A PRIVACY AND SECURITY PRESERVING FRAMEWORK FOR SMART HOME IOT NETWORK

A THESIS

Submitted in partial fulfillment of the

Requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

NEHA SHARMA

(Enrollment No.:206203)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Waknaghat, Solan, Himachal Pradesh, India-173234

February, 2026

DECLARATION BY THE SCHOLAR

I hereby declare that the work presented in the Ph.D. thesis titled **"A Privacy and Security Preserving Framework for Smart Home IoT Network"**, submitted to **Jaypee University of Information Technology, Wakhnaghat, Solan (H.P.), India – 173234**, an authentic record of my work carried out under the supervision of **Dr. Pankaj Dhiman**. I have not submitted this work for any other degree or diploma elsewhere. I am fully responsible for the contents of my Ph.D. Thesis.

Neha Sharma
Enrolment No.: 206203
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,
Solan (H.P), India, 173234
Date:14/02/2026

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled **"A Privacy and Security Preserving Framework for Smart Home IoT Network"** submitted by **Neha Sharma**, Enrollment no. 206203 at **Jaypee University of Information Technology, Waknaghat Solan (HP), India, 173234**, is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Dr. Pankaj Dhiman

Assistant Professor

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,
Solan (H.P), India, 173234
Date: 14/02/2026

ACKNOWLEDGEMENTS

With the providential grace of **"Almighty God"**, my Ph.D. journey has come to an end, and I am filled with gratitude for everyone who provided support, faith, and effort to help me complete this path. I am immensely pleased to express my profound gratitude towards my supervisor **Dr. Pankaj Dhiman, Assistant Professor**, Department of Computer Science & Engineering and Information Technology, JUIT,
(Wakhnaghat), graciously allowed me to work under their guidance. I am thankful for his patience, continuous support, optimistic approach, never-ending deliberations, time-to-time guidance, and liberty throughout this course. I will always stay indebted to him for bearing my shortcomings with their immense sense of awareness, maturity, thorough knowledge of the specific field, and consistency.

I am grateful to our Honourable Vice Chancellor **Prof. (Dr.) Rajendra Kumar Sharma** and Dean (Research and Internationalization) **Prof.(Dr.) Sudhir Kumar** to promote the research and facilitate resources in the institution. I would also like to thank the DPMC members, **Dr. Aman Sharma**, **Dr. Nishant Sharma**, and **Prof. (Dr.) Shruti Jain**, for their thought-provoking interactive assessments, queries and opinions. Their valuable motivation, help, suggestions, affirmative vision, magnificent supervision, and enormous confidence in my abilities helped me face challenging circumstances during the research.

I am deeply grateful to my entire family for their unwavering support throughout my PhD journey. Their belief in me has kept my spirits high and my motivation strong during this process. I would also like to thank all the faculty and staff of the Department of Computer Science and Engineering and Information Technology for their scholarly guidance and support. Additionally, I appreciate my fellow PhD friends for their consistent help and valuable discussions.

TABLE OF CONTENTS

DECLARATION BY THE SCHOLAR iii

CERTIFICATE v

ACKNOWLEDGEMENT vii

LIST OF TABLES xiii

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xvii

ABSTRACT xix

1 CHAPTER - 1
INTRODUCTION 1

1.1 Internet of Things . . . 3
1.2 Characterstics of IoT . . . 3
1.3 Key Layers of IoT Architecture . . . 5
1.3.1 IoT Applications . . . 7
1.3.2 Challenges in IoT . . . 10
1.3.3 IoT Security Requirements . . . 12
1.4 Background of the Authentication in IoT . . . 13
1.4.1 Security Properties of Authentication . . . 16
1.4.2 Security Threats in IoT . . . 18
1.4.3 Identification of Authentication Schemes for IoT Applications . 20
1.5 Motivation . . . 21
1.6 Problem Statement . . . 22
1.7 Thesis Organization . . . 23

2 CHAPTER 2
LITERATURE REVIEW 25

2.1 Taxonomy of Authentication Schemes . . . 25
2.1.1 Key-Based Authentication . . . 26
2.1.2 Symmetric Key-Based Authentication . . . 26
2.1.3 Asymmetric Key-Based Authentication . . . 28

2.1.4 Identity-Based Authentication . . . 28
2.1.5 Digital Signature-Based Authentication . . . 29
2.1.6 Privacy-Preserving Authentication . . . 29
2.1.7 Factor-Based Authentication . . . 30
2.2 Formal Security Evaluation . . . 32
2.2.1 Attacker Model . . . 32
2.2.1.1 Dolev-Yao (DY) Threat Model . . . 33
2.2.2 AVISPA Tool . . . 34
2.2.3 Burrows–Abadi–Needham (BAN) Logic . . . 35
2.2.4 Real-or-Random (ROR) Model . . . 35
2.3 Simulator Tools . . . 36
2.3.1 NS-3 . . . 36
2.4 Related Work . . . 36
2.4.1 Security Methods . . . 37
2.4.2 Existing Lightweight Authentication Schemes . . . 38
2.4.3 Global Addressing Methods . . . 50
2.4.3.1 IoT address allocation . . . 52
2.4.4 Existing Addressing Schemes . . . 52
2.5 Research Gaps . . . 55
2.6 Research Objectives . . . 56
2.7 Research Contribution . . . 56
2.8 Summary . . . 58

3 CHAPTER 3
Multifactor Unidentified Remote User Authentication Scheme for IoT Network . . . 59
3.1 Introduction . . . 59
3.2 Main Contribution of the Proposed Scheme . . . 60
3.3 Related Work . . . 61
3.4 Networking Model and Authentication Mechanism . . . 62
3.4.1 Authentication Procedure . . . 63
3.4.2 Bio-Hash functions . . . 64
3.5 Proposed Scheme . . . 64
3.5.1 Registration Phase . . . 65
3.5.2 Login and Authentication Phase . . . 66
3.5.3 Password Change Phase . . . 68
3.5.4 Revocation Phase . . . 68
3.5.5 Security and Efficiency . . . 69
3.6 BAN Logic-Based Authentication Proof . . . 69
3.6.1 BAN Logic Notations . . . 70
3.7 AVISPA tool simulation for formal security verification . . . 73
3.8 Performance Evaluation . . . 73
3.9 Summary . . . 77

4 CHAPTER - 4
Secure Mutual Addressing Authentication Mechanism for Smart IoT Home Network **79**
4.1 Introduction . . . 79
4.2 Main Contribution of the Proposed Scheme . . . 81
4.3 Related Work . . . 82
4.3.1 Addressing Techniques . . . 82
4.3.2 Security Techniques . . . 83
4.4 System Model . . . 84
4.5 Attacker Model . . . 85
4.6 Proposed Scheme . . . 86
4.6.1 Phase 1: Installation and Network Configuration . . . 88
4.6.2 Phase 2: Addressing and Identification . . . 89
4.6.3 Phase 3: Registration Phase . . . 89
4.6.4 Phase 4: Authentication, Login, and Key Agreement Phase . . . 90
4.6.5 Phase 5: Password Update Phase . . . 92
4.7 Informal Security Analysis . . . 93
4.7.1 Mobile User Impersonation Attack . . . 93
4.7.2 Home Gateway Impersonation Attack . . . 94
4.7.3 Smart Device Impersonation Attack . . . 94
4.7.4 Session Key Disclosure Attack . . . 94
4.7.5 Replay and MITM Attacks . . . 95
4.7.6 Anonymity and Untraceability . . . 95
4.7.7 Offline Guessing Attack . . . 95
4.7.8 Stolen Device Attack . . . 96
4.7.9 Mutual Authentication . . . 96
4.7.10 Perfect Forward Secrecy . . . 96
4.7.11 Desynchronization Attack . . . 97
4.8 Network Simulation Using NS-3 . . . 97
4.8.1 Simulation Setup . . . 98
4.8.2 Throughput Analysis . . . 100
4.8.3 End-to-End Delay (E2E) . . . 100
4.9 Formal Security Analysis . . . 100
4.9.1 Formal Security Analysis Using ROR Model . . . 100
4.9.2 Formal Security Analysis Using AVISPA Tool . . . 102
4.10 Performance Evaluation . . . 104
4.10.1 Functionality Comparison . . . 105
4.11 Communication Cost . . . 106
4.12 Computational Cost . . . 108
4.13 Conclusion . . . 109

5 CHAPTER 5
Reliable and Secure Addressing with Authentication in IoT Networks **111**
5.1 Introduction . . . 111

5.2 Main Contribution of the Proposed Scheme . . . 112
5.3 Related Work . . . 113
5.3.1 Addressing Techniques . . . 113
5.3.2 Security Techniques . . . 114
5.4 System Model and Attacker Model . . . 115
5.4.1 System Model . . . 115
5.4.2 Attacker Model . . . 116
5.4.3 Assumptions and Notation . . . 117
5.5 Proposed Scheme . . . 117
5.5.1 Network Phase . . . 118
5.5.2 Addressing Phase . . . 119
5.5.3 System Installation Phase . . . 119
5.5.4 Registration and Login Phase . . . 120
5.5.5 Key Establishment and Authentication Phase . . . 120
5.5.6 Password Update Phase . . . 123
5.6 Informal Security Analysis . . . 124
5.6.1 Mutual Authentication Attack (MAA) . . . 124
5.6.2 Device Impersonation Attack (DIA) . . . 124
5.6.3 Anonymity Attack . . . 125
5.6.4 Untraceability . . . 125
5.6.5 Replay Attack . . . 126
5.6.6 Man-in-the-Middle Attack (MITM) . . . 126
5.6.7 Denial-of-Service Attack (DoS) . . . 126
5.7 Formal Security Analysis Using RoR Model . . . 127
5.7.1 Formal Security Verification Using AVISPA Tool . . . 129
5.8 Performance Evaluation . . . 132
5.8.1 Communication Cost Analysis . . . 133
5.8.2 Computational Cost Analysis . . . 134
5.9 Summary . . . 135

6 CHAPTER - 6
CONCLUSIONS AND FUTURE RESEARCH SCOPE **137**
6.1 Conclusion and Future Work . . . 137

LIST OF PUBLICATIONS **139**

References **141**

LIST OF TABLES

2.1 Comprehensive review of Remote Authentication and Security Schemes 45
2.2 Comparison of techniques across different approaches 54

3.1 List of symbols and their descriptions 65
3.2 Registration Phase: Secure initialization of identity and parameters . . . 66
3.3 Login and Authentication Phase: Establishing mutual trust and session key 67
3.4 Password Change Phase: Secure update of user credentials 68
3.5 Revocation Phase: Recovery from compromised or lost credentials . . . 69
3.6 Functionality and Security Attribute Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol 75

4.1 Symbols and Their Descriptions 88
4.2 Mobile User Registration Phase in the Proposed Authentication Protocol 89
4.3 Smart Device Registration Phase in the Proposed Authentication Protocol 90
4.4 Login and Key Establishment in the Proposed Authentication Protocol . 91
4.5 Password Update Phase in the Proposed Authentication Protocol 93
4.6 NS-3 Simulation Parameters 98
4.7 Comparison of Performance and Security Features 106

5.1 List of Symbols and Their Descriptions 118
5.2 Registration and Login Phase 121
5.3 Key Establishment and Authentication Phase 122
5.4 Password Update Phase 123
5.5 Comparison of Functionality and Security Attributes 135

LIST OF FIGURES

1.1 Components of IoT 4
1.2 Specific characteristics of IoT 5
1.3 Layered architecture of IoT 6
1.4 Various applications in the IoT architecture 7
1.5 Authentication challenges in the IoT 10
1.6 Security Requirements of IoT 13
1.7 Background of the authentication in IoT 14
1.8 Key properties of authentication in IoT systems 17

2.1 Global unicast address 51
2.2 Link local address 51
2.3 Unique-local address 52
2.4 EUI-64 based IID 53

3.1 Proposed Model for user authentication in the IoT 60
3.2 Role for user and gateway node 72
3.3 Role for session and environment 73
3.4 OFMC output 74
3.5 Communication Cost Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol 75
3.6 Computation Cost Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol 76

4.1 Appliances for smart home systems 80
4.2 Modified IPv6 Protocol 82
4.3 Illustration of IPv6 Address Format Assignment for Smart Home Devices and User Devices in a Secure IoT Network 86
4.4 Stages of a unique addressing scheme 87
4.5 Network topology using NS-3 98
4.6 Network simulation results 99
4.7 Role for User 104
4.8 Role of Session and Environment 105
4.9 AVISPA result using OFMC and CL-AtSe 106
4.10 Comparison of Communication cost 107
4.11 Comparison of Computation cost 109

5.1 Role for user 130
5.2 Role of Home Gateway and session 131
5.3 AVISPA result using OFMC 132
5.4 Comparison of Communication cost 133
5.5 Comparison of Computation cost 134

LIST OF ABBREVIATIONS

ICT	:	Information and Communications Technology
IID	:	Interface Identifier
MIID	:	Modified Interface Identifier
SHG	:	Smart Home Gateway
MIMA	:	Man in the Middle Attack
IoT	:	Internet of Things
IP	:	Internet Protocol
GUA	:	Global Unicast Address
M2M	:	Machine to Machine
SH-IoT	:	Smart Home Internet of Things
IIoT	:	Industrial Internet of Things
IoV	:	Internet of Vehicles
V2V	:	Vehicle-to-Vehicle
V2R	:	Vehicle-to-Roadside
SSN	:	Social Security Number
IETF	:	Internet Engineering Task Force
SHA	:	Secure Hash Algorithm
SP	:	Service Provider
DSRC	:	Dedicated Short Range Communications
UAF	:	Unicast Address Format
GWN	:	Gateway Node
RSA	:	Rivest–Shamir–Adleman
VPN	:	Virtual Private Networks
LLA	:	Link-local address
ULA	:	Unique Local Address
WSN	:	Wireless Sensor Networks

FNC	:	Federal Networking Council
AES	:	Advanced Encryption Standard
HMAC	:	Hash-based Message Authentication Code
VANET	:	Vehicle Ad Hoc Networks
AUSS	:	Authenticated Unidentified Security Scheme
SUMAS	:	Secure and Unique Addressing with Mutual Authentication Scheme
SLAPSH	:	Secure and Lightweight Authenticated Protocol for Smart Home

ABSTRACT

Global advances in connectivity and digital technology have created a growing demand for seamless data sharing and interaction between devices, systems, and people. The Internet of Things (IoT) is at the forefront of this trend, linking physical objects such as home appliances, vehicles, sensors, and industrial equipment to the Internet, where they can communicate and exchange information. While the IoT delivers increased efficiency, cost-effectiveness, and sustainability, especially in environments with limited resources, it also faces challenges due to the constrained processing power, memory, and energy of many devices. With forecasts estimating that IoT-connected devices will surpass 40 billion by 2025, the risk of security breaches and insecure is also arises in IoT environment. Malicious individuals may take advantage of vulnerabilities, which could compromise user privacy and result in the trade of sensitive data.

To address these security concerns, the first goal of this thesis is the development of the Authenticated Unidentified Security (AUSS) Scheme. AUSS implements robust security through multifactor authentication, utilizing biometrics, passwords, and device identifiers, while preserving the privacy of lightweight IoT devices through efficient cryptographic techniques. This scheme is designs to resist various attacks, such as impersonation, replay, offline guessing, man-in-the-middle, denial-of-service threats, and many more. AUSS's security is thoroughly analyzed using the ROR model and formal verification with the AVISPA tool.

The second objective introduces the Secure and Lightweight Authenticated Protocol for Smart Homes (SLAPSH). SLAPSH uses and modifies the standard IPv6 interface identifier to enable destination-specific identity verification, maintaining compatibility with existing standards. By assigning a unique 64-bit identifier to each smart device or appliance, the protocol allows for fine-grained authentication and secures communication using session keys. The security of SLAPSH is evaluated using both mathematical analysis and formal tools, including OFMC and CL-AtSe. This scheme demonstrates effective resistance to attacks such as offline password guessing, forward secrecy breaches, session key disclosure, and denial-of-service, with its performance evaluated against existing methods based on throughput, latency, packet loss, and access times, resulting in notable gains in both security and efficiency.

The final and most comprehensive contribution of this thesis is the development of the Secure and Unique Addressing with Mutual Authentication (SUMAS) scheme, which

introduces a novel approach to structuring IPv6 addresses specifically to improve identity management and authentication in smart home IoT networks. Unlike traditional schemes that treat interface identifiers as opaque values, SUMAS redefines the 64-bit IPv6 interface identifier by logically partitioning it into a 48-bit Owner ID, which uniquely identifies the user, and a 16-bit Device ID, which corresponds to the individual smart device. This structural refinement facilitates more granular control over user-device relationships and enables scalable and secure address assignment mechanisms. By embedding authentication-relevant information directly within the addressing framework, SUMAS significantly simplifies the mutual authentication process between smart devices and central servers or gateways. It eliminates the need for external identity verification services and reduces protocol overhead, making it particularly suited for environments with constrained computational and energy resources. To ensure the robustness of the security scheme, a two-tiered evaluation was conducted. First, an informal analysis using the ROR model demonstrated that SUMAS is theoretically resilient to key-based attacks. Second, formal verification with the AVISPA tool provided automated proofs of its resistance against various known network threats, including masquerading, device compromise, man-in-the-middle attacks, and replay attacks. Together, these evaluations confirm that SUMAS offers a reliable, lightweight, and scalable solution for secure identity and access management in modern IoT-enabled smart homes.

This thesis summarises the key contributions from the design, development, and thorough evaluation of three novel security schemes—AUSS, SLAPSH, and SUMAS. Each scheme specifically addresses vulnerabilities within smart home IoT environments. Additionally, the thesis outlines future directions for enhancing secure communication in these settings. The proposed roadmap emphasizes the importance of continuously improving privacy and security measures to keep pace with the ongoing evolution of IoT technology.

CHAPTER - 1
INTRODUCTION

”Internet of Things” (IoT) refers to a network of interconnected devices equipped with sensors and communication technology. These devices can collect and share data over the Internet using various communication methods. It is rapidly expanding, creating a constantly growing network of physical devices connected and has the potential to collect, distribute, and analyse data. It can significantly change the dynamics of smart IoT technologies. The IoT offers solutions to a range of global issues, including enhancing energy efficiency, alleviating traffic congestion, and improving both the accessibility and quality of healthcare. However, the IoT ecosystem also raises numerous concerns regarding privacy and security. These issues stem from the diversity of IoT devices, the variety of data sources, the differing requirements of participants, and the seamless sharing of data among multiple users and devices [1] As the IoT ecosystem expands, the volume and variety of connected devices are increasing rapidly. In 2020, the global IoT market was valued at approximately 250 billion and is expected to rise to around 1.1% trillion by 2026, representing a compound annual growth rate (CAGR) of about 27%. The number of IoT devices worldwide has exceeded 30 billion and is projected to approach 50 billion by 2030. In the industrial sector, nearly 70% of businesses are currently employing or planning to adopt IoT solutions, while smart home devices have reached over 40% of households in developed countries. The IoT has a significant economic impact, potentially contributing up to 15% trillion to the global economy by 2030 due to increased efficiency, cost savings, and new revenue opportunities [2]. IoT devices encompass a wide range of technologies, from home appliances to industrial sensors. They find applications in various areas, including healthcare, smart homes, and industrial automation. When processing the collected data—whether at the edge or in the cloud—it is essential to prioritise security. This includes implementing measures such as encryption and authentication. The rise of these devices generates vast

amounts of data, which raises significant privacy concerns, especially since this data can include sensitive personal and health information [3]. For example, smart energy meters can reveal how individuals use household appliances. Also, energy conservation is a critical challenge for battery-powered devices, making effective energy management strategies, such as low-power modes and optimised communication protocols, essential for extending battery life and reducing the need for replacements [4]. These strategies are vital in remote or resource-constrained environments where IoT transforms by delivering real-time data and enabling remote monitoring of homes, workplaces, and critical infrastructure. In such contexts, specific applications like the Internet of Military/Battlefield Things (IoMT/IoBT) are increasingly being deployed across sectors, including agriculture, defence, and hazardous environments, where both energy efficiency and reliable communication are paramount. However, security remains a significant concern; many IoT devices are vulnerable to hacking due to the presence of overlooked security features. There is an urgent need to encrypt sensors in previously unprotected areas while balancing functionality with energy efficiency. Protecting user privacy, ensuring service availability, and maintaining data integrity are crucial objectives. Authentication is the primary security mechanism, and ongoing research focuses on lightweight encryption solutions for low-power devices. Implementing robust authentication and anomaly detection is vital for improving IoT security [5].

Standards and protocols are essential for ensuring interoperability among IoT devices. Authentication plays a critical role in protecting sensitive data and preventing unauthorized access. However, traditional security solutions often fail to address the unique challenges posed by IoT devices. Many of these devices remain vulnerable due to the use of unencrypted communications and weak passwords. Cyberattacks, such as the 2016 Mirai botnet attack, highlight the risks associated with insecure IoT devices, particularly in industries where security vulnerabilities can have severe consequences [6].

Moreover, many IoT devices are in unprotected areas, making them susceptible to physical attacks. For instance, on-chip non-volatile memory (NVM) that stores secret keys can be compromised, potentially leading to unauthorized access. Therefore, thorough research into IoT security is critical to protect against these risks. This thesis examines authentication systems and secure data-sharing solutions for the IoT environment to protect sensitive information from unauthorized access. This work aims to develop scalable authentication solutions for IoT based on existing literature, empirical research, and real-world scenarios [7].

1.1 Internet of Things

IoT has revolutionized device communication through technological advancements. It refers to a network of connected items, including physical objects and smart devices equipped with sensors and software to share data over the Internet. IoT improves efficiency and service quality in various industries. The concept originated in the 1980s when David Nichols developed a program to remotely monitor Coke vending machines, one of the earliest known IoT devices. In the 1990s, the growth of the Internet and wireless sensor networks led to widespread adoption in the manufacturing, transportation, and healthcare sectors. The term ”Internet of Things” was coined in 1999 to describe the ability to connect physical objects online [8, 9]. In the early 2000s, the advent of low-cost wireless sensors and RFID tags enabled the large-scale deployment of the IoT, creating a global network of billions of devices. Smart cities, wearable technology, healthcare, and traffic monitoring are key applications. Real-time data access is crucial for IoT applications, especially in critical scenarios such as surveillance. Cloud platforms provide essential storage and processing capabilities for IoT devices [10]. Cloud computing enables on-demand access to resources from anywhere, allowing organizations to store and analyze data remotely. The advantages include cost savings, scalability, and ease of use. An IoT network comprises end-users, an interface (or gateway), Internet-connected objects, and the cloud. IoT sensors collect data and transmit it to the cloud via wireless sensor networks. The gateway node aggregates this data and connects to the Internet, allowing communication with the cloud. Users can access cloud-stored data via mobile applications and web portals [11, 12].

1.2 Characterstics of IoT

IoT has distinct traits that distinguish it from traditional computing paradigms. These traits allow IoT systems to operate autonomously, intelligently collect and exchange data, and deliver actionable insights across various domains. Understanding these features is essential for designing secure, efficient, and scalable IoT architectures. They are vital for developing IoT applications. Figure 1.1 illustrates the core components of the IoT, while Figure 1.2 outlines its key characteristics. The key characteristics of IoT include:

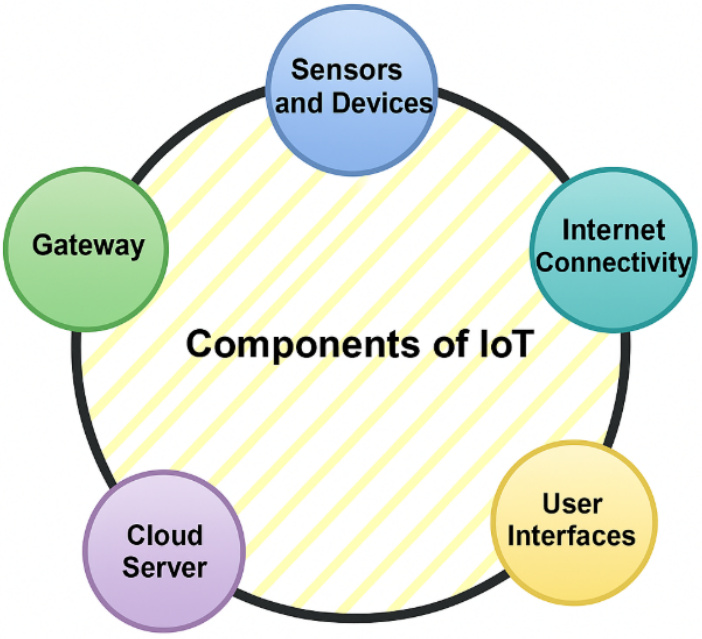


FIGURE 1.1: Components of IoT

1. Sensors: IoT devices use multiple sensors to collect data from their surroundings. These sensors measure temperature, humidity, light, motion, and physical events, depending on the device's intended use [12].

2. Connectivity: IoT devices use WiFi, Bluetooth, or cellular networks to connect to the Internet.

3. Identity: IoT devices require a unique identity to be recognized and authenticated by other devices and systems. Devices can be identified using many factors, such as their MAC address, IP address, or serial number.

4. Communication: Communicate between the IoT devices and other systems can be done via the Internet. It enables them to exchange information, coordinate operations, and respond to events.

5. Data Acquisition and Processing: IoT devices use sensors to collect and process environmental data. The data can be processed on the device or sent to a cloud server.

6. Dynamic Nature: The dynamic properties of IoT devices refer to their ability to respond to changing situations.

7. Privacy: In the context of IoT, privacy refers to how devices and systems collect, use, and share personal data.

8. Scalability: As the number of IoT nodes in a network grows, solutions must efficiently manage the resulting surge in data traffic. IoT systems must respond to environmental changes while maintaining performance and dependability.

9. Security: The constant Internet connectivity of IoT devices makes them vulnerable to cyberattacks, making security a key concern. Encryption, authentication, and access control can be used to secure IoT devices [13].

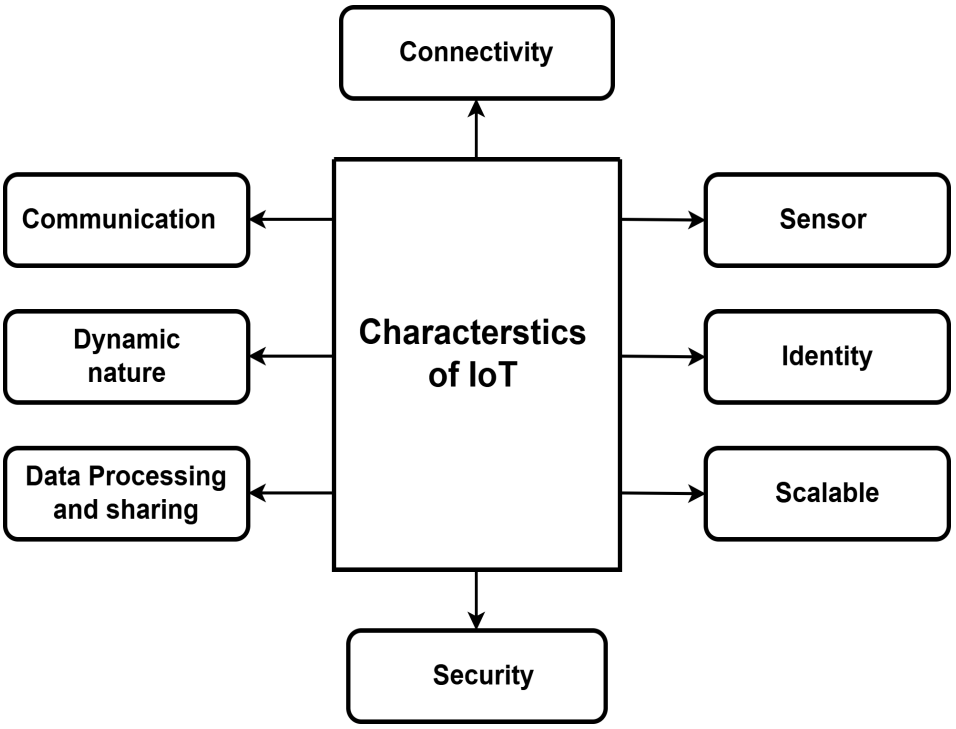


FIGURE 1.2: Specific characteristics of IoT

1.3 Key Layers of IoT Architecture

The IoT architecture refers to the design and structure of devices that facilitate communication with other systems and devices. It delineates the components, functions, and interactions of an IoT system. Figure 1.3 illustrates the IoT architecture, which includes the device, communication, processing, and application layers, with each layer managing a specific phase in the data lifecycle. The data collected may be utilized to automate operations or deliver user services. The IoT architecture comprises the following layers [14].

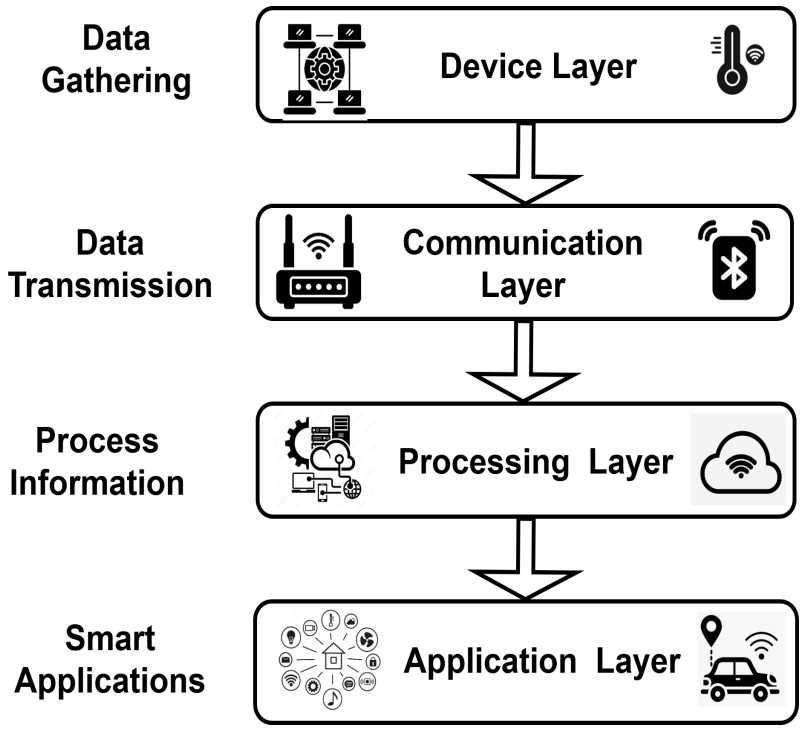


FIGURE 1.3: Layered architecture of IoT

1. Perception/Device Layer: The Perception/Device Layer, the first layer of IoT architecture, collects data from sensors and devices in the physical environment. Sensors can measure many environmental factors, including air quality, mobility, and temperature.

2. Network/Communication Layer: It transfers data from the device layer to the cloud or other devices. IoT devices can communicate via Wi-Fi, Bluetooth, Low-Power, Wide-Area Networks, and wired connections.

3. Middleware/Processing Layer: It stores and processes enormous amounts of data transmitted from the network layer. The processing layer can utilise this data to enhance IoT system performance and provide new services to users.

4. Application Layer: It delivers services to users based on data collected and processed by other layers. An IoT application can give users real-time energy consumption and health information. IoT applications can automate processes like building temperature control and room lighting [15, 16].

1.3.1 IoT Applications

IoT applications have significantly enhanced convenience, safety, and efficiency across various areas. Smart healthcare gadgets use IoT technology to monitor vital signs and diagnose ailments remotely, promoting well-being and sustainability, or smart agricultural systems that maximize crop yields while preserving resources. Connected vehicles revolutionize mobility by decreasing accidents and congestion. Smart cities benefit from efficient energy usage and real-time monitoring of their infrastructure. The IoT has transformed companies, homes, and public services, enhancing quality of life, promoting economic growth, and tackling global concerns [27]. Figure 1.4 illustrates the multi-layered structure of an IoT system, spanning from physical sensing elements to high-level smart applications across various domains. The following are some real-world applications of IoT.

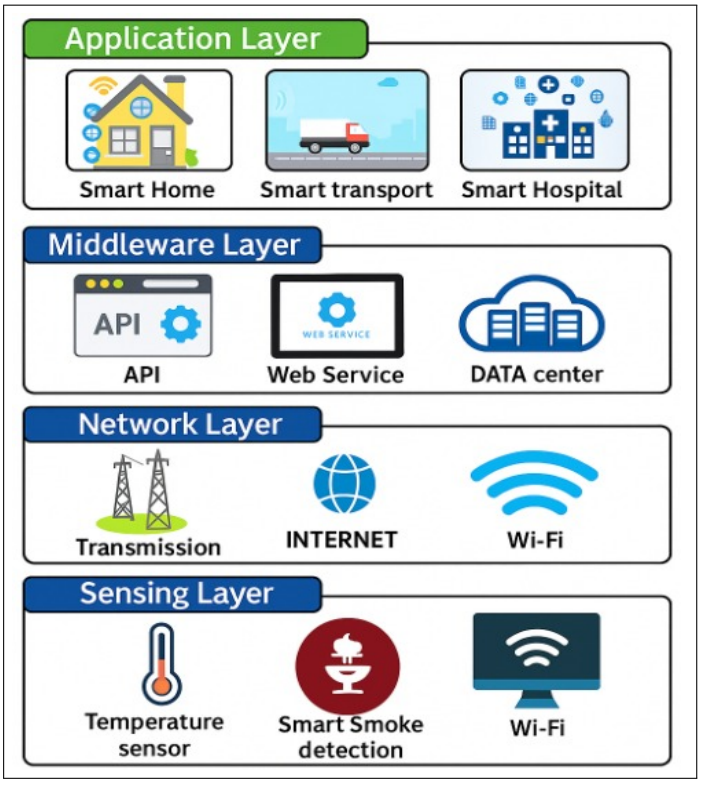


FIGURE 1.4: Various applications in the IoT architecture

1. **Smart Home**: Smart technology and IoT devices can automate and organise multiple household tasks. Homes can remotely monitor and manage appliances, lighting, security cameras, thermostats, and other equipment using a centralised hub or mobile app. Smart home automation improves convenience, energy efficiency,

and security, making homes more comfortable and effective. Smart sensors can analyse physiological data to assess an individual's emotional state and modify the environment accordingly. Furthermore, an intelligent electronic heater may adjust the room temperature autonomously. Smart home apps, such as the CURB energy intelligence system and Philips Hue wireless lighting system, are available on the market. The CURB technology enables customers to remotely manage their home's temperature and anticipate future energy expenses. The Philips Hue system allows for voice control, scheduling, routine creation, and colour experimentation through mobile apps [28].

2. **IoT-based e-Healthcare System**: E-healthcare and IoT are interrelated concepts with the potential to revolutionise the healthcare business. The IoT in healthcare integrates devices, systems, and stakeholders to deliver better patient care, increased efficiency, and proactive management through real-time monitoring. Electronic health records (EHRs) store patient health data in distributed data centres and the cloud, allowing for remote access and quick diagnosis. IoT devices can capture patient physiological information, including ECG, blood pressure, and temperature [29]. IoT technology is crucial for healthcare applications, enabling operations that can be conducted without human intervention. Healthcare monitoring and services rely on heterogeneous networks of IoT devices communicating through WSN technologies. The BioStrap is a wearable wristband and shoe clip that measures blood oxygen saturation, heart rate, and sleep quality. BioStrap's smartphone apps provide a quick and effective way to measure health and fitness progress. IoT devices rely on gateway nodes, such as mobile phones or routers, for communication due to their limited computational power and capacity. Due to their power and capacity, IoT devices require a gateway node, such as a mobile phone or router, for communication [30].

3. **Smart Grid IoT**: The IoT plays a crucial role in transforming traditional power grids into smart grids, thereby enhancing consumer energy efficiency and reliability. Smart networks optimise the utilisation of distributed energy resources and electric vehicles, improving energy storage capacity while reducing carbon dioxide emissions. Implementing bidirectional communication networks and smart meters enhances communication between utility companies and their customers, facilitating a more efficient and effective exchange of information. IoT integration facilitates the deployment of smart meters in homes and buildings, linking them

to smart grid communication networks. Smart meters are designed to monitor energy output, storage, and consumption. They can contact utility companies and report their clients' energy usage [31].

4. **Industrial IoT**: IIoT refers to the application of IoT in industrial contexts, often referred to as Industry 4.0. IIoT involves linked devices and sensors collecting data from industrial machinery, equipment, and production processes. Cloud analytics can enhance industrial operations by providing insights for predictive maintenance, process optimisation, and resource allocation, resulting in increased efficiency and reduced downtime. Cloud-based IIoT offers remote monitoring and control, allowing operators to optimise processes from anywhere. Strong security measures are necessary to secure sensitive industrial data and prevent cyber threats [32].

5. **Internet of Vehicles (IoV)**: IoV evolved from IoT's integration with ITS. Vehicles in the IoV utilise communication technologies such as Wi-Fi, cellular networks, and Dedicated Short Range Communications (DSRC) to connect to the Internet and exchange information with infrastructure and cloud platforms. ITS seeks to enhance safety and comfort for passengers and drivers while optimizing road traffic management systems. IoV vehicles employ On-Board Units (OBU) to communicate with RSUs and other vehicles via DSRC. The IoV architecture supports Vehicle-to-Vehicle (V2V) and Vehicle-to-Roadside (V2R) communication. V2V allows bidirectional information exchange, including position, speed, traffic conditions, and collision data. Vehicles can broadcast their location, speed, and direction up to 10 times per second, offering a comprehensive view of their surroundings [33]. According to [34], drivers and passengers can access Internet services via V2R. VANET, an ITS program that utilises an ad-hoc wireless connection, has limitations due to changing network topology, resulting in service interruptions and reduced application accuracy. The IoV improves the safety and convenience of V2V and V2R interactions by upgrading the VANET system. Implementing IoV can reduce road traffic congestion by 60% and increase short-distance transit efficiency by 70%, as reported in [35, 36]. IoV is vital for connecting cars, sensors, actuators, and humans, providing a user-friendly experience with flexible, scalable, and seamless connections while consuming low power.

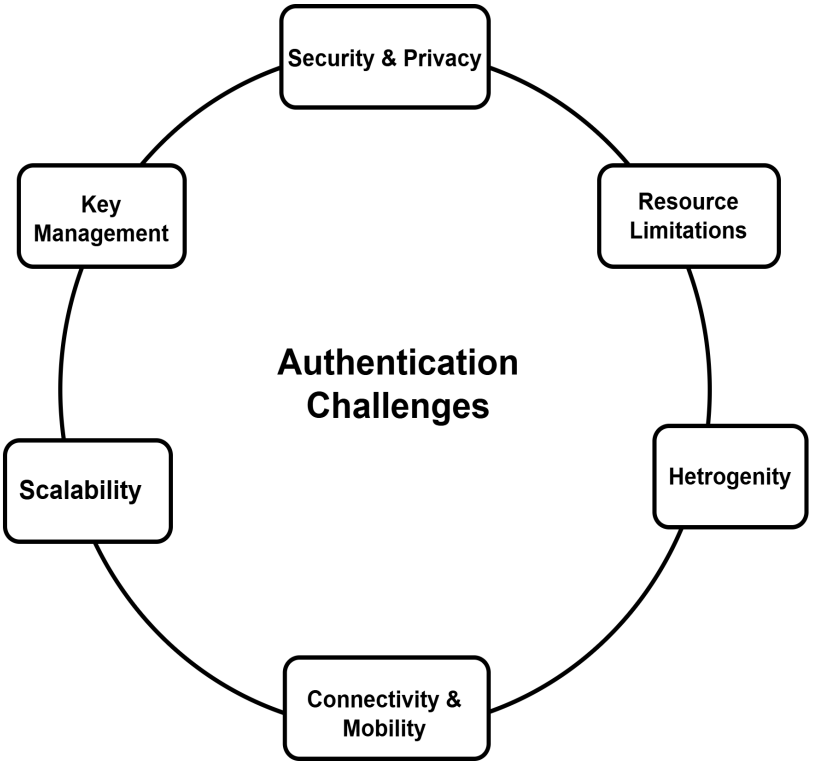


FIGURE 1.5: Authentication challenges in the IoT

1.3.2 Challenges in IoT

Ensuring security within the IoT ecosystem is complex due to its massive scale, the heterogeneity of devices, and the extensive interconnectivity. Figure 1.5 highlights the fundamental difficulties encountered when developing robust and efficient authentication solutions for IoT environments.

1. Scalability: The IoT ecosystem encompasses a wide range of devices, from small sensors to large-scale industrial systems. Designing authentication procedures for numerous devices and users in an IoT environment is challenging due to the need for scalability and flexibility. Authentication systems must efficiently manage and authenticate multiple devices and users while maintaining performance and security.

2. Resource Limitation: Authentication solutions for IoT devices should be optimised for limited computational resources, such as processor power, memory, and energy. This optimisation aims to reduce both computational overhead and

energy consumption. Using lightweight cryptographic protocols, efficient authentication algorithms and optimized communication protocols is crucial for efficient authentication in resource-constrained IoT devices [53].

3. Heterogeneity: Authentication in the IoT ecosystem is challenging due to the diversity of devices, operating systems, platforms, and communication protocols. Interoperability and compatibility between these heterogeneities are crucial to secure communication and seamless authentication.

4. Connectivity and Mobility: Authentication mechanisms for IoT devices must adapt to dynamic network conditions, including frequent connection and disconnection, changing network configurations, and device mobility. Ensuring continuous and secure authentication is crucial as devices move within the IoT infrastructure. This requires seamless handover and re-authentication mechanisms.

5. Security and Privacy: IoT devices handle sensitive data, making security and privacy a vital concern. Authentication systems must resist various threats, including password guessing, impersonation, eavesdropping, and brute-force attacks. Authentication mechanisms and transferred data must be available, confidential, and of high integrity. Addressing privacy concerns about user authentication credentials and personal information is crucial to avoid unauthorized access or data breaches [54].

6. Key Management: IoT authentication uses cryptographic keys for secure communication and protocols. Managing keys securely across multiple IoT devices is a challenging task. Effective key generation, distribution, revocation, and update processes prevent unauthorised access, compromise, or misuse. Key management solutions should be scalable, efficient, and resistant to key-based attacks [55]. Encryption is critical in the IoT to protect the privacy, security, and integrity of data sent through networked devices. Encryption protects sensitive information collected and transmitted by IoT devices, reducing the danger of unauthorized access and interception. It is critical for combating cyber risks, including data tampering, unauthorized access, and eavesdropping, since it protects communication channels, authenticates devices, and enforces access control. Beyond addressing specific cybersecurity concerns, encryption facilitates regulatory compliance, particularly in businesses with stringent data protection regulations. Encryption is a fundamental component of the IoT security framework, providing robust protection against vulnerabilities and ensuring the trustworthiness of the

entire IoT ecosystem. Encryption must be used on unused resources and when the application processor is running; therefore, it should be as lightweight as possible [56, 57].

