Investigating Approaches for Improving Security in Remote User Authentication Schemes for IoT Paradigm

Thesis

Submitted in partial fulfillment of the requirement of

Doctor of Philosophy

by

Chintan Patel

Roll No. 17RCP003

Under the guidance of

Dr. Nishant Doshi

Department of Computer Science and Engineering

School of Technology

Pandit Deendayal Energy University

Gandhinagar – 382426. Gujarat - India

August, 2021.

Approval Sheet

This thesis entitled **"Investigating Approaches for Improving Security in Remote User Authentication Schemes for IoT Paradigm"** by **Chintan Patel** is recommended for the degree of Doctor of Philosophy.

Examiners

_____

_____

_____

Supervisor

_____

_____

_____

H.o.D/Chairman

_____

Date: _____

Place: _____

Student Declaration

I, **Chintan Patel**, hereby declare that this written submission represents my ideas in my own words and where others' idea or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Pandit Deendayal Energy University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Chintan Patel

Roll No : 17RCP003

Date: _____

Acknowledgment

I feel truly privileged to be a doctorate student in the Computer Science and Engineering (CSE) department of the School of Technology (SoT) of the Pandit Deendayal Energy University (PDEU), Gandhinagar. My doctorate journey in PDEU has taught me certain values such as academic integrity, honesty, patience and hard work above everything else. It has been one of the most challenging as the same time fruitful phase of my life. In this journey, i came across many people who have inspired me and motivated me for my work. I would like to take this opportunity to thank all those who have helped me and encouraged me during the course of working on this thesis.

First, I express deep sense of gratitude towards my guide **Dr. Nishant Doshi**. I am obliged to him for giving me freedom of expression beyond the limits. I feel happy to be his first doctorate Scholar. I will always cherish and follow his ideals of keeping patience, being humble and doing hard work in a smarter way during any critical situations. Dear Sir. I truly feel honoured to my external examiner **Prof. Maniklal Das**, Professor of Computer Science at Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, India, for providing his valuable suggestions during every phase of this submission and word might not suffice to express my deep gratitude and respect towards him.

I would like to express my sincere gratitude to **Prof. S. Sundar Manoharan** (DG, PDEU), **Prof. Sunil Khanna**, (Director, SoT) and **Prof. Rajesh Patel**, (Dean, SoT). I would like to express my sincere thanks to **Dr. Samir Patel** (HoD, CSE). Respected panel members, Your honest support and fruitful suggestions

helped me a lot during this journey. I will always remain honoured towards the confidence you have shown in me. You have steered my career to a path which will always inspire me to touch a sky in my work.

I would also like to thank **Dr. Manish Chaturvedi**, IITRAM, Ahmedabad, India for his valuable suggestions, continuous support in this journey. I am so much thankful to other faculties of CSE department, specially **Dr. Rudresh Dwivedi**, **Dr. Payal Chaudhari** and faculties from other department specially **Dr. Vivek Patel**, Mechanical Engineering Department for their support and encouragement. I would like to thank PDEU management for providing facilities and administrative support during this research journey.

I cannot miss this opportunity to express my heartfelt gratitude to the friends who always inspired and motivated me for this journey. Dear **Ms. Hemani Parikh** (my positive thinker), **Mr. Kanhaiya Sharma** (my anytime helpline) , and **Mr. Pruthivish Rajput** (my research helpline), I can not visualize any moment of this journey without you people and thank you so much from my bottom of the heart for all your support. Thanks are due to my dear friends **Pratik Patel, Mumukshu Trivedi, Ankit Oza, Ruchita Shah, Hardik Jani, Ravi Patel & Devanshi Dave**. I will cherish the tea & lunch breaks and other fun at Ph.D. workspace. Thank you all for making me realise my strengths and for being there in moments of weakness.

Lastly, I would like to thank the actual drivers of this journey. My Parents. **Jayprakash Patel (father)** and **Leelaben Patel (mother)**, your love, support and sacrifices made me what I am today. You are the forces that kept me going through the program. Dad, you always said that your duty is far better than my duty and i know you will be happy to see me a successful person in the scientific field. I am

also thankful to three other most special entities of my life, My wife **Jagruti Patel** (engine who always supported my vehicle in any situation by burning herself) and My sister **Dr. Hetal Patel** (oil of my vehicle) for motivating me to be best version of myself. I am so much thankful to my son **Aarush Patel** for highest scarifies and indirect support in this journey. I am also thankful to my other family members for their kind support.

