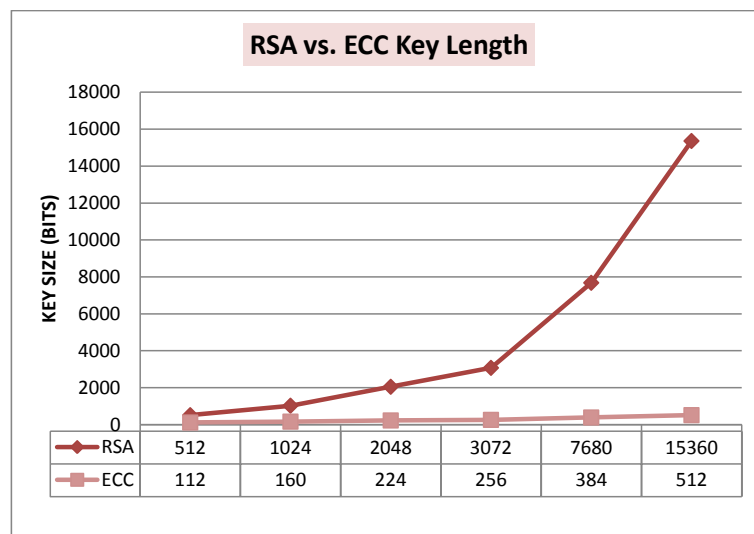


FIGURE C.1: Key Length Comparison.

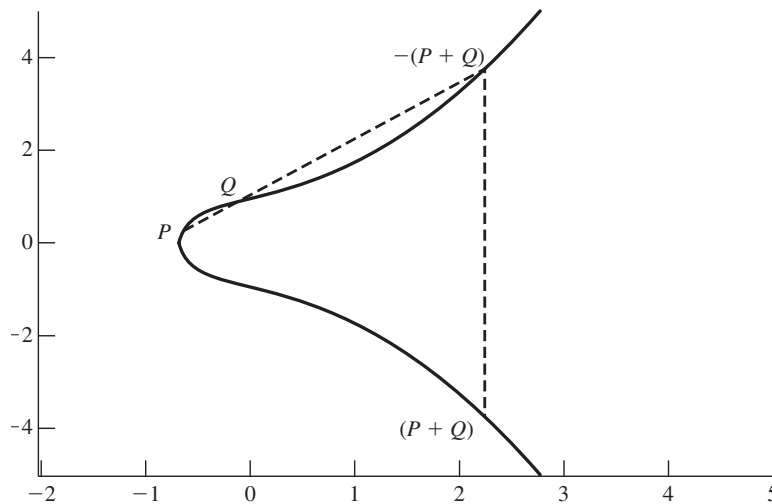


FIGURE C.2: Example of elliptic curve:  $y^2 = x^3 + x + 1$

The elliptic curves are represented through the *Weierstrass equation*. The elliptic curves are comprised of 2 variables and coefficients. It is worth noting that in cryptography the elliptic curves are restricted to finite fields. The *Weierstrass equation* for elliptic curves can be expressed as:

$$y^2 \bmod p = (x^3 + ax + b) \bmod p$$

Fig. C.2 depicts an example of elliptic curve that meets the following necessary condition of coefficient set,

$$4a^3 + 27b^2 \neq 0$$

## C.2 Diffie Hellman Key Exchange (DHKE) Algorithm

Diffie Hellman Key Exchange is a process to safely negotiate the keys between two parties. DHKE enables the parties to generate a common symmetric key which can be used for ciphering and other purposes. Consider an instance shown in Table C.1 where Alice and Bob want to negotiate the keys. Both of them use the global public elements to select the private value,  $X_A$  and  $X_B$ . Alice and Bob calculates the public key for dissemination,  $Y_A = \alpha^{X_A} \bmod q$  and  $Y_B = \alpha^{X_B} \bmod q$ . Thereafter, Alice and Bob exchange the public keys with each other,  $Y_A$  and  $Y_B$ . Lastly, Alice and Bob calculates symmetric session key,  $K = Y_B^{X_A} \bmod q$ ,  $K = Y_A^{X_B} \bmod q$ . Hence, both Alice and Bob can exchange the secret keys without disclosing any secret credentials over the vulnerable wireless medium.