1.3.3 IoT Security Requirements

As IoT continues to expand across various critical sectors, including healthcare, smart homes, the military, and industrial automation, securing its infrastructure has become a top priority. Due to the vast scale, heterogeneity, and resource constraints of IoT devices, conventional security mechanisms often fail to provide adequate protection. Therefore, specific security requirements must be addressed to ensure the protection of data, devices, and communication networks. Key security considerations in the IoT domain include:-

1. Availability: Availability is a critical security requirement for IoT, emphasizing protection against disruptions, especially DoS attacks. Ensures that connected devices remain accessible and responsive, safeguarding continuous operation despite possible malicious attempts to disrupt services. This requirement is essential to ensure the overall reliability and proper functioning of the IoT ecosystem [23].

2. Confidentiality: To ensure confidentiality in IoT, encryption prevents unauthorized access, secures data, and protects personal information from breaches.

3. Integrity: IoT devices and systems require secure data collection and transmission. This involves using cryptographic hash algorithms to detect attempts at tampering and falsification. These IoT techniques authenticate the data, ensuring its integrity and security.

4. Authentication: In IoT, confirming the identities of the devices and users is essential to ensure secure and authorized access to network resources. Cryptographic techniques, such as digital signatures and certificates, can help achieve this goal [25].

5. Authorization: It refers to providing or limiting access permissions to devices or users based on their authenticated identities. After adequately authenticating a device or user, authorization establishes their level of access and permissions in the IoT ecosystem. This ensures that only authorized organizations can perform specific actions or access resources [24].

6. Access Control: The access control system enforces permissions and prevents unauthorized resource access. Figure 1.6 illustrates the security requirements for the IoT [26].

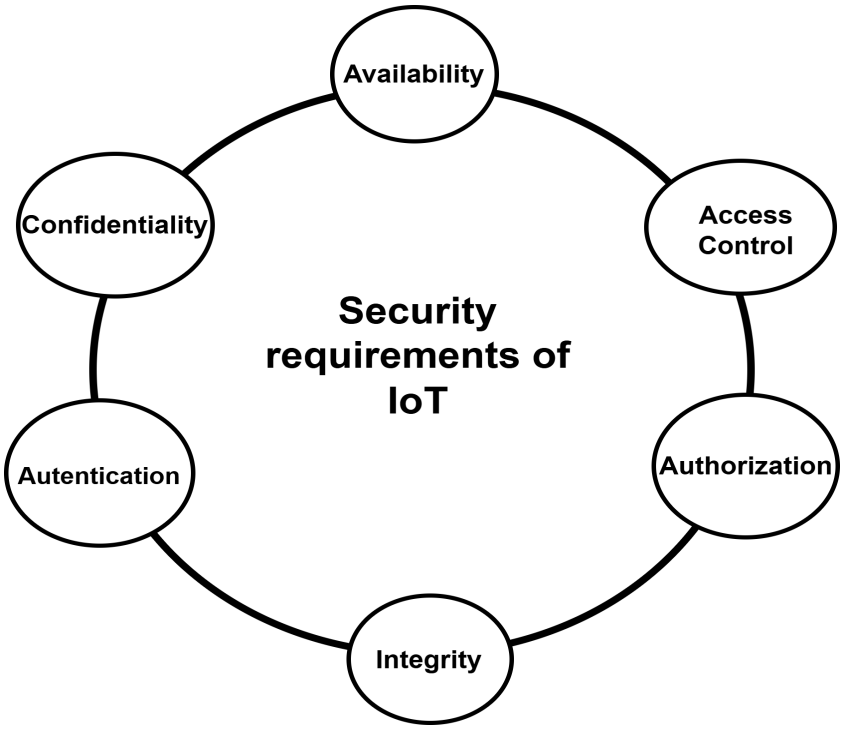


FIGURE 1.6: Security Requirements of IoT

1.4 Background of the Authentication in IoT

Authentication schemes have evolved significantly over time, with their origins dating back to ancient civilizations. The earliest forms of authentication involved obtaining verification from a trusted source that provided clear evidence of a person's identity. For example, in ancient times, a monarch would grant a seal to a trusted advisor to confirm their identity. Over the years, various authentication systems have been developed to ensure data confidentiality [41]. Authentication is a foundational security mechanism that allows only verified and trusted entities, such as users, devices, or systems, to access protected resources or networks. It plays a crucial role in maintaining the confidentiality, integrity, and availability of data in digital environments. The evolution of authentication is illustrated in Figure 1.7.

1. In the 1960's:- In the 1960s, passwords were first employed for authentication. They offer a straightforward and efficient method for verifying users. Upon initial

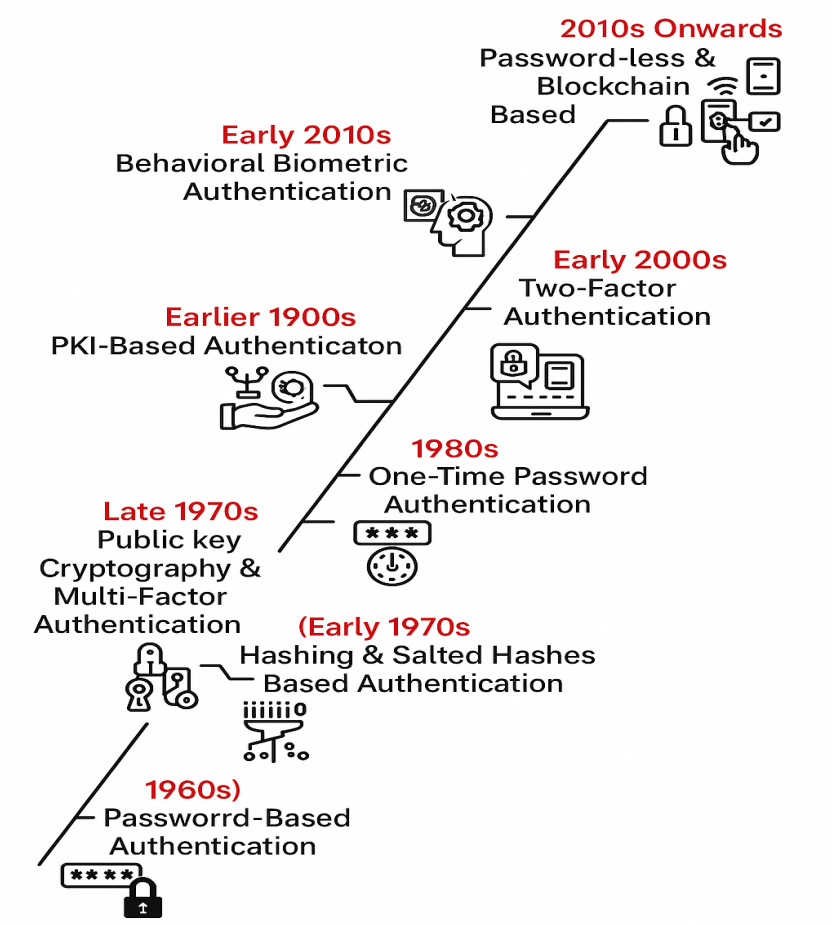


FIGURE 1.7: Background of the authentication in IoT

login to a service, users create a password. This password confirms the user's identity while using the service. However, it is also susceptible to attacks. Passwords can be guessed using brute-force, dictionary, or rainbow table techniques. Additionally, they can be compromised through spyware or phishing operations. Furthermore, storing the password database requires additional memory space. The original password authentication technique was designed for use over insecure communication channels [42].

2. In the early 1970s:- Robert Morrison introduced salted hashing to address the security issue of storing passwords in plaintext. This technique involves adding a random string, called a 'salt', to a password. Salting enhances security by making it more difficult for attackers to crack password hashes, even when they have access to the database. Hashing is a one-way function that simplifies the calculation of password hashes but complicates the recovery of the original password. Random and user-specific salts ensure that each password hash is unique, thereby

bolstering password security. Salted hashes can still be cracked; however, adding salt significantly raises the difficulty of breaking a password compared to plaintext [43].

3. In the mid-1970s:- Asymmetric cryptography is also known as Public Key Infrastructure. It utilizes both a secret key and a public key. The private key remains confidential, while the public key is disseminated widely. This dual-key approach enables the generation of digital signatures, which are crucial for verifying the authenticity and integrity of messages and documents. Digital certificates serve to authenticate individuals and organizations. CAs, or Trusted Third Parties, issue certificates [44].

4. In the 1980s:- The security sector aimed for stronger authentication methods compared to static passwords. Cybercriminals can illegally access systems and data by stealing, intercepting, or guessing these passwords. To address this issue, a new authentication approach known as One-Time Passwords (OTPs) was introduced. OTPs are randomly generated and valid for just one login attempt, making unauthorized access more challenging. There are two key issues in developing OTP authentication:

   - How can you create random passwords that the system recognizes as legitimate?
   - How do you provide passwords to users?

   The first problem was addressed by utilising a time-based technique to create passwords. The second difficulty was addressed by giving credentials via a physical token or mobile app. OTP specifications now include hash-based, event-based, and challenge/response techniques. OTP distribution has been altered, eliminating the need for physical tokens. OTPs can now be sent via SMS, email, or mobile app [45].

5. In the late 1990s:- With the advent of the World Wide Web, the late 1990s saw the necessity of PKI to address security challenges in online transactions and information exchange. PKI became essential for verifying user and website identities and encrypting data in transit. The SP4 protocol, later renamed TLS, was introduced for secure internet communication, utilizing asymmetric cryptography. The SSL protocol, a precursor to TLS, developed in the early 2000s, is still used on some websites, providing server authentication and data encryption during transit.

PKI is still widely used for Internet security by organisations such as banks and government agencies [46].

6. In the 2000s, there was a growing popularity of Multi-Factor Authentication (MFA) and Single Sign-On. MFA combines two or more authentication factors, including something the user knows (like a password), something they have (like a security token), or something they are (like a fingerprint), enhancing security by making unauthorized access more challenging. Single Sign On streamlines access by eliminating the need to enter a username and password for each website or service. Instead, users authenticate once with a TTP like Google or Microsoft, automatically gaining access to all authorized websites and services without needing repeated logins [47].

7. In the early 2010s:- Biometrics is widely recognized as a robust authentication method, particularly in highly secure applications, because it can identify individuals based on distinguishing physiological or behavioral traits, such as iris patterns or fingerprints. It is more secure than passwords since biometric features are challenging to replicate. Moreover, biometrics are unique to each individual and can authenticate users without relying on a password database [48]. This reduces storage space and enhances the security of the authentication process. However, biometric authentication can also result in false positives and negatives. Therefore, the system should be thoroughly tested before it is deployed.

8. In the late 2010s:- Behavioural authentication is a promising idea for increasing online security. Behavioural biometrics, a kind of biometric authentication, identifies distinct patterns in user behaviour. This involves analysing behaviours such as screen touches for mobile phone unlocking, typing habits, locomotion, and other unique qualities. According to an IBM survey [49], biometrics were deemed the most secure authentication method by 44% of global security reviewers, while 66% found it more efficient.

1.4.1 Security Properties of Authentication

From a security perspective, authentication is a crucial defense mechanism that ensures only verified and authorized entities can access sensitive systems. This reduces threats such as impersonation, data breaches, and unauthorized control in IoT contexts. Authentication is the process of verifying a user's, device's, or system's identity, and it

is fundamental to maintaining security in computing and communications. The key characteristics of an effective authentication mechanism are illustrated in Figure 1.8, highlighting the importance of secure and reliable access in IoT systems.

1. Identity Verification: Authentication relies on identity verification to verify the stated identity of entities, such as users or devices. The main goal is to verify that the entity requesting access is truly who it claims to be.

2. Confidentiality: Confidentiality in authentication protects sensitive information, such as user credentials or authentication tokens, during transmission and storage. Secure communication channels (e.g., SSL/TLS) and robust key management processes are critical to preventing unauthorised access or interception of confidential data by implementing strong encryption algorithms [50].

3. Non-Repudiation: It allows you to prove the legitimacy of a message. Furthermore, it prevents the sender from denying any involvement in the message.

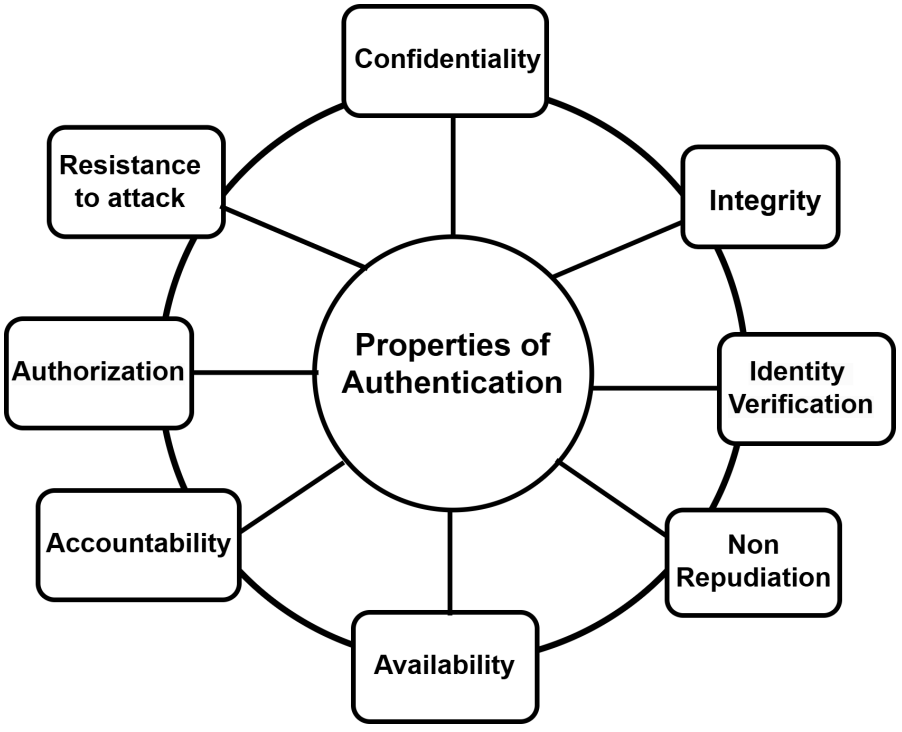


FIGURE 1.8: Key properties of authentication in IoT systems

4. Availability: It refers to consistent and reliable access to authentication services and systems. It guarantees that the authentication mechanisms are operational and accessible as needed [51].

5. Authorization: Authorization involves authenticating the device or user and determining their permissions. Authorization determines the level of access and rights granted to the verified entity within a system or application.

6. Resistance to Attacks: Authentication techniques must protect against common threats such as password guessing, brute-force attacks, and credential theft. This involves creating account lockout procedures and utilising strong, unique passwords.

7. Accountability: Accountability in authentication is the capacity to trace and assign accountability for activities and access to specified entities within a system. Authentication mechanisms must be transparent and auditable for security reasons [52].

1.4.2 Security Threats in IoT

IoT devices pose various security threats because of their interconnected nature and the vast amounts of data they generate. Significant security concerns in the IoT landscape include

1. Replay attacks exploit legitimate authentication data to impersonate authorised users or devices. Attackers may intercept and replay authentication messages, tokens, or credentials to gain unauthorized access. Implementing secure session management, timestamping, and message integrity checks can prevent replay attacks.

2. Man-in-the-Middle attacks: Malicious users intercept and change communication between the authentication server and IoT device. An attacker can modify authentication data to impersonate authorised entities or eavesdrop on sensitive information. Use secure connection protocols (SSL/TLS) to prevent MITM attacks and validate server certificates.

3. Device Spoofing/Impersonation Attacks: Attackers may try to imitate IoT devices to get unauthorized access. This can be accomplished through cloning device identities or exploiting flaws in authentication mechanisms. Implement robust authentication mechanisms, device certificates, and secure bootstrapping processes to prevent device spoofing attacks.

4. Side-Channel Attacks: This attack exposes information about devices' physical attributes or implementation. Attacks on cryptographic algorithms, key generation procedures, or timing information can compromise authentication and reveal sensitive data. To prevent side-channel attacks, safe deployments, protection from power analysis, and timing attacks can be used [17].

5. Phishing and Social Engineering Attacks: Malicious individuals may use fraudulent websites or social engineering to obtain login credentials or access IoT devices. Educating users about phishing dangers, implementing email validation, and raising awareness of social engineering strategies can help combat these attacks.

6. Physical Attacks: Physical attacks entail gaining physical access to IoT devices to harvest passwords, tamper with authentication systems, or modify firmware or software. Implementing physical security measures, tamper-resistant technology, and secure provisioning processes can help mitigate the risk of physical attacks.

7. Modification Attacks: Tampering attacks occur when an attacker modifies data or system components to gain unauthorised access, distort information, and disrupt operations. Modification attacks alter data or system behaviour to benefit the attacker or harm the target system or users [18].

8. Stolen Verifier Attacks: A security breach occurs when a malevolent person acquires unauthorised access to a verifier, which contains secret or confidential information needed in authentication methods. The attacker can gain access to the verifier by unauthorised means, such as theft or eavesdropping on the store system. Once attackers know the identity of the verifier, they can impersonate the original user. Gain access to the system or sensitive resources [19].

9. Denial of Service (DoS) Attacks: Malicious user attacks aim to prohibit authorised users from accessing computer systems, networks, or online services by interfering with their functionality. A DoS attack aims to overwhelm the target system's resources or traffic, limiting its ability to handle genuine requests.

10. Masquerading Attacks: The masquerading attack is akin to impersonation. A masquerading attack involves an adversary duplicating a node's genuine identity and tricking the receiver into believing two senders have the same identity [20].

11. Sniffing attacks: Sniffing attacks, also known as network or packet sniffing, are cyberattacks that intercept and collect data packets sent over a network. The attacker utilises specialised software or hardware tools, such as hacking network switches or routers, to analyse network traffic and get access to sensitive data, including private information, financial data, and login credentials. The attacker aims to capture unencrypted packets by exploiting network protocol weaknesses or gaining unauthorised access to network devices [21].

12. Privileged Insider Attacks: An insider threat happens when a person with authorised access and privileged permissions within an organisation abuses their position to carry out destructive activities. Cryptography saves credentials for all entities within a Trusted Third Party (TTP). The TTP will also generate encryption and decryption keys for the entities [22].

1.4.3 Identification of Authentication Schemes for IoT Applications

Authentication systems are classified into five categories: factor-based, digital signature-based, privacy-preserving, identity-based, and key-based authentication. Classifying authentication techniques helps to understand their strengths, shortcomings, and applicability in various contexts. Identifies the optimal method based on security requirements, user convenience, and risk tolerance. This classification aids in establishing effective security measures, mitigating vulnerabilities, and complying with requirements to prevent unauthorized access or breaches. This work analyzes different authentication protocols and their essential features that facilitate secure interactions between interconnected devices [58].

1. Factor-Based Authentication: This approach employs one or more factors to verify a person's identity. These factors include: • Something you have: smart cards, security tokens, or mobile devices; • Something you know: passwords, PINs, or answers to security questions; and • Something you are: biometric information such as fingerprints, facial recognition, or iris scans.

2. Digital Signature-Based Authentication: This technique verifies a sender's identity using digital signatures. A private key creates a digital signature, which is validated with a public key, ensuring non-repudiation, authenticity, and data integrity [59].

3. Privacy-Preserving Authentication: This approach minimises the disclosure of personal information, with a primary focus on identity verification. It operates by using strategies such as anonymous credentials or zero-knowledge proofs for verification without revealing private details. Benefits include enhanced user privacy and protection against identity theft [60].

4. Identity-Based Authentication: This method uses unique identifiers for users, such as email addresses, usernames, or digital certificates, to authenticate them. It works by using the identifier to retrieve the user's credentials, which are then verified. The benefits include simplifying the management of user identities and integrating with existing identity management systems.

5. Key-Based Authentication: This technique verifies the identity of individuals or devices using cryptographic keys, which can be symmetric or asymmetric [61].

    - Operation: Users authenticate using either a pair of keys (public and private in asymmetric schemes) or a shared secret key (in symmetric schemes). By showing that you have the correct key, you can see that you have the right key. This offers robust security and is frequently utilized in secure communications.

1.5 Motivation

The IoT encompasses RFID tags, sensor nodes, embedded systems, and more, designed to work together for large-scale applications with minimal human intervention. Sensors are utilized in smart grids, environments, homes, cities, and beyond, densely distributed to analyze their surroundings collaboratively. Once deployed, the sensor nodes self-organize to optimize performance and extend network life. Critical deployment aspects involve strategically placing these nodes for task efficiency and coverage. Data from these nodes is converted into digital signals and processed to extract insights about the environment and network status. Due to the limited transmission ranges of individual nodes, they often cannot communicate directly with the sink node. Consequently, relay nodes facilitate communication through a multi-hop architecture, ensuring reliable data transmission and enhanced connectivity. However,sensor nodes encounter several challenges, including mobility, energy depletion, security risks, and network failures. To address these issues, robustness is essential in lightweight IoT networks, necessitating

an optimized model for durability and resilience in the face of node failures, which is vital for sensor IoT networks.

IoT applications enable smart devices to communicate with each other. By 2030, it is projected that over 5 billion real-world smart items will be connected to the Internet, while the global population currently exceeds 7 billion. As a result, it is essential to adopt secure and unique addressing mechanisms to facilitate seamless communication between end devices and users. The growing popularity of IoT reflects users' interest in connecting and managing their devices, providing wireless communication channels that facilitate Internet connectivity and machine-to-machine (M2M) interactions. As a result, communication is considered the most critical component of the IoT network. To ensure secure user access, the communication channel must be protected and capable of safeguarding the network against attacks.

1.6 Problem Statement

The rapid proliferation of IoT devices in smart home environments has created critical security vulnerabilities that existing authentication schemes fail to address adequately. Manufacturers often overlook security best practices, releasing millions of vulnerable devices that generate substantial volumes of sensitive data. Current security mechanisms suffer from fundamental limitations, which include:

1. The lack of a strong mutual authentication security mechanism between devices and gateways considerably weakens the system, making it susceptible to impersonation attacks and unauthorized access.
2. The vulnerability to Denial of Service (DoS) attacks can disable entire smart home networks.
3. High computational and communication overhead incompatible with resource-constrained IoT devices, resulting in poor scalability that cannot accommodate the exponential growth in connected devices.

These limitations make current security schemes ineffective in real-world IoT deployments. Hence, we need to develop a lightweight and robust authentication protocol that provides comprehensive security while minimising resource consumption, ensuring scalability, and maintaining high performance across various smart home ecosystems.

1.7 Thesis Organization

This thesis is structured into five chapters, each focusing on specific subject matter and proposed solutions based on the research. At the end of each chapter, a summary of the findings and key insights is provided. The general organization of the thesis is as follows.

1. **Chapter 2** presents the Authenticated Unidentified Security Scheme (AUSS), a multifactor authentication protocol specifically designed for IoT networks. The Authenticated Unidentified Security Scheme (AUSS) leverages three-factor authentication—comprising knowledge (password), possession (device credentials), and inherence (biometrics via biometric hash technology)—within a three-tier architecture that includes mobile nodes, gateways, and sensor nodes. The protocol is formally verified using BAN logic and the AVISPA tool, demonstrating resilience against various attacks.

2. **Chapter 3** introduces the Secure and Lightweight Authenticated Protocol for the Smart Home (SLAPSH), which addresses privacy and authentication in Smart Home IoT (SH-IoT). SLAPSH uses a unique IPv6 addressing scheme that assigns each device a verifiable address for secure communication with IoT servers. It provides robust user authentication to avert unauthorised access. NS-3 simulations indicate low end-to-end delay, and security tests with the ROR model and AVISPA tool confirm SLAPSH's resilience against attacks. These findings establish SLAPSH as a reliable protocol for smart home IoT systems.

3. **Chapter 4** presents the Secure and Unique Mutual Authentication Scheme to enhance the security of smart home IoT systems. This scheme addresses vulnerabilities such as unauthorized access and data breaches by modifying the IPv6 format to assign unique device addresses. Additionally, SUMAS incorporates mutual authentication to resist network attacks. Informal analysis and formal verification with the AVISPA tool confirm that SUMAS effectively resists various security threats, ensuring secure communication in IoT environments.

4. **Chapter 5** The thesis concludes by summarising the key findings of this research. It also explores potential directions for future research.

CHAPTER 2
LITERATURE REVIEW

Numerous studies have highlighted the significant challenges associated with authenticating devices, primarily due to the wide range of connected IoT devices. Over the years, various authentication techniques specifically designed for the IoT have been developed and implemented. This chapter provides a comprehensive review of the existing literature on the challenges of authentication and their solutions in the IoT domain. Examines security assessments and discusses the complexities of safeguarding privacy during the authentication process while ensuring secure data transmission in IoT environments. Furthermore, this chapter provides a detailed overview of key research efforts, helping scholars understand current approaches and identify several unresolved technical issues.

2.1 Taxonomy of Authentication Schemes

Authentication is widely recognised as a fundamental security mechanism [63], essential for safeguarding OT networks. Verifying the legitimacy of entities accessing the network helps defend against prevalent threats such as replay, man-in-the-middle, impersonation, and denial-of-service attacks. However, when authentication is implemented improperly, it can create vulnerabilities that allow attackers to steal critical credentials, compromising the entire network's security. Therefore, robust authentication enables trusted communication and secure data sharing between interconnected IoT devices. Validates both users and devices within the network. Authentication is a fundamental aspect of establishing secure communication between IoT users and devices. It ensures that the identity of each user or device is verified adequately within the IoT ecosystem. Failing to negotiate the device's identity properly can leave the network vulnerable

to various attacks [64]. Authentication is critical for verifying device identities, securing communication, preventing unauthorised access, and maintaining data integrity and confidentiality. Furthermore, it enhances overall system security. Organising these techniques enables the enhancement of security protocols, the reduction of potential vulnerabilities, and adherence to regulatory standards, thereby helping to prevent unauthorised access and security breaches [65]. This work explores different authentication approaches and the requirements for maintaining secure communication between interconnected devices. Subsequently, authentication mechanisms are classified, and a review of related literature is presented.

2.1.1 Key-Based Authentication

There are two types of authentication schemes: symmetric and asymmetric key-based.

2.1.2 Symmetric Key-Based Authentication

Throughout the authentication process, these systems employ the same secret key for encrypting and decrypting data [66, 67]. The approaches must create and validate a matching cryptographic hash or MAC to ensure data integrity and authenticity. However, managing shared keys can be challenging. This solution effectively protects communication in ad hoc networks by establishing a secure channel for key exchange and authentication. Symmetric key-based authentication techniques are classified as follows:

1. Message Authentication Code- This method generates a fixed-size authentication tag using a hash function and a secret key, enabling receivers to verify the integrity and authenticity of messages. To enhance security and efficiency in Vehicle Ad Hoc Networks (VANETs), an accelerated message authentication protocol employs a keyed Hash-based Message Authentication Code (HMAC) for faster revocation checks, replacing traditional Certificate Revocation List (CRL) methods. Each On-Board Unit (OBU) has a unique secret key to create a Message Authentication Code (MAC), which the receiving OBU uses to verify the authenticity of the message [68].

   The Enhanced Message Authentication Protocol (EMAP) employs probabilistic key distribution to secure information sharing among non-revoked OBUs [69],

improving communication reliability in vehicular networks. To further address the challenge of authentication in VANETs while preserving user anonymity, an alternative solution replaces Certificate Revocation List (CRL) verification with HMAC-based computation. This approach leverages pseudonyms to enhance user privacy and employs HMACs to ensure message integrity and enable efficient batch authentication. However, despite its advantages, the method incurs high computational costs, which limits its scalability and makes Roadside Units (RSUs) vulnerable to Denial-of-Service (DoS) attacks.

Lightweight systems combining RSA, AES, and HMAC facilitate mutual authentication and session key verification, although they do not fully address data privacy concerns. In e-health applications, HMAC and nonces are used for sensor and base station authentication, requiring substantial processing resources. The Time-Efficient and Secure Vehicular Communications (TSVC) approach utilises MAC authentication to minimise message loss ratios. However, it requires considerable computing and storage resources, which makes it less suitable for extensive IoT setups [70].

2. Hash Function based Authentication- Authentication employs hash functions to generate a fixed-size output from messages, ensuring their integrity. The hash is sent alongside the message, enabling the recipient to verify the message by recalculating the hash and checking for a match. A proposed multi-server authentication technique eliminates verification tables, utilizing nonces and hash functions for integrity and authenticity. This method is ideal for distributed networks due to its low computational load, but it lacks scalability for batch authentication and does not protect user identity privacy. In the context of Vehicle Ad Hoc Networks (VANETs), a privacy-preserving method employing hash functions enhances security and optimizes key distribution. Each vehicle is assigned a unique random identifier that could undermine unlinkability while also reducing computational demands. However, this scheme faces challenges with scalability for large-scale IoT deployments. A smart card-based authentication strategy utilizes a secure hash function for user validation, though it can be computationally intensive and encounters scalability challenges when managing numerous hash values. An IoT-based cloud architecture for authentication integrates IoT and cloud computing for secure access control. Nevertheless, it remains susceptible to replay, impersonation, man-in-the-middle, and insider threats, while neglecting batch authentication and user privacy concerns [71].

2.1.3 Asymmetric Key-Based Authentication

It employs public and private keys to authenticate users or entities. The public key encrypts data or messages, while the private key decrypts them. This approach provides secure communication without requiring a shared secret key between parties. Elliptic curve, PKI-based authentication, and digital signatures utilize public-key authentication to verify user or device identities across security protocols and systems [72].

1. PKI-Based Authentication Scheme- PKI-based authentication utilises public key cryptography to safeguard communications and authenticate users. In this system, users generate a public-private key pair. A reliable CA receives the public key for validation. After verification, a digital certificate is signed with the CA's private key, the user's public key, and any necessary identifying information [73]. To establish secure communication, users provide a certificate, which others can verify by inspecting the CA's signature and ensuring it has not been revoked. Depending on the CA, it results in a single point of failure. The PKI scheme has issues with certificate administration, including storage, distribution, revocation, and verification [74].

2. Elliptic Curve Based Authentication Scheme- It employs elliptic curves to generate key pairs for authentication. ECC is widely regarded as more secure and efficient than RSA. ECC is better suited for IoT devices since it requires fewer computational resources compared to RSA. Elliptic Curve Cryptography (ECC) delivers equivalent security to RSA while utilizing significantly smaller key sizes, resulting in improved speed and efficiency. ECC-based authentication utilizes elliptic curves to construct keys, making it more effective and secure in resource-constrained environments, such as mobile devices or IoT devices [75].

2.1.4 Identity-Based Authentication

This encryption method eliminates the need for a public-private key combination, allowing users to use simple information as their public key. Users can use a username or email address to identify their public key. The solutions significantly minimise communication overhead by enabling authentication without the requirement for certificates. These methods reduce the cost of certificate management and distribution. A TTP, a KGC, generates the private key associated with a

user's identification. The KGC generates a unique private key based on the user's identification and securely delivers it to the user. Messages or data signed with the user's private key can be validated using their public key [76].

2.1.5 Digital Signature-Based Authentication

Using cryptographic methods, digital signatures ensure the legitimacy, consistency, and irreversibility of digital data. Digital signatures allow recipients to authenticate the authenticity of documents and messages. This method uses the sender's public key to decode a hash value from an encrypted document or message. Recipients can verify the integrity of communication and the sender's identity by comparing the decrypted hash value to a newly generated hash. Digital signature-based authentication covers both certificate-based and certificateless authentication methods. Certificate-based authentication involves the sender creating a unique digital signature with their private key. The receiver validates the signature using the sender's public key, which was acquired from the CA, to ensure communication integrity and validity. Certificate-less authentication utilises a unique signature generated by the sender, rather than standard certificates. The receiver can validate a signature without utilising a certificate authority by rebuilding the sender's public key with standard parameters. Although this method simplifies certificate administration, it requires strong cryptography precautions [77]. Certificateless signature methods address the key escrow issue in identity-based encryption while maintaining the convenience and security of traditional public key infrastructure (PKI). This method improves security and accelerates key distribution in cases where obtaining traditional certificates is challenging or impossible [78].

2.1.6 Privacy-Preserving Authentication

This relates to protecting sensitive information while authenticating a user or entity's identity. This technique prioritises user privacy by limiting the amount of personal information shared during login. Privacy-preserving strategies, integrated with authentication schemes, address privacy concerns by utilising cryptographic protocols and privacy-enhancing technologies. Privacy-preserving authentication enables users to securely authenticate their identities without disclosing sensitive information or revealing their genuine identities. This strategy balances identity verification with user privacy using new cryptography, decentralised systems, and privacy-enhancing technology during

authentication. Techniques like anonymous credentials, zero-knowledge proofs, decentralised identity systems, and attribute-based authentication help maintain user privacy. Using privacy-preserving authentication systems meets security criteria and protects individuals' privacy rights, creating a more trustworthy and secure digital environment. This integration offers numerous benefits, including user anonymity, reduced risk of identity theft, data minimisation, increased trust, compliance, and enhanced security. Cryptographic approaches, such as group signatures, ring signatures, and pseudonyms, enhance authentication security and privacy [79].

2.1.7 Factor-Based Authentication

In the past, transactions were primarily confirmed in person, requiring the individual executing the transaction to be physically present. While this method of transaction authentication was secure, it proved inconvenient. There are various methods for user authentication, including passwords, PINs, biometrics, and two-factor authentication. These technologies offer greater security and convenience than authentication based on physical presence. Initially, users were authenticated using a single factor (SFA), which was favoured for its simplicity and ease of use. Users needed only to provide one piece of documentation to verify their identity, typically a password or PIN. However, SFA is susceptible to password sharing, dictionary attacks, and social engineering, making it the least secure level of authentication [80].

1. Two-Factor Authentication (2FA): To overcome the shortcomings of Single-Factor Authentication (SFA), Two-Factor Authentication (2FA) was developed. With 2FA, users must enter a password and a one-time code sent to their phone, thereby strengthening security. This approach makes unauthorized access to online accounts and devices more difficult, offering greater protection than SFA. Using ECC, a 2FA protocol for smart homes was presented in [81]. This technique is vulnerable to various attacks, including replay, insider, session key disclosure, offline password guessing, impersonation, and the absence of mutual authentication, as demonstrated by an improved two-factor authentication approach [82]. A proposed two-factor anonymous authentication technique [84] protects patient identity privacy. Their approach is inefficient in terms of computing, storage, and transmission costs. A proposed scheme [85, 86] for anonymous two-factor

authentication (2FA) in medical systems aims to protect patient privacy and authenticity. However, the scheme has been proven vulnerable to DoS attacks, offline identity and password guessing, and user impersonation. A lightweight two-factor authentication solution for the Internet of Medical Things (IoMT) has been proposed, utilizing hash chains within wireless sensor network (WSN) environments [87]. This scheme meets various security criteria and effectively prevents potential breaches. However, the scheme presented in [88] has identified vulnerabilities in the approach discussed in [87], including the risks of stolen verifiers and physical captures of sensors. Additionally, a secure and lightweight authentication key management process for medical systems has been proposed in [89]. This method utilizes a trusted third party (TTP) to verify the legitimacy of system entities; however, the dependence on a TTP makes the system model less robust.

2. Multi Factor Authentication (MFA)- Using weak passwords can render 2FA ineffective. Even with two-factor authentication enabled, attackers can readily guess weak passwords. Two-factor authentication (2FA) systems using PINs and smart cards can be compromised by attackers who guess the user's PIN or smart card credentials. Without strong passwords, two-factor authentication (2FA) cannot provide sufficient security [90]. A three-factor authentication mechanism has been suggested. The three major categories of factors are:

    - (a) Knowledge-based factors: Information the user is aware of, like a password or personal identification number (PIN).
    - (b) Possession-based factors: Items the user owns, such as a security token or mobile device.
    - (c) Inherence factors: Unique characteristics of the user, including fingerprints or facial recognition.

    Multi-factor authentication (MFA) enhances security by integrating two-factor (2FA) and three-factor (3FA) methods. This approach significantly reduces unauthorized access risks, even if a user's password is compromised. One MFA method involves secret sharing for key distribution, but it faces challenges such as denial-of-service (DoS) attacks, replay attacks, and desynchronization attacks. To enhance security in cloud-based IoT, a lightweight mechanism validates biometric data using a dynamic index and a fuzzy extractor algorithm, while preserving user anonymity. However, concerns about reliability arise from the involvement of a Trusted Third Party (TTP). A blockchain-based multi-factor device authentication

system has been developed for cross-domain Industrial Internet of Things (IIoT), addressing privacy issues and securely storing authentication data to protect sensitive user information and minimize impersonation risks [91].

3. Three Factor Authentication (3FA)- During the three-factor authentication (3FA) process, users provide proof of their identity through ”something they know,” ”something they have,” and ”something they are.” While 3FA enhances security against unauthorized access, it can also be burdensome for users. The authors [92] present a 3FA framework for telemedicine information management systems designed to guard against insider threats, password guessing, and stolen smartcards. However, it remains vulnerable to insider attacks and faces challenges with password updates and patient privacy. For patient monitoring systems, a lightweight authentication key protocol (AKP) scheme [93] using hash and XOR algorithms was created. This method, however, is vulnerable to sensor key leakage, mobile device theft, and desynchronization. An enhanced end-to-end AKP approach [94] addresses some previous security flaws but remains susceptible to DoS attacks and insider threats.

2.2 Formal Security Evaluation

To identify vulnerabilities in an authentication method, it is essential to evaluate it using various security analysis metrics. These metrics provide a comprehensive assessment of the method's robustness against different types of attacks. Therefore, a detailed security analysis is necessary to confirm the stability and reliability of the proposed authentication scheme. This evaluation measures the scheme's resilience to common threats.

2.2.1 Attacker Model

A significant challenge in securing data lies in ensuring its safe exchange over vulnerable wireless networks. It is essential to utilise robust encryption techniques, such as public-key cryptography and secret key exchanges, to safeguard sensitive information. When evaluating security protocols, attention to detail is critical, and identifying weaknesses can be complicated without clear explanations. Using precise terminology is essential for defining security goals and protocols, especially in software development.

Additionally, understanding attacker behavior in the context of cryptography is vital for validating security measures [100].

2.2.1.1 Dolev-Yao (DY) Threat Model

The Dolev-Yao (DY) adversarial model, introduced by Danny Dolev and Andrew Yao, characterises attackers as highly capable, able to perform both passive and active threats such as intercepting, modifying, fabricating, or deleting network messages. Within the context of smart home environments, it is vital to safeguard devices against such powerful adversaries to maintain the security and reliability of the network. If a device is lost or stolen, there is a risk that malicious actors could impersonate legitimate users and gain unauthorised access to confidential information. Therefore, robust data protection mechanisms are essential. The proposed approach establishes a secure authentication protocol tailored for IoT networks, designed to withstand various threats, including message tampering, replay attacks, identity spoofing, device compromise, denial-of-service attacks, brute-force password attempts, and man-in-the-middle attacks. It guarantees mutual authentication, data integrity, confidentiality, and forward secrecy, with a particular focus on confidentiality within two-party protocols, while also highlighting key security features [96].

1. Secrecy properties: Assume James receives a message (M) as input. He starts exchanging communications with Bond, stressing the importance of keeping the message (M) confidential. The security property ensures that adversaries cannot capture message (M), even if they intentionally interfere with the protocol. No additional security properties have been considered.

2. Stateless Parties: Authentic parties are stateless, meaning they can only send messages based on the most recent one and their prior knowledge, without referencing earlier messages. This limitation is unique to authentic parties, while attackers can maintain state information for future exchanges. This lack of state aids honest parties and reflects real-world scenarios, such as servers using cookies to track session states without needing clients to resend information.

3. Concurrent execution: The attacker can initiate several protocol executions. The game involves many parties and allows each player to participate in multiple concurrent executions. Computational cryptography began to address concurrency issues in the 1990s.

4. Public-key cryptography and infrastructure: A public table with each user's public key and name is assumed to be published publicly. This table and the user's secret decryption key are part of their basic knowledge.

2.2.2 AVISPA Tool

AVISPA is a role-based language in which each agent has a specific function during the execution of a protocol. The AVISPA tool utilizes the HLPSL to define security protocols, ensuring secure message exchanges between agents, which encompass both authentication and data confidentiality. HLPSL features a section dedicated to defining security attributes and evaluates the protocol against predefined goals to determine its SAFE status. Furthermore, the HLPSL2IF translator converts HLPSL specifications to Intermediate Format (IF), providing precise input for various back-ends of the AVISPA toolkit [98]. Some back-end tools are listed below:

1. On-the-Fly Model-Checker: The OFMC employs symbolic methods and algebraic properties to investigate a demand-driven state space.
2. CL-AtSe: It is a Constraint-Logic-based Attack Searcher that turns security protocol specifications expressed in the IF language into a set of constraints. This allows for the effective detection of protocol attacks.
3. The SAT-Based Model Checker (SATMC): SATMC generates a propositional formula from the transitional states described in an Intermediate Format specification. This formula can reveal violations of security properties, indicating potential vulnerabilities or attacks.
4. Tree Automatic Approximations for Security Protocol Analysis: The TA4SP predicts the accuracy or susceptibility of a protocol by accurately estimating the capabilities of the intruder.

The output format (OF) consists of six components that indicate whether the scheme is secure and if the protocol is under attack. The following is the verification summary:

1. SUMMARY: This segment indicates whether the tested scheme is secure, insecure, or inconclusive.

2. DETAILS: This section includes detailed information on the tested protocol. What conditions make the protocol safe? Which assaults make it unsafe? Why is the verification summary inconclusive?

3. PROTOCOL: This segment specifies the intermediate format of HLPSL code for tested protocols.

4. GOAL: This section shows whether or not the test protocol's goals were met.

5. BACKEND: It specifies the backend name used to analyze the protocol.

6. STATISTICS: This section outlines the vulnerabilities of the tested protocol. Identify weaknesses and provide relevant statistics and comments [97].

2.2.3 Burrows–Abadi–Needham (BAN) Logic

Robust authentication methods are essential for maintaining security within distributed systems. The behavior and evolving perspectives of trusted entities in authentication protocols can be modeled using a logical framework with established deduction rules [99]. BAN logic is a formal technique for protocol verification, designed to evaluate the correctness of protocols. This logic supports the analysis of public and shared key operations, as well as the notion of message freshness, thereby enabling the formal representation of challenge-response mechanisms. BAN logic facilitates addressing several key questions:

1. What conclusions can be drawn from the protocol?

2. What assumptions must be met for the protocol to function correctly?

3. Are there redundant steps in the protocol that could be eliminated?

4. Does the protocol unnecessarily encrypt information that could be safely transmitted in plain text?

2.2.4 Real-or-Random (ROR) Model

Real or Random Model (ROR) validates the protocol's security and the session key. In this example, the communicating entities may include IoT mobile users, IoT end

node devices, a gateway server (GS), or an authentication server (AS). The ROR model assumes that the adversary can read, edit, eavesdrop, insert, and create new messages during transmission [100].

2.3 Simulator Tools

Over the years of conducting IoT security research, we have discovered that one of the most popular tools is the Network Simulator NS-3. This helpful tool helps explore and enhance the understanding of IoT protocol performance.