Chintan Patel

Dedicated to

Grand Father and Grand Mother*... You lightened it*

Papa*...You envisioned this.*

Your determination and belief made this a reality..

Mummy*...Your faith and prayers have lead us here..*

Your patience & support made this a reality..

Jagruti & Hetal & Aarush*.. You made me laugh at the lowest points.. You made me believe that I can..*

Abstract

Internet of Things (IoT) is a network of interconnected tiny resource constraint devices. These devices can be sensors, actuators, gateways or other microcontrollers and microprocessors. An intra-device communication (a.c.a. Machine to Machine Communication (M2M)), as well as an inter-device communication in the IoT, must be protected from both, active and passive attackers. Traditional cryptography protocols (i.e. RSA or DES or AES) provides well suited reliable security mechanism in the current internet topology that is built up using highly resource capable devices such as computers and servers. Fundamental different between internet cryptography and IoT cryptography lies in types of devices used to build up the topology. The internet is built up using resource capable devices such as computers and servers while the IoT network is built up using resource constraint devices such as sensors and actuators. Devices in the IoT suffers from copious limitations such as poor battery backup, less processing capabilities and lower storage capabilities. Hence, it is highly erratic and inept at using traditional internet cryptography in IoT communication.

Remote Use Authentication (RUA) is a backbone of any trusted and secure communication. The RUA provides mutually trusted session key for further encrypted communication between communicating parties. In the IoT, user devices (i.e. wearable devices or mobile devices) receives data from the sensing devices through series of gateway devices. In these devices, gateway devices are only devices which are resource capable devices and can perform complex cryptography operations while other devices may not be capable of performing those operations.

Hence, it is trivial to establish a mutually trusted session key between sensing devices and user devices using lightweight cryptography operations such as Elliptic Curve Cryptography (ECC), XOR operations and Lightweight hash functions.

The fundamental objective of the proposed work is to propose a highly efficient and computationally reliable authentication mechanism for mutually authenticated session key generation between either User - GateWay (U-GW) or between User - GateWay - Sensing device (U-GW-S) in the constraint environment. In this thesis, we propose a novel key exchange schemes for U-GW model as well as U-GW-S model using an XOR operation, lightweight hash function and the ECC encryption-decryption as a preliminary cryptography operation.

In this work, we also propose a novel authentication mechanism called a Level Dependent Authentication for Generic IoT (LDA-GIoT). In LDA-GIoT, every user and sensing devices get a level number based on their deployment on the ground. The system user with a lower level compared to the sensing level can only establish a session key for the communication with sensor. An implementation and a security analysis of the proposed LDA-GIoT using the classical tools proves that the proposed LDA-GIoT is an efficient and reliable algorithm for the IoT devices authentication.

Further, we provide a security analysis for the proposed work using a widely adopted Dolev-Yao channel and random oracle based Real or Random (ROR) Model. We also prove a mutual authentication for the proposed schemes using a widely adopted Burrows–Abadi–Needham logic (BAN) Logic. We have used a python language for the implementation and raspberry pi and nodeMCU for the test-beds generation of the proposed work. For the communication purpose, we have used MQTT (Message Queuing Telemetry Transport) protocol.