TABLE C.1: Diffie-Hellman Key Exchange process

Action	Alice	Bob
Global Public Elements	$q, \alpha$	$q, \alpha$
Select Private	$X_A < q$	$X_B < q$
Calculate Public	$Y_A = \alpha^{X_A} \bmod q$	$Y_B = \alpha^{X_B} \bmod q$
Alice transmits $Y_A$ to Bob Bob transmits $Y_B$ to Alice		
Calculate Secret Key	$K = Y_B^{X_A} \bmod q$	$K = Y_A^{X_B} \bmod q$
Secret Key established successfully		

TABLE C.2: ECC Diffie-Hellman Key Exchange process

Action	Alice	Bob
Global Public Elements	$E_q(a, b), G$	$E_q(a, b), G$ (order: $n$ )
Select Private	$n_A < n$	$n_B < n$
Calculate Public	$P_A = n_A \times G$	$P_B = n_B \times G$
Alice transmits $P_A$ to Bob Bob transmits $P_B$ to Alice		
Calculate Secret Key	$K = n_A \times P_B$	$K = n_B \times P_A$
Secret Key established successfully		

### C.3 ECC-Diffie Hellman Key Exchange Algorithm (ECDH)

ECDH is a key agreement process that uses elliptic curve based public-private key pair to generate the symmetric session key. Consider an instance as shown in Table C.2 where the Alice and Bob use the global public element  $(E_q(a, b), G$  (order: $n$ )) to select the private values,  $n_A < n, n_B < n$ . Post selection, Alice and Bob calculate and disseminate the public key,  $P_A = n_A \times G, P_B = n_B \times G$ . Finally, Alice and Bob computes the secret symmetric key using the obtained public keys,  $K = n_A \times P_B, K = n_B \times P_A$ . Hence, Alice and Bob exchanged the secret key safely. It is noteworthy that strength of the ECDH approach depends upon the complexity of elliptic curves.



# Alphabetical Index

- access point, 76
- ad-hoc, 27
- adversarial threats, 84
- adversary, 21
- anonymity, 22, 23
- authentication, 20
- automatic guided vehicles, 48
- AVISPA, 21
  
- base station, 75
- bit-wise XOR, 82
- bogus request, 87
- botnet, 49
- Burrows-Abadi-Needham logic, 82
- business disruption, 52
  
- catastrophic, 74
- cellular network, 75
- Certificate Renewal, 53
- Certification Authority (CA), 50
- ciphering, 22
- cloning, 21
- cloud, 53
- commit/open pair, 78
- compatibility, 48
- computation efficient, 19
- confidentiality, 22
- counterfeiting, 81
- Cryptanalysis, 79
- cryptography, 19
- cryptography primitives, 82
- cyberattacks, 26
  
- decipher, 75
- decreased efficiency, 52
  
- Denial of Service (DoS), 22
- Device to Device (D2D), 74
- Diffie Hellman Key Exchange, 21
- disclosure attacks, 49
- Dolev-Yao, 33
- Dynamic Host Configuration Protocol, 76
  
- eavesdropping, 23
- ECC, 22
- ECQV implicit certificates, 49
- Elliptic Curve Diffie Hellman, 23
- Elliptic Curve Discrete Logarithm, 21
- Elliptic Curve Qu-Vanstone, 23
- ephemeral secret leakage attacks, 38
- execution time, 42
  
- FANUC 2000 IC robots, 53
- fatal, 74
- fifth-generation mobile networks (5G), 74
- financial and reputation loss, 52
- flood attack, 75
  
- Generator point, 55
- group owner (GO), 76
  
- HLPSL script, 33
  
- IEEE 802.11, 51
- IEEE 802.3, 51
- IEEE 802.11z, 76
- illegal access, 48
- illegitimate users, 49
- impersonation, 22
- implicit certificates, 23
- Industrial IoT networks, 20

- industrial revolution, 47
- integrity, 22
- intercepts, 38
- internet, 47
- interoperability, 48
  
- key agreement, 84
- key exchange, 19
- key management protocol, 23
- keyed-hash, 49
- known-key attack, 21
  
- latency, 74
- legitimate, 35
- Lightweightness, 53
- low powered modules, 51
  
- M2M, 19
- malicious, 49
- malware, 51
- man-in-the-middle (MITM), 19
- masquerade, 23
- Media Access Control (MAC), 85
- message freshness, 22
- mutual authentication, 19
  
- nonce, 22
  
- offline, 55
- on-the fly model-checker, 33
- overheads, 19
  
- peer-to-peer (P2P), 76
- physical capturing, 23
- physically unclonable functions, 21
- plain text, 40
- privacy, 20
- prototype, 40
  
- random number, 38
- Real-Or-Random (RoR) model, 22
- Received Signal Strength Indicator, 85
- reliability, 48
- replay attack, 22
  
- resource constrained, 21
- resource-hungry nodes, 43
- resource-rich devices, 49
- revoked certificates, 53
  
- security and privacy paradigms, 49
- security properties, 19
- session key agreement, 19
- smart environment, 25
- smart factory, 48
- symmetric key cryptography, 49
  
- tamper proof, 28
- TelosB, 40
- threat model, 23
- TinyECC, 40
- trusted channel, 79
  
- unauthorised access, 22
- untraceability, 27
  
- vulnerabilities, 20
  
- WiFi (wireless fidelity), 74
- WiFi Direct, 74
- WiFi Protected Access (WPA), 79
- WiFi Protection Setup (WPS), 76
- Wireless Sensor Networks, 22
  
- Z-Wave, 51
- Zigbee, 51