2.3.1 NS-3

NS-3 is an open networking research platform that employs discrete event network modelling to simulate complex systems [101] effectively. It serves as a simulation engine focused on Internet protocols and network modelling. NS-3 supports external animators, data analysis, and visualisation tools, enabling development in C++ and Python to meet diverse research needs. The platform analyzes crucial network metrics such as delay, throughput, packet loss, latency, and packet delivery ratio. NS-3 includes several modular components, with nodes representing fundamental computing devices in the simulation. These customizable nodes are managed through a Node Container, which organizes the nodes for constructing the simulated network topology.

2.4 Related Work

Researchers have recently studied IoT frameworks, focusing on resilient approaches, lightweight authentication schemes, global addressing techniques, and security schemes. We summarise and discuss IoT authentication schemes, with Table 5.1 comparing and analysing existing schemes, highlighting their contributions and limitations.

2.4.1 Security Methods

Several safety techniques for single servers are inadequate for distributed frameworks with multiple servers, as clients need separate registrations and passwords. Various two-factor security mechanisms for IoT environments include a new multi-server method that employs biometrics and key agreement for secure access; however, it has vulnerabilities. While secure single-server solutions exist, they should not be used in distributed systems with several servers [148, 151]. Alternatively, users are required to register individually and maintain distinct passwords for every server. In recent times, various two-factor authentication approaches based on cryptographic methods have been introduced for use in IoT applications. The authors of [156] propose a multi-server approach to mitigate vulnerabilities in IoT networks. They implement key agreement authentication using biometrics, enabling secure access to the server. Additionally, they develop an ECC Diffie-Hellman key scheme to prevent repeated attacks, although this method remains susceptible to MIMA attacks. The use of a silicon ID as a unique identifier for devices is inadequate due to potential security risks; it can disrupt overall communication within the network. This study utilizes ECC to connect multiple IoT devices, based on the assumptions of ECDLP (Elliptic Curve Discrete Logarithm Problem) and ECCDH (Elliptic Curve Diffie-Hellman) [155]. The protocol implements authentication for both users and the home server. In [140], a secure authentication scheme for smart IoT devices is described. However, as noted in [136], attackers may still intercept sensitive information, potentially leading to communication disruptions. Researchers validate privacy and security strategies to mitigate cyber threats, helping organisations protect their IoT devices [153]. Additionally, attackers can send hidden communications that are easily intercepted, blocking communication and tracking device location via static hardware memory [134, 137]. Existing security approaches do not recognise that altering the IPv6 format can provide authentication without additional computational overhead, often leading to increased security and computational costs [121]. The Chebyshev Chaotic Maps (CCM) privacy method leverages lightweight networks to enable mutual authentication [157]. In [179], a privacy-preserving authentication (PPA) scheme was introduced to address various security challenges. The authors assessed the PPA mechanism using BAN logic and compared its security effectiveness with other existing methods. The study in [159] presents a lightweight privacy technique utilizing a LAM-CloT-based fuzzy extractor, which effectively mitigates active attacks and lowers computational demands in IoT environments. Furthermore, as detailed in [160], this approach protects against data leakage during communication. Secure IoT access is

further reinforced through a multi-factor key agreement protocol and the use of bilinear pairing techniques.

2.4.2 Existing Lightweight Authentication Schemes

This section reviews user authentication methods proposed in recent years for wireless sensor networks (WSNs), which are easily adaptable to IoT environments, particularly in smart home applications. One such approach is the Lightweight Authentication Scheme for Smart IoT Home Networks, which exemplifies the use of efficient, low-overhead techniques suited for resource-constrained IoT devices. In 2009, **Hsiang and Shih et al.** [102] introduced a secure dynamic ID enhancement within a multiserver framework, grounded in a remote user authentication system. The enhanced scheme's computational cost, security, and efficiency are well-suited for practical application environments. However, it was vulnerable to erroneous password modifications, replay attacks, and impersonation threats. In 2010, **Li and Hwang et al.** [146] presented a low-cost biometric authentication scheme for remote users using smart cards. This secure scheme combines biometric authentication, a smart card, and a one-way hash function. It enables users to change passwords freely, ensuring mutual authentication with the remote server. Unlike many remote schemes that require synchronised clocks to prevent replay attacks, their design uses random numbers, eliminating the need for clock synchronisation. **Yeh et al.** [104] devised an elliptical curve-based remote user authentication technique in 2011. An aspect of the proposed protocol that protects both internal and external security is mutual authentication. Furthermore, it enhances the security of WSN authentication by outperforming other protocols and inheriting the advantages of the Elliptical Curve Cryptography-based technique. Therefore, the protocol is better suited to settings utilizing wireless sensor networks (WSNs). Nevertheless, it was found that the computational cost was significant compared to other procedures.

In 2011, **Vaidya et al.** [105] introduced an ECC-based authentication system for Smart Energy Home Area Networks (SE-HAN). This approach consists of five stages: pre-deployment, initialization, authenticated key agreements, user-controlled key renewal, and key revocation. Because the technique meets several standard security criteria, it is more reliable and secure than current authenticated key protocols. However, key-share attacks pose a common threat to this approach. The author **Xue et al.** [106] introduced a mutual authentication protocol in 2012 that requires only hash and XOR computations. The user, gateway, and sensor nodes share temporal credentials, with the user's

credentials stored securely or openly on a smart card. However, the credentials associated with the sensor node identity must be stored securely. Evaluations of both security and performance indicate that the proposed scheme strengthens security features while maintaining minimal impact on communication, computational, and storage resources. However, implementation is vulnerable to attacks like server spoofing and smart card theft. **Y. Chang et al.** [107] introduced a biometric user authentication protocol for linked healthcare in 2013, ensuring privacy. The plan prevents tracking of users via transmitted data, granting access only to authorised users. A medical server generates a new random identity for users at the start of each session, making it impossible for malicious actors to trace a specific individual. However, the protocol lacks multiple security levels and strong authentication. Moreover, remote users must remember their identities and passwords for each server, making this scheme unsuitable for distributed systems such as the IoT. In 2013, **Gubbi et al.** [108] developed a unified IoT authentication protocol for multiple levels and nodes. The scheme uses element extraction and hashing. This approach satisfies IoT authentication security requirements by incorporating irreversibility into the extraction procedure. In 2014, **Ndibanje et al. [109]** advanced authentication and access management for IoT devices by analysing and revising current authentication and access control methods. They enhanced device authentication by introducing a simple, efficient, and secure key establishment protocol utilizing elliptic curve cryptography. Access rights within IoT network applications are managed through role-based access control. However, the protocol incurs substantial communication costs for IoT sensor nodes, and its security evaluation has yet to be confirmed through real-world experimentation.

In 2015, **Chen et al.** [110] proposed a secure user authentication protocol for wireless sensor networks (WSNs) that employs symmetric key methods to guard against smart card loss attacks. They outline the threats and security requirements for 2-factor authentication. The scheme withstands smart-card loss attacks but is vulnerable to DoS attacks due to inefficient verification. Furthermore, it does not ensure user anonymity, as identity is transmitted in plaintext during the login process. The scheme also consumes resources due to delays in identifying incorrect login credentials, including passwords. **Das et al.** 2016 [111] developed a three-factor multi-gateway WSN-based user authentication system in 2016 because generic WSNs impose significant costs on the gateway and consume more energy than multi-gateway WSNs. Based on well-known BAN logic, they demonstrated that their approach provides secure mutual authentication. The security of the proposed scheme was verified using the AVISPA tool, a trusted framework for

assessing network security protocols, which confirmed the protocol's robustness. The protocol was demonstrated to be resilient against several cryptographic threats, including impersonation and sensor capture attacks. However, it was found to be susceptible to user tracking vulnerabilities. Additionally, each of the three participants utilizes a distinct session key.

In 2016, **Amin and Biswas et al.** [112] proposed a smart card-based authentication mechanism to enable secure access to private cloud servers in remote settings. The security of this protocol was established through BAN logic, which verified its ability to deliver mutual authentication and establish a secure session key. Furthermore, simulation using the AVISPA tool, with both OFMC and CL-AtSe models, confirmed its robustness. Performance comparisons underscore the scheme's simplicity and practicality for real-world deployment. **Farash et al.** [113] presented an advanced user authentication and key exchange protocol designed explicitly for heterogeneous Wireless Sensor Networks (WSNs). Their solution, aimed at strengthening the User Authentication and Key Agreement System (UAKAS), introduces a four-phase process that enables registered users to interact directly with sensor nodes, thereby eliminating the need for an intermediary gateway. Security analysis revealed strong protection against prevalent wireless threats, and formal validation using BAN-logic and AVISPA further reinforced the protocol's security. However, the scheme remains susceptible to certain vulnerabilities, including stolen smart cards, user impersonation, offline password attacks, and compromise of session-specific data.

In the same year, **Kaul and Awasthi et al.** [114] developed a remote user authentication protocol using smart cards for client-server authentication. Their approach, which is resource-efficient in terms of storage, computation, and communication, achieves mutual authentication and is considered practical for deployment. Security assessments with the AVISPA tool confirmed its resistance to both active and passive attacks, although it remains exposed to user impersonation risks.

Roy et al. [115] introduced a lightweight three-factor authentication scheme for IoT, incorporating smart cards, passwords, and biometrics. The protocol avoids computationally intensive operations, such as those involving elliptic curves or modular exponentiation, thereby ensuring efficiency. Security verification was performed using Proverif, along with further analysis via real-or-random and BAN logic methods, confirming its resilience and practicality. In 2017, **Dhillon and Kalra et al.** [116] proposed a multifactor authentication and key agreement protocol designed for remote users in

IoT environments. This scheme enables authorised users to securely retrieve real-time sensor data from IoT nodes, mandating authentication from both the gateway and the target device. The protocol was rigorously evaluated and demonstrated its effectiveness in IoT scenarios.

Wazid et al. [118] introduced an efficient remote user authentication and key management scheme for smart homes, utilizing elliptic curve cryptography (ECC). This approach achieves mutual authentication, user anonymity, and perfect forward secrecy, while defending against replay, impersonation, and insider attacks. Formal verification was conducted using BAN logic and AVISPA, and a performance analysis confirmed its suitability for resource-constrained IoT devices, providing both privacy and robust security. **Nikooghadam et al.** [119] developed a lightweight key agreement and authentication scheme that features strong BAN logic correctness and robust security at a low computational cost. The protocol is tailored for environments with limited resources, supporting secure key agreement and authentication. **Kang et al.** [120] introduced a biometric-based authentication and authorization scheme for IoT infrastructures, employing dynamic identifiers and combining biometric data with hashing to ensure user anonymity and resist impersonation and offline password attacks. Their method also mitigates replay and time synchronization threats via single-use numbers, although it remains vulnerable to insider threats and disclosure of temporary secrets.

Shah et al. [121] developed Secure Vault, an authentication protocol designed to enhance communication between IoT devices and servers. The protocol supports mutual authentication and resists side-channel attacks; however, it has certain authentication gaps and is susceptible to physical compromise of IoT devices, which could render them inoperable. **Chandrakar et al.** [122] designed a three-factor authentication scheme for TMIS, demonstrating security in both logical and causal contexts and providing resistance to various attacks. Security was validated through BAN logic, but the scheme is not immune to perfect forward secrecy violations and replay attacks. **Amin et al.** [123] proposed an anonymous, untraceable password-based authentication protocol for WSNs, offering strong defense against active and passive attacks while maintaining low communication overhead. Despite its practicality for low-power sensor deployments, the protocol remains susceptible to insider threats, replay attacks, and password-guessing attacks.

Park et al. [124] developed a multicast-based approach for large-scale IoT communication using telemetry transport, reducing congestion and latency in distributed edge

environments. While it achieves mutual authentication verified by BAN logic, its session key security and three-factor authentication are insufficient, leaving it vulnerable to KXNTI attacks. **Lu et al.** [125] presented a three-factor anonymous key exchange protocol for WSNs based on ECC, ensuring comprehensive security with minimal communication and processing cost. The system is efficient, resistant to multiple threats, and offers increased security features at lower storage requirements. **Ostad-sharif et al.** [126] introduced a secure, lightweight authentication and key negotiation technique for IoT-based WSNs. The scheme guarantees perfect forward secrecy, efficiency, and improved security with reduced communication and storage burdens. **Garg et al.** [127] proposed the OAuth protocol, enabling user access to middleware via a combination of credentials and tokens, acting as a bridge to sensor data through a REST API. BAN logic analysis confirmed its strong security and implementation simplicity; however, reliance on credential-based authentication introduces potential risks of data breaches.

Banerjee et al. [128] introduced an authentication protocol for smart home environments that is efficient, secure, and preserves user anonymity. The system requires resilience against various threats, demonstrating the superior security and utility benefits of the suggested scheme. A network simulation was run to get practical insight into the proposed scheme's application. However, it cannot withstand anonymity or trace attacks. **Suresh et al.** [129] improved Telecare Medical Information Systems in 2020 by implementing a chaotic map for mutual authentication and key exchange. The system was informally tested for multiple security threats and employs formal BAN logic for mutual authentication, thereby protecting both privacy and security. However, this strategy is computationally demanding and vulnerable to desynchronisation attacks. **Bae and Kwak et al.** [130] introduced a smart card-based authentication protocol designed for environments with multiple servers. The protocol's security was evaluated using the AVISPA tool, which demonstrated its effectiveness against user impersonation and session key exposure, making it applicable for smart card-driven key exchange. However, the scheme still has weaknesses, including vulnerability to traceability, impersonation, and session key leakage, as well as inadequate mutual authentication. **Deebak et al.** [131] developed a lightweight authentication and key management (LAKM) approach for smart IoT-enabled systems. Their method incorporates continuous user authentication, facilitating rapid processing while persistently monitoring devices to improve battery efficiency. The proposed scheme enhances security by ensuring strong connections between computing systems, maximizing forward secrecy, reducing both communication and computation overhead, and lowering storage requirements. **Hinden et al.** [132]

proposed a global and aggregatable IPv6 address format to improve routing and address management, in line with the IPv6 Protocol and "IPv6 Addressing Architecture." While this design aims for scalable Internet routing, the increasing complexity of the Internet has outpaced the capabilities of current address management tools and technologies, complicating effective IPv6 administration.

Sabir et al. [133] presented an innovative subnetting technique for Class C IP addresses to better utilize address space. Their research covered both Fixed-Length and Variable-Length Subnet Masking (VLSM), introducing a new aggregation method. Although VLSM supports more efficient IP address allocation, it can sometimes result in suboptimal usage and the waste of available address space. **Narten et al.** [134] extended the IPv6 stateless address autoconfiguration process by incorporating unique identifiers derived from IEEE interfaces. This enables automatic configuration in large-scale IPv6 networks. Despite the scheme's effectiveness for address allocation, it fails to provide a robust defence against denial-of-service (DoS) attacks, thereby limiting its security applicability. **Hinden and Haberman.** [135] specified a unique local address format for IPv6 (Unique Local IPv6 Unicast Addresses), which are intended for local interactions within private networks. These addresses are also known as Local IPv6 Addresses. While effective for local routing, they require border routers to filter such addresses from the global Internet to avoid routing conflicts, which adds to the management complexity. **Judmayer et al.** [136] proposed 6HOP, a lightweight and secure addressing technique for IoT environments. It offers ease of implementation, incurs no significant computational cost, and provides basic protection against network attacks. However, this approach does not adequately address the challenges posed by the simultaneous operation of multiple servers or integration with other advanced hopping techniques. **Gont and Chown et al.** [137] focused on improving IPv6 network security by replacing RFC 5157 [155]. Their study highlighted the limitations of traditional IPv6 reconnaissance techniques and proposed strategies to strengthen early-stage IPv6 deployments. Nonetheless, without robust security, a dual-stack IPv6/IPv4 implementation remains susceptible to various forms of reconnaissance, necessitating advanced protection mechanisms before widespread adoption. **Dunlop et al.** [140] proposed the Moving Target IPv6 Defense (MT6D), a security approach at the network layer that takes advantage of the expansive IPv6 address space to generate more address variations than attackers can feasibly target. The method allows for frequent address changes without modifying the IPv6 protocol, facilitating its deployment within networks and supporting defense-in-depth strategies. Despite enhancing network-layer protection, MT6D does not address

threats at higher protocol layers and is impractical for IPv4 networks due to the limited address space. **Tsai et al.** [141] proposed a novel anonymous authentication mechanism based on smart cards. Their scheme ensures privacy-preserving verification through the use of a hash function and elliptic curve cryptography (ECC). Though secure and privacy-centric, the proposed method incurs higher computational and communication costs compared to conventional protocols, which could hinder its practical deployment. **Nicanfar et al.** [142] developed an authentication protocol for smart meter and server communication, utilizing an initial password. The method reduces communication steps and secures distance-based exchanges better than standard password schemes. However, it is inefficient due to large cryptographic keys and distribution overhead. **Kumar and Chouhan et al.** [138] introduced the Secure Addressing and Mutual Authentication (SAMA) protocol, designed to safeguard medical IoT networks. The scheme uniquely identifies smart medical monitoring devices through a specialised addressing and identification mechanism, enabling secure mutual authentication. However, the heavy reliance on central authority poses a significant risk; therefore, future systems should aim to reduce such dependency to improve robustness. **Wang et al.** [143] proposed a secure communication framework for the IoT that ensures robust connections between embedded devices and servers. The system leverages formal methods to guarantee security against well-defined challenging problems. Nonetheless, the scheme remains susceptible to replay and message forgery attacks, limiting its reliability in adversarial settings. **Hu et al.** [144] devised a novel approach using event-triggered transmissions to optimise network resource usage and defend against non-periodic Denial-of-Service (DoS) attacks. Their system incorporates a time-delay technique to filter out malicious behaviour through switched system modelling. Despite these innovations, the approach remains vulnerable to specific denial-of-service (DoS) threats, necessitating further enhancements. **Kaur and Kumar.** [145] focused on mitigating offline password guessing and related attacks by developing a user authentication mechanism resilient to replay, gateway bypass, and key agreement vulnerabilities. While the solution addresses many security concerns, it demands significant computational, storage, and communication resources, indicating the need for a more lightweight and efficient implementation. **Li et al.** [146] introduced a three-factor anonymous authentication approach for Wireless Sensor Networks (WSNs) in IoT settings. Their method utilises a fuzzy commitment scheme to safeguard users' biometric data securely. However, the proposed model is hindered by high communication and computational overhead, rendering it less suitable for resource-constrained applications. Finally, Jiang et al. **Jiang et al.** [147] developed an unlinkable upgraded authentication technique aimed at reducing computational

costs. Although this approach enhances anonymity and resource efficiency, it remains vulnerable to DoS attacks and incurs significant performance trade-offs due to its high communication demands.

TABLE 2.1: Comprehensive review of Remote Authentication and Security Schemes

Author	Contribution	Result Obtained	Drawbacks
Hsiang and Shih et al.,[102]	Proposed a secure remote user authentication system using dynamic ID augmentation in a multiserver scenario.	Increased computing cost. Improved the security and efficiency of the authentication scheme.	Passwords are vulnerable to improper changes, impersonation attacks, and replay attacks.
Li and Hwang et al., [146]	Developed a robust and efficient remote user authentication method that integrates biometric verification with smart card technology.	Reduced computation cost using random numbers instead of timestamps, preventing time synchronization issues. Users can update their passwords at will and achieve mutual authentication with the remote server.	Insufficient security enhancement.
Yeh et al., [104]	Developed an authentication technique for remote users using elliptic curve cryptography.	Mutual authentication ensures internal and external security. Improves authentication security.	Increase in computing costs.

Continued on next page

Author	Contribution	Result Obtained	Drawbacks
Vaidya et al., [105]	Introduced a remote user authentication protocol utilizing biometrics and smart cards, designed to offer both high efficiency and strong security.	Reduces computation costs by discarding timestamps, eliminating serious time synchronization issues with random numbers.	Lack of security enhancement.
Xue et al.,[106]	Developed an efficient mutual authentication and key agreement protocol that uses temporal credentials	Provides excellent security without adding communication, processing, and storage overhead.	Susceptible to server impersonation and smart card theft.
Chang et al.,[107]	Developed a biometric authentication mechanism for connected healthcare.	Ensures anonymity and uniqueness. Secure against various attacks and ensure authorized data access.	Not suitable for distributed systems like IoT due to limited protection layers.
Gubbi et al., [108]	Focused on creating a uniform authentication mechanism for the IoT.	Prevents jamming attacks and secures IoT devices.	No evidence that the method enhances data security.
Ndibanje et al., [109]	Presented a secure key establishment strategy for IoT using ECC.	Enhanced device authentication and access control.	Increase in computing costs.
Chen et al., [110]	Developed a secure user authentication strategy for WSNs.	Ensures high security and avoids smart-card loss attacks.	Prone to DoS attacks and credential delays.
Das et al. 2016., [111]	Developed a three-factor user authentication technique.	Immune to sensor capture and impersonation.	Vulnerable to tracking-based attacks.

Continued on next page

Author	Contribution	Result Obtained	Drawbacks
Amin and Biswas et al.,[112]	Smart card-based authentication framework for distributed cloud.	Ensures anonymity, efficiency, and password flexibility.	Vulnerable to tracking, session key leakage, and impersonation.
Farash et al., [113]	Improved key exchange protocol for WSNs.	Resists a variety of attacks.	Still vulnerable to smart card and password-based attacks.
Kaul and Awasthi et al., [114]	Smartcard-based remote authentication protocol.	Efficient with mutual authentication.	Vulnerable to impersonation attacks.
Roy et al. 2017 [115]	Three-factor remote authentication using biometrics and chaotic map.	Lightweight, efficient, suitable for healthcare.	No evidence of data security improvement.
Wazid et al., [118]	Authentication approach for smart homes.	Suitable for low-resource devices.	Vulnerable if gateway verification table is compromised.
Nikoooghadam et al.,[119]	Lightweight key agreement and authentication mechanism.	Ensures user anonymity and low cost.	Vulnerable to replay and password guessing attacks.
Kang et al., [120]	Biometric-based authentication and key exchange for IoT.	Prevents impersonation and replay attacks.	Vulnerable to insider and ephemeral secret leakage attacks.
Shah et al., [121]	Secure authentication using Secure Vault.	Prevents side-channel attacks.	Physical attacks at perception layer possible.
Chandrakaret et al.,[122]	Three-factor authentication for TMIS.	Provides strong security with BAN logic validation.	Vulnerable to replay attacks.

Continued on next page

Author	Contribution	Result Obtained	Drawbacks
Amin et al., [123]	Anonymous password authentication for WSNs.	High security with low power usage.	Vulnerable to replay and insider attacks.
Park et al., [124]	Direct multi-message telemetry for IoT.	Reduces network congestion.	Inadequate authentication and authorization.
Lu et al., [125]	3-factor key exchange using ECC.	Robust against multiple attacks.	Session keys lack full security.
Ostad-sharif et al.,[126]	Lightweight key agreement for IoT-WSNs.	Ensures forward secrecy and efficiency.	Vulnerable to various attacks
Garg et al., [127]	OAuth-based access control using middleware.	Secure REST API communication.	Middleware creates data breach risks.
Banerjee et al., [128]	Anonymous authentication for smart homes.	Resistant to several threats.	Fails to prevent anonymity and trace attacks.
Suresh et al.,[129]	Authentication for TMIS using chaotic maps.	Efficient and formally validated.	Computationally intensive and vulnerable to desynchronization.
Bae and Kwak et al.,[130]	Smart card-based multi-server authentication.	Prevents session key leakage.	Lacks mutual authentication and vulnerable to impersonation.
Deepak et al., [131]	LAKM for IoT-assisted systems.	Secure, low-overhead device connections.	Enhances security with minimal cost.

Continued on next page

Author	Contribution	Result Obtained	Drawbacks
Hinden et al., [132]	Proposed an IPv6 address format that is compatible with the IPv6 Protocol and Architecture	Enables scalable Internet routing and improved management	Advanced technology is required to handle the complexity.
Sabir et al.,[133]	IP address schemes based on fixed- and variable-length subnet masking and aggregation techniques	Allows for greater flexibility in IP address allocation and routing	VLSM might result in wasteful usage of address space
Narten et al., [134]	Extended IPv6 stateless auto-configuration with unique interface identifiers.	Allows for automated IPv6 address setup for devices	Ineffective against denial-of-service attacks.
Hinden and Haberman.,[135]	Proposed Unique Local IPv6 Unicast Addresses for local interactions.	Allows local prefix allocation without global routing.	Requires border routers to enforce filtering.
Judmayer et al., [136]	Developed 6-HOP addressing for secure IoT communications.	Offers simple implementation and protection against basic threats.	Lacks analysis of multi-server deployment and interoperability.
Gont and Chown .,[137]	Replaced RFC 5157 to address IPv6 reconnaissance using new strategies.	Enhances resistance to early-stage reconnaissance.	Lacks encryption-based defence; risk remains in dual-stack networks.
Kumar and Chouhan., [138]	Presented unique addressing and authentication using secure identifiers.	Strengthens authentication and encryption in endpoint communication.	Not validated on real-world testbeds; high complexity.

Continued on next page

Author	Contribution	Result Obtained	Drawbacks
Li et al., [139]	Three-factor authentication for WSNs.	Enhanced security using biometrics.	Inefficient due to high cost.
Dunlop et al. [140]	MT6D for unmodified IPv6 integration.	Resists reconnaissance attacks.	cannot prevent non-network layer attacks.
Tsai et al., [141]	Anonymous authentication using ECC.	Enhanced privacy and authentication.	High computation and communication cost
Nicanfar et al., [142]	Smart meter authentication protocol	Reduces latency and overhead.	Inefficient due to large key sizes
Wang et al., [143]	propose a secure IoT protocol using formal methods,	resulting in improved system security.	Still vulnerable to replay and forgery.
Hu et al., [144]	Anti-DoS method using time-delay modelling	Resists non-periodic jamming.	Susceptible to certain DoS types
Kaur and Kumar., [145]	Password-based authentication system.	Reduces common attack vulnerabilities.	High resource overhead
Jiang et al., [147]	Unlinkable, secure authentication scheme.	Improved unlinkability and security.	DoS vulnerability and high cost

2.4.3 Global Addressing Methods

Global addressing methods encompass address type and allocation [132].

1. Address Type The Internet Engineering Task Force (IETF) developed the IPv6 protocol in 1996 to address security concerns and to prepare for future challenges related to the IPv4 system. IPv6 offers a significantly larger pool of logical addresses compared to IPv4. Below is an outline of the different types of IPv6 addresses.

2. Global Unicast Address (GUA): IPv6 nodes on the Internet use this for routing aggregation and direct connections to service providers. The GUA has three categories: public topology, site topology, and an interface identifier (IID). Figure 2.1 shows the GUA format. The first 48 bits are the global bits, specifically the global routing prefix from the service provider (SP).

3. Link-local address (LLA): In a single routed access network, such as an Ethernet LAN, LLAs serve as both source and destination addresses for IPv6 packets, as depicted in Figure 2.2. These addresses begin with FE80, and the highest 48 bits are set to zero.

4. Unique Local Address (ULA): ULA addresses are private, routing IPv6 packets to homes or enterprises. Figure 2.3 shows the ULA format. IPv6 addresses are associated with interfaces, allowing multiple identifiers to be assigned per interface. The first 64 bits form the network prefix, which comprises a 48-bit global routing prefix and an 8-bit subnet ID that identifies the destination network node. The last 64 bits are unused.

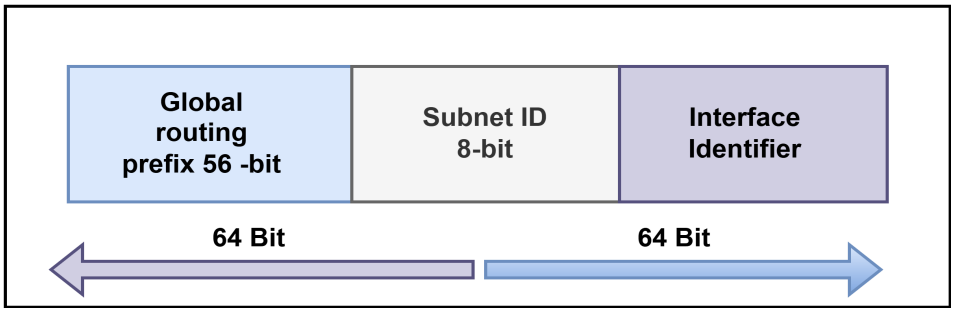


FIGURE 2.1: Global unicast address

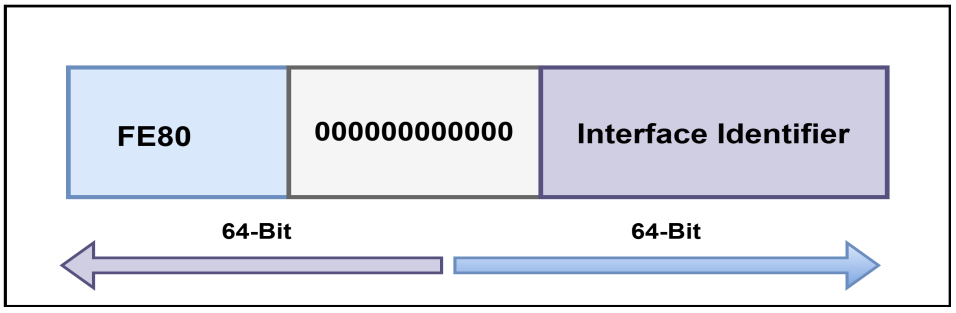


FIGURE 2.2: Link local address

2.4.3.1 IoT address allocation

The IPv6 protocol presents two options for the allocation and assignment of addresses to IoT nodes:

1. State-Less Address Auto-Configuration (SLAAC)- It dynamically assigns addresses to nodes. SLAAC enables nodes to assign themselves link-local addresses (LLAs) without requiring a router. To assign an address, the setup phase sends a router advertisement (RA) message containing the router's allocated network prefix. Several techniques for assigning 64-bit interface identifiers (IIDs) include EUI-64-based methods, temporary or privacy addresses, and cryptographically generated addresses (CGA).
2. State-Full Addresses Auto-Configuration (SFAAC): This approach, commonly known as DHCPv6, is used to assign IPv6 addresses directly. SFAAC requires a DHCP server to track and manage network addresses.

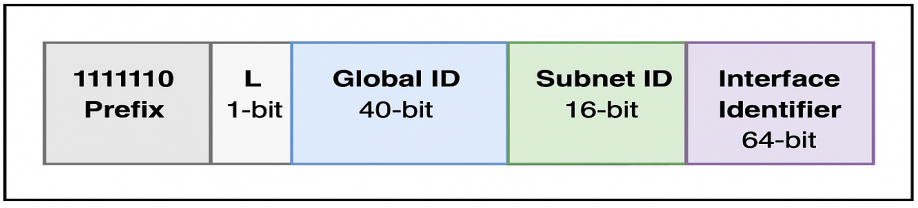


FIGURE 2.3: Unique-local address

2.4.4 Existing Addressing Schemes

Numerous methods have been developed for generating IPv6 addresses. This section reviews static, semi-static, and dynamic addressing techniques, as outlined in Table 2.2. Typically, two forms of static addresses are employed in addressing schemes, often

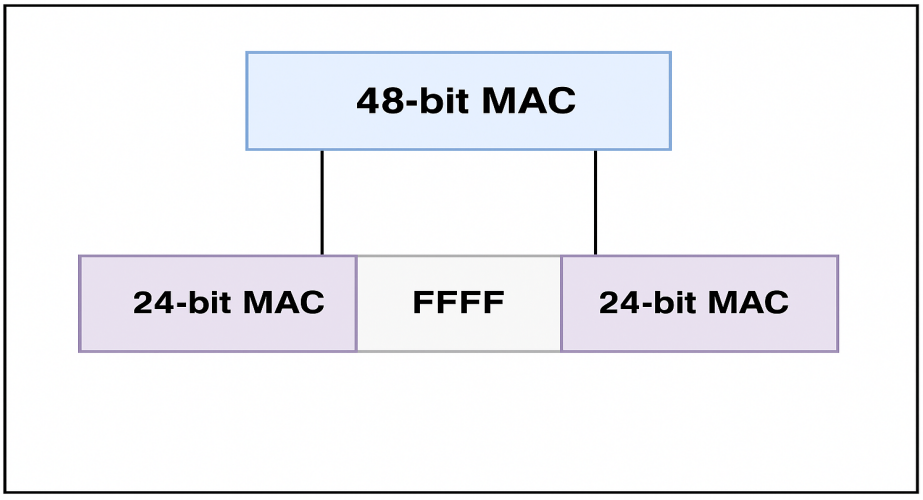


FIGURE 2.4: EUI-64 based IID

utilizing patterns based on port numbers or IPv4 addresses [149]. In [148], a new addressing framework suitable for local communications and inter-site private networks is introduced, though increased ambiguity can raise the risk of address prefix conflicts.

In [132], the authors introduce an IPv6-based Unicast Address Format (UAF) that adheres to IPv6 address allocation standards. Adjustments to the address format enable efficient packet routing across different hierarchical network layers, with these UAF addresses mainly intended for use in local smart home environments. In [150], various strategies for implementing a global unicast address format were explored. This work builds on and extends the approaches in [132] and [135] through adjustments such as registry bit modifications, support for EUI-64 (see Figure 2.4), and data aggregation techniques. However, global unicast IPv6 addressing introduces several challenges, including:

- Topological changes do not affect the IID bits of IPv6 addresses, which cause them to remain static.
- Duplicate IPv6 addresses can occur if two IIDs collide on the same network link.
- The authors note that the final 64 bits of a global unicast IPv6 address are essentially arbitrary and should be regarded as an opaque value [151], as depicted in Figure 2.4 of the standard IPv6 protocol.

In [156], a mobile IPv6 system is provided that allocates COA-based addresses to mobile nodes using the IPv6 Stateless Address Autoconfiguration technique. The mobile

Symbol	[48]	[47]	[48]	[49]	[50]
Randomization	N	N	N	N	Y
Time synchronization	Y	Y	Y	Y	N
Reconnecting	Y	Y	Y	N	Y
DoS attack	N	Y	N	N	Y
Man in the middle attack	N	N	N	N	Y

TABLE 2.2: Comparison of techniques across different approaches

IPv6 system utilizes a distinct Duplicate Address Detection (DAD) algorithm. However, other vulnerabilities have been identified, such as address conflicts and broken node sessions. In [152], the authors demonstrated that configuring a Care-of Address (COA) for mobile IPv6 is effective for enabling real-time packet routing. The IPv6 stateless address configuration mechanism allows devices to self-configure and obtain network addresses automatically. However, in wireless environments, unauthorised nodes can join the network, making it vulnerable to risks such as man-in-the-middle (MITM) attacks, denial-of-service (DoS) attacks, and other security threats. The Cryptographically Generated Address (CGA) approach outlined in [153] aims to ensure the confidentiality and integrity of messages, but its computational complexity requires significant bandwidth. The IPv6 privacy extension and stateless address auto-configuration proposed in [134] offer an alternative approach for IPv6 address assignment. Nevertheless, attackers may exploit the limited user pool by forging server addresses to compromise smart IoT devices. Additionally, if the IoT system relies on static or extended unique identifiers, adversaries could obtain the user's MAC address through these smart devices. In [140], the Moving Target IPv6 Defense (MT6D) method was developed to encapsulate IPv6 packets and shield containers from denial-of-service (DoS) attacks. However, due to its significant overhead, synchronization delays, and slow packet transmission speed, MT6D is not suitable for IoT environments.

The EUI-64 format employs a specific pattern to limit the address space available for probing, thereby enhancing security. This 64-bit identifier generates host addresses derived from Ethernet MAC addresses [137]. However, static address assignments are susceptible to denial-of-service (DoS) attacks and do not provide sufficient protection against them. In [154], a stateless IPv6 address auto-configuration (SLAAC) technique with semantically opaque identifiers was proposed. This system generates unique interface identifiers by deterministically hashing network prefixes into addresses, thereby

enhancing security against malicious attacks. Despite this, such addresses remain vulnerable to DoS threats. Additionally, when a user changes locations, their address must also change, requiring a new interface identifier to be generated. In [121], the authors propose a novel approach to address the assignment that is less computationally demanding than the method in [153]. Their technique is claimed to be secure against denial-of-service (DoS) and device impersonation attacks. However, a key security issue arises from the interface identifier being derived from the static Media Access Control (MAC) address. Additionally, [136] proposes a lightweight IPv6 address hopping (6-HOP) strategy specifically for IoT networks that minimises overhead while offering protection against surveillance and DoS attacks. However, this address-hopping method is not well suited for IoT environments with a large number of smart devices, as it does not consistently maintain optimal performance in all cases.

2.5 Research Gaps

The literature review highlights that security is the foremost concern for both researchers and practitioners in the Internet of Things (IoT). Various deficiencies in this area have been identified, underscoring the urgent need for enhanced protection measures and protocols to secure IoT devices and networks.

1. The exponential growth of the IoT has led to millions of new devices that collect and transmit data; however, many manufacturers are not following best security practices. Millions of potentially vulnerable devices could be exploited, making them susceptible to attacks.

2. Smart home IoT devices generate a significant amount of data that complicates threat detection. An efficient scheme is needed to protect the network from attacks and vulnerabilities.

3. Many current security protocols lack mutual authentication and are susceptible to DoS attacks, excessive overhead, lengthy synchronization times, and slow packet transmission, among other limitations.

4. Due to the rapid growth of smart IoT devices, these schemes often fail to deliver consistent performance in diverse IoT environments. Furthermore, current methods increase communication and computation costs during data transmission.

2.6 Research Objectives

The rapid expansion of interconnected devices has increased the complexity of achieving secure communication, authentication, and identification, particularly given the diversity and limited resources of IoT systems. This research aims to develop efficient, lightweight, and scalable security mechanisms specifically designed for IoT networks. The main objectives include:

- Develop a robust multifactor authentication scheme for IoT environments.
- Design a secure addressing and identification framework for IoT networks.
- Design secure session key establishment utilising unique addressing within IoT infrastructures.

2.7 Research Contribution

The contribution of each work in the thesis is as follows:

1. To address the evolving security challenges in IoT environments, we propose the Authenticated Unidentified Security Scheme (AUSS)—a robust and lightweight multifactor authentication protocol tailored for resource-constrained networks. AUSS employs a three-factor authentication mechanism that combines knowledge (user passwords), possession (device-specific credentials), and inherence (biometric traits) using advanced biometric hash technology. This layered approach significantly strengthens identity verification and minimises the risk of unauthorised access.

   The protocol operates within a three-tier architecture comprising mobile nodes, a gateway, and sensor nodes, enabling efficient distribution of computational load across the system. This architectural design ensures that resource-limited devices, such as sensor nodes, are shielded from heavy cryptographic operations, thereby maintaining optimal performance and energy efficiency.

   To validate its strong security guarantees, AUSS is formally analysed using both BAN logic and the AVISPA tool. The evaluation confirms that the protocol is

"SAFE" and demonstrates resilience against numerous advanced threats, such as replay, impersonation, and man-in-the-middle attacks.

2. To improve security in IoT networks, particularly within smart home environments, we present SLAPSH (Secure and Lightweight Authenticated Protocol for Smart Home)—a novel and efficient security architecture specifically tailored for Smart Home IoT (SH-IoT) systems. The core contribution of this research lies in the design of a lightweight authentication protocol that ensures mutual authentication between IoT devices and servers, thereby effectively preventing unauthorised access.

   A key innovation in SLAPSH is the introduction of a unique addressing scheme that modifies the traditional IPv6 protocol, enabling each device to possess a secure and distinct identity. This enhancement not only facilitates safe and reliable communication but also strengthens device-level security and traceability. This protocol ensures data confidentiality, message integrity, and end-to-end privacy, all while maintaining low computational and communication demands, making it particularly suitable for smart home devices with limited resources.

   To validate its security, SLAPSH undergoes rigorous formal analysis, which is conducted using the ROR model and is tested with the AVISPA verification tool. These evaluations demonstrate the protocol's resilience against various attacks, such as replay, impersonation, and man-in-the-middle attacks. Additionally, network performance is evaluated through NS-3 simulations, with a focus on key metrics, including end-to-end delay and throughput.

   The simulation results and security analyses collectively demonstrate that SLAPSH achieves high security and operational efficiency, establishing it as a scalable and dependable solution for securing smart home IoT networks now and in the future.

3. To strengthen IoT network security at the foundational level, the third objective focuses on the development of Secure and Unique Addressing with Mutual Authentication Scheme (SUMAS) and Secure and Lightweight Authenticated Protocol for Smart Home (SLAPSH). These schemes are designed to embed security directly into the network addressing architecture, offering a seamless and efficient means of identification and authentication within IoT environments.

   Both SUMAS and SLAPSH achieve this by modifying the standard IPv6 address structure, specifically by restructuring the 64-bit Interface Identifier (IID) into two components: a 48-bit owner ID, derived from a securely hashed unique identifier,

and a 16-bit device ID. This innovative format integrates authentication information within the address, enabling destination-based verification without requiring additional message overhead. Moreover, this design ensures full compatibility with existing IPv6 infrastructure, facilitating easy deployment across heterogeneous networks. The systems also include timestamp-based verification procedures to improve security against network-level hazards, hence reducing man-in–the-middle (MITM) assaults during device communications and replay attacks. Particularly in settings with limited resources, SUMAS and SLAPSH provide a scalable and effective way to protect IoT networks by combining secure addressing with lightweight mutual authentication.

2.8 Summary

This chapter gives an overview of current practices in privacy preservation and authentication, along with the associated security solutions. It discusses various systems designed to enhance the security and privacy of smart home environments. Different frameworks and security strategies are explored to improve the efficiency of smart home systems. Additionally, the proposed models and network-based approaches aim to reduce communication failures, minimise response times, improve device coordination, and enhance data dissemination within smart home networks.

Despite these measures to protect privacy and security, smart home systems remain vulnerable to various security attacks. These include replay attacks, man-in-the-middle attacks, modifications, impersonation, key guessing, and issues related to non-repudiation. Thus, there is a pressing need for new techniques and algorithms to strengthen the security, privacy, and efficiency of smart home networks. The chapter also highlights technical research gaps identified through a literature survey.

CHAPTER 3
Multifactor Unidentified Remote User Authentication Scheme for IoT Network

3.1 Introduction

The IoT consists of a network of resource-constrained nodes densely deployed across various environments. These devices are designed to function autonomously, often making quick decisions without human intervention. Due to the variety of architectures and platforms used in IoT development, each system exhibits unique characteristics and constraints, introducing significant complexity, particularly within smart home applications, as noted by the author [161]. These systems operate continuously, independently of location or time, and serve critical roles in various domains, including healthcare, home automation, industrial manufacturing, and urban infrastructure. The advent of 5G technology has raised expectations for seamless interconnectivity between mobile devices and everyday objects, growing faster and more reliable data sharing. However, this increased connectivity also amplifies the risks associated with the misuse of IoT technologies, particularly in residential settings, where it can potentially compromise both safety and privacy. As suggested by the author [162], security and privacy must be prioritised. One of the primary means of achieving this is through robust authentication protocols that verify the identities of users and servers before any data exchange. Smart home networks are especially vulnerable due to the coexistence of various communication standards and the challenge of adapting to rapidly evolving cyber threats. Consequently, securing IoT systems is crucial for protecting user data and maintaining trust in connected environments. Secure IoT networks are essential for protecting user's

privacy and safeguarding against emerging threats. To achieve comprehensive protection, it is necessary to implement strong security mechanisms that encompass virtual network protection, data confidentiality, service availability, and data integrity. Furthermore, authentication protocols must comply with rigorous security and performance standards to ensure reliable communication within IoT ecosystems. The AUSS scheme proposed in this work is particularly well-suited for IoT environments as it delivers both computational and communication efficiency at a low cost.