Contents

List of Figures

List of Tables

Abbreviations

IoT ..... Internet of Things

M2M ..... Machine to Machine

D2D ..... Device to Device

RUA ..... Remote User Authentication

RFID ..... Radio Frequency IDentification

ECC ..... Elliptic Curve Cryptography

SC ..... Smart Card

SCR ..... Smart Card Reader

SK ..... Session Key

NFC ..... Near Field Communication

DES ..... Data Encryption Standard

ECDLP ..... Elliptic Curve Discrete Logarithm Problem

SG ..... Smart Grid

SH ..... Smart Home

SHe ..... Smart Healthcare

DOS ..... Denial of Services

DDOS ..... Distributed Denial of Service

MITM ..... Man-In-the-Middle Attack

SI ..... Smart Industry

HTTP ..... Hyper Text Transfer Protocol

ECDHE ..... Elliptic Curve Diffie-Hellman Encryption

STS ..... Smart Transportation System

RoR Model ..... Real or Random Model

BAN Logic ..... Burrows–Abadi–Needham Logic

LDA ..... Level Dependent Authentication

BLE ..... Bluetooth Low Energy

MQTT ..... Message Queuing Telemetry Transport

CoAP ..... Constraint Application Protocol

XMPP ..... Extensible Messaging and Presence Protocol

SM ..... Smart Meter

AVISPA .. Automated Validation of Internet Security Protocols and Applications

Nomenclature

$H(.)$ ..... Hash Function

Chapter 1

Introduction of IoT Paradigm

This chapter provides an introduction for IoT paradigm and RUA. ***Section 1.1*** provides introduction to IoT. ***Section 1.2*** provides an overview on layered architectures adopted by different entities working on IoT expansion. ***Section 1.3*** presents overview on RUA and different authentication models adopted in IoT paradigm. ***Section 1.4*** presents security issues faced by IoT eco-system and it's applications. ***Section 1.5*** provides introductory discussion about IoT applications that makes complete IoT eco-system as an interconnected eco-system. ***Section 1.6*** highlights existing research gaps and major contributions of this thesis. ***Section 1.7*** provides detailed thesis preview in terms of chapter wise overview.

1.1 Introduction

An IoT network build-ups using a highly homogeneous, globally dynamic, deeply deployed, and the comparatively resource-constrained devices to provide "*Any type*" service at "*Any location*" to "*Anyone*" on "*Any time*". The scale of IoT data

generation is directly proportional to the growing quantity of Internet connected devices. As per the recent predictions by the global giant of telecommunications and market intelligence agency International Data Cooperation (IDC), there will be approx 42 billion deployed devices that will generate approx 80 ZettaByte data by 2025 [Atzori et al. (2010)].

The journey of IoT was started with the RFID technology in which RFID tag transmit the identification data to the RFID reader. This approach is broadly expanded and now a days IoT can be defined as an "interconnected network of uniquely identifiable tiny resource constraint device those communicate with each other in the environment of mesh topology". Most of the devices deployed in the IoT networks are tiny resource constraint sensing devices and actuators which are having capability of tracing the surrounding biological environment as well as the nonbiological environment. These devices use technologies such as Bluetooth, RFID, Zigbee, and WiFi to transmit the sensed data to the cloud server via nearest gateway devices. The major IoT devices are resource constraint devices in terms of storage cost, computation capabilities, communication capabilities and power utilization.

The traditional Internet uses complex cryptography mathematical operations for to protect the data over communication channel. These operations require ample storage space and high computation memory with huge power consumption. Thus, due to the highly constraint availability of numerous IoT devices, it is indispensable to prototype a lightweight security mechanism for the end-to-end data communication in IoT Model. For IoT network, there is a strong need to design a lightweight security mechanism that must be efficient in terms of the computation capabilities, optimized in terms of memory and time utilization, and robust

against the traditional and non-traditional security attacks.

A recent prediction by IoT business news shows that IoT devices will grow from 7.6 billion to 24.1 billion between 2019 and 2030, with estimated revenue will reach to USD 1.5 trillion [Hatton (2020)]. Fuqaha et al. highlighted that by 2022, more than 45% Internet connected devices will be IoT devices that may lead to economical valuation upto $6.2 trillion by 2025 [Al-Fuqaha et al. (2015)]. Hence, there is a need of strong and justifiable standardization and architecture that provide a clear direction to the IoT learners, developers, researchers, industries, and all other entities involved with IoT in either ways.

Rapid growth in connected devices also invites exponential growth to zero day vulnerabilities in IoT eco-system. Major challenges in IoT eco-system includes (1) heterogeneity of IoT devices in terms of manufacturers and technology used by them. (2) providing unique identity to billions of Internet connected devices. (3) establishing reliable communication among these devices. (4) managing resources based on energy consumption, storage requirement, computation cost and communication cost for complete eco-system (5) providing better services to all the types of users through improved data analytic and intelligent decision making system. (6) providing strong security and privacy system for the complete eco-system.

In general, IoT eco-system is build up using three types of devices. i.e. (1) Sensing devices those sense the data and pass those data to the intermediary devices. (2) The intermediary devices, those may be gateway devices, cluster heads or other devices whose work is to forward the received data to either user for real time monitoring or to the cloud for further processing and storage. (3) The user devices that receive data from the intermediary devices and take decisions accord-

ingly. Among the numerous security parameters, authentication provides mutual trust among these devices in terms of identity and assures each other that they are communicating with the valid devices.

The vast applicability of IoT eco-system attracted government agencies, researchers and other entities and inspired to gear up development towards making non-smarter applications to the smarter applications. We can list out some of the IoT based applications such as smart healthcare, smart manufacturing, smart transportation, smart retails and logistics, smart agriculture, smart home and so on. Every IoT applications developed in a layered manner from device deployment to user applications. Next section 1.2 presents different layered architectures adopted by different entities.

1.2 Layered Architecture

The layered architecture of any technology helps researchers and learners to understand the development hierarchy of full system. Till today many different organization such as ITU, IETF, IEEE have proposed tentative reference models for IoT eco-system, but still, none of the architecture got a world wide acceptance to become reference model for IOT. In 2014, CISCO, IBM and Intel jointly presented a seven layer architecture at world forum [Cisco (2014)]. Researchers in [Atzori et al. (2010)] presented three layered architectures that consists of perception layer, transport layer and application layer. Authors in [L. D. Xu et al. (2014)] presented four layered architecture that adds one extra layer called as a middle ware in the three layered architectures. Some other authors in [Al-Fuqaha et al. (2015)] presented five layered architecture that consists of perception layer,

network layer, support layer, application layer. Over all, an IoT eco-system consists of things which sense the data, network that transmit the data, middle ware that process the data and application layer that uses the data. In this section, First, seven layered architecture proposed by CISCO is over viewed [Cisco (2014)] and then four layered architecture presented in [Patel & Doshi (2018)] is discussed with security vulnerabilities at each layer.

A Seven Layered Architecture

Following Fig. 1.1 presents seven layer architecture of IoT presented by CISCO.

Layer 1: Physical Devices and Controllers

The Layer-1 of the CISCO model deals with the IoT devices those collects the data from the environment.

- This layer basically discusses about "things" in the IoT. Things can be devices such as sensors and actuators or controllers such as arduino and raspberry pi. These devices collect the data from the surroundings and forward it to near by controllers. The nearby controllers forward those data to the valid destination at edge level.

- Examples of these devices and controllers are: sensors, microcontrollers, micro processors, cameras, RFID Tags, RFID readers, bluetooth devices, cars, wearable devices, intelligent machines and so on.

Major Security concerns in this layer are confidentiality and integrity of data, physical device security, authentication between devices, accountability and non repudiation of the data transmitting devices.

Layer 2: Connectivity

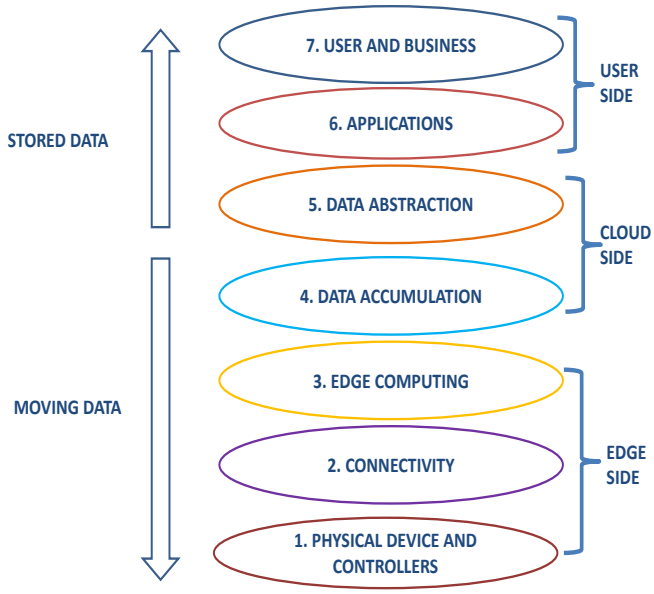

Figure 1.1. Seven Layered Architecture by CISCO [Cisco (2014)]

The Layer-2 of the CISCO model provides connectivity between physical devices and edge devices. This layer deals with communication among:

- Computing devices.
- Computing devices and controllers.
- Controllers and gateway device/edge device as well as
- Gateway and cloud.
- Users/application and cloud.