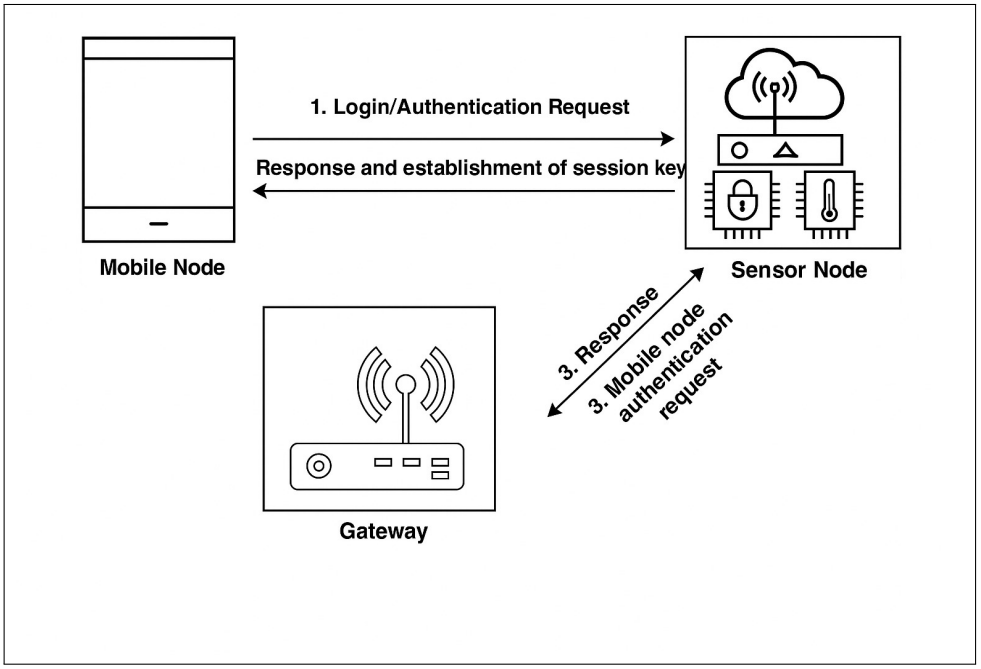


FIGURE 3.1: Proposed Model for user authentication in the IoT

3.2 Main Contribution of the Proposed Scheme

This work improves upon the user authentication method proposed by the authors [116] by addressing its security vulnerabilities. Figure 3.1 illustrates the proposed model, called the Authenticated Unidentified Security Scheme (AUSS). We present a more robust approach that has been validated through both informal and formal security evaluations using BAN logic and the AVISPA tool. The AUSS model demonstrates strong resistance to various types of attacks and meets essential security requirements. Moreover, performance evaluations indicate that the method is lightweight, efficient, and suitable for low-cost IoT devices in practical applications.

1. **Enhanced Multi-Factor Authentication:** A lightweight authentication scheme is proposed for IoT networks that combines three authentication factors: password, biometrics and device identity, offering greater protection compared to conventional two-factor approaches.

2. **User Privacy and Anonymity:** The scheme maintains user anonymity and ensures unlinkability by employing hashed and encrypted credentials, preventing identity tracing and activity profiling.

3. **Session Key Agreement and Revocation:** The system includes robust session key generation and supports revocation of credentials, allowing secure handling of lost or compromised devices.

4. **Resistance to Known Attacks:** Using BAN logic for formal analysis and the AVISPA tool for simulations, we evaluated the proposed protocol's security. The findings confirm that it effectively resists impersonation, replay, insider threats, and stolen verification attacks, showcasing its robustness against vulnerabilities.

5. **Efficiency for IoT Devices:** The scheme employs a range of efficient operations, including hashing, XOR (exclusive OR), and symmetric encryption. These techniques are not only computationally lightweight but also designed to optimise performance, making the system particularly well-suited for resource-constrained IoT environments where processing power and memory are limited.

3.3 Related Work

The IoT's fast expansion raises demand for safe authentication solutions meant for devices with limited resources. Maintaining data privacy, preserving system integrity, and guaranteeing the consistent running of the system depend on strong user authentication. Particularly for wireless sensor networks (WSNs), which are very vital in IoT networks, many light-weight, safe, and privacy-oriented solutions have developed during the past twenty years. The author [165] developed a mutual authentication protocol that utilises lightweight operations, such as hashing and XOR. While this method improved computation time and was suitable for IoT devices, it lacked strong privacy guarantees, including user anonymity and untraceability. The author [166] created a two-factor user authentication scheme via a gateway (GW). To address these challenges, author [117] proposed a password-based protocol for Wireless Sensor Networks (WSNs). However,

[166] revealed vulnerabilities, such as impersonation risks if a user shares a login ID or if the verifier is compromised. These issues required enhancements, like two-factor authentication using gateways for user-sensor node interactions. Despite improvements, many protocols still fail to ensure essential properties, such as mutual authentication and resilience against insider threats. To enhance authentication within wireless sensor networks (WSNs), [167] put forth a provably secure and flexible scheme tailored for ad hoc environments. Although their solution achieved efficiency and adaptability, it lacked mechanisms to address dynamic identity management. To resolve this issue, [168] presented an innovative key agreement protocol using dynamic credentials for secure authentication as the node identities changed. To enhance authentication in wireless sensor networks (WSNs), Chang and Le proposed a flexible scheme optimized for ad hoc environments. While their solution achieved efficiency, it lacked identity management mechanisms. Addressing this, Yang et al. introduced a key agreement protocol with dynamic credentials for secure authentication as node identities evolved. Additionally, [169] developed a smart card–based user authentication scheme utilizing biometric data to improve resilience against common attacks. Nonetheless, challenges like computational overhead and scalability in resource-constrained environments persist. A recent study by [164] proposed a three-factor authentication model that combines passwords, biometrics, and mobile devices to prevent password-guessing attacks, denial-of-service attacks, and spoofing. However, this model lacks secure session key agreement and revocation mechanisms, which could lead to impersonation in the event of device theft. These findings highlight vulnerabilities in existing protocols, which often prioritize computational efficiency over security. Many of these protocols lack formal proofs of correctness, do not adequately resist contemporary cyberattacks, and are impractical for real-time IoT systems. Therefore, there is an urgent need for a user authentication framework that ensures mutual authentication, forward secrecy, privacy protection, resilience against attacks, and reduced system overhead.

3.4 Networking Model and Authentication Mechanism

The Authenticated Unidentified Security Scheme (AUSS) framework is structured around the typical architecture of an IoT network, featuring three main components: the mobile user node (M_n), the sensor node (S_n), and the gateway node (GW). Every component serves a certain purpose guaranteeing the security of the network. User-operated devices having biometric sensors that create authentication requests are mobile nodes.

Data collecting sensor nodes help to enable network connectivity. Acting as a trusted authority, the gateway guarantees safe interactions between nodes and helps users to be authenticated. Network communication takes place across several hops. The process starts when a mobile node phones a sensor node to start an authentication request. The gateway is then passed this confirmational request. After the request has been validated, the gateway provides cryptographic tools to generate a safe session between the user and the sensor node, therefore ensuring safe communication and hence circumventing IoT device limitations.

3.4.1 Authentication Procedure

The proposed authentication protocol is structured into a series of systematic steps to ensure mutual authentication and the secure establishment of session keys. This protocol is divided into four main phases: registration, login and authentication, password update, and revocation.

1. **Registration Phase:** In this initial phase, the user selects a unique identity (ID_i), sets a password (Pw_i), and submits biometric information (Bio_i). These details are processed using cryptographic hash functions to generate secure tokens, which are transmitted to the gateway via a secure communication channel. The gateway, in turn, generates random nonces and secret keys, establishes cryptographic links, and returns the necessary tokens to the user for storage on their device.

2. **Login and Authentication Phase:** During this phase, the user provides their credentials, which are used to recreate verification parameters. These are sent to the sensor node and then forwarded to the gateway. The gateway validates the submitted data by comparing it with previously stored values and computes fresh cryptographic tokens. Once the information is authenticated, a session key is established and securely distributed between the user and the sensor node.

3. **Password Update Phase:** If the user wishes to change their password, the system first verifies the old credentials. Once validated, the new password is used to compute updated tokens that replace the previous values in both the mobile device and the gateway records.

4. **Revocation Phase:** In the event of loss or compromise of the device, the user can request revocation of their account. The user submits both old and new identifiers

and credentials. The gateway verifies the request and, if valid, generates a new set of secure parameters. These are sent back to the mobile node to restore access securely.

The design ensures critical security properties, including user anonymity, mutual authentication, unlinkability, and resistance to various attacks, such as impersonation, replay, and stolen verifier attacks. By relying on computationally lightweight operations such as hashing, symmetric encryption, and XOR, the protocol remains efficient and suitable for low-power IoT environments.

3.4.2 Bio-Hash functions

The suggested AUSS system uses biometric hash to improve authentication security in Internet of Things systems. From biometric data, this technique generates small, non-reversible binary codes. Particularly fit for low-power devices, these bio-hashes are robust against environmental fluctuations and noise. A fused credential is produced by combining the user's password and biometric data using a secure hash function during registration. This mix increases defense against spoofing and impersonation assaults.

Encrypted to facilitate multi-factor authentication—which includes user ID masking and random token generation—the bio-hashed output is then This method guarantees the robust identity assurance required for safe IoT authentication together with making the credentials revocable, storage-efficient, and privacy-preserving.

3.5 Proposed Scheme

This section discusses AUSS, a lightweight and secure authentication technique intended exclusively for IoT devices. The AUSS system seeks to get beyond the restrictions of earlier multi-factor authentication systems. While guaranteeing low processing cost, it combines biometric data, passwords, and device identities. There are four primary phases to the system: registration, login and authentication, password update and revocation. Every phase is meticulously crafted to preserve security while using IoT devices' resources as minimally possible. Table 3.1 provides a list of symbols along with their definitions.

TABLE 3.1: List of symbols and their descriptions

Symbol	Description
Sn_i	Represents the sensor node
Mn_i	Refers to the mobile node
Id_i	Identifier for the mobile device
Pw_i	Password associated with the mobile node
Id_i, Sn_i	Unique identities of Sn_i and Id_i
Bio_i	Biometric information of Mn_i
T_x	Denotes a timestamp value
n_x, r_x	Random values generated during protocol steps
SK	Session key shared by Mn_i and Sn_i
$E_K(.)$, $D_K(.)$	Symmetric encryption and decryption with key K
$H(.)$	Cryptographic hash function
$\|\|$	Concatenation operation
$\oplus$	Bitwise XOR operation
K_{gu}	Private key belonging to Mn_i
K_{gn}	Secret key jointly held by Sn_i and gateway (GW)

3.5.1 Registration Phase

In this phase, the mobile user selects a unique identity (ID_i), a password (Pw_i), and provides biometric input (Bio_i). The device computes a hashed combination of the password and biometric input.

$$PwBi = H(Pw_i \,||\, H(Bio_i)), \quad Mid_i = H(ID_i \,||\, H(Bio_i))$$

This parameter $\langle ID_i, PwBi, Mid_i \rangle$ is securely transmitted to the Gateway (GW). The gateway generates random numbers r_{gu} and r_d, then computes the following values:

$$Rid_i = E_K(ID_i), \quad Pid_i = E_K(ID_i \,||\, r_d)$$

$$X_i = H(ID_i \,||\, PwBi), \quad Y_i = H(ID_i \,||\, PwBi \,||\, r_{gu}) \oplus H(K_{gu} \,||\, ID_i)$$

The parameters $\langle Pid_i, X_i, Y_i, r_{gu} \rangle$ are securely sent back to the user for storage on their mobile device. The gateway stores the tuple (Rid_i, Mid_i) in its secure database. Table 3.2

Entity	Operation
Mobile Node (M_n)	Chooses ID_i, Pw_i, and Bio_i Performs computations: $PwBi = H(Pw_i \| H(Bio_i))$ $Mid_i = H(ID_i \| H(Bio_i))$ Transmits $\langle ID_i, PwBi, Mid_i \rangle$ to the Gateway over a secure connection
Gateway (GW)	Creates random values r_{gu} and r_d Calculates: $Rid_i = E_K(ID_i)$ $Pid_i = E_K(ID_i \| r_d)$ $X_i = H(ID_i \| PwBi)$ $Y_i = H(ID_i \| PwBi \| r_{gu}) \oplus H(K_{gu} \| ID_i)$ Saves: (Rid_i, Mid_i) for future reference Sends $\langle Pid_i, X_i, Y_i, r_{gu} \rangle$ back to M_n
Mobile Node (M_n)	Securely stores the parameters received on the device

TABLE 3.2: Registration Phase: Secure initialization of identity and parameters

illustrates the step-by-step operations carried out by the mobile node and the gateway during the registration phase.

3.5.2 Login and Authentication Phase

Table 3.3 illustrates that the login and authentication process requires several cryptographic operations between the mobile node, sensor node, and gateway to create a secure session key. When accessing the network, the user inputs ID_i, Pw_i, and Bio_i into their device. The stored and newly computed values are compared to ensure consistency. A nonce n_i is generated, and the following values are computed:

$$A_i = Y_i \oplus H(ID_i \,\|\, PwBi \,\|\, r_{gu}), \quad UN_i = H(A_i \,\|\, Pid_i \,\|\, n_i), \quad UZ_i = n_i \oplus A_i$$

The authentication request $M_1 = \langle Pid_i, UN_i, UZ_i, T_1 \rangle$ is sent to the sensor node (S_n), which forwards it to the gateway.

The gateway verifies timestamp freshness and validates credentials by recalculating values. If successful, it generates new cryptographic tokens and shares them securely with the user and sensor node. A session key is derived using:

Entity	**Operation**
Mobile Node (M_n)	Inputs ID_i, Pw_i, Bio_i Computes $PwBi$, $X_i^* = H(ID_i \| PwBi)$ Verifies $X_i^* = X_i$ Generates random n_i Computes: $A_i = Y_i \oplus H(ID_i \| PwBi \| r_{gu})$ $UN_i = H(A_i \| Pid_i \| n_i)$ $UZ_i = n_i \oplus A_i$ Sends $M_1 = \langle Pid_i, UN_i, UZ_i, T_1 \rangle$ to Sensor Node
Sensor Node (S_n)	Forwards M_1 to Gateway Verifies timestamp Generates n_j and computes: $x_j = y_j \oplus H(K_{gn} \| r_j \| Nid_j)$ $A_j = H(x_j) \oplus n_j$ $B_j = H(x_j \| n_j)$ Sends $M_2 = \langle M_1, Nid_j, A_j, B_j \rangle$ to Gateway
Gateway (GW)	Verifies credentials Computes session key: $F_j = H(ID_i \| n_i^*)$ $G_j = F_j \oplus x_j^*$ $R_{ij} = n_j^* \oplus n_i^*$ $H_j = H(x_j^* \| n_j^* \| n_i^* \| F_j)$ Sends $M_3 = \langle Pid_i^{new}, G_j, R_{ij}, H_j \rangle$ to M_n
Sensor Node (S_n)	Validates H_j and computes: $L_j = H(Nid_j \| n_i^*) \oplus m_j$ $SK_{ji} = H(F_j^* \| n_i^* \| m_j)$ $SV_j = H(SK_{ji} \| T_1 \| T_2)$ Sends $M_4 = \langle Pid_i^{new}, L_j, SV_j, T_2 \rangle$ to M_n
Mobile Node (M_n)	Verifies SV_j, computes session key: $m_j^* = L_j \oplus H(Nid_j \| n_i)$ $SK_{ij} = H(H(ID_i \| n_i) \| n_i \| m_j^*)$ $SV_i = H(SK_{ij} \| T_1 \| T_2)$ Session key is established if $SV_i = SV_j$

TABLE 3.3: Login and Authentication Phase: Establishing mutual trust and session key

$$SK_{ij} = H(F_j \| n_i \| m_j), \quad SV_j = H(SK_{ij} \| T_1 \| T_2)$$

Both the user and sensor node independently compute and confirm the session key, ensuring mutual authentication.

3.5.3 Password Change Phase

The user initiates the password change process by providing their current identity, the old password, the new password, and their biometric data. The system verifies the validity of the old credentials. If the credentials are correct, it generates new hashed values and updates the stored records on both the mobile device and the gateway. Table 3.4 outlines the operations performed during the password change phase, ensuring secure updates of credentials on the mobile node.

Entity	**Operation**
Mobile Node (M_n)	Inputs: ID_i, Pw_i^{old}, Pw_i^{new}, Bio_i Verifies: $X_i^* = H(ID_i \| H(Pw_i^{old} \| H(Bio_i)))$ If valid, computes: $A_i = Y_i \oplus H(ID_i \| Pw_i^{old} \| r_{gu})$ $PwBi^{new} = H(Pw_i^{new} \| H(Bio_i))$ $X_i^{new} = H(ID_i \| PwBi^{new})$ $Y_i^{new} = H(ID_i \| PwBi^{new} \| r_{gu}) \oplus A_i \oplus Y_i$ Updates X_i and Y_i locally

TABLE 3.4: Password Change Phase: Secure update of user credentials

3.5.4 Revocation Phase

Table 3.5 describes the revocation process that allows a mobile node to securely update its identity and password in the event of a credential compromise. If a user's device is compromised or their credentials need to be updated, the revocation mechanism is activated. The user must provide their old identity information, new identity information, and biometric data. The gateway verifies the old credentials against its records. Once validated, it generates new secure parameters as follows:

$$PwBi_{new} = H(Pw_i^{new} \| H(Bio_i)), \quad Mid_i^{new} = H(ID_i^{new} \| H(Bio_i))$$

Entity	Operation
Mobile Node (M_n)	Inputs: $ID_i^{old}, ID_i^{new}, Pw_i^{new}, Bio_i$ Computes: $PwBi^{new} = H(Pw_i^{new} \| H(Bio_i))$ $Mid_i^{old} = H(ID_i^{old} \| H(Bio_i))$ $Mid_i^{new} = H(ID_i^{new} \| H(Bio_i))$ Sends: $\langle ID_i^{old}, ID_i^{new}, Mid_i^{old}, Mid_i^{new}, PwBi^{new} \rangle$ to Gateway
Gateway (GW)	Verifies: $Rid_i^{old} = E_K(ID_i^{old})$ If valid, generates: $Pid_i^{new} = E_K(ID_i^{new} \| r_d^{new})$ $Rid_i^{new} = E_K(ID_i^{new})$ $X_i^{new} = H(ID_i^{new} \| PwBi^{new})$ $Y_i^{new} = H(ID_i^{new} \| PwBi^{new} \| r_{gu}^{new}) \oplus H(K_{gu} \| ID_i^{new})$ Updates its database and sends updated parameters to M_n
Mobile Node (M_n)	Stores updated values securely on the device

TABLE 3.5: Revocation Phase: Recovery from compromised or lost credentials

New encrypted identifiers and authentication tokens have been implemented to enhance security, replacing the outdated credentials. These new tokens are securely stored to ensure that only authorized users can access sensitive information. The old credentials have been discarded and are now deemed invalid.

3.5.5 Security and Efficiency

The AUSS scheme offers robust security features, including anonymity, forward secrecy, mutual authentication, and protection against impersonation, replay, and man-in-the-middle attacks. Its use of hash functions, symmetric encryption, and XOR operations ensures computational efficiency, making it ideal for deployment on devices with limited resources.

3.6 BAN Logic-Based Authentication Proof

To formally validate the correctness and establish mutual trust through the proposed AUSS scheme, we utilise the Burrows–Abadi–Needham (BAN) logic [99]. This framework outlines the reasoning behind the beliefs held by the parties involved in a security protocol. Our objective is to demonstrate that both the mobile user (M_n) and the sensor node (S_n) mutually trust the freshness and authenticity of the established session key.

3.6.1 BAN Logic Notations

The following notations are used in our BAN logic analysis:

- $P \mid\equiv X$: Principal P believes statement X.
- $P \lhd X$: Principal P sees or receives message X.
- $\#(X)$: Statement X is fresh (not reused).
- $P \mid\sim X$: Principal P once said X.
- $P \leftrightarrow_K Q$: Principals P and Q share a secret key K.
- $\{X\}_K$: Message X is encrypted with key K.

Authentication Goals:

The primary goals of the protocol are:

1. $M_n \mid\equiv M_n \leftrightarrow_{SK} S_n$ (G1)
2. $S_n \mid\equiv M_n \leftrightarrow_{SK} S_n$ (G2)
3. $M_n \mid\equiv S_n \mid\equiv M_n \leftrightarrow_{SK} S_n$ (G3)
4. $S_n \mid\equiv M_n \mid\equiv M_n \leftrightarrow_{SK} S_n$ (G4)

Assumptions and Initial Beliefs:

- $M_n \mid\equiv \#(T_1)$: M_n believes T_1 is fresh.
- $S_n \mid\equiv \#(T_2)$: S_n believes T_2 is fresh.
- $GW \mid\equiv \#(K)$: Gateway believes key K is fresh.
- $M_n \leftrightarrow_{SK} S_n$: M_n and S_n share a session key.
- $GW \mid\equiv M_n \leftrightarrow_K S_n$: The gateway certifies that M_n and S_n share a valid key.

Idealized Protocol Messages:

Let us express key messages from the authentication exchange in idealized form:

- M_1: $M_n \rightarrow S_n$: $\{A_i, Pid_i, T_1\}_{n_i}$
- M_2: $S_n \rightarrow GW$: $\{M_1, Nid_j, A_j, B_j\}_{x_j}$
- M_3: $GW \rightarrow S_n$: $\{F_j, n_j, n_i, K\}_{x_j}$
- M_4: $S_n \rightarrow M_n$: $\{Pid_i^{new}, L_j, SV_j, T_2\}_{m_j}$

Proof Outline:

1. From M_1, S_n receives Pid_i, A_i, T_1 and believes:

$$S_n \mid\equiv M_n \mid\sim (Pid_i, A_i, T_1) \quad \text{(by Rule 1)}$$

2. With $S_n \mid\equiv \#(T_1)$, it derives:

$$S_n \mid\equiv \#(Pid_i, A_i, T_1) \quad \text{(Rule 4: freshness)}$$

3. Hence, S_n concludes:

$$S_n \mid\equiv M_n \mid\equiv (Pid_i, A_i, T_1) \quad \text{(Rule 2: nonce-verification)}$$

4. Similarly, from M_4, M_n receives L_j, SV_j, T_2, and believes:

$$M_n \mid\equiv S_n \mid\sim (L_j, SV_j, T_2)$$

5. Using freshness of T_2 and proper derivation of SV_j, M_n infers:

$$M_n \mid\equiv S_n \mid\equiv (SK_{ij}, T_1, T_2)$$

6. Therefore, both M_n and S_n believe they share the fresh session key SK:

$$M_n \mid\equiv M_n \leftrightarrow_{SK} S_n \quad \text{(G1)}$$

$$S_n \mid\equiv M_n \leftrightarrow_{SK} S_n \quad \text{(G2)}$$

$$M_n \mid\equiv S_n \mid\equiv M_n \leftrightarrow_{SK} S_n \quad \text{(G3)}$$

$$S_n \mid\equiv M_n \mid\equiv M_n \leftrightarrow_{SK} S_n \quad \text{(G4)}$$

We have employed BAN logic analysis to demonstrate that the proposed authentication scheme satisfies key security objectives, including mutual authentication and secure session key agreement. As a result, both the mobile user and the sensor node can trust the authenticity and freshness of the session key. This trust enables secure communications in IoT environments.

```
Role alice (Ui, GWN. SNj: agent,
H: hash func.
SKuigwn: symmetric_key,
Snd, Rcv: channel(dy))
played by Ui
def= local State : nat,
IDi, IDsnj, K, PWi, Bi, T1, T2, T3: text,
Xs, EKi. Ki, Request, R, RPWi : text,
Gen, Rep: hash, func
const alice server_t1, server_bob_t2,
bob_ alice_t3, sub1, sub2, sub3, sub4 : protocol_id
init State := 0
transition
1. Stale = 0 /\ Rcv(start) =|>
% Registration phase
State' = 1 A K'= new()
/\ secret((PWi,Bi,K'),sub1.Ui)
/\ secret(EKi,sub2,{Ui,GWN})
/\ RPWi'= H(IDi.PWi.K')
% Ui sends login message to GWN securely
/\ Snd({IDi.RPWi'.EKi}.SKuigwn)
% Ui receives the smart card from GWN securely
2. State = 1 /\ Rcv ((H.Gen. Rep.H(xor(IDi.H(Xs)))_SKuigwn) =|>
% Login phase
State' := 2 /\ secret(Xs,sub3,GWN)
% Ui sends the login message to the GWN
/\ Snd(IDi.Request)
% Authentication and key agreement phase
% Ui reccives the message <R> from GWN
3. State = 2 /\ Rcv(R') =|>
State' = 3 /\ T1' := new()
% Ui sends the message <E_eki(R,T1,IDsnj)> to GWN
/\ Snd((R'.T1'.IDsnj}_ EKi)
% Ui has freshly generated the value T1 for GWN
/\ witness(Ui, GWN ,alice_server_ti, TI')
% Ui receives the message from sensor node SNj
2. State = 3 /\ Rcv (H (H(H (IDsnj.H (xor (IDi,H(Xs)))).
IDi.IDsnj.T1'.T3').T3')) =|>
% Ui's acceptance of the value T3 generated for Ui by SNj
State':= 4 /\ request(SNj, Ui, bob_alice_t3, T3')
end role
role bob (Ui, GWN, SNj: agent,
H: hash_func,
SKuigwn: symmetric_key,
Snd, Rcv: channel(dy))
played_by SNj
def=
local State: nat,
IDi, IDsnj, K, PWi, Bi, TI, T2, T3: text,
Xs, EKi, Kj, Request, R, RPWi: text,
Gen, Rep: hash_func
const alice_server_t1, server_bob t2
```

```
bob_alice_t3, sub1, sub2, sub3, sub4: protocol_id
init State := 0
transition
% Authentication and key agreement phase
% Receive the message from the GWN
1. State = 0 /\ Rcv(IDi. (IDi.IDsnj.TI'.
T2'.H(IDsnj.H(xor(IDi,H(Xs))))] _ Kj)=|>
State': = 1 /\ T3' := new()
/\ secret((PWi,Bi,K),sub,Ui)
/\ secret(EKi,sub2,{Ui, GWN})
/\ secret(Xs,sub3,GWN)
/\ secret(Kj,sub4, {GWN,SNI})
% Send the message to Ui
/\ Snd(H(H(H (IDsnj.H(xor(IDi,H (Xs)))).
IDi.IDsnj,TI'.T3').T3'))
% SNj has freshly generated the value T3 for SNj
/\ witness(SNj.Ui,bob_alice_t3.T3')
% SNi's acceptance of the value T2 generated for SNj by
GWN
/\ request(GWN, SNj, server_bob_t2, T2')
end role
role server (Ui, GWN, SNj: agent,
H: hash_func,
SKuigwn: symmetric_key,
Snd, Rcv: channel(dy))
played_by GWN
def=
local State: nat,
IDi, IDsnj, K, PWi, Bi, T1, 12, T3: text,
Xs, EKi, Kj, Request, R, RAWi: text,
Gen, Rep: hash_func
const alice_server_t1, server_bob_t2,
bob_alice_t3, sub1, sub2, sub3, sub4 : protocol_id
init State := 0
transition
end role
% Registration phase
% GWN receives login message from UI securely
1. State = 0 /\ Rcv((IDi.H(IDi.PWi.K').EKi)_SKuigwn)=|>
State' := 1 /\ secret (PWi,Bi,K'},sub,Ui)
% GWN sends the smart card to Ui securely
/\ Snd({H.Gen.Rep.H(xor(IDi,H(Xs)))]_SKuigwn)
% Login phase: receive the login request message from Ui
2. State = 1 /\ Rcv(IDi.Request)=|>
State': = 2 /\ R' = new()
/\ secret(EKi,sub2, {Ui, GWN})
/\ secret(Xs,sub3,GWN)
/\ secret(Kj,sub4, {GWN,SNj})
% Authentication and key agreement phase
% GWN sends the message to Ui
/\ Snd(R')
end role
```

FIGURE 3.2: Role for user and gateway node

3.7 AVISPA tool simulation for formal security verification

This section presents a formal security evaluation of the proposed AUSS scheme, conducted using the AVISPA tool. Among AVISPA's four available back-end options, this evaluation specifically employs the On-the-Fly Model Checking (OFMC) back end. The protocol is described using the High-Level Protocol Specification Language (HLPSL) to assess its resilience to common security threats [98]. The CAS+ specifications are translated into HLPSL within AVISPA via the SPAN animator tool, which, in intruder mode, produces a message sequence chart (MSC). AVISPA and SPAN are widely used by researchers and practitioners for validating protocol security. Figure 3.2 illustrates the user and home gateway roles, Figure 3.3 presents the session and environment roles, and the OFMC analysis results are presented in Figure 3.4.

```
role session(Ui, GWN,SNj: agent,
% H is hash function
H: hash_func,
SKuigwn: symmetric_key)
def=
local US, UR, SS, SR, VS, VR: channel (dy)
composition
alice(Ui, GWN, SNj, H, SKuigwn, US, UR)
/\ server(Ui, GWN, SNj, H, SKuigwn, SS, SR)
/\ bob(Ui, GWN, SNj, H, SKuigwn, VS, VR)
end role
role environment)
def=
const ui, gwn, snj: agent,
h, gen, rep: hash_func,
skuigwn: symmetric_key,
idi, idsnj, t1, t2, t3 : text,
alice_server_tl, server_bob_t2,
bob_alice_13, sub1, sub2,
sub3, sub4 : protocol_id
intruder_knowledge = (idi,h,gen,rep,t3 )
composition
session (ui, gwn, snj, h, skuigwn)
session(ui, gwn, snj, h, skuigwn)
/\ session(ui, gwn, snj, h, skuigwn)
end role
```

FIGURE 3.3: Role for session and environment

3.8 Performance Evaluation

We evaluated the proposed AUSS scheme authentication protocol to assess its computational and communication efficiency. Designed for IoT devices with limited processing

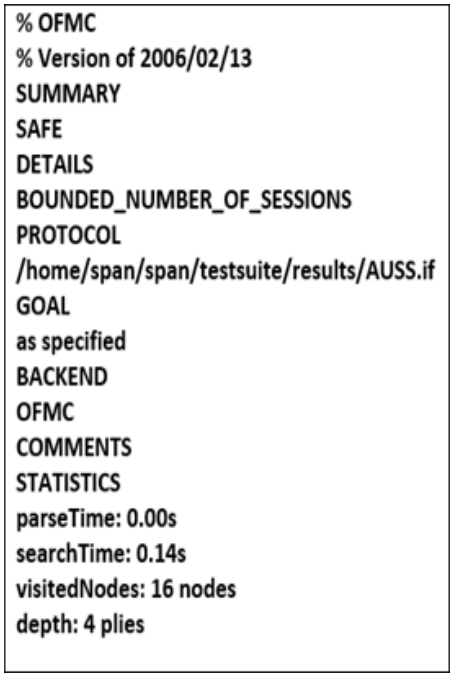


FIGURE 3.4: OFMC output

power and energy, the scheme is lightweight and scalable, making it ideal for various applications. The evaluation utilized three representative platforms to simulate the real-world IoT environment:

- **Mobile Node:** A Galaxy Note 9 device running Android 9.0, powered by an Octa-Core processor (2.7GHz + 1.7GHz) with 8 GB RAM. Development and testing were conducted using Android Studio and its associated software development kits (SDKs).
- **Sensor Node:** An LPC1768 microcontroller with an ARM Cortex-M3 CPU running at 100 MHz, 512 KB flash memory, and 64 KB SRAM.
- **Gateway Node:** A PC equipped with an Intel Pentium G4600 (3.60 GHz) processor and 8 GB RAM, operating on 64-bit Windows 10. The Crypto++ library was used for cryptographic implementations in Visual Studio 2017.

To quantify the costs associated with cryptographic operations, we referenced benchmark results obtained from Abbasinezhad-Mood and Nikooghadam [171]. The average execution times for cryptographic primitives are as follows:

- **Mobile Node:** $T_e \approx 28.48\,\mu s$, $T_s \approx 74.2\,\mu s$, $T_h \approx 104.38\,\mu s$

- **Sensor Node:** $T_e \approx 1264\,\mu s$, $T_h \approx 14.5\,\mu s$
- **Gateway Node:** $T_e \approx 2224\,\mu s$, $T_s \approx 5.41\,\mu s$, $T_h \approx 4.95\,\mu s$

TABLE 3.6: Functionality and Security Attribute Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol

Scheme	UAA	UUA	SMDA	MA	SKAA	UIA	RA	UVA	SVA	PIA	PCA	FSA	SNIA	RPA
[164]	✓	×	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×
[165]	✓	×	✓	✓	✓	✓	×	✓	✓	✓	×	✓	✓	×
[166]	✓	✓	×	✓	✓	✓	✓	×	✓	✓	×	✓	✓	×
[167]	✓	✓	×	✓	✓	✓	×	×	✓	✓	×	✓	✓	×
[168]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[169]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[170]	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AUSS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: ✓ = Yes (Supported), × = No (Not Supported). **UAA:** User Anonymity Attack, **UUA:** User Untraceability Attack, **SMDA:** Stolen Mobile Device Attack, **MA:** Mutual Authentication, **SKAA:** Session Key Agreement Attack, **UIA:** User Impersonation Attack, **RA:** Revocation Attack, **UVA:** User Verification Attack, **SVA:** Sensor Verification Attack, **PIA:** Private Information Attack, **PCA:** Password Change Attack, **FSA:** Forward Secrecy Attack, **SNIA:** Sensor Node Impersonation Attack, **RPA:** Replay Attack.

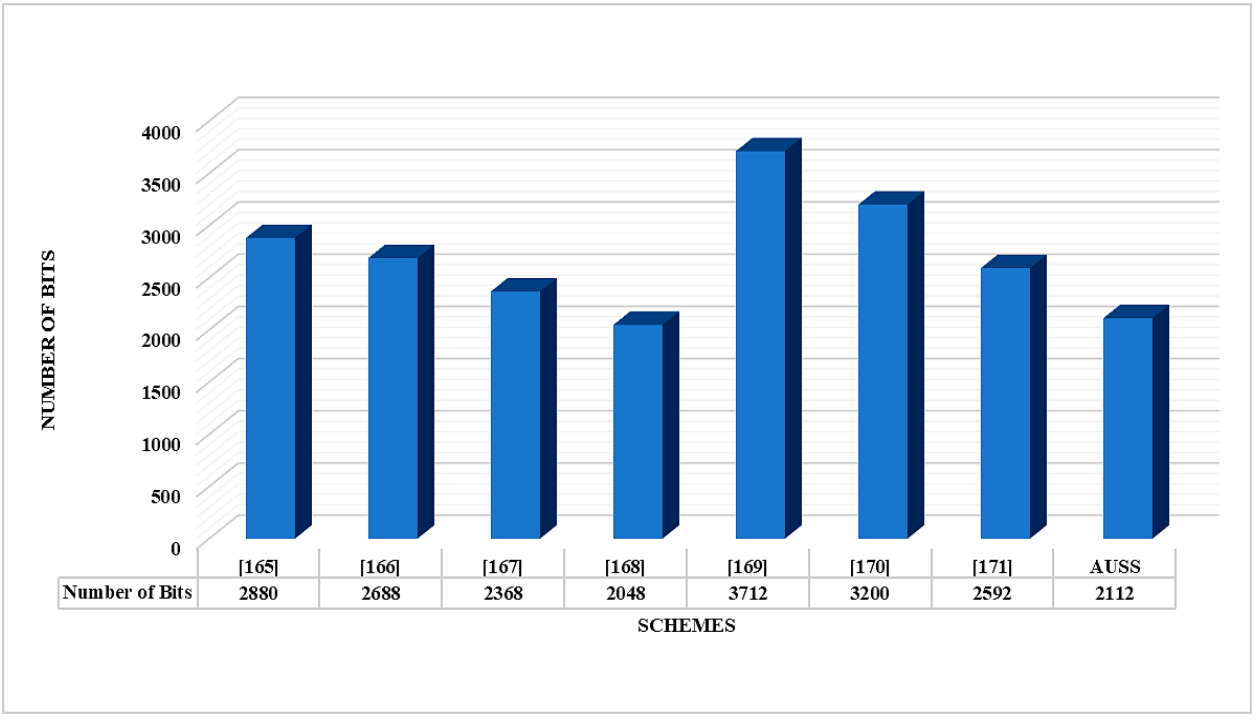


FIGURE 3.5: Communication Cost Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol

The findings reveal that the scheme by Turkanovic et al. [165] achieves reduced computational complexity but remains susceptible to attacks, as highlighted by Farash et

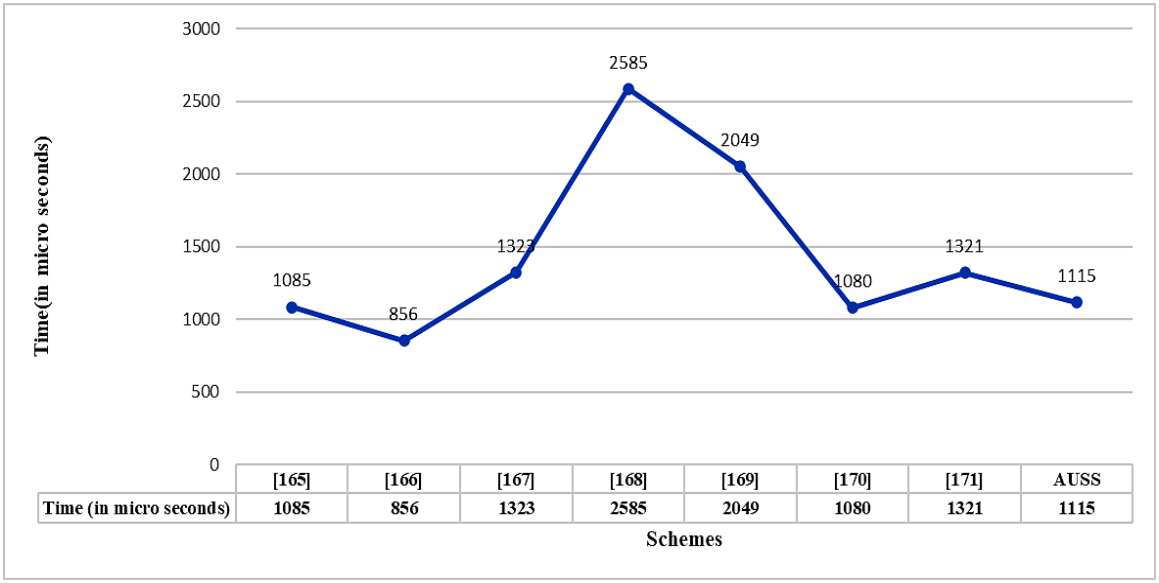


FIGURE 3.6: Computation Cost Comparison of Existing Authentication Schemes and the Proposed AUSS Protocol

al. [113]. On the other hand, our approach demonstrates lower computational overhead compared to the methods proposed by Das et al. [166], Chang et al. [167], Yang et al. [168], and Wu et al. [170]. Furthermore, although the protocol by Banerjee et al. [169] ranks just behind ours, it does not incorporate a revocation mechanism.

We further evaluated the communication overhead during the login and authentication stages. The communication cost of our scheme is 2,112 bits, which exceeds that of Chang et al. [167]; however, this increase is justified by the improved security features our approach provides. Our analysis shows that, although the computational and transmission costs of our scheme are marginally higher than specific other approaches when evaluated on hardware models for IoT environments, its use of XOR and hash functions ensures compatibility with low-cost IoT devices.

Table 3.6 provides a comparison of the functionality and security attributes of the proposed SLAPSH scheme against several existing protocols. Additionally, Figure 3.5 shows a comparison of communication costs between the proposed AUSS scheme and other existing schemes. Meanwhile, Figure 3.6 presents a comparative analysis of computational costs, emphasizing the efficiency of the proposed approach.

To evaluate the communication costs for the login and authentication stages, we applied the methodology described in [172, 173]. We assumed a length of 128 bits for the identity, 32 bits for the timestamp, and 64 bits for the random number. The outputs

generated by the symmetric key encryption, elliptic multiplication operation, and hash function are 256 bits, 360 bits, and 160 bits, respectively.

3.9 Summary

Our work introduces a secure and efficient multifactor authentication protocol designed specifically for IoT environments, particularly those with limited computational and energy resources. Unlike several previous schemes, our approach addresses critical issues such as the absence of session key negotiation and the vulnerability to impersonation when mobile devices are compromised.

The proposed scheme AUSS integrates passwords, biometric features, and mobile device identifiers to authenticate users while maintaining anonymity and preventing traceability. We employ a combination of lightweight cryptographic operations, such as hashing and symmetric encryption, to ensure that the protocol is practical for real-world IoT applications. Formal verification using BAN logic, along with simulations conducted with the AVISPA tool, confirms that the protocol meets essential security requirements, including mutual authentication, freshness, forward secrecy, and resistance to various attacks. Based on our performance analysis, the proposed approach has low costs for both computation and communication compared to other protocols.

In summary, our authentication framework combines strong security with a realistic implementation, making it suitable for a wide range of IoT applications, including smart homes, industrial automation, and healthcare systems.

CHAPTER - 4
Secure Mutual Addressing Authentication Mechanism for Smart IoT Home Network

4.1 Introduction

The growth of Information and Communication Technology (ICT) has significantly transformed our daily lives, leading to the widespread adoption of smart home environments powered by the IoT. These homes feature various interconnected sensors and smart devices shown in Figure 4.1, which enable real-time monitoring, automation, and remote control. This integration enhances user convenience, improves energy efficiency, and enhances the overall quality of life. As smart homes become increasingly common, especially in residential areas, our reliance on these systems has grown significantly.

As Smart Home Internet of Things (SH-IoT) systems become increasingly prevalent, concerns about security and privacy are also growing. These systems consist of interconnected devices and sensors that collect and process sensitive user data. A typical SH-IoT setup comprises a home gateway that connects devices to the Internet, a registration authority for authentication, and user devices such as smartphones for interaction. Communication occurs over open wireless channels, such as Wi-Fi or Zigbee, which can expose the network to cyber threats, including impersonation attacks, message manipulation, and replay attacks [174].

To mitigate these risks, it is essential to establish secure and reliable communication in SH-IoT environments while also protecting user privacy. It is essential to implement strong encryption, reliable authentication protocols, and privacy-protecting measures to safeguard user data. Addressing these challenges is vital for building user trust and

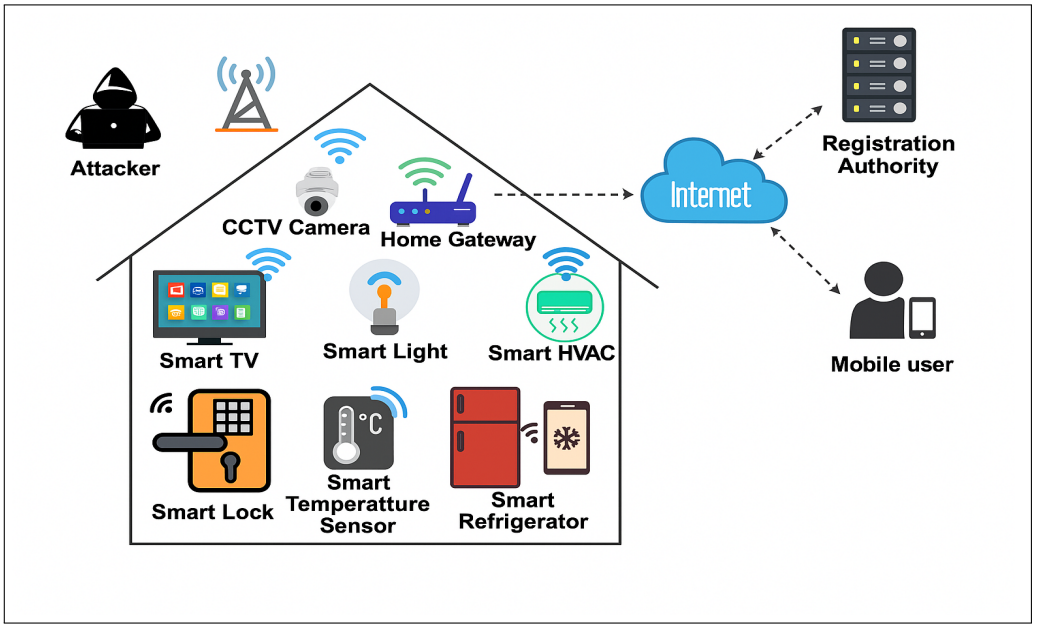


FIGURE 4.1: Appliances for smart home systems

enabling the responsible adoption of SH-IoT technology. This approach will allow users to enjoy greater convenience without compromising their safety.

Numerous cryptographic techniques and security protocols have been developed for IoT environments. However, these often prove inadequate in resource-constrained smart home settings. Limitations in computational power, memory, and energy make traditional, resource-heavy security measures impractical [175]. A significant challenge is the absence of a robust addressing mechanism that enables efficient communication between devices while also supporting authentication and access control. This is important to prevent unauthorized access and data breaches.

To address these challenges, we propose SLAPSH, which features an IPv6-based addressing framework and a lightweight mutual authentication scheme tailored for smart home IoT (SH-IoT) environments. Each device is given a verifiable address-bound identity, enhancing protection against identity spoofing and unauthorised control.

Designed especially to run effectively on IoT devices with limited resources, the SLAPSH system Strong resilience to a range of assaults has shown from both informal study and formal approaches like the ROR model and AVISPA tool, so completely evaluating its security. Furthermore doing simulations on the NS-3 network simulator reveals that the SLAPSH approach adds only minor communication cost, thereby assuring that general system performance is mainly unaltered.

By properly balancing lightweight implementation with strong security measures, the proposed SLAPSH scheme presents a solid and safe option for smart home IoT networks overall. It is a potential method for improving the security of smart home environments since it greatly strengthens the authentication process and addresses weaknesses connected to device addressing.

4.2 Main Contribution of the Proposed Scheme

We propose a novel secure communication framework optimized for Smart Home IoT (SH-IoT) environments, focusing on enhanced addressing techniques and robust mutual authentication. The core contributions of this work can be outlined as follows:

- We introduce a modified IPv6-based addressing scheme, illustrated in Figure 4.2, that assigns unique identifiers to both users and smart devices. This design significantly improves traceability and reinforces secure data exchange within SH-IoT networks.
- We create a dynamic identity mapping system whereby the identity of every device is immediately embedded into its network address. This function helps to verify recipient-side identity, therefore lowering the possibility of illegal device access.
- The proposed SLAPSH protocol achieves lightweight yet secure mutual authentication and session key establishment. It is tailored to the computational limitations of low-power IoT devices, ensuring data integrity and confidentiality with minimal processing overhead.
- The security strength of SLAPSH is validated using formal verification techniques, including the Random Oracle Model (ROR) and the AVISPA security analysis framework, confirming its resilience to various attack vectors.
- We assess the protocol's real-time performance through simulations in the NS-3 network simulator. The results indicate that SLAPSH maintains high throughput and low latency, introducing negligible communication overhead to the system.

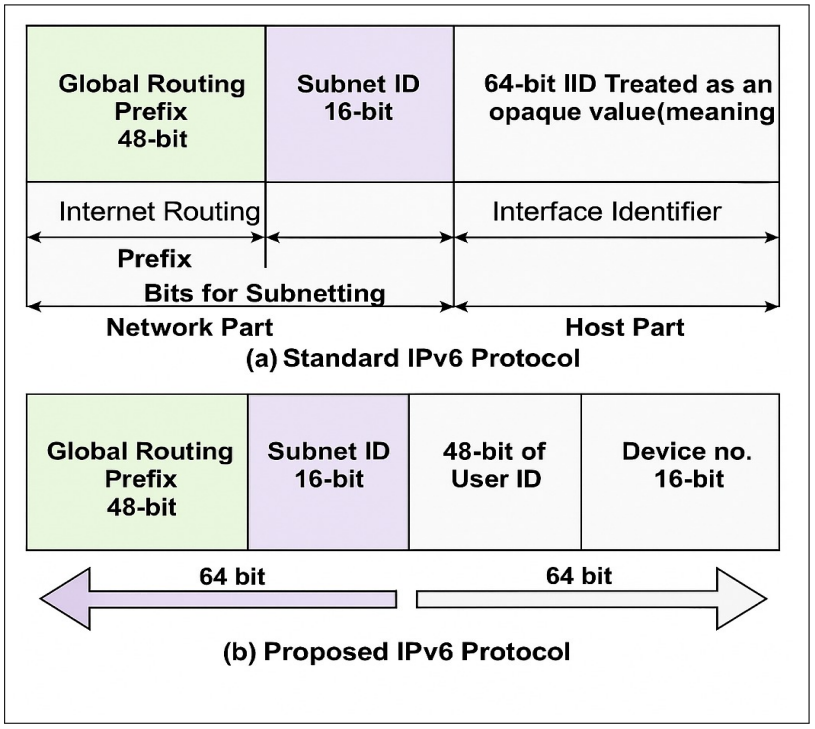


FIGURE 4.2: Modified IPv6 Protocol

4.3 Related Work

In smart home IoT (SH-IoT) systems, designing safe and effective methods to handle addressing and authentication still presents a great difficulty. The job of guaranteeing safe, dependable communication only gets more difficult as more and more devices get linked—and with many of them having limited resources. Years of research have gone toward finding strategies to increase privacy, scalability, and fault tolerance in these systems. As cyberthreats change, their aim is to safeguard user data, streamline identity management, and increase dependability of smart home networks.

4.3.1 Addressing Techniques

Researchers have extensively investigated various IPv6 addressing formats to improve scalability and manageability in large networks. Early approaches focused on aggregatable global unicast addresses, which aimed to simplify routing and protect address space [132]. However, these methods faced significant challenges, including address prefix collisions and limited scalability in dynamic environments.

To enhance device identification, researchers adopted addressing techniques that relied on static and semi-static methods, using EUI-64 formats derived from MAC addresses. While these approaches increased efficiency, their reliance on predictable patterns introduced vulnerabilities, making networks more susceptible to probing and DoS attacks [148, 149]. Consequently, although these addressing formats provided certain advantages, they also introduced new security challenges that needed further attention in the field of IPv6 addressing.

To tackle these issues, techniques such as 6HOP were proposed to enable the dynamic reassignment of IP addresses. While these methods are cost-effective and straightforward, they struggle to manage high device densities and lack scalability [150, 151]. Additionally, moving target defense (MT6D) schemes aim to leverage IPv6's vast address space to hinder attacks; however, they lead to significant synchronization delays and increased computational overhead. Alternative strategies, such as addressing autoconfiguration extensions and randomization techniques, seek to conceal device identities. However, these methods are effective only in localized settings and do not provide sufficient protection in complex multi-device smart home environments.

4.3.2 Security Techniques

In the area of security for IoT and Wireless Sensor Network (WSN) environments, many schemes have focused on lightweight authentication and key agreement protocols. Earlier methods used single-server architectures, which required users to manage multiple credentials for different services. Multi-server protocols that incorporate Elliptic Curve Cryptography (ECC) and biometric authentication have been developed, but they remain vulnerable to impersonation, replay, and MIMA attacks, particularly in resource-constrained environments.

In [179, 180], the authors introduced a novel multi-server architecture for IoT networks to address security vulnerabilities. Their approach uses biometric-based key agreement authentication for secure access to server services and implements an ECC Diffie-Hellman key exchange to protect against replay attacks, though it remains vulnerable to Man-in-the-Middle (MIMA) attacks.

The authors emphasize that using silicon identification as a unique device identifier is important, but this measure alone does not sufficiently reduce the risks of device compromise. This study utilizes Elliptic Curve Cryptography (ECC) to interconnect

multiple IoT devices, drawing on the principles of the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the ECC Diffie-Hellman (ECCDH) protocol [181, 182].

Recent proposals utilising unique device identifiers and fuzzy commitment schemes aim to protect user credentials and biometric data, thereby enhancing anonymity, but face challenges due to hardware limitations and high communication costs. Privacy-preserving protocols that utilise bilinear pairings, chaotic maps, and hash-based verifications effectively secure session keys and prevent traceability, yet still struggle with scalability, IPv6 compatibility, and computational efficiency in smart home environments. These challenges underline the need for a unified solution that provides unique device identification, efficient key management, and mutual authentication with minimal overhead. We address these issues with the SLAPSH scheme, which merges a modified IPv6 addressing model with a secure authentication framework for SH-IoT environments.

4.4 System Model

The proposed architecture for a secure smart home IoT environment includes several key components designed to support trusted communication, identity verification, and data protection. This model aligns with the IEEE 802.11 wireless networking standard and employs IPv6 for dynamic address allocation and device identification.

- **User (U):** This refers to a person or household member who uses personal devices, such as laptops or smartphones, to access and interact with smart home devices. Each user is assigned a unique ID linked to their device.
- **Smart Devices (SD):** These include various home appliances and sensors (e.g., thermostats, lights, alarms) that connect to the smart home network and perform specific tasks. Each device is assigned a unique identity and communicates securely with other devices.
- **Smart Home Server (SHS):** This central authority manages device registration, user authentication, and data packet verification. It is responsible for validating identity claims and maintaining secure sessions between users and devices.
- **Home Gateway (HG):** The HG acts as an interface between internal devices and the wider internet. It stores encrypted identity information, enforces access

control, and coordinates address generation by combining network prefixes and device identifiers to produce complete IPv6 addresses.

- **Central Control Unit (CCU):** This unit manages the registration and authentication process. It ensures that devices and users are only allowed access once their identities are successfully verified against stored credentials. The CCU maintains encrypted identity mappings for security.

Each smart device is assigned a 64-bit Interface Identifier (IID), which consists of a 48-bit encrypted user identity and a 16-bit device-specific value. These identifiers are combined with the network prefix to create unique 128-bit IPv6 addresses. The Home Gateway (HG) ensures that only registered and authenticated devices with matching IID values are allowed to communicate within the smart home network.

4.5 Attacker Model

To evaluate the security robustness of the proposed protocol, we consider an attacker model based on the Dolev-Yao framework [96]. This model assumes that the attacker (**A**) has complete control over the communication channel and can perform both passive and active operations.

- **Eavesdropping and Message Modification:** The attacker can intercept, modify, replay, or drop any message exchanged over the public channel between users and devices.
- **Physical Device Compromise: A** may gain physical access to non-tamper-resistant smart home devices and extract sensitive data stored in their memory.
- **Side-Channel Attacks:** Using methods like power analysis, an attacker may attempt to obtain credentials or secret keys from user devices.
- **Key Disclosure Capabilities:** To launch advanced attacks like session hijacking or impersonation, the attacker might temporarily obtain long-term secrets of legitimate organisations or short-term session keys.

This threat model guarantees that the protocol is assessed against a broad spectrum of attack situations, including desynchronisation attempts, key compromise, message

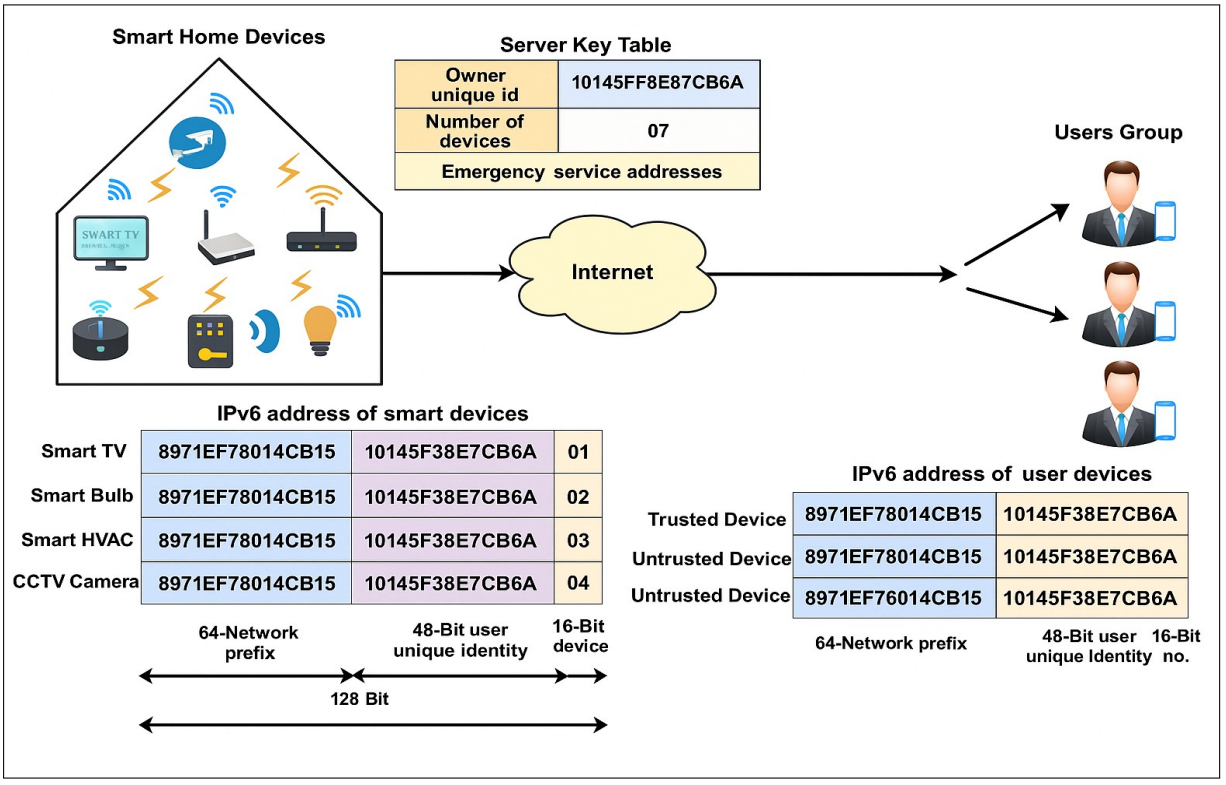


FIGURE 4.3: Illustration of IPv6 Address Format Assignment for Smart Home Devices and User Devices in a Secure IoT Network

forging, and identity spoofing. Specifically intended to identify, resist, and recover from these attacks while preserving operating efficiency are the security mechanisms in the SLAPSH scheme.

4.6 Proposed Scheme

The Secure and Lightweight Authenticated Protocol for Smart Home (SLAPSH) is a novel framework for mutual authentication. It integrates lightweight cryptographic operations with a unique identity-based addressing mechanism derived from the IPv6 protocol.

SLAPSH enhances conventional authentication methods by embedding verifiable identity information directly into the 64-bit Interface Identifier (IID) portion of the IPv6 address. The address format for smart devices is shown in Figure 4.3. This enables seamless identification and authentication without additional messaging overhead or complex key exchanges. The protocol supports end-to-end mutual authentication among mobile users, the home gateway (HG), and smart devices, while maintaining low computational

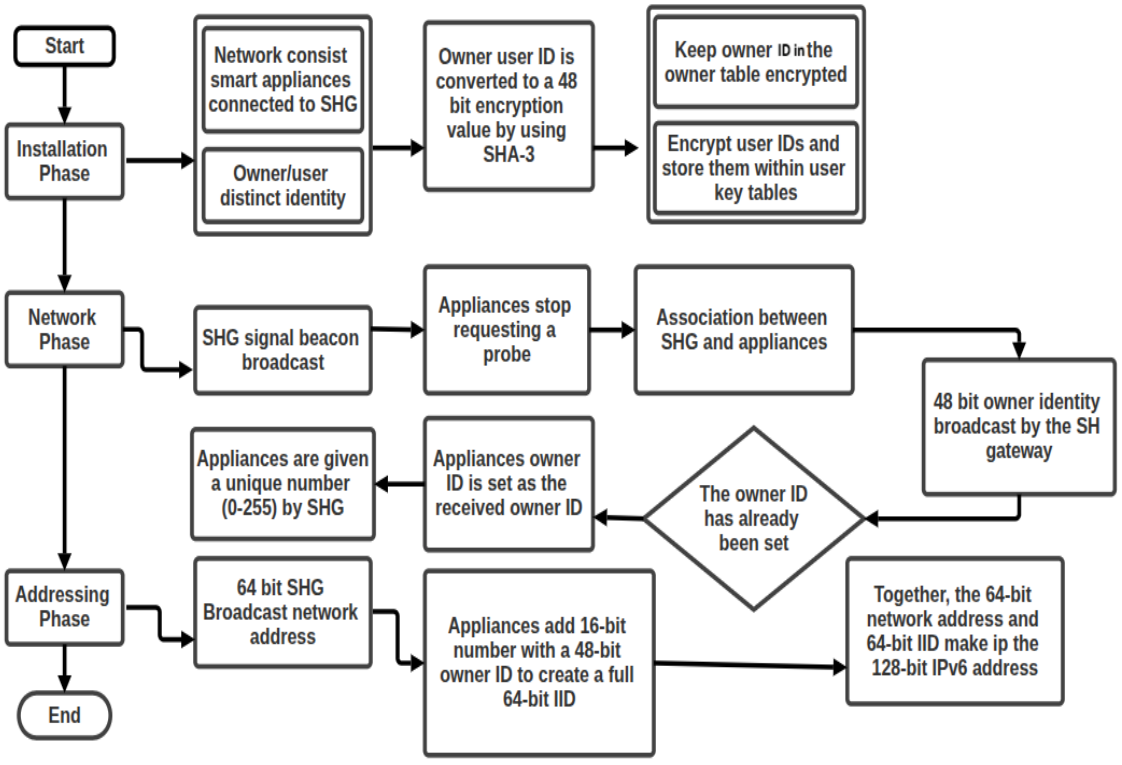


FIGURE 4.4: Stages of a unique addressing scheme

and communication costs. Table 4.1 shows the symbols used and their descriptions. And Figure 4.4 shows the stages of the SLAPSH scheme.

The proposed SLAPSH's scheme has been divided into five operational phases:

1. Installation and Network Configuration Phase
2. Addressing and Identification Phase
3. Registration Phase
4. Authentication and Key Agreement Phase
5. Password Update Phase

TABLE 4.1: Symbols and Their Descriptions

Symbol	Description
IID_u	48-bit modified interface identifier of the user
IID_a	16-bit interface identifier of the smart device
SD	Smart Device
SHS	Smart Home Server
A	Attacker
U_{user}	Mobile User
K_{Reg}	Master key of Registration Authority
K_a	Master secret key of smart device
PW_u	User's password
K_{uhg}	Shared key between user and Home Gateway
K_{hga}	Shared key between smart device and Home Gateway
RID_u	Response Identifier of the user
F_u	Hashed pseudonym of the user
F_a	Hashed pseudonym of the smart device
Sk	Session key between user and smart device
$h(.)$	One-way hash function
$\|\|$	Concatenation operation
T	Timestamp

4.6.1 Phase 1: Installation and Network Configuration

In this phase, the Registration Authority (RA) sets up the system by configuring IPv6 addresses with embedded identity information. Each user is assigned a 48-bit identifier (UID) derived from a unique identity, such as a national ID or SSN, using the SHA-3 algorithm. Smart devices are assigned a 16-bit device identifier (DID). The Interface Identifier (IID) portion of an IPv6 address is created by combining the user and device identifiers: $IID = UID||DID$. The complete IPv6 address is then formed by appending this 64-bit IID to a 64-bit network prefix: $IPv6 = Prefix_{64}||IID$. This address provides both identification and authentication capability. The UID is securely stored with the user, while the Home Gateway (HG) maintains a record of the mapping between UID and associated devices. This configuration ensures that each device in the smart home network has a unique, verifiable address linked to its owner.

4.6.2 Phase 2: Addressing and Identification

During this phase, each smart device is assigned an IPv6 address formed during the installation phase. The address structure combines the 64-bit network prefix with the 64-bit Interface Identifier (IID), which itself consists of a 48-bit user identifier and a 16-bit device identifier. This embedded addressing mechanism enables the unique identification of every device, allowing for streamlined authentication without requiring additional communication overhead. The Home Gateway uses this address structure to establish a trusted mapping between mobile users and their corresponding smart devices, enabling secure device identification during authentication and session setup processes.

4.6.3 Phase 3: Registration Phase

This phase facilitates the secure onboarding of both users and smart devices to the Home Gateway (HG) and Registration Authority (RA), ensuring their legitimacy before participation in the SLAPSH protocol.

TABLE 4.2: Mobile User Registration Phase in the Proposed Authentication Protocol

Mobile User Device	**Registration Authority (RA)**
Selects ID_u, PW_u, and generates nonce N_u Computes $F_u = h(ID_u \| N_u)$ Sends $\rightarrow F_u$	Computes $K_{uhg} = h(F_u \| K_{reg} \| N_{Reg})$ Computes $RID_u = h(F_u \| K_{hga})$ Stores $\{F_u, RID_u, K_{hga}\}$ in HG database
Computes $HPW_u = h(PW_u \| N_u)$ Computes: $Y_1 = N_u \oplus h(ID_u \| PW_u)$ $Y_2 = h(ID_u \| PW_u \| N_u \| HPW_u)$ $Y_3 = RID_u \oplus h(N_u \| HPW_u)$ $Y_4 = K_{uhg} \oplus h(RID_u \| HPW_u)$ Stores Y_1, Y_2, Y_3, Y_4, F_u	$\leftarrow \{K_{hga}, RID_u\}$

User Registration:- The mobile user initiates registration by selecting an identity ID_u and password PW_u, and generating a random nonce N_u. The smart device computes the hashed password $HPW_u = h(ID_u \| PW_u)$, and derives intermediate values: $Y_1 = N_u \oplus h(ID_u \| PW_u)$, $Y_2 = h(ID_u \| PW_u \| N_u \| HPW_u)$, $Y_3 = h(PW_u \| ID_u) \oplus K_{uhg}$, and $Y_4 = h(ID_u \| PW_u \| K_{uhg})$. A user fingerprint $F_u = h(ID_u \| N_u)$ is also created. Y_1, Y_2, Y_3, Y_4, F_u—are securely stored in the user's smart device for subsequent authentication purposes. The interaction steps are detailed in Table 4.2.

TABLE 4.3: Smart Device Registration Phase in the Proposed Authentication Protocol

Smart Device	**Registration Authority (RA)**
Generates N_a and computes $F_a = h(ID_a \| N_a)$ Sends $\{F_a, N_a\} \rightarrow$	Generates nonce N_{Reg} Computes $K_{hga} = h(F_a \| K_{hga1} \| N_a)$ Stores $\{F_a, K_{hga}, N_a\}$ in HG's database
Success: Computes: $X_1 = N_a \oplus h(ID_a \| K_a)$ $X_2 = K_{hga} \oplus h(N_a \| K_a)$ Stores $\{X_1, X_2, F_a\}$	$\leftarrow \{K_{hga}\}$

Smart Device Registration:- Similarly, the smart device begins its registration by generating a nonce N_a and computing its identity fingerprint $F_a = h(ID_a \| N_a)$. This fingerprint is sent to the Registration Authority, which generates a shared key $K_{hga} = h(F_a \| K_{hga} \| N_a)$ and stores it with F_a and N_a in the Home Gateway database. The device then computes $X_1 = K_a \oplus h(ID_a \| N_a)$ and $X_2 = h(ID_a \| K_a \| N_a)$, and securely stores X_1, X_2, F_a. The detailed registration interaction is shown in Table 4.3.

4.6.4 Phase 4: Authentication, Login, and Key Agreement Phase

This phase ensures mutual authentication and the establishment of a session key among the mobile user, HG, and SD. It begins with the user logging in, followed by nonce-based challenges and hashed verifications to defend against impersonation, replay attacks, and man-in-the-middle attacks.

The protocol starts when the user inputs their identity ID_u and password PW_u into their smart device. The smart device retrieves the stored values and computes the hashed password $HPW_u = h(ID_u \| PW_u)$, and then derives the random value $N_u = Y_1 \oplus h(ID_u \| PW_u)$. A verification token is generated as $V_1 = h(ID_u \| PW_u \| N_u \| HPW_u)$. For further authentication, it calculates $V_2 = h(F_u \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_m)$, and computes $W_u = h(F_u \| RID_u \|$
$N_m \| K_{uhg})$. The user then sends the message $\{F_u, RID_u, V_1, V_2, W_u\}$ to the HG.

Upon receiving the message, the HG verifies the credentials by checking whether $V_1 \stackrel{?}{=} h(ID_u \| PW_u \| N_u \| HPW_u)$, $V_2 \stackrel{?}{=} h(F_u \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_m)$, and $W_u \stackrel{?}{=} h(F_u \| RID_u$
$\| N_m \| K_{uhg})$. If all values are validated successfully, the HG generates a fresh nonce N_{hg}, and computes the encrypted nonce challenge $V_3 = h(F_u \| N_m \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_{hg})$,

TABLE 4.4: Login and Key Establishment in the Proposed Authentication Protocol

Mobile User (U)	**Home Gateway (HG)**	**Smart Device (SD)**
Step 1: Inputs ID_u, PW_u Computes: $N_u = Y_1 \oplus h(ID_u \| PW_u)$ $HPW_u = h(ID_u \| PW_u)$ $Y_3 = h(ID_u \| PW_u \| N_u \| HPW_u)$ Checks: $Y_2 = Y_3$ Generates N_m Computes: $RID_u = Y_4 \oplus h(N_u \| HPW_u)$ $K_{uhg} = Y_3 \oplus h(ID_u \| PW_u)$ $V_1 = h(F_u \| RID_u \| N_m \| F_u \| K_{uhg})$ $S_1 = h(ID_u \| N_m)$ $W_u = h(K_{uhg} \| N_m)$ Sends $\{F_u, V_1, S_1, W_u\}$ to HG	**Step 2:** Receives $\{F_u, V_1, S_1, W_u\}$ Restores RID_u and K_{uhg} Computes: $V_1^* = h(F_u \| RID_u \| N_m \| F_u \| K_{uhg})$ Checks: $V_1 = V_1^*$ Generates N_{hg} $V_2 = h(N_m \| N_{hg})$ $V_3 = h(F_u \| K_{hga} \| ID_a \| N_a)$ $hash = h(ID_a \| N_a)$ $S_1^* = h(ID_u \| N_m)$ $S_2 = h(hash(ID_u \| N_m))$ $W_u^* = h(K_{uhg} \| N_m)$ Checks: $W_u = W_u^*$ Sends $\{F_u, V_3, S_2, W_u\}$ to SD	**Step 3:** Computes: $N_a = X_1 \oplus h(ID_a \| K_a)$ $K_{hga} = X_2 \oplus h(N_a \| K_a)$ $V_3 = h(F_a \| K_{hga} \| N_a)$ Checks: W_a^* Computes: $hash(ID_a \| N_m)$ $S_2 = h(K_{hga} \| N_a)$ $SK = h(hash(ID_u \| N_m) \|$ $hash(ID_{hg} \| N_{hg}) \| h(ID_a \| N_a))$ $W_a = h(F_a \| F_u \| V_2 \|$ $hash(ID_a \| N_a) \| K_{hga})$ Sends $\{V_4, W_a\}$ to HG
Step 5: Computes $F_u' = hash(F_u \| N_m)$ $V_5 = hash(ID_{hg} \| N_m) \oplus hash(ID_a \| N_a) \oplus F_u'$ $V_6 = hash(F_u \| N_m)$ Sends $\{V_5, V_6\}$ to HG	**Step 4:** Computes $hash(ID_a \| N_a)$ $V_4 = hash(F_a \| K_{hga} \| N_a)$ Checks: $V_4 = V_4^*$ Computes: $W_4 = hash(F_a \| F_u \| V_2 \|$ $hash(ID_a \| N_a) \| K_{hga})$ $F_u' = hash(F_u \| N_m)$ $RID_u^* = hash(F_u' \| K_{uhg})$ $Y_1^* = N_u \oplus h(ID_u \| PW_u)$ $Y_2^* = RID_u \oplus h(N_u \| HPW_u)$ $Y_3^* = hash(ID_u \| PW_u \| N_u \| HPW_u)$ $Y_4^* = hash(ID_u \| PW_u \| K_{uhg})$ Returns $\{Y_3, Y_4, F_u\}$ to U	
	Step 6: Computes $V_6^* = hash(S_6 \| F_u')$ Checks: $V_6 = V_6^*$ If correct, deletes $\{F_u, RID_u\}$ from DB	

along with the confirmation hash $W_{uhg} = h(F_u \| RID_u \| N_m \| N_{hg} \| K_{uhg})$. These values $\{F_u, RID_u, V_3, W_{uhg}\}$ are then forwarded to the smart device.

The smart device, upon receiving the message, validates the information and generates its authentication components. It computes the following values: $V_4 = h(F_a \| K_{hga} \| N_a) \oplus h(ID_a \| K_a)$ and $W_a = h(F_u \| F_a \| V_2 \| h(ID_a \| N_a) \| K_{hga})$. These values, $\{F_a, V_4, W_a\}$, are then returned to the Home Gateway for verification. Upon receiving the response from the smart device, the Home Gateway computes $V_5 = h(F_u \| F_a \| V_2 \| N_{hg} \| K_{hga})$ and sends it to the smart device. The smart device then validates this final value and replies with $V_6 = h(F_u \| F_a \| N_a \| K_{hga})$.

After the user, the HG, and the smart device complete all validations, a session key is established. This key is derived using a combination of the identities of the involved parties and their nonces.

$$Sk = h(h(ID_u \| N_m) \| h(ID_{hg} \| N_{hg}) \| h(ID_a \| N_a)),$$

Which ensures mutual secrecy and forward security for future communications. Table 4.4 presents the detailed interaction.

4.6.5 Phase 5: Password Update Phase

To ensure long-term security, SLAPSH enables users to update their passwords periodically without requiring re-registration. This process is securely carried out between the user and the Home Gateway using the values stored on the smart device, as detailed in Table 4.5.

The user initiates the password update by providing their old password PW_u^{old} and the new password PW_u^{new}. The smart device initiates the password update process by retrieving the stored values Y_1, Y_2, Y_3, Y_4, F_u. It first recomputes the hashed version of the old password as $HPW_u^{old} = h(ID_u \| PW_u^{old})$, and uses it to derive the original nonce $N_u = Y_1 \oplus h(ID_u \| PW_u^{old})$. Next, it generates the new hashed password $HPW_u^{new} = h(ID_u \| PW_u^{new})$. Using this, the smart device updates the verification tokens.

It computes the new $Y_1' = N_u \oplus h(ID_u \| PW_u^{new})$, and the new integrity value $Y_2' = h(ID_u \| PW_u^{new} \| N_u \| HPW_u^{new})$. It then generates the updated obfuscated pseudonym $Y_3' = RID_u \oplus h(N_u \| HPW_u^{new})$, and refreshes the security token $Y_4' = K_{uhg} \oplus h(RID_u \| HPW_u^{new})$. These

updated values Y_1', Y_2', Y_3', Y_4' are securely stored back into the smart device, completing the password update without requiring communication with the Home Gateway.

The updated values Y_1', Y_2', Y_3', Y_4' replace the old ones in the smart device. Since the protocol does not require any message exchange with the HG during this phase, it ensures secure, offline password updates.

TABLE 4.5: Password Update Phase in the Proposed Authentication Protocol

Mobile User U	**HG**
User inputs old and new passwords: PW_u^{old}, PW_u^{new}	Smart device retrieves Y_1, Y_2, Y_3, Y_4, F_u Computes: $HPW_u^{old} = h(ID_u \| PW_u^{old})$ $N_u = Y_1 \oplus h(ID_u \| PW_u^{old})$ $HPW_u^{new} = h(ID_u \| PW_u^{new})$
	Updates values: $Y_1' = N_u \oplus h(ID_u \| PW_u^{new})$ $Y_2' = h(ID_u \| PW_u^{new} \| N_u \| HPW_u^{new})$ $Y_3' = RID_u \oplus h(N_u \| HPW_u^{new})$ $Y_4' = K_{uhg} \oplus h(RID_u \| HPW_u^{new})$ Stores updated values Y_1', Y_2', Y_3', Y_4'

4.7 Informal Security Analysis

The proposed SLAPSH scheme offers robust security against various common threats encountered by IoT-based smart home systems. This section details each of these threats and explains how SLAPSH effectively mitigates them through the use of cryptographic methods, ensures the uniqueness of sessions, and verifies the identities of all entities involved.

4.7.1 Mobile User Impersonation Attack

An attacker may try to impersonate a legitimate mobile user by extracting values from the user's smart device, such as Y_1, Y_2, Y_3, Y_4, and F_u. However, generating valid authentication tokens, such as $V_1 = h(F_u \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_m)$, is computationally infeasible without the password PW_u, identity ID_u, and session nonce N_m. These crucial components are not stored in plaintext on the smart device or transmitted over the network, ensuring that even with access to the device data, an attacker cannot forge valid credentials or impersonate a legitimate user.

4.7.2 Home Gateway Impersonation Attack

In an attempt to impersonate the Home Gateway, an attacker might intercept protocol messages such as F_u, V_3, S_2, and W_{uhg}. However, these intercepted values alone are insufficient to fabricate valid authentication responses. The attacker would require access to several critical components, including the Home Gateway's fresh session nonce N_{hg}, the shared secrets K_{uhg} (between the user and the HG) and K_{hga} (between the HG and the smart device), and the static identities ID_u, ID_{hg}, and ID_a. Without these elements, the attacker cannot compute values like $V_5 = h(F_u \| F_a \| V_2 \| N_{hg} \| K_{hga})$ or the necessary response hashes. Consequently, any forged responses will fail verification and be rejected by both the smart device and the user, preventing the impersonation attack.

4.7.3 Smart Device Impersonation Attack

To impersonate a legitimate smart device, an attacker must generate valid authentication parameters used by the protocol to verify the device's identity. Specifically, the attacker needs to compute $V_4 = h(F_a \| K_{hga} \| N_a) \oplus h(ID_a \| K_a)$ and $W_a = h(F_u \| F_a \| V_2 \| h(ID_a \| N_a) \| K_{hga})$. However, performing these calculations requires access to the device's secret key K_a, its identity ID_a, and the random nonce N_a. None of this information is transmitted over the network or can be derived from intercepted messages. These values are stored securely within the device and are unknown to the attacker. As a result, any attempt to spoof a smart device by fabricating these authentication tokens will inevitably fail, thus preserving the integrity of device authentication in the protocol.

4.7.4 Session Key Disclosure Attack

The proposed SLAPSH scheme derives the session key using secure one-way hash functions applied over a combination of fresh nonces and static entity identifiers. The session key is computed as $Sk = h(h(ID_u \| N_m) \| h(ID_{hg} \| N_{hg}) \| h(ID_a \| N_a))$. Even if an attacker intercepts communications the given instruction guarantees that the key is secure and tamper-proof. Using new nonces N_m, N_{hg}, and N_a for each session ensures that session keys are distinct and ensures perfect forward secrecy, safeguarding previous keys even if long-term credentials are compromised.

4.7.5 Replay and MITM Attacks

The SLAPSH scheme offer robust protection against replay and man-in-the-middle attacks by using session-specific nonces and cryptographic hash. Each session begins with unique nonces N_m, N_{hg}, and N_a, which are incorporated into authentication tokens and verification hashes. These values are linked to the session's context hence any attempt to reuse or alter intercepted messages leads to validation failures due to hash mismatches. As the nonces are unique and never sent in plaintext, attackers cannot forge responses or intercept communications.

4.7.6 Anonymity and Untraceability

The proposed SLAPSH scheme ensures user anonymity and session untraceability by substituting permanent identifiers with dynamic pseudonyms. Rather than transmitting a user's actual identity ID_u, the protocol uses a hashed pseudonym $F_u = h(ID_u \| N_u)$, where N_u is a user-generated random value that changes per session. The pseudonym changes with each authentication attempt, preventing observers from linking different pseudonyms to the same user. Secure one-way hash functions ensure that intercepted pseudonyms cannot be reversed to reveal the original identity.

4.7.7 Offline Guessing Attack

The proposed SLAPSH scheme effectively mitigates offline guessing attacks by tightly coupling the user's identity and password in its verification mechanism. To validate any guessed credentials, an attacker would need to simultaneously guess both the identity ID_u and the password PW_u correctly to produce a valid output for expressions like $Y_1 = N_u \oplus h(ID_u \| PW_u)$. Since the value of N_u is concealed via an XOR operation with a cryptographic hash and never transmitted in plaintext, it becomes computationally infeasible to isolate and test individual components. Moreover, the use of SHA-3, a robust and collision-resistant hash function, further strengthens the scheme against dictionary and brute-force attacks, rendering such offline guessing attempts practically unsuccessful.

4.7.8 Stolen Device Attack

In scenarios where a smart device or a user's smart device is physically stolen, an attacker may gain access to locally stored values, such as X_1, X_2, and $F_a = h(ID_a \| N_a)$. The SLAPSH protocol remains secure even in various situations because the values alone are insufficient for impersonating the device or user. Critical authentication computations, such as $V_4 = h(F_a \| K_{hga} \| N_a) \oplus h(ID_a \| K_a)$ and $W_a = h(F_u \| F_a \| V_2 \| h(ID_a \| N_a) \| K_{hga})$, rely on specific session-related nonces, namely N_m, N_{hg}, and N_a as well as shared secrets such as K_a and K_{hga}. Importantly, these shared secrets are never stored on the device in plaintext. Additionally, these temporary values are essential for computing or verifying session keys, which are known only to and refreshed by the Home Gateway. Therefore, even with full access to the device's internal data, an attacker cannot successfully participate in the authentication protocol or derive valid session keys, effectively preventing a system compromise due to physical theft.

4.7.9 Mutual Authentication

The proposed SLAPSH scheme ensures robust mutual authentication between the mobile user, the Home Gateway (HG), and the smart device (SD) through a series of challenge-response interactions that verify the authenticity of all parties involved. The user authenticates the HG and the smart device by validating received tokens such as $V_5 = h(F_u \| F_a \| V_2 \| N_{hg} \| K_{hga})$ and $V_6 = h(F_u \| F_a \| N_a \| K_{hga})$. Simultaneously, the HG authenticates the user by checking values like $V_1 = h(F_u \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_m)$ and $W_u = h(F_u \| RID_u \| N_m \| K_{uhg})$, and verifies the device through parameters $V_4 = h(F_a \| K_{hga} \| N_a) \oplus h(ID_a \| K_a)$ and $W_a = h(F_u \| F_a \| V_2 \| h(ID_a \| N_a) \| K_{hga})$. The smart device, in turn, confirms the consistency of the session by validating V_5 and computing its hash-based response V_6. This bidirectional verification process ensures that no session key is established unless all three entities have authenticated one another, thereby preventing impersonation and unauthorized participation in the communication session.

4.7.10 Perfect Forward Secrecy

The proposed SLAPSH scheme achieves perfect forward secrecy by ensuring that each session key is derived from fresh, session-specific nonces that are never reused or stored.

Even if an adversary obtains long-term secrets like the user-gateway key K_{uhg} or the device key K_a, past communication sessions remain secure. The session key is obtained as $Sk = h(h(ID_u \| N_m) \| h(ID_{hg} \| N_{hg}) \| h(ID_a \| N_a))$, where N_m, N_{hg}, and N_a are ephemeral nonces generated independently for each session by the user, Home Gateway, and smart device, respectively. Since these nonces are never transmitted in raw form and are securely deleted after use, an attacker cannot reconstruct prior session keys even with full access to static secrets. This design guarantees that every session remains cryptographically isolated and secure.

4.7.11 Desynchronization Attack

The proposed SLAPSH scheme is inherently resistant to desynchronization attacks by deferring state updates until the successful completion of the authentication process. In particular, if the final confirmation message such as $V_6 = h(F_u \| F_a \| N_a \| K_{hga})$ is lost, corrupted, or delayed due to network issues or malicious interference, the Home Gateway refrains from updating its internal state or session variables. This approach ensures that any incomplete authentication attempt does not overwrite previously valid session data. As a result, both the Home Gateway and the smart device remain synchronised, preventing identity mismatches or session key inconsistencies. This mechanism upholds the reliability and consistency of ongoing and future sessions, even in the presence of message loss or tampering.

4.8 Network Simulation Using NS-3

In order to assess the effectiveness of the proposed SLAPSH scheme in a realistic network environment, we conducted a simulation through the use of NS-3 (Network Simulator 3), a discrete-event simulator that is frequently employed for academic analysis and research of communication protocols.

The simulation aims to measure two key network performance metrics: network throughput and end-to-end delay (E2E), under both normal and comparative conditions. The IPv6 protocol and Routing Information Protocol Next Generation (RIPng) were used in the topology design shown in Figure 4.5.

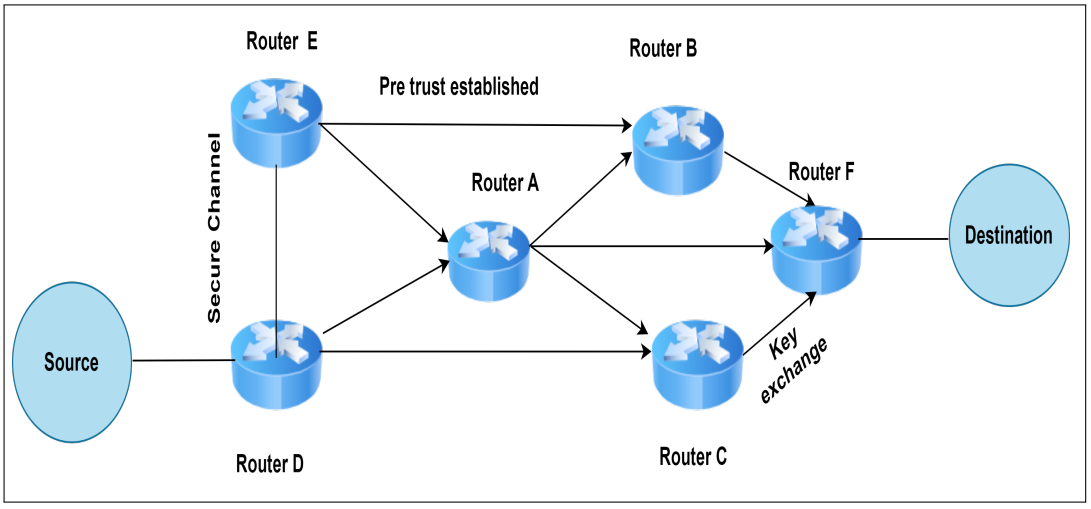


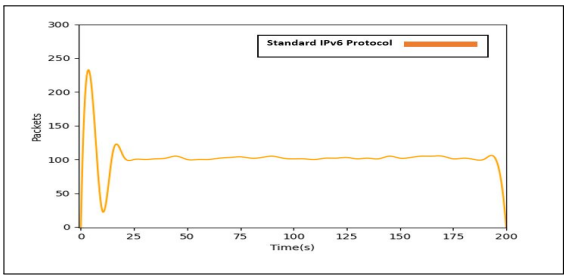
FIGURE 4.5: Network topology using NS-3

4.8.1 Simulation Setup

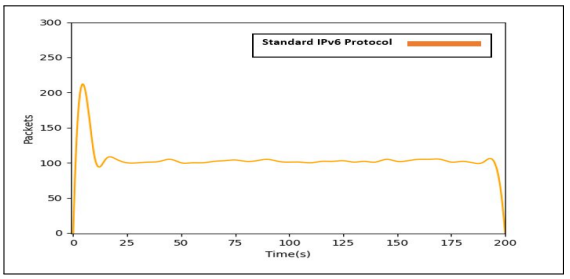
The simulated network consists of six routers configured with RIPng, arranged in a mesh topology. These routers connect the source and destination nodes through intermediate routers labelled A to F. The simulation uses IPv6 addressing for all nodes, and packet flow is monitored using ICMPv6 (ping) to observe transmission behaviors. The simulation parameters are detailed in Table 4.6.

TABLE 4.6: NS-3 Simulation Parameters

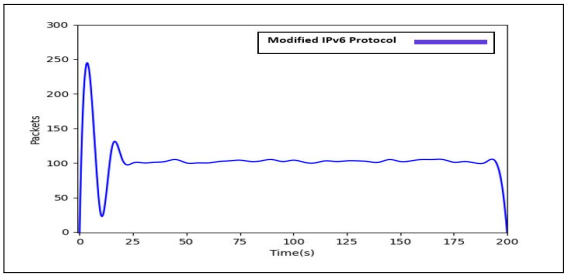
Parameter	Value
Number of RIPng Routers	6
Addressing Protocol	IPv6
Packet Size	1024 bytes
Total Packets Transmitted	100
Channel Data Rate	5000 kbps
Propagation Delay	2 milliseconds
Channel Type	CSMA (Carrier Sense Multiple Access)
Packet Interval	1.0 second
Simulation Duration	200 seconds
Number of Iterations	10 runs



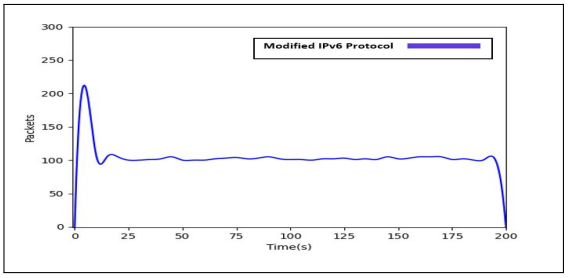
Throughput of the source Node using standard IPv6 Protocol



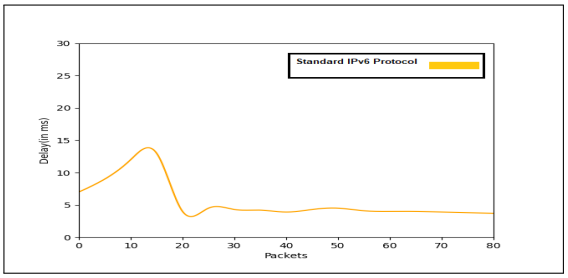
Throughput of the destination Node using Modified IPv6 Protocol



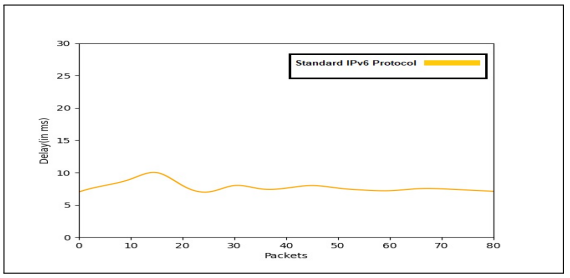
Throughput of the source Node using standard IPv6 Protocol



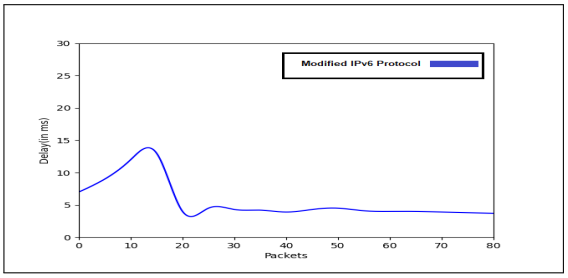
Throughput of the destination Node using Modified IPv6 Protocol



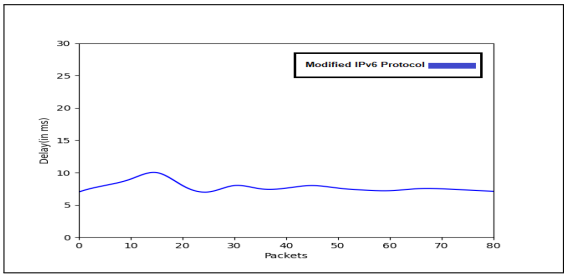
Delay to packets plot at source using Standard IPv6 Protocol



Delay to packets plot at destination using Modified IPv6 Protocol



Delay to packets plot at source using Modified IPv6 Protocol



Delay to packets plot at destination using Standard IPv6 Protocol

FIGURE 4.6: Network simulation results

4.8.2 Throughput Analysis

Throughput is defined as the rate at which packets are successfully delivered from the source to the destination. In our simulation, data transmission begins after an initial setup delay of 3 seconds. During the 200-second interval, the packet delivery rate steadily increases, reaching its peak at the end of this period. The SLAPSH scheme demonstrates higher and more stable throughput compared to the standard IPv6 protocol, all without any additional transmission overhead.

4.8.3 End-to-End Delay (E2E)

End-to-End Delay refers to the average time it takes for a data packet to travel from the source node to the destination node. In this simulation, a temporary network disruption occurs when Router A is disabled, simulating a fault scenario that requires the protocol to reroute data through alternative paths. Once Router **A** resumes operation, the protocol dynamically recovers and maintains stable latency levels. The results indicate that SLAPSH scheme introduces only negligible additional delay compared to the baseline IPv6, even during routing changes.

The NS-3 simulation results, presented in Figure 4.6, confirm that SLAPSH scheme not only improves authentication and security but also sustains high network performance. It ensures consistent throughput and low latency making it a practical and secure solution for real-world smart home IoT environments.

4.9 Formal Security Analysis

The proposed SLAPSH scheme is further evaluated through formal security verification using two standard methodologies: the ROR model for session key security and the AVISPA tool for automated protocol validation.

4.9.1 Formal Security Analysis Using ROR Model

The ROR model is employed to evaluate the SLAPSH scheme. The ROR model is a standardized framework that is employed to demonstrate the indistinguishability of

session keys in adversarial environments. In this model, an attacker $\mathscr{A}$ can interact with the protocol participants through a series of oracle queries. The goal is to determine whether $\mathscr{A}$ can distinguish a real session key from a randomly generated one with a non-negligible advantage.

The SLAPSH scheme involves three principal entities: the mobile user $\mathscr{P}_U^{tp1}$, the Home Gateway $\mathscr{P}_{HG}^{tp2}$, and the smart device $\mathscr{P}_{SD}^{tp3}$. Each party engages in one of three corresponding phases, and the adversary can interact with them over an open channel using the following queries:

- **Execute**($\mathscr{P}_U^{tp1}, \mathscr{P}_{HG}^{tp2}, \mathscr{P}_{SD}^{tp3}$): This passive query returns the transcript of messages exchanged during an honest session.
- **Send**($\mathscr{P}_U^{tp1}, V$): Sends a message V to a party and receives the resulting response, enabling impersonation and reflection attempts.
- **Reveal**($\mathscr{P}_U^{tp1}$): Returns the session key from a completed session, testing the forward secrecy of the protocol.
- **CorruptMD**($\mathscr{P}_U^{tp1}$): Models the compromise of a mobile device or smart device, exposing values such as Y_1, Y_2, Y_3, Y_4, and F_u.
- **Test**($\mathscr{P}_U^{tp1}$): The crucial challenge query; returns either the actual session key or a random value depending on a hidden coin flip $f_c \in \{0,1\}$. The adversary's objective is to guess f_c.

To demonstrate the protocol's resilience, we define a sequence of Game GM_0 through GM_3. Let $\Pr[Succ_{\mathscr{A}}, GM_i]$ represent the success probability of the attacks in Game i.

In the initial Game GM_0, the attacks interact with all parties using all permitted queries, including Execute, Send, and Reveal. The advantage of distinguishing the key is defined as:

$$Adv_{\mathscr{A}}^{SLAPSH} = 2 \cdot \Pr[Succ_{\mathscr{A}}, GM_0] - 1.$$

In Game GM_1, the adversary issues the Test query during a session. This query appears identical to the one in Game 0 from the adversary $\mathscr{A}$'s perspective, resulting in the same probability of success in both games.

$$\Pr[Succ_{\mathscr{A}}, GM_1] = \Pr[Succ_{\mathscr{A}}, GM_0].$$

In Game GM_2, we treat the hash oracle as a random function to simulate its behavior. The adversary may try to reverse-engineer nonces or create valid responses to messages. The limitations on the adversary's success probability are defined by:

$$\Pr[Succ_{\mathscr{A}}, GM_2] - \Pr[Succ_{\mathscr{A}}, GM_1] \leq \frac{q_{send}^h}{|h(\cdot)|}.$$

Game GM_3 simulates the CorruptMD query, granting the adversary access to compromised values such as $Y_1 = N_u \oplus h(ID_u \| PW_u)$ and $Y_2 = h(ID_u \| PW_u \| N_u \| HPW_u)$. Despite this, due to the infeasibility of reversing secure hashes and the entropy of the nonce N_u, the adversary's best guess probability reduces to:

$$\Pr[Succ_{\mathscr{A}}, GM_3] = \frac{1}{2}.$$

The final advantage of the adversary is bounded using the triangle inequality:

$$Adv_{\mathscr{A}}^{SLAPSH} \leq \frac{q_h^2}{|h(\cdot)|} + 2 \cdot (C \cdot q_{send}^h),$$

where q_h is the number of hash queries and C is a small constant tied to complexity assumptions.

Based on our analysis, we conclude that the SLAPSH protocol achieves semantic security within the ROR model. Even with extensive adversarial capabilities, such as observing session transcripts, compromising smart device, and accessing hash oracles, there is no practical advantage in identifying a genuine session key from a random value. The protocol's layered approach, which includes the use of nonces, secure hash functions, and identity obfuscation, ensures that the SLAPSH scheme effectively resists offline dictionary, replay attacks, and key-compromise impersonation threats.

4.9.2 Formal Security Analysis Using AVISPA Tool

To thoroughly evaluate the security features of the SLAPSH scheme, we utilized the AVISPA tool. AVISPA operates on the Dolev-Yao intruder model and is a widely recognized formal verification framework. It assesses protocols for vulnerabilities related to various attacks, including replay attacks, man-in-the-middle (MITM) attacks, message forgery, and authentication failures [98].

The SLAPSH protocol was specified using HLPSL, which captures the roles, transitions, and cryptographic operations of the participating entities. The model defines three primary roles: the User, the Home Gateway (HG), and the Smart Device (SD), each with its associated knowledge and communication behaviors. A dedicated session role orchestrates the interactions among these entities. The protocol model explicitly includes nonce generation (e.g., N_u, N_m, N_{hg}, N_a), hash-based computations such as $V_1 = h(F_u \| RID_u \| K_{uhg}) \oplus h(K_{uhg} \| N_m)$, and secure exchanges over a public channel.

The HLPSL specification is modular, and the roles are defined independently. For example, the logic of the User role is outlined in Figure 4.7, while the Session and Environment behavior is illustrated in Figure 4.8. These HLPSL components together describe the complete behavior and interactions of the SLAPSH protocol within the AVISPA framework.

The security goals defined in the AVISPA goal section include:

- **Secrecy of the session key** *Sk*, shared among the user, HG, and SD.
- **Mutual authentication** between all pairs of communicating entities.
- **Freshness of sessions**, ensuring resilience against replay attacks by incorporating nonces into protocol logic.
- **MITM resistance**, verifying that messages cannot be intercepted or altered without detection.

The protocol was evaluated using AVISPA's two main verification engines: OFMC and the CL-AtSe. OFMC utilizes symbolic execution combined with dynamic constraint solving, whereas CL-AtSe converts HLPSL specifications into logical constraints to systematically examine all possible protocol states.

The results, depicted in Figure 4.9, confirm that the SLAPSH protocol satisfies all defined security goals under both backends.

Verification Results:

- **OFMC**: *SAFE* – No attack traces discovered; protocol operations satisfy the defined goals.

```
%%%%% Role for User %%%%%%%
role user(U, SD, HG, RA: agent, SKmura, SKsdra:
symmetric_key,
H: hash_ func, SND, RCV: channel(dy))
played_by U
def=
local State: nat,
MIIDu, PWu, Rmu, PIDu, Kmug, Kra, Rra,
RIDmu,
HPWu, A1, A2, A3, A4, IDsd, Rsd, PIDsd, Kgsd,
B1, B2, Ksd: text,
RNmu, M1, C1, Vmu, IDg. RNg, M2, M3, C2,
Vmug, RNsd, SK, M4, Vsd, MS, Vgsd, PIDunew,
RIDunew, A3new, A4new, M6: text
const spl, sp2, sp3, sp4, u_hg_rnmu,
hg_sd_rng, sd_hg_rnsd, hg_u_rng, hg_u_rnsd:
protocol_id
init State:= 0
transition
%%%%%Registration phase%%%%%%%
1. State = 0 /\ RCV (start) =|>
State' := 1 /\ Rmu' := new() /\ PIDu' :=
H(IDu.Rmu')
/\SND({PIDu'} _SKmura)
/\ secret({MIIDu,PWu), sp1, {U})
2. State = 1 /\ RCV
({H(H(IDu.Rmu').Kra.Rra).H(H(MIIDu.Rmu').H(H
(MIIDu.Rmu').Kra.Rra)).Rsd')_SKmura)=|>
State' := 2 /\ HPWu':= H(PWu.Rmu')
/\ A1' := xor(Rmu',H(MIIDu.PWu))
/\ A2' := H(IDu.PWu.Rmu'.HPWu')
```

```
/\A3':=xor(H(H(MIIDu.Rmu').H(H(MIIDu.Rmu').KraR
ra)), H (Rmu'HPWu)))
/\ A4' := xor (H(H(MIIDu.Rmu'). Kra.Rra),
H(H(H(IDu.Rmu').H(H(MIIDu.Rmu').Kra.Rra)).HPWu'
))
%%%%%Authentication & Key agreement
phase%%%%%%%
/\ RNmu':= new()
/\ M1' := xor(H(H(MIIDu.Rmu')
H(H(IDu.Rmu').H(H(MIIDu.Rmu').Kra.Rra))
H(H(IDu.Rmu').Kra.Rra)),H(RNmu'.H(IDsd.Rsd))
/\ C1' := xor(H(MIIDu.RNmu),H(H(H(MIIDu
Rmu').Kra.Rra).RNmu'))
/\ Vmu':= H(H(IDu.Rmu').H(H(IDu.Rmu').H(H(IDu
Rmu').Kra.Rra)).RNmu'.H(IDsd.Rsd').H
(H(IDu.Rmu').Kra.Rra))
/\ SND(H(MIIDu. Rmu').M1'C1'.Vmu')
/\ witness(U,HG,u_hg_rnmu,RNmu')
3. State = 2 /\
RCV(xor(H(H(H(IDu.Rmu').H(H(MIIDu.Rmu').Kra.Rra
)).RNmu'),(H(IDg.RNg').H(IDsd.RNsd').H
(H(IDu.Rmu'.RNmu'))).H(H(IDu.Rmu').RNmu'.H(IDg.
RNg').H
(IDsd.RNsd').H(H(MIDu.RmuKra.Rra)))=|>
State' := 3 /\ SK':=
H(H(MIIDu.RNmu').H(IDg.RNg').H(IDsd.RNsd'))
/\ M6' := H(SK'.H(H(MIIDu.Rm').RNmu'))
/\ SND(M6')
/\ request(HG,U,hg_u_rng.RNg')
/\ request(HG,U,hg_u_rnsd,RNsd')
end role
```

FIGURE 4.7: Role for User

- **CL-AtSe**: *SAFE* – No violations of secrecy, authentication, or freshness were found.

These outcomes confirm that the SLAPSH protocol achieves its intended security objectives under formal analysis. The protocol withstands message tampering, identity spoofing, and key compromise scenarios, thereby reinforcing its robustness for smart home IoT networks.

4.10 Performance Evaluation

We evaluate the SLAPSH scheme's effectiveness in three key areas: functional capabilities, communication costs, and computational overhead. Our goal is to highlight

```
role session(U, SD, HG, RA : agent, SKmura, SKsdra: symmetric_key,
H: hash_func)
def=
local SN1, SN2, SN3, SN4, RV1, RV2, RV3, RV4 : channel(dy)
composition
user(U, SD, HG, RA, SKmura, SKsdra, H, SN1, RV1)
/\ smartdevice(U, SD, HG, RA, SKmura, SKsdra, H, SN2, RV2)
/\ homegateway(U, SD, HG, RA, SKmura, SKsdra, H, SN3, RV3)
/\ registrationauthority(U, SD, HG, RA, SKmura, SKsdra, H, SN4, RV4)
end role
role environment def=
const u, sd, hg, ra: agent, skmura, sksdra: symmetric_Key,
h: hash_func,
miidu, idsd, idg, kra, ksd, kmug, kgsd, pidu, pidsd: text,
u_hg_rnmu, hg_sd_rng, sd_hg_rnsd, hg_u_rng, hg_u_rnsd: protocol
id, spl, sp2, sp3, sp4: protocol_id
intruder_knowledge = {u, sd, hg, ra, pidu, pidsd, h}
composition
session(u, sd, hg, ra, skmura, sksdra, h)
/\session(sd, hg, ra, skmura, sksdra, h)
/\session(u,  hg, ra, skmura, sksdra, h)
/\ session(u, sd, ra, skmura, sksdra, h)
/\session(u, sd, hg,  skmura, sksdra, h)
end role
goal
secrecy_of spl, sp2, sp3, sp4
authentication_on u_hg_rnmu
authentication_on hg_sd_rng
authentication_on sd_hg_rnsd
authentication_on hg_u_rng
authentication_on hg_u_rnsd
end goal
environment()
```

FIGURE 4.8: Role of Session and Environment

SLAPSH's strengths and the efficiency-security trade-offs in IoT security, comparing it to established authentication schemes.

4.10.1 Functionality Comparison

Table 4.7 compares SLAPSH with similar protocols, emphasizing its effectiveness in protecting against known security risks. The procosed SLAPSH meets all the evaluated security features, such as mutual authentication, complete forward secrecy, and resistance to impersonation, session attacks and desynchronization attacks. This demonstrates its robust defense capabilities against various types of attacks.

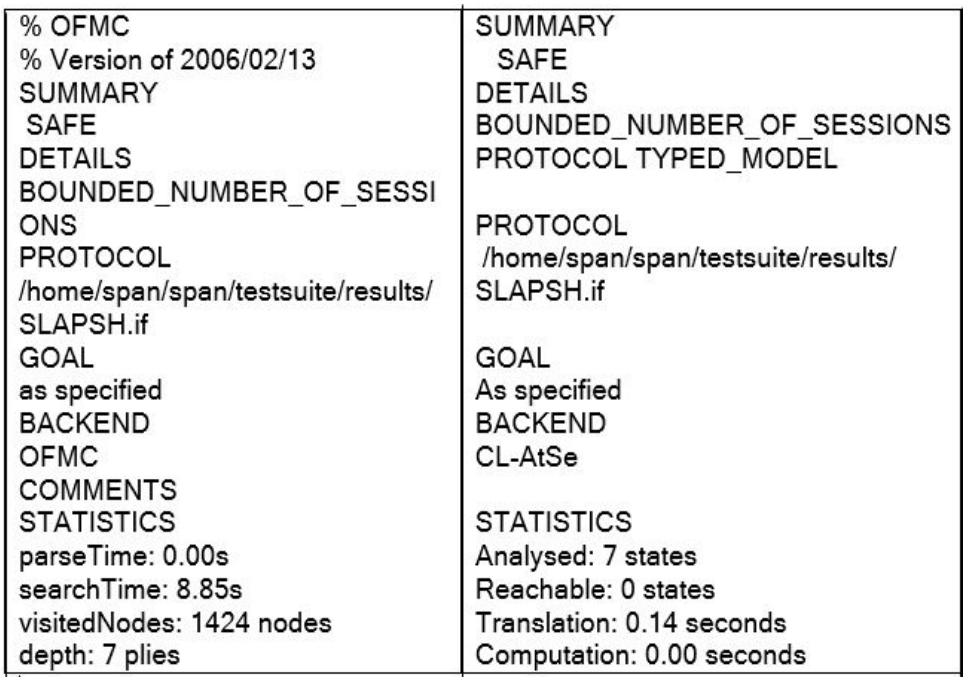


FIGURE 4.9: AVISPA result using OFMC and CL-AtSe

TABLE 4.7: Comparison of Performance and Security Features

Attacks	[193]	[192]	[185]	[186]	[187]	[188]	[189]	[190]	SLAPSH
MUI	Y	N	Y	Y	Y	Y	Y	Y	Y
HGI	N	Y	N	Y	N	Y	Y	Y	Y
SDI	Y	Y	Y	–	Y	Y	Y	Y	Y
SKD	Y	Y	Y	–	Y	Y	Y	Y	Y
RM	Y	Y	N	Y	Y	–	Y	–	Y
AU	Y	Y	Y	Y	Y	Y	Y	Y	Y
OG	–	Y	Y	N	Y	Y	–	Y	Y
SSD	Y	Y	Y	–	Y	Y	–	–	Y
MA	–	N	–	Y	–	Y	Y	Y	Y
PFS	–	Y	–	–	N	–	N	–	Y
DS	–	Y	Y	–	N	–	–	–	Y

4.11 Communication Cost

This section presents a detailed analysis of the communication costs incurred by the proposed SLAPSH scheme and compares them with those of existing authentication

mechanisms. The study is based on the length of transmitted messages during the authentication and key agreement phases.

The proposed SLAPSH scheme achieves a total communication cost of 1440 bits, accomplished with only four messages exchanged. This is significantly lower than many existing protocols, as shown in Figure 4.10.

The cryptographic elements used in SLAPSH—such as identities, random numbers, hash functions, timestamps, elliptic curve points, and symmetric encryption blocks—are based on the following bit lengths: 160, 128, 160, 160, 128, and 32 bits, respectively.

The four exchanged messages and their respective contents are as follows:

- **MSG$_1$** = $\{ F_u, V_1, S_1, W_u \}$: 320 bits
- **MSG$_2$** = $\{ F_u, V_3, S_4, W_2 \}$: 320 bits
- **MSG$_3$** = $\{ M_1, V_3, W_3 \}$: 320 bits
- **MSG$_4$** = $\{ V_6 \}$: 16 bits

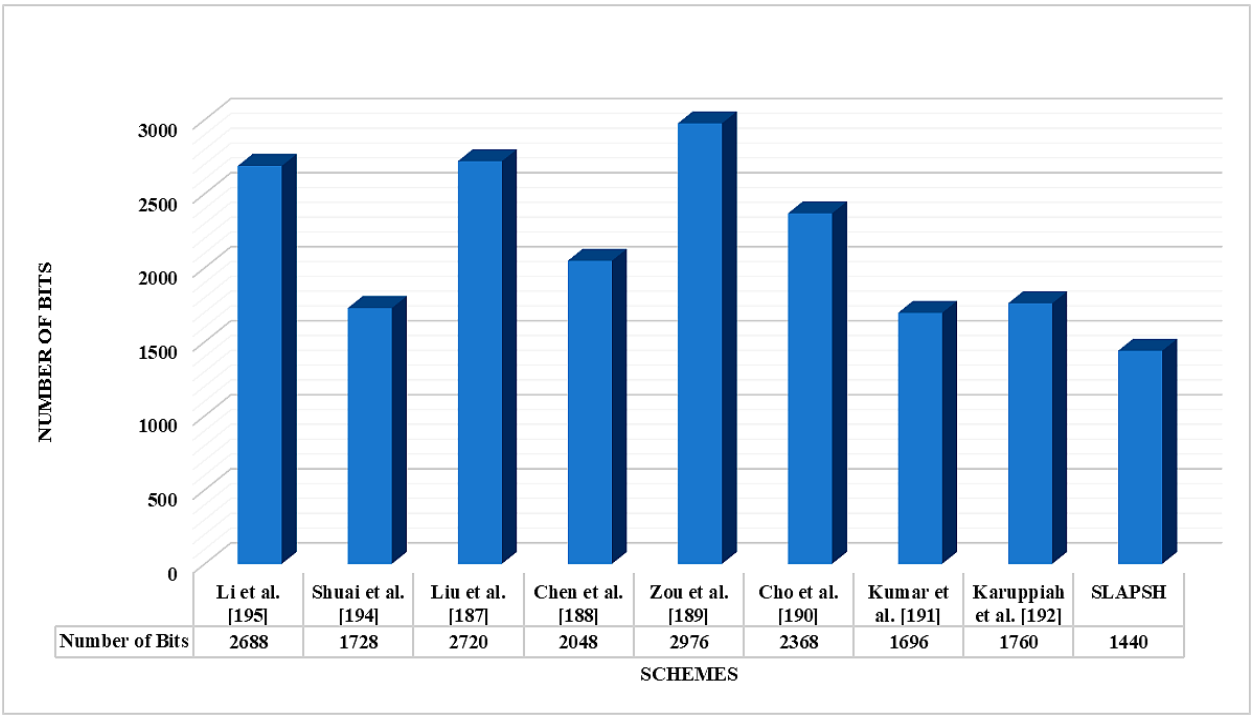


FIGURE 4.10: Comparison of Communication cost

The total communication cost is computed as:

$$320 + 320 + 320 + 64 + 16 = 1440 \text{ bits}$$

SLAPSH scheme enhances privacy by avoiding fixed user or device identities in transmitted packets, which also reduces packet size. The Interface Identifier (IID) in the IPv6 address format is integrated into the protocol's addressing strategy without adding extra transmission costs.

In general, the proposed SLAPSH scheme ensures lightweight and secure communication while maintaining anonymity, forward secrecy, and resistance to known attacks. Additionally, it offers lower communication costs compared to similar schemes.

4.12 Computational Cost

To evaluate the efficiency of the proposed SLAPSH scheme, we compare its computational overhead with that of existing schemes. The computational cost is calculated based on the number of cryptographic operations required and their respective time complexities.

In the computation cost analysis, T_m denotes the time required for symmetric key encryption or decryption, which is approximately 0.1303 milliseconds. The time taken for executing a hash function is represented by T_h, measured at around 0.0004 milliseconds. Similarly, T_E indicates the time for performing elliptic curve point multiplication, which is approximately 7.3529 milliseconds. Lastly, T_f accounts for the time consumed by fuzzy extractor operations.

The SLAPSH scheme achieves a total computational cost of:

$$T = 42T_h + T_E + T_f + T_m$$

which equates to approximately 0.2806 milliseconds. This cost is significantly lower than that of competing schemes as shown in Figure 4.11, highlighting SLAPSH's suitability for time-sensitive and resource-constrained smart home environments.

The proposed SLAPSH protocol offers lower computational and communication overhead compared to existing schemes, while providing robust security features such as

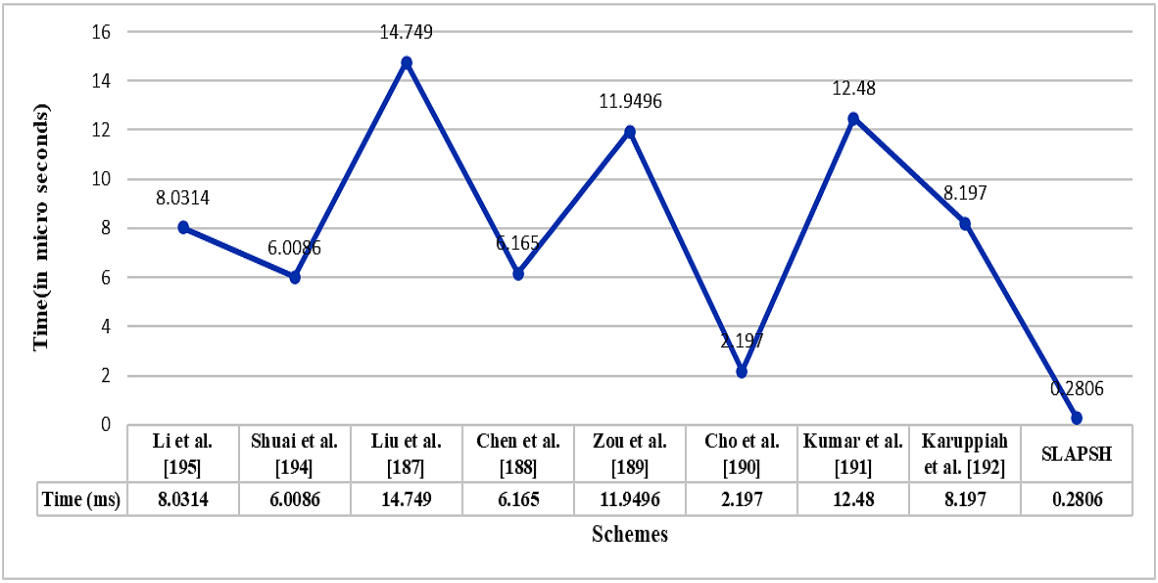


FIGURE 4.11: Comparison of Computation cost

mutual authentication, anonymity, and resistance to known attacks. Its lightweight design makes the SLAPSH protocol particularly effective for real-time smart home applications, where minimizing latency and resource usage is crucial.

4.13 Conclusion

Smart home technologies are transforming daily living, but simultaneously introduce significant challenges regarding privacy, authentication, and secure communication. To address these challenges, this work introduces SLAPSH tailored for smart home scenarios. SLAPSH incorporates an enhanced IPv6 addressing method and a mutual authentication framework to facilitate secure and efficient interactions among users, devices, and the home gateway.

The protocol extends the standard IPv6 format by embedding verifiable identity information within device addresses, thus strengthening traceability and authentication without increasing transmission overhead. SLAPSH leverages hash-based computations and nonce-based challenge-response mechanisms to counter threats such as impersonation, replay, man-in-the-middle attacks, and session key compromise.

Comprehensive security analysis, including formal validation with the ROR model and the AVISPA tool, confirms the robustness of SLAPSH. Network simulations using NS-3 demonstrate minimal communication delays and consistent throughput. Comparative

studies demonstrate that SLAPSH provides significant reductions in computational and communication expenses, making it particularly suitable for IoT devices with limited resources.

Although the proposed SLAPSH scheme delivers a robust and scalable solution for contemporary smart homes, future research will explore its applicability in broader IoT ecosystems, integration with post-quantum cryptography, and evaluation in practical deployment environments. These developments aim to further enhance its resilience and adaptability in evolving intelligent systems.

CHAPTER 5
Reliable and Secure Addressing with Authentication in IoT Networks

5.1 Introduction

The swift evolution of information and communication technologies (ICT) has fueled the proliferation of smart homes that leverage the Internet of Things (IoT). This progress enables users to remotely manage and monitor household devices, resulting in greater flexibility, enhanced efficiency, and a noticeable improvement in quality of life. Within the IoT environment, a diverse array of devices—such as wearables, sensors, and actuators—work together to collect data and enable intelligent automation within homes, as depicted in Figure 4.1. Components like smoke alarms, smart lighting, and climate control systems are typically managed by a central gateway, often under the supervision of a registration authority.

While the advantages of cost savings and streamlined automation are evident, these innovations also introduce new security concerns. Risks such as unauthorized device access, data breaches, and identity fraud jeopardize user privacy and the overall integrity of the network. It is essential to address these security issues to protect smart home users and ensure the reliability and trustworthiness of these interconnected systems.

To ensure a secure process, robust user and device authentication mechanisms are essential. However, achieving secure communication in IoT networks remains challenging due to the dynamic, distributed, and resource-constrained nature of these environments. Devices typically rely on open wireless channels, making them susceptible to attacks

such as identity forgery, replay, and man-in-the-middle attacks [175]. Moreover, the architectural complexity of IoT networks—especially when involving cloud-based three-tier structures—exacerbates the difficulty of maintaining trust and privacy. Although various cryptographic techniques and secure protocols have been proposed, many fall short in terms of scalability, efficiency, or adaptability to evolving threats. The depletion of IPv4 addresses and the need for seamless integration of an ever-growing number of devices further necessitate a transition to IPv6 and the development of enhanced addressing and authentication schemes [132].

We addresses these concerns by proposing a secure and unique addressing scheme tailored for smart home IoT networks. The proposed solution enhances authentication and ensures data integrity by embedding verifiable user and device identities into the IPv6 address structure, thereby facilitating secure and efficient communication without excessive overhead.

5.2 Main Contribution of the Proposed Scheme

The key contributions of this research are outlined below:

- We have presented a novel authentication framework known as Secure and Unique Addressing with Mutual Authentication Scheme (SUMAS). This protocol presents a modified IPv6 interface identification structure that embeds both user and device IDs without increasing the packet size, therefore enabling improved traceability and smooth deployment inside smart home IoT scenarios.
- SUMAS effectively combines unique identities into the lower-order bits of the IPv6 address to remove the packet overhead usually brought about by identity encapsulation. This method guarantees exact and safe addressing for every included smart device.
- The protocol incorporates a robust mutual authentication mechanism based on dynamically generated session keys, ensuring strong resistance against impersonation, replay attacks, and other common threats. An informal security analysis substantiates the protocol's resilience.

- Comprehensive formal analysis is done using ROR model and the AVISPA verification tool, both of which confirm the protocol's effectiveness and soundness in maintaining authentication integrity.
- A detailed performance evaluation—comparing computational and communication costs with existing protocols—demonstrates that SUMAS offers reduced overhead while preserving essential security guarantees. Its lightweight design makes it particularly well-suited for the constrained resources of IoT devices.

5.3 Related Work

The field of IoT security has attracted considerable research attention, particularly in the development of secure addressing techniques and robust authentication mechanisms. This section reviews related studies, which can be divided into two main categories: addressing techniques for unique device identification and security mechanisms for mutual authentication, along with measures to protect against various threats.

5.3.1 Addressing Techniques

To manage the exponential growth of connected devices, several addressing methods have been developed based on the IPv6 protocol. The authors [135] and other IETF reports [148] proposed unique unicast IPv6 address formats for local and global communication in smart environments. These formats aim to assign distinct identities to devices while complying with IPv6 allocation frameworks.

One standard method for automatic address generation uses the EUI-64 format [150], which constructs the interface identifier (IID) from the MAC address of networked devices. However, this approach has security drawbacks. Because MAC addresses are static and globally unique, they make the device vulnerable to tracking and targeted attacks. Moreover, static IPs are susceptible to DoS attacks due to their predictability.

To enhance privacy in network communications, Stateless Address Auto Configuration (SLAAC) mechanisms were improved with opaque identifiers that generate randomized Interface Identifiers (IIDs) [137, 151]. These identifiers help reduce the risk of user tracking. Building on this, the author [134] introduced privacy extensions to SLAAC

that create short addresses for anonymous communications, allowing users to interact without leaving a traceable footprint.

Further addressing techniques, such as MT6D (Moving Target IPv6 Defence) [135, 148] and IPv6 hopping [148, 150], aim to increase the difficulty for attackers to locate and target devices by dynamically changing addresses. However, these schemes often increase processing and communication overhead.

Despite these advancements, significant limitations persist. For instance, address duplication can occur when multiple devices on the same link generate the same IID. Additionally, the 64-bit space used for the IPv6 interface identifier often lacks entropy when derived from predictable sources, weakening its effectiveness against collision and inference attacks [151].

5.3.2 Security Techniques

In parallel with addressing challenges, extensive research has been conducted to develop secure authentication schemes for IoT networks. Traditional authentication protocols using biometric-based credentials and smart cards [191, 192] have been extended to support multi-server environments. However, many suffer from vulnerabilities to attacks such as man-in-the-middle (MITM), impersonation, and offline guessing.

Some existing research explores the use of device-specific physical properties, such as silicon IDs, for unique identification. However, these identifiers can be compromised during tampering, which limits their effectiveness in situations involving attackers.

Elliptic Curve Cryptography (ECC) has been widely adopted in lightweight IoT authentication schemes due to its balance of security and computational efficiency [193, 194]. ECC-based schemes typically rely on the ECDLP (Elliptic Curve Discrete Logarithm Problem) and ECCDH (Elliptic Curve Computational Diffie-Hellman) assumptions for secure key exchange. However, several ECC-based schemes have been found to lack full-fledged identity verification steps, making them vulnerable to spoofing and impersonation attacks [195].

To enhance user privacy and data integrity, systems like Priv-Home [196] were introduced, focusing on secure entity and data authorization in smart environments. Other approaches leverage physically unclonable functions (PUFs) for key agreement protocols, providing resistance to device cloning [197].

Many smart home protocols still struggle to protect against sophisticated attacks, such as energy manipulation, message traceability, and session hijacking. These systems often require significant computational resources or lack scalability. Recent works such as [199] attempt to bridge these gaps by introducing anonymous two-factor authentication and key agreement protocols designed for constrained smart home environments. However, most existing methods still do not integrate user and device authentication with address-level security at the network layer.

The proposed SUMAS scheme addresses these issues by incorporating identity and authentication data into the IPv6 address through a modified interface identifier (MIID). By embedding hashed user and device identifiers, it facilitates traceable and verifiable communication without adding extra packet overhead. This dual functionality—providing unique addressing and enabling secure mutual authentication—makes the SUMAS scheme particularly suitable for low-power, latency-sensitive IoT environments.

5.4 System Model and Attacker Model

In this section, we describe the architecture of the proposed SUMAS scheme by presenting both the system model and the attacker model. The system is designed for a smart home IoT environment, where secure communication and reliable authentication are essential for protecting user data and device interactions. The system model and address format for devices are shown in Figure 4.3.

5.4.1 System Model

The system model is structured around three primary entities: the user, the home gateway (HG), and a centralized control unit. These elements work together to facilitate secure management and authentication of smart devices within the home environment.

- **User (U):** The end-user operates smart home appliances and services through personal devices such as smartphones or tablets. Each user is assigned a unique 48-bit identity to distinguish their requests and actions.
- **Home Gateway (HG):** The HG serves as a central hub that connects smart devices to the internet. It plays a critical role in assigning identities, verifying login

credentials, managing device access, and enforcing security policies. It communicates with the control unit and other devices via the IEEE 802.11 wireless standard.

- **Control Unit:** This device keeps the smart home network in general setup. It controls flow of communication, allocates addresses, and keeps special identities. It embeds the user's 48-bit identity and a 16-bit device ID during device initialising to create a 64-bit modified interface identification (MIID), which is applied in the IPv6 address construction. This integration guarantees safe identification without enlarging the packet header size.

5.4.2 Attacker Model

To analyze the security of the SUMAS protocol, we employ the well-known Dolev-Yao (DY) threat model [96]. This established adversarial framework assumes that an attacker has complete control over all public communication channels. Consequently, the attacker can engage in various malicious activities, including:

- **Eavesdropping:** Intercepting all messages transmitted between legitimate entities.
- **Modification:** Altering, injecting, or deleting messages in real time.
- **Replay Attacks:** Resending previously intercepted messages to trick the system.
- **Man-in-the-Middle (MITM):** Inserting themselves between communicating parties to manipulate or monitor data exchanges.
- **Power Analysis:** Accessing stored credentials from a compromised device by analyzing its power consumption patterns.
- **Session Compromise:** Attempting to derive secret session keys or impersonate legitimate devices and users.

Under the DY model, it is assumed that the attacker cannot break the underlying cryptographic primitives (e.g., hash functions, encryption algorithms) but can manipulate messages freely within the network. The proposed scheme is evaluated against these threat vectors to ensure confidentiality, integrity, and authenticity.

5.4.3 Assumptions and Notation

The following assumptions and notations are used throughout the protocol:

- All smart devices (SD), users (U), and the home gateway (HG) are pre-configured with symmetric cryptographic capabilities.
- The HG and service provider (SP) are considered fully trusted entities with unlimited computational and memory resources.
- Symbols such as ID_u, ID_{hg}, SK (session key), TK (token), and $MIID$ (modified interface identifier) are defined to represent various security parameters.
- The system supports essential features, including energy monitoring, appliance control, safety protocols, and secure communication between mobile users and IoT devices.

5.5 Proposed Scheme

The proposed SUMAS scheme aims to enhance security and ensure unique identification in smart home IoT networks. It accomplishes this by incorporating a secure addressing mechanism along with a lightweight mutual authentication process. The protocol is divided into six distinct phases:

1. Network Phase
2. Address Phase
3. System Installation Phase
4. Registration and Login Phase
5. Key Establishment and Authentication Phase
6. Password Update Phase

Each phase plays a crucial role in ensuring device integrity, enabling secure communication, and providing protection against various types of attacks. Table 5.1 lists the symbols used in the protocol and their descriptions, while Figure 4.4 illustrates the stages of the SUMAS scheme.

TABLE 5.1: List of Symbols and Their Descriptions

Symbols	Descriptions
$MIId_i$	48-bit identifier uniquely representing a user interface
$MIID_a$	16-bit identifier assigned to a smart device interface
ID_{hg}	Unique identifier for the home gateway
$E_k[m]$	Message m encrypted using key k
$D_k[m]$	Message m decrypted using key k
S_k	Session-specific cryptographic key
T_K	Authentication token allocated to a smart device
PW_i	Password of user i
U_i	User i's unique identity
SP	Service provider entity
b	Random value generated by user U_i
h(.)	Cryptographic hash function
$\|\|$	Concatenation operation
T_{id}	Identifier for a particular timestamp
$\oplus$	Bitwise XOR operation
T	Timestamp value
N_r	Unique security parameter for the home gateway

5.5.1 Network Phase

The network phase is initiated during the initial setup of the smart home environment. In this phase, each user's identity is cryptographically mapped into a 48-bit Modified Interface Identifier (MIID) using the SHA-3 hash function. This hashed identity serves as a unique and anonymized reference, which is securely stored within the Home Gateway (HG). When a user attempts to initiate communication with the network, the HG broadcasts an initialization signal across all smart devices. Devices previously associated with the corresponding MIID respond to the signal by updating their local session and state information. In contrast, devices that do not recognize the MIID discard the signal, ensuring that only relevant devices remain responsive.

To enable device-level granularity, each smart device, including the user's primary control unit, is assigned an additional 16-bit device identifier during this phase. By combining the user's 48-bit identity with the device-specific 16-bit identifier, we create a unique 64-bit Multi-Identity Identifier (MIID) for each node in the network. This MIID serves as the foundation for secure addressing and authentication within the SUMAS protocol, allowing for lightweight and collision-resistant identity representation.

5.5.2 Addressing Phase

After the network initialization, the Home Gateway begins the addressing phase by broadcasting a 64-bit IPv6 network prefix to all authorized devices connected to its network. Each device then creates its complete IPv6 address by appending its 64-bit Modified Interface Identifier (MIID) to the network prefix. his process results in a globally unique 128-bit IPv6 address, as shown in Figure 4.2. This address structure ensures that each device has a verifiable and consistent identifier that is linked to both its user and specific hardware instance.

During this phase, the HG securely encrypts the user's identity and stores it in a protected key table for reference in future authentication sessions. The 16-bit device identifier previously assigned during the network phase is now permanently linked to the 48-bit user identity to maintain consistency and traceability. The resultant 64-bit MIID not only enables secure and decentralized addressing but also functions as a cryptographic anchor for authentication within the proposed SUMAS framework. This design eliminates the need for a centralized identity resolution mechanism while preserving privacy and supporting dynamic device discovery.

5.5.3 System Installation Phase

Before deploying any smart device within a smart home environment, a secure registration process must be carried out with an offline Service Provider (SP). This phase establishes the foundational cryptographic context necessary for all subsequent authentication and communication procedures. During installation, the SP assigns a globally unique Modified Interface Identifier to the device, denoted $MIID_a$, which serves as its persistent identity across the network. To ensure entropy and security, the SP also selects two secret values, p and q, which are used to derive key cryptographic hashes.

Specifically, the SP computes $h(p\|q)$ as a global entropy hash and $h(ID_{hg}\|h(p))$, which binds the device to the Home Gateway (HG) through its identifier ID_{hg}. These hashes are securely stored in the memory of the HG to facilitate future identity resolution and mutual authentication.

In addition to these hashed values, the SP generates a set of device-specific parameters essential for protocol operations. These include the key K, the random value R_i, and the device secret S_i, all of which are crucial for deriving temporary session keys and for validating message authenticity. The values $\{K, R_i, S_i\}$ are securely embedded in the device and are also retained in the SP backend database for recovery and integrity checks. This dual-record strategy ensures resilience and supports secure authentication even if the device undergoes a partial reset or transfer within a trusted domain.

By doing these prior computations and registrations offline, the installation phase ensures that all participating devices have the required cryptographic credentials to interact securely inside the proposed SUMAS framework, so reducing dependency on dynamic initialization at runtime.

5.5.4 Registration and Login Phase

During registration, the user U_i selects their identifier ID_i, password PW_i, and a random number b, computing $h(b \oplus PW_i)$. The user sends $X_1 = h(MIID_i\|N_r)$ to the HG, which computes $X_2 = X_1 \oplus h(MIID_i\|h(b \oplus PW_i))$ and stores both X_1, X_2, and $H_1 = h(X_1)$.

For login, the user enters their identity and password. The HG retrieves $X_1 = X_2 \oplus h(MIID_i\|h(b \oplus PW_i))$ and verifies $H_1^* = h(X_1)$. If H_1^* matches the stored H_1, the user is authenticated. The interaction between device and HG can be seen in Table 5.2.

5.5.5 Key Establishment and Authentication Phase

The key establishment and authentication phase of the SUMAS schemes enables secure mutual verification between the smart device and the Home Gateway (HG), ultimately leading to the generation of a session-specific symmetric key that safeguards subsequent communications. This phase is crucial for establishing a trust relationship and ensuring that both parties are legitimate participants in the protocol. The process initiates when the smart device generates a fresh random nonce N_A, which introduces randomness

TABLE 5.2: Registration and Login Phase

Device A (User)	**Home Gateway (HG)**
Registration Phase: Selects identity ID_i, password PW_i, and random number b Computes $h(b \oplus PW_i)$ Computes $X_1 = h(MIID_i \| N_r)$ Sends X_1 to HG	Receives X_1 Computes $X_2 = X_1 \oplus h(MIID_i \| h(b \oplus PW_i))$ Computes $H_1 = h(X_1)$ Stores $\{X_1, X_2, H_1\}$ for future login
Login Phase: Inputs ID_i and PW_i Computes $h(b \oplus PW_i)$ Sends $\{MIID_i, PW_i\}$	Receives login request Computes $X_1 = X_2 \oplus h(MIID_i \| h(b \oplus PW_i))$ Computes $H_1^* = h(X_1)$ Verifies $H_1^* = H_1$ If valid, login is successful; otherwise, authentication fails

and ensures uniqueness for the session. This nonce serves as a foundational input for computing multiple authentication parameters that are cryptographically bound to both the device's and the gateway's identities. These parameters are designed to preserve the confidentiality, integrity, and freshness of the session while preventing impersonation and replay attacks. Through the exchange and verification of these computed values, the protocol ensures that only authorised devices and the registered HG can derive the same session key and proceed with secure data transmission.

Firstly, the device computes the value V_1 as $V_1 = h(ID_{hg} \| h(p)) \oplus N_A \oplus T_1$, where ID_{hg} is the identity of the HG, p is a secret value shared with the device, and T_1 is the current timestamp. To preserve identity privacy, the device creates a concealed identity tag CID_i given by $CID_i = S_i \oplus h(h(ID_{hg} \| h(p)) \| N_A \oplus T_1)$, where S_i is a device-specific secret. A temporary key KT is derived using $KT = h(R_i) \oplus N_A$, with R_i being a pre-assigned random number. This key is used to encrypt session metadata including the device identity, gateway identity, nonce, session token TK, and timestamp, resulting in the ciphertext $C_1 = E_{KT}[ID_a \| ID_{hg} \| n \| TK \| T_1]$.

The device sends the tuple $\{V_1, CID_i, C_1, T_1, ID_i, MIID_a\}$ to the HG. Here, ID_i represents the user's identity, and $MIID_a$ is the device's modified interface identifier. Upon receiving this message, the HG checks the freshness of the request by verifying if the time difference $|T_2 - T_1|$ is within the threshold ΔT. It extracts the nonce N_A using the formula: $N_A = V_1 \oplus h(ID_{hg} \| h(p)) \oplus T_1$. With this nonce, the HG reconstructs the temporary key $KT = h(R_i) \oplus N_A$ and decrypts C_1 to retrieve ID_a^*, ID_{hg}^*, n^*, TK, and T_1^*.

TABLE 5.3: Key Establishment and Authentication Phase

Device A (User)	**Home Gateway (HG)**
Step 1 Generates random nonce N_A Computes: $V_1 = h(ID_{hg} \| h(p)) \oplus N_A \oplus T_1$ $CID_i = S_i \oplus h(h(ID_{hg} \| h(p)) \| N_A \oplus T_1)$ $KT = h(R_i) \oplus N_A$ $C_1 = E_{KT}[ID_a \| ID_{hg} \| n \| TK \| T_1]$ Sends $\{V_1, CID_i, C_1, T_1, ID_i, MIID_a\}$ to HG	**Step 2** Checks freshness: $\|T_2 - T_1\| \leq \Delta T$ Computes: $N_A = V_1 \oplus h(ID_{hg} \| h(p)) \oplus T_1$ $KT = h(R_i) \oplus N_A$ Decrypts $C_1 \rightarrow ID_a^*, ID_{hg}^*, n^*, TK, T_1^*$ Verifies: $ID_{hg}^* = ID_{hg},\ T_1^* = T_1$ Computes session key: $SK = h(N_A \| N_{hg} \| ID_{hg} \| TK)$ Computes: $V_2 = N_{hg} \oplus S_i \oplus h(R_i) \oplus T_2$ $C_2 = E_{SK}[ID_{hg}, T_1, T_2]$ Sends $\{C_2, V_2, T_2\}$ to Device A
Step 3 Receives $\{C_2, V_2, T_2\}$ Verifies: $\|T_3 - T_2\| \leq \Delta T$ Computes: $N_{hg} = V_2 \oplus S_i \oplus h(R_i) \oplus T_2$ $SK = h(N_A \| N_{hg} \| ID_{hg} \| TK)$ Decrypts C_2 to retrieve: ID_{hg}^*, T_1^*, T_2^* Checks: $ID_{hg}^* = ID_{hg},\ T_1^* = T_1,\ T_2^* = T_2$ Sends confirmation of key agreement	**Step 4** Receives confirmation Verifies: $ID_{hg}^* = ID_{hg},\ T_1^* = T_1,\ T_2^* = T_2$ Recomputes: $SK = h(N_A \| N_{hg} \| ID_{hg} \| TK)$ Mutual key agreement is finalized

It verifies the sender's legitimacy by comparing the extracted values with stored ones; mutual authentication is confirmed if $ID_a^* = ID_a$, $ID_{hg}^* = ID_{hg}$, and $T_1^* = T_1$.

Following successful verification, the HG generates a new nonce N_{hg} and derives the session key using the formula $SK = h(N_A \| N_{hg} \| ID_{hg} \| TK)$. To reply securely, it computes a masked version of N_{hg} using $V_2 = N_{hg} \oplus S_i \oplus h(R_i) \oplus T_2$, and constructs an encrypted message $C_2 = E_{SK}[ID_{hg}, T_1, T_2]$, where T_2 is the HG's current timestamp. The HG then transmits the tuple $\{C_2, V_2, T_2\}$ back to the device.

On receiving the response, the device validates the freshness of T_2 by ensuring that $|T_3 - T_2| \leq \Delta T$, where T_3 is the device's local time. It extracts the gateway nonce using $N_{hg} = V_2 \oplus S_i \oplus h(R_i) \oplus T_2$, and recalculates the session key with the same formula $SK = h(N_A \| N_{hg} \| ID_{hg} \| TK)$. Finally, the device decrypts C_2 using this key and verifies that the contents, namely ID_{hg}, T_1, and T_2, match the expected values.

If all verifications succeed, the smart device and HG agree on the session key SK. This mutual authentication protects the communication channel from impersonation, replay

attacks, and unauthorized access while maintaining low overhead. The key establishment and authentication phase is presented in Table 5.3.

5.5.6 Password Update Phase

The password update phase enables users to securely change their login credentials without requiring re-registration. This mechanism is essential for maintaining account security and preventing unauthorized access, especially in long-term deployments of smart home devices. The password update phase is presented in Table 5.4.

To initiate the update process, the user first provides their current password, PW_i^{old}, and associated identifier, $MIID_i$, over a secure communication channel to the Home Gateway (HG). The HG then verifies the legitimacy of the request using the same hash-based validation procedure employed during the login phase. Specifically, the HG retrieves the stored values X_1, X_2, and H_1, and recomputes X_1 as $X_2 \oplus h(MIID_i \| h(b \oplus PW_i^{old}))$. If the hash $h(X_1)$ matches the stored value H_1, the current credentials are considered valid.

TABLE 5.4: Password Update Phase

Device A (User)	**Home Gateway (HG)**
Enters current ID and old password PW_i^{old} Computes: $h(b \oplus PW_i^{old})$ Sends $\{MIID_i, h(b \oplus PW_i^{old})\}$	Receives login request Computes: $X_1 = X_2 \oplus h(MIID_i \| h(b \oplus PW_i^{old}))$ Verifies: $H_1^* = h(X_1)$ against stored H_1 If valid, proceeds to next step, else aborts
Enters new password PW_i^{new} and new random b^{new} Computes: $h(b^{new} \oplus PW_i^{new})$ Sends $\{MIID_i, h(b^{new} \oplus PW_i^{new})\}$	Computes: $X_2^{new} = X_1 \oplus h(MIID_i \| h(b^{new} \oplus PW_i^{new}))$ Replaces old X_2 with X_2^{new} Updates stored record

Once authenticated, the user selects a new password PW_i^{new} along with a new random number b^{new}. The user then computes the hash $h(b^{new} \oplus PW_i^{new})$ and transmits it to the HG. The HG, upon receiving this hash, updates the stored value of X_2 to reflect the new password using the equation $X_2^{new} = X_1 \oplus h(MIID_i \| h(b^{new} \oplus PW_i^{new}))$. This new value replaces the previous X_2 in the HG's secure storage.

By employing this method, the SUMAS system guarantees that authorized users may safely implement password updates without disclosing private information including

passwords and random seeds. Especially the usage of XOR operations and hash functions preserves lightweight computational efficiency and provides good resilience against credential spoofing and offline dictionary attacks.

5.6 Informal Security Analysis

Using an informal analytical approach, this section assesses the resilience of the proposed SUMAS scheme against prevalent security concerns. Under many attack situations, the study shows that the system guarantees confidentiality, integrity, anonymity, and mutual authentication.

5.6.1 Mutual Authentication Attack (MAA)

The SUMAS scheme defends against mutual authentication attacks by requiring the smart device and the Home Gateway (HG) to verify identities. A session is established only if both confirm each other's legitimacy. The HG verifies the device's identity by reconstructing a random value R_i from installation, validating it with $R_i = E_k(MIID_a, n)$, where E_k is symmetric encryption with session key k, $MIID_a$ is the modified interface identifier, and n is a nonce. This confirms the message is from a legitimate, preregistered device. The smart device checks the HG by comparing the identity and timestamp from the gateway with expected values. After decryption, it confirms the identity matches $ID^*_{hg} = ID_{hg}$ and the timestamp aligns with $T^*_1 = T_1$. These checks ensure the response is current and from the authorized HG. This protects SUMAS from impersonation and replay attacks, preventing attackers from initiating authentication without device-specific secrets and session tokens. Only legitimate devices and registered HG can establish a secure session key.

5.6.2 Device Impersonation Attack (DIA)

The SUMAS scheme resists device impersonation attacks with dynamic, confidential parameters bound to each device. An adversary must fabricate a valid authentication message with a correctly encrypted payload to impersonate a smart device. The attacker generates ciphertext $C_1 = E_{KT}(ID_a \| ID_{hg} \| n \| TK \| T_1)$, where E_{KT} is symmetric encryption with temporary key KT. The key KT is derived as $KT = h(R_i) \oplus N_A$, using

device-specific random value R_i and nonce N_A. Since R_i is embedded in the device and N_A is never sent in plaintext, the attacker cannot reconstruct KT. Without this key, they cannot create a valid ciphertext C_1 that decrypts correctly or passes verification at the Home Gateway. If an attacker replays an intercepted message, the timestamp T_1 will fail the freshness check by the gateway. Thus, the SUMAS scheme prevents unauthorized entities from mimicking registered devices, ensuring only legitimate nodes with valid credentials can authenticate requests.

5.6.3 Anonymity Attack

The SUMAS scheme ensures strong anonymity protection by dynamically masking the device's identity in every communication session, preventing external observers from linking messages to a specific user or device. During each authentication attempt, the user generates a session-specific identifier that conceals its true identity through a combination of cryptographic operations and nonce randomness.

The user identifier, CID_i, is computed as $CID_i = S_i \oplus h(h(ID_{hg} \| h(p)) \| N_A \oplus T_1)$, where S_i is a static secret tied to the device, ID_{hg} is the Home Gateway identity, p is a shared key, N_A is a random nonce, and T_1 is the timestamp. This ensures that while S_i is constant, the masked identifier varies with each session due to N_A and T_1. Since N_A and T_1 are unique and not reused in sessions, CID_i appears statistically independent across protocol runs. Thus, an eavesdropper monitoring the network cannot link multiple CID_i values to a specific device or user, even with many observed sessions. This ensures the protocol protects identity and maintains session unlinkability, crucial for user privacy in IoT smart homes.

5.6.4 Untraceability

The SUMAS scheme ensures untraceability by utilizing randomness and time variance in authentication sessions, making it challenging for attackers to associate sessions with specific devices or users. This aspect is vital for maintaining user privacy in smart homes where devices frequently interact.

In each authentication cycle, the protocol uses unique parameters: a device nonce (N_A), a random Home Gateway nonce (N_{hg}), a session token (TK), and the gateway's identity (ID_{hg}) to compute the session key as $SK = h(N_A N_{hg} ID_{hg} TK)$. If an attacker intercepts

several encrypted messages, the session keys and payloads remain independent of one another. Without access to the internal nonce values, an outsider cannot link the communications to the same user or device. This provides strong protection against traceability attacks and ensures that each session is cryptographically unlinkable.

5.6.5 Replay Attack

Upon receiving an authentication request, the Home Gateway (HG) verifies the freshness of the message by comparing the received timestamp (T_1) with its current time (T_2). The protocol enforces $|T_2 - T_1| \leq \Delta T$, where ΔT is a predefined window that accounts for minor delays. This check ensures that only recent messages are processed.

If the condition fails, indicating a delayed or reused message, the HG discards the request. Because the timestamp is embedded in the encrypted payload and identity computations, tampering without the original secrets is infeasible. Thus, even if an attacker captures a message, any attempt to reuse it in a different session will be invalidated by the timestamp verification, neutralizing the replay threat.

5.6.6 Man-in-the-Middle Attack (MITM)

The encrypted communication from the Home Gateway (HG) to the smart device, $C_2 = E_{SK}(ID_{hg}, T_1, T_2)$, is an example of this defensive mechanism. E_{SK} is the encryption technique that uses the session key SK, which is calculated as a hash of session-specific inputs such as device nonce N_A, gateway nonce N_{hg}, gateway identity ID_{hg}, and session token TK. External observers find the key SK unexpected and distinct since its values change across sessions and are never in plaintext. As a result, an attacker attempting to intercept the communication will be unable to decode or modify C_2 unless they have access to the secret key SK. Unauthorised alterations would result in decryption failures or invalid identity and timestamp values, which would therefore be rejected.

5.6.7 Denial-of-Service Attack (DoS)

Early cryptographic hash checking during the authentication process serves as a protective technique. After receiving X_1 from the user, the Home Gateway (HG) calculates a hash, $H_1^* = h(X_1)$, and compares it to the reference value H_1 stored in the system. If

a discrepancy is detected, the authentication attempt is denied, preventing further processes such as decryption or session key creation. With minimal computing expense, this early departure mechanism efficiently rejects fraudulent or invalid requests.

Additionally, before executing authentication, SUMAS checks that $|T_2 - T_1| \leq \Delta T$ by evaluating the timestamp's freshness. This dual-layered approach, which combines hash-based verification and time-bound validation, effectively filters out unauthenticated or replayed messages. It preserves computational resources while maintaining protocol responsiveness in the event of a denial-of-service (DoS) attack.

5.7 Formal Security Analysis Using RoR Model

To validate the security of the SUMAS scheme, we utilize the Real-or-Random (ROR) model, which is a well-established framework for analyzing authentication and key exchange protocols. This model evaluate the likelihood that an adversary can distinguish between a real session key and a randomly generated one. The ROR model consists of several key components.:

1. **Participants:** User (U), Home Gateway (HG), and Smart Device (SD) are the protocol entities involved in each session.
2. **Sessions:** Each instance of a protocol run is denoted as Int_U^i, Int_{HG}^j, and Int_{SD}^k for user, gateway, and device respectively.
3. **Session Key Freshness:** A session key is considered *fresh* if it is unknown to the adversary through Reveal, Corrupt, or any previous queries.
4. **Adversary Capabilities:** The adversary J can intercept messages, initiate protocol runs, reveal session keys, and corrupt participants, but cannot break underlying cryptographic primitives.

The attacker J can interact with the protocol through the following queries:

1. Send(U/HG, M): Sends a message M to either U or HG and observes the response.
2. Reveal(Int^i): Returns the session key used in session Int^i.

3. Corrupt(Int^i): Reveals long-term secrets of the entity.

4. Execute(U, HG): Passively eavesdrops on honest execution.

5. Test(Int^i): Returns either the actual session key or a random string based on a hidden coin flip.

Security Goal

The goal is to assess the adversary's ability to differentiate the actual session key from a random string. This advantage is defined as:

$$Adv^{\text{ROR}}_{\text{SUMAS}}(J) = 2 \cdot \Pr[J \text{ guesses correctly}] - 1$$

1. **Game-Based Proof Sequence**

   The security of the **SUMAS** protocol is analyzed through a sequence of games:

2. **Game 0 (Real Execution)**

   In this game, the protocol runs honestly, and the Test query returns the actual session key. The adversary tries to distinguish it from a random value.

$$\Pr[\text{GM}_0] = \Pr[\text{Correct guess with real session key}]$$

3. **Game 1 (Hash Query Limit)**

   We simulate all hash function queries. If the adversary finds a collision or inversion in the hash function, the game ends. The advantage is bounded by:

$$|\Pr[\text{GM}_0] - \Pr[\text{GM}_1]| \leq \frac{q_h^2}{|H|}$$

   where q_h is the number of hash queries and $|H|$ is the hash output space.

4. **Game 2 (Message Forgery)**

   We simulate the adversary forging a valid session key or message. This involves computing:

$$SK = h(N_A \| N_{hg} \| MIID_a \| ID_{hg} \| TK)$$

Without access to N_A, N_{hg}, or TK, the adversary cannot forge SK due to the collision resistance of $h(\cdot)$. The advantage of success is negligible.

$$|\Pr[\mathrm{GM}_1] - \Pr[\mathrm{GM}_2]| \leq \frac{q_s}{2^n}$$

Where q_s is the number of sent queries and n is the key length in bits.

5. **Game 3 (Dictionary Attack)**

   Assuming low-entropy passwords, the adversary might attempt to guess the password using offline dictionary attacks. The success probability is bounded by:

$$|\Pr[\mathrm{GM}_2] - \Pr[\mathrm{GM}_3]| \leq \frac{q_s \cdot q_h}{|D|}$$

   where $|D|$ is the dictionary size.

6. **Game 4 (Corruption and Key Reveal)**

   Even if the adversary corrupts specific sessions or reveals keys, they cannot infer a fresh session key unless they corrupt both participants involved in the session. Thus:

$$\Pr[\mathrm{GM}_4] \leq \frac{1}{2}$$

7. **Overall Advantage Bound**

   Combining all the above games, the final advantage of the adversary is:

$$Adv^{\mathrm{ROR}}_{\mathrm{SUMAS}}(J) \leq \frac{q_h^2}{|H|} + \frac{q_s}{2^n} + \frac{q_s \cdot q_h}{|D|}$$

This demonstrates that the **SUMAS** scheme provides strong session key indistinguishability and withstands known cryptographic attacks under the ROR model assumptions.

5.7.1 Formal Security Verification Using AVISPA Tool

To enhance the validation of the robustness of the SUMAS protocol, we employ the AVISPA framework. AVISPA is a well-established toolset that provides formal verification for various security properties, including confidentiality, authentication, and resilience against attacks [98].

The protocol is modelled using the High-Level Protocol Specification Language (HLPSL), which outlines the roles, transitions, and communication flows among the involved entities: the User, Home Gateway (HG), and Smart Device. Each participant in the protocol is characterised by their local variables, states, and the specific security goals they are expected to achieve.

```
%%%%%Role for A%%%%
Role role_A (A, HG: agent,
        Hash:hash_func,
        K:symmetric_key,
        SK:symmetric_key,
      SND, RCV: channel (dy))
Played by A def =
Local State: Nat,
MIIDi, IDg, Ai, Bi, Alpha, T1, T2, X, Y: text,
VI,V2,CIDi,TK,C1,C2: message,
Hash:hash_func
const
deviceA_hg_Ra, hg_deviceA_Rg,
device_hg_MIIDi,
hg_deviceA_IDg:protocol_id,
deviceA_hg_T1, hg_deviceA_T2, sub1, sub2,
sub3:protocol_id
Init State: = 0
transition
1. State=0/\RCV (start) =|>
State':=1/\Ra':=new ()
/\T1':= new ()
/\V1':=xor (Hash (IDg.Hash(x)), xor (Ra', T1'))
/\CIDi':=xor (Bi, Hash (Hash
(IDi.Hash(x)).Ra'.T1'))
/\TK':=xor (Hash (Ai), Ra')
/\C1':={ MIIDi.IDg.N.Alapha.T1'} _TK'
/\SND (Vi, CIDi', CI', T1')
/\secret ({MIIDi, IDg}, sub1, {A, HG})
/\secret ({X, Y}, sub2, {A, HG})
/\secret (Alpha, sub2, {A, HG})
/\witness (A, HG, deviceA_hg_T1, T1')
%device A has freshly generated the value of
T1 for HG
/\Witness (A, HG, deviceA_hg_Ra, Ra')
%device has freshly generated the value of
T1 for HG
2. State= 3/\ RCV
({IDg.T1.T2'}_SK,xor(Rg',xor(Bi,xor(Hash(Ai,T
2')))),T2')=|>
State':=5 /\Ra':=new()
/\T2':= new()
/\V1':=xor(Hash(IDg.Hash(x)),xor(Ra',T2'))
/\CIDi':=xor(Bi,Hash(Hash(MIIDi.Hash(x)).Ra'.
T2'))
/\TK':=xor(Hash(Ai),Ra')
/\C1':={ MIIDi.IDg.N.Alpha.T2'} _TK'
/\SND(Vi,CIDi',CI',T2')
/\secret({MIIDi,IDg},sub1,{A,HG})
/\secret ({X, Y}, sub2, {A, HG})
/\secret (Alpha, sub2, {A, HG})
/\witness (A, HG, deviceA_hg_T2, T2')
%device A has freshly generated the value of
T2 for HG
/\Witness (A, HG, deviceA_hg_Ra, Ra')
%device A has freshly generated the value of
T2 for HG
end role
```

FIGURE 5.1: Role for user

The roles are then composed in the HLPSL environment module to simulate the interactions, forming a complete session scenario. These sessions include:

- User Role: Initiates communication by sending the identity and hashed credentials shown in Figure 5.1.
- HG Role: Verifies identities, timestamps, and generates session keys.

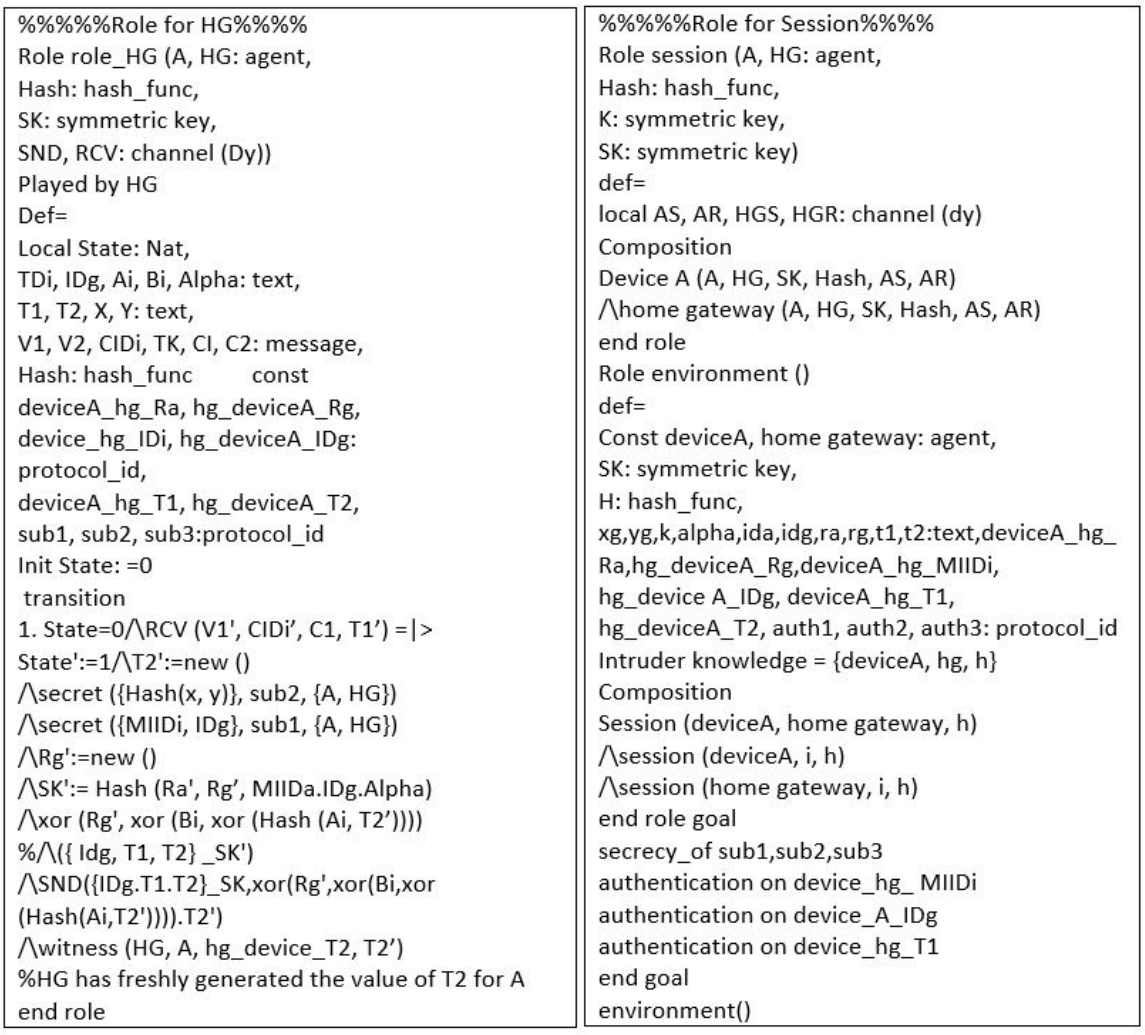


FIGURE 5.2: Role of Home Gateway and session

- Session Role: Models the interaction between User and HG based on the message flow shown in Figure 5.2.

AVISPA provides a suite of four back-end analysis tools: OFMC, CL-AtSe, SATMC, and TA4SP. In our evaluation, we specifically used the On-the-Fly Model Checker (OFMC) [98]. This powerful tool is designed to efficiently explore the state spaces of various protocols, allowing for a systematic assessment of their behaviour. Its main strength is its capacity to detect security violations in real-time, thus helping to assure the integrity as well as the safety of the protocols under examination.

The SUMAS scheme was verified against various attack models, and the simulation results confirmed that the protocol is SAFE in all modeled threat scenarios. Specifically, OFMC validated the following aspects.

FIGURE 5.3: AVISPA result using OFMC

- **Secrecy of session keys:** Session keys are not accessible to unauthorized parties.
- **Authentication:** Both mutual and one-way authentication requirements are satisfied.
- **Replay Resistance:** Messages with expired timestamps are discarded.
- **MITM Resistance:** Any attempt to alter or forge messages results in protocol termination.

The simulation was conducted with a depth of four plies and examined sixteen protocol states. The OFMC analysis was completed in approximately 0.04 seconds, thereby demonstrating the protocol's efficiency and scalability, as illustrated in Figure 5.3.

5.8 Performance Evaluation

This section presents a comprehensive performance evaluation of the proposed SUMAS scheme, comparing it with several well-established authentication methods recognised for their effectiveness. We examine the resilience of the protocol against various security threats, as outlined in Table 5.5. In this detailed analysis, we emphasise the practical benefits of the SUMAS scheme, especially in resource-limited settings like

smart homes and IoT networks, where effective and secure authentication is crucial for optimal performance.

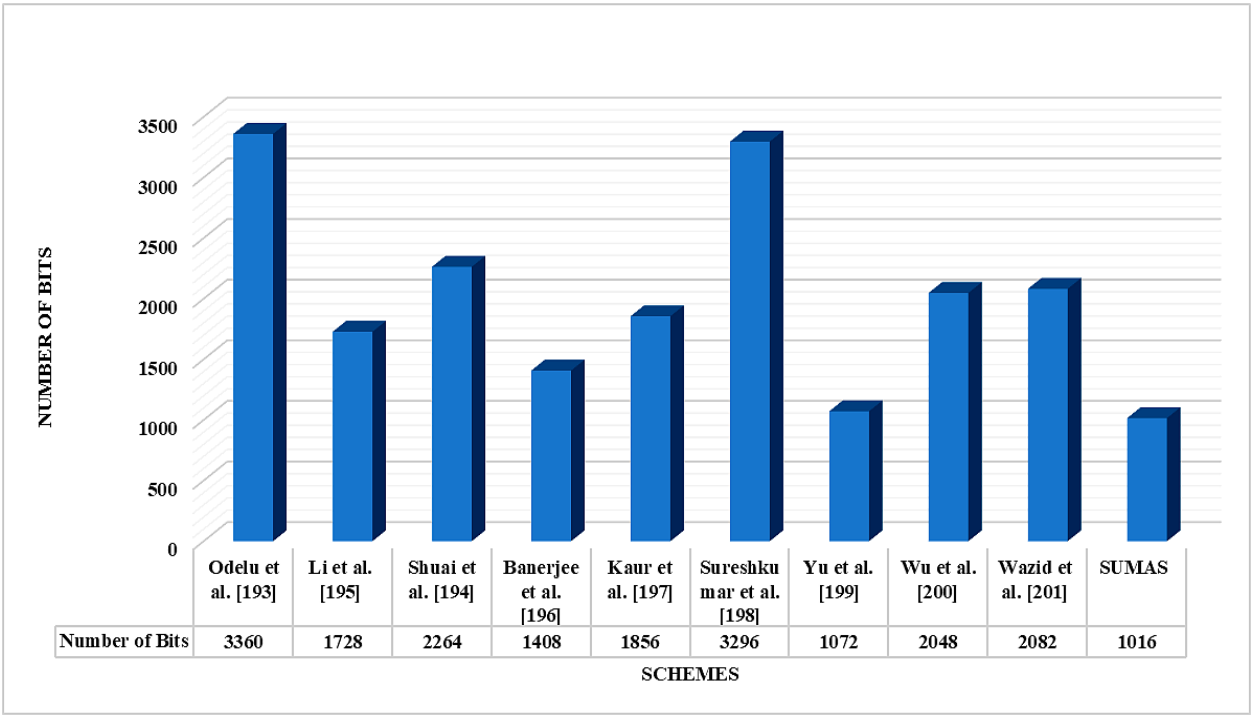


FIGURE 5.4: Comparison of Communication cost

5.8.1 Communication Cost Analysis

The communication cost refers to the total number of bits exchanged during the execution of the authentication and key-establishing phases. In the proposed SUMAS scheme, there are four primary message exchanges between the user device and the home gateway:

- **Message 1:** User sends their identity and hashed password: 176 bits
- **Message 2:** User transmits login request data: 16 bits
- **Message 3:** Device sends multiple authentication values including V_1, CID_i, C_1, T_1, ID_i, and $MIID_a$: 448 bits
- **Message 4:** Gateway responds with session key data SK, encrypted message C_2, value V_2, timestamp T_2, and random nonce N_{hg}: 376 bits

The total communication overhead is the sum of these message sizes:

$$\text{Total Communication Cost} = 176 + 16 + 448 + 376 = \textbf{1016 bits}$$

Compared to the other schemes shown in Figure 5.4, the SUMAS scheme has a lower communication overhead, making it ideal for bandwidth-sensitive environments.

5.8.2 Computational Cost Analysis

The computational cost includes the number and type of cryptographic operations required during the protocol execution. The SUMAS scheme uses lightweight symmetric operations, optimized for constrained devices. The total computational operations involved are:

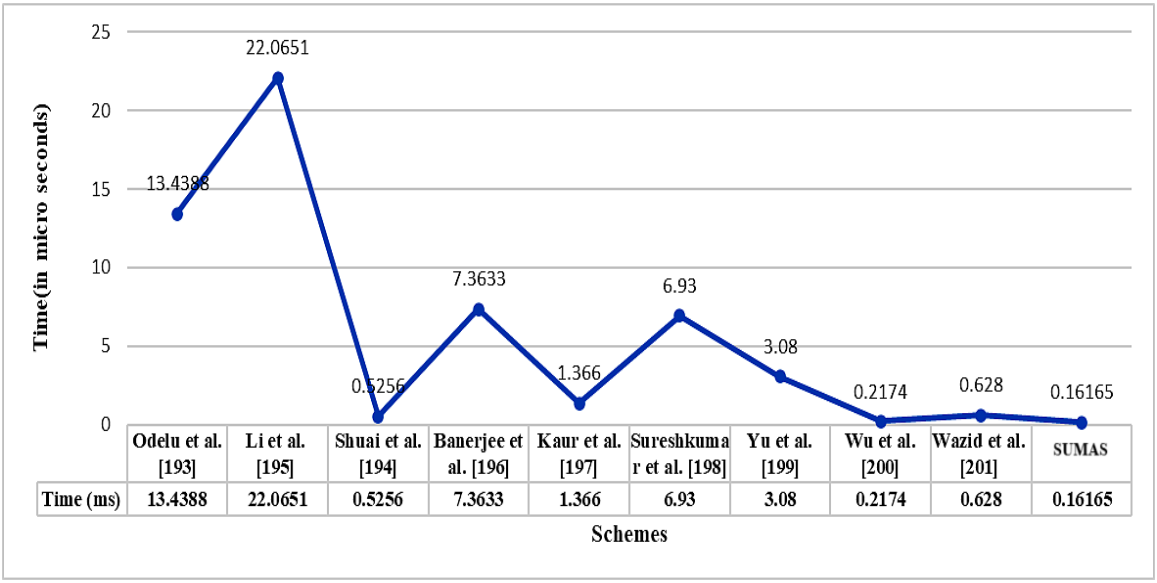


FIGURE 5.5: Comparison of Computation cost

- 7 Hash operations (T_h)
- 25 XOR operations
- 1 Encryption operation (T_{enc})
- 1 Decryption operation (T_{dec})

Assuming benchmark values:

- $T_h = 39$ ms
- $T_{enc} = 3.5$ ms
- $T_{dec} = 41.15$ ms
- XOR operations = 106 ps (negligible)

The estimated cumulative execution time for the SUMAS scheme is approximately 0.16 milliseconds, which is significantly faster than traditional ECC or biometric-based protocols.

The evaluation demonstrates that the proposed SUMAS scheme effectively balances security and efficiency as shown in Figure 5.5. With a communication overhead of 1,016 bits and an average computation time of approximately 0.16 milliseconds, it significantly outperforms other schemes that depend heavily on public-key cryptography or complex biometric computations.

TABLE 5.5: Comparison of Functionality and Security Attributes

Attacks / Scheme	**MAA**	**DIA**	**AA**	**UA**	**RA**	**MITM**	**DoSA**
Odelu et al. [191]	✓	✓	✓	–	✓	×	✓
Shuai et al. [192]	✓	✓	×	✓	✓	×	✓
Li et al.[193]	×	✓	✓	✓	✓	✓	✓
Banerjee et al. [194]	✓	✓	×	×	✓	×	✓
Kaur et al. [195]	✓	–	✓	✓	✓	–	–
Suresh et al. [196]	✓	✓	×	✓	✓	–	✓
Wu et al. [198]	✓	✓	✓	✓	✓	✓	–
Yu et al. [197]	✓	✓	✓	–	✓	×	–
Wazid et al. [199]	×	×	×	–	✓	×	×
Our Scheme (SUMAS)	✓	✓	✓	✓	✓	✓	✓

5.9 Summary

The proposed SUMAS (Secure and Unique Mutual Authentication Scheme) addresses several limitations found in existing IoT authentication protocols, such as inefficient

identity verification, high communication overhead, and vulnerability to impersonation and replay attacks. By integrating unique user and device identifiers directly into a modified IPv6 address format, SUMAS facilitates secure addressing and efficient mutual authentication without increasing the size of the packet header.

The protocol uses lightweight cryptographic operations—including SHA-3 hash, XOR functions, and symmetric encryption techniques—to suit the resource limits of IoT devices. By means of both informal analysis and formal validation techniques including simulations under the Dolev-Yao adversarial model and analysis utilising the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool, its security robustness has been rigorously verified. These tests demonstrate the scheme's resistance to various attacks, including man-in-the-middle (MITM) attacks, replay attacks, anonymity breaches, and traceability concerns.

Performance criteria show that SUMAS is more efficient than current standards. With an execution time of just 0.16 milliseconds and a minimal communication overhead of 1,016 bits, it Furthermore quite appropriate for use in developing smart home environments is the scheme's provision of crucial security features including password updates, session key establishment, and dynamic device authentication.

Looking ahead, the SUMAS framework will be enhanced by incorporating context-aware trust evaluation and machine learning-based anomaly detection. Future work will also focus on evaluating its real-world implementation, energy efficiency, and adaptability in dynamic and heterogeneous IoT ecosystems.

CHAPTER - 6
CONCLUSIONS AND FUTURE RESEARCH SCOPE

6.1 Conclusion and Future Work

This research examines the primary challenges of secure authentication, unique addressing, and lightweight protection mechanisms in IoT networks, with a focus on smart home environments. Three proposed frameworks—AUSS, SLAPSH, and SUMAS—aim to address various limitations of traditional schemes. These limitations include difficulties in handling impersonation attacks, message replay, threats from stolen devices, and inefficient addressing in environments with multiple devices.

The AUSS (Authenticated Unidentified Security Scheme) proposes a robust three-factor authentication approach using passwords, biometrics, and mobile devices. It ensures user anonymity, unlinkability, secure session key generation, and resistance to known attacks while remaining suitable for resource-constrained IoT environments. The scheme's strength was validated via BAN logic, AVISPA simulation, and comparative performance evaluation, confirming its suitability for real-time IoT applications.

The Secure and Lightweight Authenticated Protocol for Smart Home (SLAPSH) scheme is an innovative approach to security in smart home ecosystems. SLAPSH effectively projects unique device identities directly into the expansive address space by innovatively altering the traditional IPv6 addressing structure. This strategic improvement not only simplifies the entire process by eliminating the necessity for additional communication overhead, but also enables secure and seamless device authentication. In simulated network environments using NS-3, SLAPSH demonstrates exceptional resilience against a variety of security threats by utilising sophisticated formal verification tools such as AVISPA and ROR models, all while ensuring low latency and high throughput.

By improving security within IPv6-based addresses, the SUMAS system—also known as Secure and Unique Mutual Authentication system—complements the SLAPSH technique. It does this by including within a 64-bit modified interface identification (MIID) both device IDs and user identities. While keeping low data packet size during transmission, this architecture allows strong mutual authentication between smart appliances and the home gateway. SUMAS guards against major security risks including MIMA attacks, message forging, and device impersonation rather successfully. The protocol is a quite effective approach since it not only increases security coverage but also maximises computing expenses.

Particularly in smart home situations, the SLAPSH and SUMAS protocols used together build a strong security architecture especially for IoT systems. They guarantee best network performance and a balanced approach that preserves user privacy by means of robust authentication systems. Eventually, this mix produces for users a more dependable and safer smart home experience.

LIST OF PUBLICATIONS

Journal Publications:

1. **Sharma, N.,** Dhiman, P., "A Survey on IoT Security: Challenges and Their Solutions Using Machine Learning and Blockchain Technology", *Cluster Computing*, Springer, 2025. (DOI: 10.1007/s10586-023-05208-0) **[SCIE, Impact Factor: 3.6]**

2. **Sharma, N.,** Dhiman, P., "Design of a Multifactor Unidentified Remote End User Authentication Mechanism for IoT Network", *Informatica*, 49(10), pp. 191–204. (DOI: 10.31449/inf.v49i10.5518) [**Scopus Indexing**]

3. **Sharma, N.,** Dhiman, P., "A Secure Addressing Mutual Authentication Scheme for Smart IoT Home Networks", *Multimedia Tools and Applications*, Springer, 2024. (DOI: 10.1007/s11042-024-19898-y) **[SCI, Impact Factor: 3.0]**

4. **Sharma, N.,** Dhiman, P., "Design of Secure and Unique Addressing with Mutual Authentication Scheme in IoT Networks", *Peer to Peer Networking*, Springer, 2024. (DOI: 10.1007/s12083-024-01882-w) **[SCIE, Impact Factor: 3.3]**

Conference Publications:

1. **Sharma, N.**, Dhiman, P., "Secure Authentication Scheme for IoT-enabled Smart Homes," in *Emergent Converging Technologies and Biomedical Systems*, Springer (ETBS 2023), pp. 611–618.

2. **Sharma, N.**, Dhiman, P., "Design and Analysis of Authentication in IoT based Smart Homes," in *Seventh International Conference on Image Information Processing (ICIIP 2023) IEEE*, JUIT, Waknaghat, Solan (H.P).

3. **Sharma, N.**, Dhiman, P., "Privacy in smart homes with remote user authenticated key establishment protocol," in *Procedia Computer Science*, (2024), pp. 119-128.

Published Book Chapters

1. **Sharma, N.,** Dhiman, P., "Fingerprint Security and the Internet of Things (IoT) in the Digital Era," in *Leveraging Computer Vision to Biometric Applications, CRC Press Taylor and Francis 2024*.

2. **Sharma, N.,** Dhiman, P., "Industrial Internet of Things Security Architecture and Protection Techniques," in *Industrial Internet of Things Security: Protecting AI-Enabled Engineering Systems in Cloud and Edge Environments, CRC Press Taylor and Francis 2024*.

References

[1] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," IEEE Commun. Surveys Tuts., vol. 17, no. 3, pp. 1294–1312, 2015.

[2] D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," San Jose, CA, USA, Cisco Internet Business Solutions Group, White Paper, Apr. 2011. [Online]. Available: `https://www.cisco.com/`

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," IEEE Internet Things Journal, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[4] J. Lin, Y. Wei, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: Architecture, enabling technologies, security privacy, and applications," IEEE Internet Things Journal, vol. 4, no. 5, pp. 1125–1142, 2017.

[5] Z. Huang, L. Zhang, X. Meng, and K.-K. R. Choo, "Key-free authentication protocol against subverted indoor smart devices for smart home," IEEE Internet Things Journal, vol. 7, no. 2, pp. 1039–1047, 2019.

[6] B. H. Taher, H. Liu, F. Abedi, H. Lu, A. A. Yassin, and A. J. Mohammed, "A secure and lightweight three-factor remote user authentication protocol for future IoT applications," J. Sensors, vol. 2021, pp. 1–18, 2021.

[7] F. M. Umar, W. Muhammad, M. Sadia, K. Anjum, and K. Talha, "A review on Internet of Things (IoT)," Int. J. Comput. Appl., vol. 113, no. 1, pp. 1–7, 2015.

[8] T. Venkat Narayana Rao, Shaik Khasim Saheb, and A. Janiki Ram Reddy, "Design of the architecture for efficient integration of Internet of Things and cloud computing," Int. J. Adv. Res. Comput. Sci., vol. 8, pp. 392–396, 2017.

[9] K. Shapla, A. I. Bin, I. I. M. Yamani, J. M. Hisham, and M. S. A. Q. Bin, "A survey of security challenges, attacks taxonomy and advanced countermeasures in the Internet of Things," IEEE Access, vol. 8, pp. 219709–219743, 2020.

[10] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain," J. Inf. Secur. Appl., vol. 45, pp. 156–175, 2019.

[11] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IoT) technologies, applications and challenges," in Proc. IEEE Smart Energy Grid Eng. (SEGE), 2016, pp. 381–385.

[12] R. Soderbery, "How many things are currently connected to the 'Internet of Things (IoT)?" Forbes, Forbes Media, Jersey City, NJ, 2013.

[13] Muhammad Waqas, Kashif Kumar, Aijaz Ali Laghari, Usman Saeed, Mirza Munawar Rind, Abid Ali Shaikh, Farhan Hussain, Ameer Rai, and Abdul Qadir Qazi, "Botnet attack detection in Internet of Things devices over cloud environment via machine learning," Concurrency Computation Practice. Exper., vol. 34, no. 4, p. e6662, 2022.

[14] Dragos Mocrii, Yin Chen, and Petros Musilek, "IoT-based smart homes: a review of system architecture, software, communications, privacy and security," Internet Things, vol. 1, pp. 81–98, 2018.

[15] Muhammad Ajmal, Muhammad Hameed Ashraf, Muhammad Shakir, Yousaf Abbas, and Farman Ali Shah, "Video summarization: techniques and classification," in Proc. Int. Conf. Comput. Vision Graph. Berlin, Germany: Springer, Sep. 2012, pp. 1–13.

[16] Zareen Fatima, Abdul Ullah Rehman, Rizwan Hussain, Saqib Karim, Muhammad Shakir, Kashif Ali Soomro, and Aijaz Ali Laghari, "Mobile crowdsensing with energy efficiency to control road congestion in internet cloud of vehicles: a review," Multimedia Tools Appl, pp. 1–26, 2023.

[17] Shun-Feng Tzeng, Shi-Jinn Horng, Tzong-Wann Li, Xiaohua Wang, Po-Hsuan Huang, and Muhammad Khurram Khan, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," IEEE Trans. Veh. Technology vol. 66, no. 4, pp. 3235–3248, Apr. 2015.

[18] Jingwei Li, Yushu Zhang, Xiaofeng Chen, and Yang Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[19] Christos Stergiou, Kostas E. Psannis, Byung-Gyu Kim, and Brij B. Gupta, "Secure integration of IoT and cloud computing," Future Gener. Comput. System, vol. 78, pp. 964–975, 2018.

[20] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash, "Security analysis of emerging smart home applications," in Proc. IEEE Symp. Security Privacy (SP), 2016, pp. 636–654.

[21] Md Mahmudul Hossain, Mohammad Fotouhi, and Rameez Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in Proc. IEEE World Congr. Services, Jun. 2015, pp. 21–28.

[22] Kai Zhao and Lina Ge, "A survey on the Internet of Things security," in Proc. 9th International Conference on Computing Intelligence. Security, Dec. 2013, pp. 663–667.

[23] Zhiwei Yan, Peng Zhang, and Athanasios V. Vasilakos, "A survey on trust management for Internet of Things," Journal of Network Computing. Appl., vol. 42, pp. 120–134, Jun. 2014.

[24] Mouhamed Ammar, Giancarlo Russello, and Bruno Crispo, "Internet of Things: A survey on the security of IoT frameworks," Journal of Information Security Applications, vol. 38, pp. 8–27, Feb. 2018.

[25] Michele Frustaci, Pasquale Pace, Giuseppe Aloi, and Gianluca Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," IEEE Internet Things Journal, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[26] Wenqi Zhou, Yifei Jia, Anping Peng, Yilei Zhang, and Ping Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," IEEE Internet Things Journal, vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[27] Peter Fremantle and Paul Scott, "A survey of secure middleware for the Internet of Things," Peer Journal Computer Science, vol. 3, p. e114, May 2017.

[28] Rolf H. Weber, “Internet of Things—New security and privacy challenges,” Comput. Law Security Revocation, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[29] Cristiano Pielli, Davide Zucchetto, Andrea Zanella, Lorenzo Vangelista, and Michele Zorzi, “Platforms and protocols for the Internet of Things,” EAI Endorsed Trans. Internet Things, vol. 15, no. 1, p. e5, Oct. 2015.

[30] Ankit Singh, Nitin Chawla, Jung Hyun Ko, Manabendra Kar, and Subhas Mukhopadhyay, “Energy efficient and side-channel secure cryptographic hardware for IoT-edge nodes,” IEEE Internet Things Journal, vol. 6, no. 1, pp. 421–434, Feb. 2019.

[31] Amir Mosenia and Niraj K. Jha, “A Comprehensive Study of Security in the Internet of Things,” IEEE Transactions Emerg. Topics Computation, vol. 5, no. 4, pp. 586–602, Oct./Dec. 2017.

[32] Eran Ronen and Adi Shamir, “Extended functionality attacks on IoT devices: The case of smart lights,” in Proc. IEEE Eur. Symp. Security Privacy, Mar. 2016, pp. 3–12.

[33] William Stallings, Cryptography and Network Security: Principles and Practice, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2014.

[34] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman, “Cryptographic communications system and method,” U.S. Patent US4 405 829 A, Sep. 20, 1983.

[35] Information Technology Laboratory, “Lightweight Cryptography Project,” NIST, 2019. [Online]. Available: https://csrc.nist.gov/projects/lightweight-cryptography (accessed Jun. 2019).

[36] Bilel Halak, Michal Zwolinski, and Mehdi S. Mispan, “Overview of PUF-based hardware security solutions for the Internet of Things,” in Proc. IEEE 59th Int. Midwest Symp. Circuits Syst, Oct. 2016, pp. 1–4.

[37] Oscar Arias, Jens Wurm, Khoa Hoang, and Yier Jin, “Privacy and security in Internet of Things and wearable devices,” IEEE Trans. Multi-Scale Comput. Syst., vol. 1, no. 2, pp. 99–109, Apr./Jun. 2015.

[38] Abdelhakim Rghioui, Abdelkrim Khannous, and Mohammed Bouhorma, “Denial-of-service attacks on 6LoWPAN-RPL networks: Issues and practical solutions,” Journal of Advanced Computing Science Technology, vol. 3, no. 2, pp. 143–153, 2014.

[39] Antoine Mayzaud, Romain Badonnel, and Isabelle Chrisment, "A taxonomy of attacks in RPL-based Internet of Things," Int. J. Netw. Security, vol. 18, no. 3, pp. 459–473, May 2016.

[40] Yehuda Y. Lindell, "Comparison-based key exchange and the security of the numeric comparison mode in Bluetooth v2.1," in Proc. Cryptograph. Track RSA Conf., Apr. 2009, pp. 66–83.

[41] R. Snader, R. Kravets, and A. F. Harris, III, "CryptoCoP: Lightweight, energy-efficient encryption and privacy for wearable devices," in Proc. Workshop Wearable Syst. Appl., Jun. 2016, pp. 7–12.

[42] Luca Coppolino, Vincenzo D'Alessandro, Salvatore D'Antonio, Lionel Levy, and Luigi Romano, "My Smart Home Is Under Attack," in Proc. IEEE 18th Int. Conf. Comput. Sci. Eng., Oct. 2015, pp. 145–151.

[43] Feng Li, Yanjun Shi, Amarsagar Shinde, Jun Ye, and Weizhao Z. Song, "Enhanced cyber-physical security in Internet of Things through energy auditing," IEEE Internet Things Journal, vol. 6, no. 3, pp. 5224–5231, Feb. 2019.

[44] Hedi Sedjelmaci, Said M. Senouci, and Tarik Taleb, "An accurate security game for low-resource IoT devices," IEEE Trans. Veh. Technology, vol. 66, no. 10, pp. 9381–9393, Oct. 2017.

[45] Jie Li, Zhipeng Zhao, Rui Li, and Honggang Zhang, "AI-based two-stage intrusion detection for software-defined IoT networks," IEEE Internet Things Journal, vol. 6, no. 2, pp. 2093–2102, Apr. 2019.

[46] Gianluca Caparra, Marco Centenaro, Nicola Laurenti, Stefano Tomasin, and Lorenzo Vangelista, "Wireless physical-layer authentication for the Internet of Things," in Proc. Glob. Internet Things Summit, Jun. 2017, pp. 390–417.

[47] Carlo Medaglia and Andrea Serbanati, "An overview of privacy and security issues in the Internet of Things," in The Internet of Things, Springer, New York, NY, USA, 2010, pp. 389–395.

[48] Rolf Weber, "Internet of Things—New security and privacy challenges," Comput. Law Security Revisions, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[49] IBM Corporation, "How modern enterprises are using IoT data to spur innovation," IBM, 2025. [Online]. Available: `https://www.ibm.com/think/insights/how-modern-enterprises-are-using-iot-data-to-spur-innovation`

[50] Xiangqian Chen, Kai Makki, Kang Yen, and Niki Pissinou, "Sensor network security: A survey," IEEE Commun. Surveys Tuts., vol. 11, no. 2, pp. 52–73, 2009.

[51] Albert C. Jose and Reza Malekian, "Improving smart home security: Integrating logical sensing into the smart home," IEEE Sensors Journal, vol. 17, no. 13, pp. 4269–4286, Jul. 2017.

[52] Tariq A. Ahanger, Abdullah Aljumah, and Mohammed Atiquzzaman, "State-of-the-art survey of artificial intelligence techniques for IoT security," Computer Network, vol. 206, pp. 1–56, 2022.

[53] Yue Yang, Li Wu, Guangjie Yin, Lihua Li, and Hong Zhao, "A survey on security and privacy issues in Internet-of-Things," IEEE Internet Things Journal, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[54] Amir Mosenia and Niraj K. Jha, "A comprehensive study of the security of Internet-of-Things," IEEE Trans. Emerg. Topics Comput., vol. 5, no. 4, pp. 586–602, Dec. 2016.

[55] Anh H. Ngu, Mario Gutierrez, Vassilis Metsis, Surya Nepal, and Quan Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," IEEE Internet Things Journal, vol. 4, no. 1, pp. 1–20, Feb. 2016.

[56] Iqbal U. Din, Mohsen Guizani, Byeong-Seok Kim, Syed Hassan, and Muhammad Khurram Khan, "Trust management techniques for the Internet of Things: A survey," IEEE Access, vol. 7, pp. 29763–29787, 2018.

[57] Md Mahmudul Hossain, Mohammad Fotouhi, and Rameez Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in Proc. IEEE World Congr. Services, Jun. 2015, pp. 21–28.

[58] Ruben Román-Castro, Javier López, and Stelios Gritzalis, "Evolution and trends in IoT security," Computer, vol. 51, no. 7, pp. 16–25, 2018.

[59] Irene Andrea, Chrysostomos Chrysostomou, and George Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in Proc. IEEE Symposium on Computers and Communications (ISCC), Jul. 2015, pp. 180–187.

[60] Filippo Meneghello, Michele Calore, Davide Zucchetto, Michele Polese, and Andrea Zanella, "IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices," IEEE Internet Things Journal, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.

[61] Xiaoxu Su, Zhiyong Wang, Xiaosong Liu, Cheol-Ho Choi, and Dong-Uk Choi, "Study to improve security for IoT smart device controller: Drawbacks and countermeasures," Secure Communication Network, vol. 2018, pp. 1–15, May 2018.

[62] Anthony Dean and Michael O. Agyeman, "A study of the advances in IoT security," in Proc. 2nd Int. Symp. Comput. Sci. Intell. Control, Sep. 2018, pp. 1–6.

[63] Haoyu Si, Chunlei Sun, Yu Li, Hao Qiao, and Liang Shi, "IoT information sharing security mechanism based on blockchain technology," Future Gener. Computer Systems, vol. 101, pp. 1028–1040, Dec. 2019.

[64] Jing Lin, Wei Yu, Nan Zhang, Xiao Yang, Haibo Zhang, and Wei Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet Things Journal, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[65] Kai Liu, Weisheng Shen, Yong Cheng, Liang X. Cai, Qing Li, Sheng Zhou, and Zhisheng Niu, "Security analysis of mobile device-to-device network applications," IEEE Internet Things Journal, vol. 6, no. 2, pp. 2922–2932, Apr. 2018.

[66] Mohamed Amine Ferrag, Leandros Maglaras, and Abdelouahid Derhab, "Authentication and authorization for mobile IoT devices using biofeatures: Recent advances and future trends," Secure Communication Network, vol. 2019, pp. 1–21, May 2019.

[67] Mohammad A. Khan and Khaled Salah, "IoT security: Review, blockchain solutions, and open challenges," Future Gener. Comput. Systems, vol. 82, pp. 395–411, May 2018.

[68] Wen Jun, Meng Lei, and Zhi Luo, "Data security mechanism based on hierarchy analysis for Internet of Things," in Proc. Int. Conf. Innov. Comput. Cloud Comput. (ICCC), 2011, pp. 68–70.

[69] George George and Sabu M. Thampi, "Vulnerability-based risk assessment and mitigation strategies for edge devices in the Internet of Things," Pervas. Mobile Computer, vol. 59, Oct. 2019, Art. no. 101068.

[70] Arshdeep Tewari and B. B. Gupta, "Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework," Future Gener. Comput. Systems, vol. 108, pp. 909–920, Jul. 2020.

[71] Fabio Loi, Aswin Sivanathan, Hassan Habibi Gharakheili, Adam Radford, and Vijay Sivaraman, "Systematically evaluating security and privacy for consumer IoT devices," in Proc. Workshop on Internet of Things Security. Privacy, Nov. 2017, pp. 1–6.

[72] Maqsood Ali, Ranjan Dhamotharan, Ehsan Khan, Sherali U. Khan, Athanasios V. Vasilakos, Keqin Li, and Albert Y. Zomaya, "SeDaSC: Secure data sharing in clouds," IEEE System Journal, vol. 11, no. 2, pp. 395–404, 2015.

[73] Kristina Popovic and Zeljko Hocenski, "Cloud computing security issues and challenges," in Proc. 33rd Int. Conv. Inf. Commun. Technol., Electron. Microelectron, IEEE, 2010, pp. 344–349.

[74] Kui Ren, Cong Wang, and Qian Wang, "Security challenges for the public cloud," IEEE Internet Computation, pp. 1–73, 2012.

[75] Benjamin Fung, Ke Wang, Rui Chen, and Philip Yu, "Privacy-preserving data publishing: A survey of recent developments," ACM Comput. Surveys, vol. 42, pp. 1–53, 2010.

[76] M. F. Aziz, A. N. Khan, J. Shuja, I. A. Khan, F. G. Khan, and A. R. Khan, "A lightweight and compromise-resilient authentication scheme for IoTs," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, p. e3813, 2022.

[77] Dan Bogdanov, Sven Laur, and Jan Willemson, "Sharemind: A framework for fast privacy-preserving computations," in Proc. ESORICS, Malaga, Spain, Oct. 2008, vol. 13, pp. 192–206, Springer Berlin Heidelberg.

[78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, pp. 120–126, 1978.

[79] Vincent Rijmen and Joan Daemen, "Advanced encryption standard," in Proc. Federal Inf. Process. Standards Publications, NIST, 2001, p. 22.

[80] Adi Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology: Proc. CRYPTO'84, Springer Berlin Heidelberg, 1985, pp. 47–53.

[81] John Bethencourt, Amit Sahai, and Brent Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symp. Secur. Privacy (SP'07), 2007, pp. 321–334.

[82] Matt Blaze, Gerrit Bleumer, and Martin Strauss, "Divertible protocols and atomic proxy cryptography," in Proc. Security Protocols Workshop, Springer, Berlin/Heidelberg, Germany, 1998, pp. 127–144.

[83] Jiangtao Liu, Yang Xiang, Wanlei Zhou, Xuemin Huang, and Jian Ma, "Data authentication with privacy protection," in Advances in Cyber Security: Principles, Techniques, and Applications, 2019, pp. 115–142.

[84] A. Gupta, M. Tripathi, S. Muhuri, G. Singal, and N. Kumar, "A secure and lightweight anonymous mutual authentication scheme for wearable devices in medical Internet of Things," Journal Information Secure Applications, vol. 68, p. 103259, 2022.

[85] Z. Huang, L. Zhang, X. Meng, and K.-K. R. Choo, "Key-free authentication protocol against subverted indoor smart devices for smart home," IEEE Internet Things Journal, vol. 7, no. 2, pp. 1039–1047, 2019.

[86] Jian Cui, Xiaohua Tao, Jing Zhang, Yong Xu, and Hong Zhong, "HCPA-GKA: A hash function-based conditional privacy-preserving authentication and group-key agreement scheme for VANETs," Vehicular Communication, vol. 14, pp. 15–25, 2018.

[87] Lei Zhou, Xiaodong Li, Kuo-Hui Yeh, Chia-Hui Su, and Wei Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," Future Generation. Computer Systems, vol. 91, pp. 244–251, 2019.

[88] Bassem Hammi, Abderrahmane Fayad, Rachid Khatoun, and Sherali Zeadally, "A lightweight ECC-based authentication scheme for Internet of Things (IoT)," IEEE System Journal, vol. 14, no. 3, pp. 3440–3450, 2020.

[89] Abdullah A. Alamr, Faheem Kausar, Jin Kim, and Changhoon Seo, "A secure ECC-based RFID mutual authentication protocol for Internet of Things," *J. Supercomput.*, vol. 74, no. 9, pp. 4281–4294, 2018.

[90] Shashank Kalra and Sandeep K. Sood, "Secure authentication scheme for IoT and cloud servers," *Pervasive Mobile Comput.*, vol. 24, pp. 210–223, 2015.

[91] Maxim Raya and Jean-Pierre Hubaux, "Securing vehicular ad hoc networks," Journal of Computing Secure, vol. 15, no. 1, pp. 39–68, 2007.

[92] Yatindra Kondareddy, Giovanni Di Crescenzo, and Prathima Agrawal, "Analysis of certificate revocation list distribution protocols for vehicular networks," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–5.

[93] Mohammed Azees, P. Vijayakumar, and L. Jino Deboarh, "EAAP: Efficient anonymous authentication with a conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 1–10, 2017.

[94] Chin-Chen Chang, Hui-Ling Wu, and Chin-Yu Sun, "Notes on 'Secure authentication scheme for IoT and cloud servers'," *Pervasive Mobile Comput.*, vol. 38, pp. 275–278, 2017.

[95] Hsiang-Ming Chen, Jen-Wei Lo, and Chin-Kuan Yeh, "An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems," *J. Med. Syst.*, vol. 36, pp. 3907–3915, 2012.

[96] Danny Dolev and Andrew Yao, "On the security of public key protocols," IEEE Trans. Inf. Theory, vol. 29, no. 2, pp. 198–208, 1983. doi: 10.1109/TIT.1983.1056650.

[97] Alessandro Armando et al., "The AVISPA Tool for the automated validation of Internet security protocols and applications," in *CAV 2005, LNCS 3576*, Springer, 2005, pp. 281–285.

[98] AVISPA Team, "AVISPA: A tool for the automated validation of Internet security protocols and applications—User manual," 2006. [Online]. Available: `http://www.avispa-project.org`.

[99] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, Feb. 1990. [Online]. Available: `https://doi.org/10.1145/77648.77649`.

[100] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS), 1993, pp. 62–73. [Online]. Available: `https://doi.org/10.1145/168588.168596`.

[101] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in Modelling and Tools for Network Simulation, K. Wehrle, M. Güneş, and J. Gross, Eds. Berlin,

Heidelberg: Springer, 2010, pp. 15–34. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_2

[102] Hsiang-Chun Hsiang and Wen-Kung Shih, "Improvement of the secure dynamic ID-based remote user authentication scheme for multi-server environment," *Comput. Stand. Interfaces*, vol. 31, no. 6, pp. 1118–1123, Nov. 2009. DOI: 10.1016/j.csi.2008.11.002.

[103] Chin-Tser Li and Min-Shiang Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 33, no. 1, pp. 1–5, Jan. 2010. DOI: 10.1016/j.jnca.2009.08.001.

[104] Hsin-Lung Yeh, Ting-Hao Chen, Pei-Chi Liu, Tae-Ho Kim, and Hsin-Wen Wei, "A secured authentication protocol for wireless sensor networks using elliptic curve cryptography," Sensors, vol. 11, no. 5, pp. 4767–4779, May 2011. DOI: 10.3390/s110504767.

[105] Bipin Vaidya, Dimitrios Makrakis, and Hussein T. Mouftah, "Device authentication mechanism for smart energy home area networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, Oct. 2011, pp. 380–385. DOI: 10.1109/SmartGridComm.2011.6102338.

[106] Kaiping Xue, Peilin Hong, and Changgen Ma, "A lightweight dynamic pseudonym identity-based authentication and key agreement protocol without verification tables for multi-server architecture," Journal of Computer System Science, vol. 78, no. 2, pp. 483–500, Mar. 2012. DOI: 10.1016/j.jcss.2011.05.004.

[107] Yen-Fu Chang, Wei-Cheng Ku, and Shi-Jinn Lin, "A secure authentication scheme for telecare medicine information systems," Journal of Medical System, vol. 37, no. 6, pp. 9982–9988, Dec. 2013. DOI: 10.1007/s10916-013-9982-2.

[108] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Gener. Comput. System, vol. 29, no. 7, pp. 1645–1660, Sep. 2013. DOI: 10.1016/j.future.2013.01.010.

[109] Jean Paul Ndibanje, Hyunseung Lee, and Heekuck Oh, "Security analysis and improvements of authentication and key agreement protocol for sensor networks," Sensors, vol. 14, no. 5, pp. 7804–7823, 2014. DOI: 10.3390/s140507804.

[110] Liang Chen, Fang Wei, and Changgen Ma, "A secure user authentication scheme using symmetric cryptography for wireless sensor networks," Secure Communication Network, 2015, Article ID 704502. DOI: 10.1155/2015/704502.

[111] Manik Lal Das, "A secure and efficient user authentication protocol for wireless sensor networks using three-factor authentication," IEEE Trans: Wireless Communication System, vol. 15, no. 1, pp. 346–357, Jan. 2016.

[112] Rashid Amin and Goutam Biswas, "A secure lightweight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," Ad Hoc Networks, vol. 36, pp. 58–80, Feb. 2016.

[113] Mohammad Soleimani Farash, "A new user authentication and key agreement scheme for heterogeneous wireless sensor networks tailored for the Internet of Things environment," Ad Hoc Networks, vol. 36, pp. 152–176, Feb. 2016.

[114] Nidhi Kaul and Ajay K. Awasthi, "An efficient and secure smart card-based remote user authentication scheme for multi-server environment," Wireless Personal Communication, vol. 90, no. 2, pp. 589–609, Sep. 2016.

[115] Souvik Roy, Sudip Chatterjee, and Ashok Kumar Das, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," Future Gener. Comput. System, vol. 76, pp. 212–226, Nov. 2017.

[116] P. K. Dhillon and S. Kalra, "Secure multi-factor remote user authentication scheme for internet of things environments," *International Journal of Communication Systems*, vol. 30, no. 16, p. e3323, 2017. [Online]. Available: `https://doi.org/10.1002/dac.3323`

[117] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput. (SUTC)*, vol. 1, IEEE Computer Society, 2006, pp. 244–251.

[118] Mohit Wazid, Ashok Kumar Das, Rasheed Hussain, Gautam Srivastava, and Young-Sik Park, "Secure remote user authentication and key management for smart home environment," IEEE Access, vol. 5, pp. 16486–16501, 2017.

[119] Maryam Nikooghadam and Hamid Amintoosi, "A lightweight authentication and key agreement protocol preserving user anonymity," Multimedia Tools

Applications, vol. 76, no. 11, pp. 13059–13084, 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-3704-8.

[120] Jungtae Kang, Jung Hee Moon, and Jin Kwak, "An efficient and secure biometric-based user authenticated key agreement scheme with anonymity," Secure Communication Network, vol. 2017, 2017. [Online]. Available: https://doi.org/10.1155/2017/5198794.

[121] Tariq Shah and Subramanian Venkatesan, "Authentication of IoT device and IoT server using secure vaults," in Proc. 17th IEEE Int. Conf. Trust, Secur. Priv. Comput. Commun. (TrustCom), New York, NY, USA, Aug. 2018, pp. 819–824. [Online]. Available: https://doi.org/10.1109/TrustCom/BigDataSE.2018.00117.

[122] Pradeep Chandrakar and Hari Om, "An extended ECC-based anonymity-preserving 3-factor remote authentication scheme usable in TMIS," International Journal of Communication Systems, vol. 31, no. 8, e3540, 2018. [Online]. Available: https://doi.org/10.1002/dac.3540.

[123] Rashid Amin, Sheikh R. Islam, Neeraj Kumar, and Kim-Kwang Raymond Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," Journal of Network and Computer Applications, vol. 104, pp. 133–141, 2018. [Online]. Available: https://doi.org/10.1016/j.jnca.2017.12.012.

[124] Jong Hyuk Park, Jaeyeon Kim, and Yongwan Kim, "DM-MQTT: An efficient MQTT based on SDN multicast for massive IoT communications," Sensors, vol. 18, no. 9, p. 3071, 2018. [Online]. Available: https://doi.org/10.3390/s18093071.

[125] Wei Huang, "ECC-based three-factor authentication and key agreement scheme for wireless sensor networks," Science Reports, vol. 14, no. 1, p. 1787, 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-52134-z.

[126] Ali Ostad-Sharif, Hossein Arshad, and Mohammad Hossein Yaghmaee, "Three-party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," Future Generation Computer System, vol. 96, pp. 428–438, 2019. [Online]. Available: https://doi.org/10.1016/j.future.2019.02.005.

[127] Saurabh Garg and Ashok Kumar Das, "An OAuth-based access control model for securing IoT devices," Journal of Network and Computer Applications, vol. 132, pp. 97–111, 2019. [Online]. Available: https://doi.org/10.1016/j.jnca.2019.01.009.

[128] Subhadeep Banerjee, Vanga Odelu, Ashok Kumar Das, Suvra Chattopadhyay, and Young-Sik Park, "An efficient, anonymous and robust authentication scheme for smart home environments," Sensors, vol. 20, no. 4, p. 1215, 2020. [Online]. Available: https://doi.org/10.3390/s20041215.

[129] V. Sureshkumar, Rashid Amin, Mohammad S. Obaidat, and I. Karthikeyan, "An enhanced mutual authentication and key establishment protocol for TMIS using chaotic map," Journal of Information Secure Applications, vol. 53, p. 102539, 2020. [Online]. Available: https://doi.org/10.1016/j.jisa.2020.102539.

[130] Woong-In Bae and Jin Kwak, "Smart card-based secure authentication protocol in multi-server IoT environment," Multimedia Tools Applications, vol. 79, no. 23–24, pp. 15793–15811, 2020. [Online]. Available: https://doi.org/10.1007/s11042-017-5548-2.

[131] B. D. Deebak and Fayez A. Al-Turjman, "Lightweight authentication and key management in mobile-sink for smart IoT-assisted systems," *Sustain. Cities Soc.*, vol. 53, p. 102416, 2020. [Online]. Available: https://doi.org/10.1016/j.scs.2019.102416.

[132] R. Hinden, M. O'Dell, and S. Deering, "An IPv6 aggregatable global unicast address format," Technical Report, 1998.

[133] M. Sabir, M. Mian, K. Sattar, and M. Fahiem, "IP address space management using aggregated fixed length subnet masking," in *Proc. Int. Conf. Electrical Engineering*, 2007, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICEE.2007.4287630

[134] T. Narten, R. Draves, and S. Krishnan, "Privacy extensions for stateless address autoconfiguration in IPv6," RFC 4941, Technical Report, 2007. [Online]. Available: https://doi.org/10.17487/RFC4941

[135] R. Hinden and B. Haberman, "Unique local IPv6 unicast addresses," RFC 4193, Technical Report, 2005. [Online]. Available: https://doi.org/10.17487/RFC4193

[136] A. Judmayer, J. Ullrich, G. Merzdovnik, A. G. Voyiatzis, and E. Weippl, "Lightweight address hopping for defending the IPv6 IoT," in Proc. 12th Int. Conf. Availability, Reliability and Security, 2017, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3098954.3103156

[137] F. Gont and T. Chown, “Network reconnaissance in IPv6 networks,” RFC 7707, Technical Report, 2016. [Online]. Available: https://doi.org/10.17487/RFC7707.

[138] P. Kumar and L. Chouhan, “Design of secure session key using unique addressing and identification scheme for smart home internet of things network,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 5, p. e3993, 2021. [Online]. Available: https://doi.org/10.1002/ett.3993

[139] 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,” *J. Inf. Secur. Appl.*, vol. 52, p. 102494, 2020. [Online]. Available: https://doi.org/10.1016/j.jisa.2020.102494

[140] Michael Dunlop, Scott Groat, Ryan Marchany, and Joseph Tront, “MT6D: A moving target IPv6 defence,” in *AFCEA/IEEE Military Commun. Conf. (MILCOM)*, 2011. [Online].

[141] Jiun-Liang Tsai and Nai-Wei Lo, “A privacy-preserving authentication scheme for SIP using ECC,” *Comput. Commun.*, vol. 33, no. 9, pp. 1094–1104, 2010. [Online]. Available: https://doi.org/10.1016/j.comcom.2010.01.020.

[142] Mehdi Nicanfar, Payam Jokar, and Victor C. M. Leung, “Smart grid authentication and key management for unicast and multicast communications,” in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, 2011. [Online].

[143] Fei Wang, Jian Liu, and Yunchuan Zhang, “ProFactory: Improving IoT security via formalized protocol implementation,” in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 1–16. [Online]. Available: https://www.usenix.org/system/files/sec22-wang-fei.pdf.

[144] Heng Hu, Yanan Wang, and Zhiqiang Li, “Resilient event-triggered control of networked switched systems under denial-of-service attacks,” *Inf. Sci.*, vol. 580, pp. 1–15, 2021. [Online]. Available: https://doi.org/10.1016/j.ins.2021.08.001.

[145] Kanika Kaur and Praveen Kumar, “A secure and lightweight authentication protocol for IoT-based smart homes,” *Sensors*, vol. 21, no. 5, p. 1503, 2021. [Online]. Available: https://doi.org/10.3390/s21051503.

[146] Xiaofeng Li, Yanan Zhang, and Jun Chen, “An efficient ECC and fuzzy verifier based user authentication scheme for wireless sensor networks,” Sci. Rep, vol.

11, no. 1, p. 1787, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-81334-3.

[147] Yue Jiang, Xiaoming Ma, and Lin Liu, "An efficient ticket-based authentication protocol with unlinkability for wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/WCSP.2017.8170923.

[148] R. M. Hinden and S. E. Deering, "IPv6 global unicast address format," IETF RFC 3587, 2003.

[149] T. Chown and F. Gont, "Network reconnaissance in IPv6 networks," *IETF RFC 7707*, 2016.

[150] R. Hinden and B. Haberman, "Unique local IPv6 unicast addresses," *IETF RFC 4193*, 2005.

[151] R. Hinden and S. Deering, "IP version-6 addressing architecture," IETF RFC 4291, 2006.

[152] Y.-H. Han and S.-H. Hwang, "Care of address provisioning for efficient IPv6 mobility support," Computer Communications, vol. 29, no. 9, pp. 1422–1432, 2006.

[153] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Requirements for IP flow information export (IPFIX)," *IETF RFC 3917 (Informational)*, 2004.

[154] F. Gont, A. Cooper, D. Thaler, and W. Liu, "Recommendation on stable IPv6 interface identifiers," *IETF RFC 8064*, 2017.

[155] B. Carpenter and S. Jiang, "Significance of IPv6 interface identifiers," *IETF RFC 7136, 2014.

[156] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," *IETF RFC 3775*, 2004.

[157] S. Wu, Y. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Security and Communication Networks*, vol. 5, no. 2, pp. 236–248, 2012.

[158] E.-J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

[159] H. Kim, W. Jeon, K. Lee, Y. Lee, and D. Won, “Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme,” in *Proceedings of the International Conference on Computational Science and Its Applications*. Springer, 2012, pp. 391–406.

[160] D. Wang, D. He, P. Wang, and C.-H. Chu, “Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[161] C. Wang, G. Xu, and W. Li, “A secure and anonymous two-factor authentication protocol in multiserver environment,” Security and Communication Networks, vol. 2018, pp. 1–15, Apr. 2018. DOI: 10.1155/2018/4012820.

[162] Y. Park, “A secure user authentication scheme with biometrics for IoT medical environments,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 11, pp. 607–615, 2018. DOI: 10.14569/IJACSA.2018.091173.

[163] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, “Security for 5G and beyond,” *IEEE Communications Surveys & Tutorials*, 2019. DOI: 10.1109/COMST.2019.2916180.

[164] P. K. Dhillon and S. Kalra, “Secure multi-factor remote user authentication scheme for Internet of Things environments,” *International Journal of Communication Systems*, vol. 30, no. 16, p. e3323, 2017. DOI: 10.1002/dac.3323.

[165] M. Turkanović, B. Brumen, and M. Hölbl, “A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion,” *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014. DOI: 10.1016/j.adhoc.2014.03.009.

[166] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, “Provably secure user authentication and key agreement scheme for wireless sensor networks,” *Security and Communication Networks*, vol. 9, no. 16, pp. 3670–3687, 2016. DOI: 10.1002/sec.1575.

[167] I.-P. Chang, T.-F. Lee, T.-H. Lin, and C.-M. Liu, “Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks,” *Sensors*, vol. 15, no. 12, pp. 29841–29854, 2015. DOI: 10.3390/s151229782.

[168] Z. Yang, J. Lai, Y. Sun, and J. Zhou, “A novel authenticated key agreement protocol with dynamic credential for WSNs,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, p. 22, 2019. https://doi.org/10.1145/3303704.

[169] S. Banerjee and D. Mukhopadhyay, “Symmetric key-based authenticated querying in wireless sensor networks,” in *Proc. 1st Int. Conf. on Integrated Internet Ad Hoc and Sensor Networks*, ACM, 2006, p. 22. DOI: 10.1145/1189355.1189378.

[170] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari, L. Wu, and J. Shen, “A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks,” *Future Generation Computer Systems*, vol. 82, pp. 727–737, 2018.

[171] D. Abbasinezhad-Mood and M. Nikooghadam, “Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications,” *Future Generation Computer Systems*, vol. 84, pp. 47–57, 2018.

[172] A. G. Reddy, A. K. Das, V. Odelu, and K. Y. Yoo, “An enhanced biometric-based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography,” *PLoS ONE*, vol. 11, no. 5, p. e0154308, 2016.

[173] S. Kumari, M. K. Khan, and M. Atiquzzaman, “User authentication schemes for wireless sensor networks: a review,” *Ad Hoc Networks*, vol. 27, pp. 159–194, 2015.

[174] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, “Botnet attack detection in Internet of Things devices over cloud environment via machine learning,” *Concurr. Comput. Pract. Exp.*, vol. 34, no. 4, p. e6662, 2022.

[175] N. Chikouche, P.-L. Cayrel, E. H. M. Mboup, and B. O. Boidje, “A privacy-preserving code-based authentication protocol for Internet of Things,” *J. Supercomput.*, vol. 75, pp. 8231–8261, 2019.

[176] J. Gao, M. Liu, P. Li, A. A. Laghari, A. R. Javed, N. Victor, and T. R. Gadekallu, “Deep incomplete multi-view clustering via information bottleneck for pattern mining of data in extreme-environment IoT,” *IEEE Internet Things J.*, 2023.

[177] D. Kaur, K. K. Saini, and D. Kumar, “Cryptanalysis and enhancement of an authentication protocol for secure multimedia communications in IoT-enabled wireless sensor networks,” *Multimed. Tools Appl.*, vol. 81, no. 27, pp. 39367–39385, 2022.

[178] R. Soltani and S. Pashazadeh, “A lightweight improvement of PEDAAC protocol for 6LoWPAN in the Internet of Things,” *Multimed. Tools Appl.*, vol. 80, pp. 31467–31486, 2021.

[179] E.-J. Yoon and K.-Y. Yoo, “Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem,” *J. Supercomput.*, vol. 63, pp. 235–255, 2013.

[180] S. Uppuluri and G. Lakshmeeswari, “Secure user authentication and key agreement scheme for IoT device access control based smart home communications,” *Wirel. Netw.*, vol. 29, no. 3, pp. 1333–1354, 2023.

[181] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” Bus Horizons, vol. 58, no. 4, pp. 431–440, 2015.

[182] P. Kumar and L. Chouhan, “A privacy and session key-based authentication scheme for medical IoT networks,” Computer Communications, vol. 166, pp. 154–164, 2021.

[183] 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 Things J.*, vol. 5, no. 3, pp. 1606–1615, 2017.

[184] M. Shuai, N. Yu, H. Wang, and L. Xiong, “Anonymous authentication scheme for the smart home environment with provable security,” *Comput. Secur.*, vol. 86, pp. 132–146, 2019.

[185] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, “A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems,” *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, 2020. [Online]. Available: https://doi.org/10.1109/JSYST.2019.2899580

[186] Y. Chen and J. Chen, “An efficient mutual authentication and key agreement scheme without password for wireless sensor networks,” Journal Super computing, vol. 77, no. 12, pp. 13653–13675, 2021.

[187] S. Zou, Q. Cao, C. Wang, Z. Huang, and G. Xu, "A robust two-factor user authentication scheme-based ECC for smart home in IoT," IEEE System Journal, vol. 16, no. 3, pp. 4938–4949, 2022. [Online]. Available: https://doi.org/10.1109/JSYST.2021.3127438

[188] Y. Cho, J. Oh, D. Kwon, S. Son, J. Lee, and Y. Park, "A secure and anonymous user authentication scheme for IoT-enabled smart home environments using PUF," *IEEE Access*, vol. 10, pp. 101330–101346, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3208347

[189] 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, 2015.

[190] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *J. Supercomput.*, vol. 74, pp. 6428–6453, 2018.

[191] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1953–1966, 2015.

[192] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for the smart home environment with provable security," *Comput. Secur.*, vol. 86, pp. 132–146, 2019.

[193] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K. K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw.*, vol. 129, pp. 429–443, 2017.

[194] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, and Y. Park, "An efficient, anonymous and robust authentication scheme for smart home environments," *Sensors*, vol. 20, no. 4, p. 1215, 2020.

[195] D. Kaur and D. Kumar, "Cryptanalysis and improvement of a two-factor user authentication scheme for smart home," *J. Inf. Secur. Appl.*, vol. 58, p. 102787, 2021.

[196] V. Sureshkumar, R. Amin, V. R. Vijaykumar, and S. R. Sekar, "Robust secure communication protocol for smart healthcare system with FPGA implementation," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 938–951, 2019.

[197] B. Yu and H. Li, "Anonymous authentication key agreement scheme with pairing-based cryptography for home-based multi-sensor internet of things," *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 9, p. 1550147719879379, 2019.

[198] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks," *Multimedia Syst.*, vol. 23, pp. 195–205, 2017.

[199] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, 2020.