Major Security concerns in this layer are side channel attacks, confidentiality and integrity of data, physical device security, authentication between devices, accountability and non repudiation of data transmitting devices.

Layer 3: Edge Computing

The Layer-3 of the CISCO architecture deals with the edge computing or fog computing where the data will be processed locally. We can summarize the operations of this layer as follows:

- Basic motive of this layer is to process the data at local layer with the objective of reduction in the computation overhead over the cloud.
- This layer helps in traffic reduction at above layers and also improve the accuracy in decision making.

Major security challenge in this layer is to protect the data from the internal adversaries who have access of edge devices as well as from the external adversaries who try to access edge device with a bad motive. Hence, some security vulnerabilities such as physical security, Denial of Service and Man in the middle attacks need to take care.

Layer 4: Data Accumulation

The layer-4 of the CISCO architecture deals with data storage and local filtering. We can conclude functionalities of this layer as follow:

- To store the data that is not going to use in instant processing but may be later on required.
- To forward stored data as per requirements and queries from the above layers.
- To convert the data from packets to tables and schemas.
- To provide conversion from the event-based data generation to query-based data consumption. It also reduces the data through filtering and selective storing.

The major security issue in this layer is to protect the stored data from the attackers.

Layer 5: Data Abstraction

The Layer-5 of the CISCO architecture supports in the development of simple and easy to use IoT applications by better rendering and the storage of the data. We can summarize functionalities of this layer as follows:

- Gathers data from the multiples sources and try to convert it into easy to use format for application developer and user.
- Performs filtering, selecting, projecting, and reformatting the data to serve the client applications .
- Provides security to data through appropriate authentication and authorization
- Performs normalization and denormalization operations over indexed data in such way that application user can access it in minimal time.

Major security requirement of this layer is to provide better authentication, authorization and access control.

Layer 6: Application

The Layer-6 of the CISCO architecture deals with basically two types of applications that are either controlling applications or monitoring applications. Major works carried out by this layer can be summarised as follows:

- Mobile applications through which user can control the devices or can monitor the data.

- Communicates with the data accumulation and data abstraction layer for receiving the data.
- Applications that performs critical analytic or received data for processing.

The major security challenge in this layer is to deal with identity and authentication of the application users.

Layer 7: Collaboration & Processes

This layer supports access of huge IoT among many people through better collaboration and simple business processes.

The CISCO presented seven layer architecture based on business processing and applicability of IoT in industry. With this many other organizations and authors presented other layered architecture based on different perspective. In 2018, Patel et al. [Patel & Doshi (2018)] presented four layer architecture with the perspective of device deployment, applications and security issues. Next subsection discusses this architecture and presents security issues at every layer.

B Four Layered Architecture

Patel et al. [Patel & Doshi (2018)] presented four layered architecture (Fig. 1.2) for the IoT eco-system based on functionalities.

1. ***Physical Layer / Perception Layer / Object Layer*** Basic Functionalities of this layer includes,
    - To deploy the sensors, actuators, RFID readers and sensor gateway on the field.

- To focus on sensing environment data such as location, temperature, pressure, humidity, motion, air pressure, pollution level and also collecting data from humans and machines for certain application.
- Assigning universally accepted unique identification number to each devices.
- To focus on capability enhancement of various IoT sensors in terms of cost,size,energy consumption,resource,communication and security.
- To handle scalability and heterogeneity of devices and data, Reliability of communication.
- **Security challenges :**
  - Physical security of the deployed devices.
  - Confidentiality and integrity of the data during transmission to above layer.
  - Authentication and access control of the devices.

2. ***Network Layer / Communication Layer*** Basic Functionalities of this layer includes,

   - Data collection from the physical or perception layer.
   - Local data processing at fog devices and forwarding data for further processing such as decision making and high level services.
   - Forwarding control signals to the ground level sensors.
   - Performs long distance communication with functionality such as connecting smart things, network devices, and cloud servers.

- Taking care of the routing, discovering and mapping dynamically happening changes inside the network and devices topology.
- **Security challenges :**
  - Access control and authentication among the networking devices such as fog devices and routers, accountability and non-repudiation of the received and transmitted data, authentication between sensor devices and routers.
  - Denial of Service is the major challenge for this layer. if attackers successfully deploy flooding attack and results in to denial of service then it can cause complete damage for the sub-system as well as IoT eco-system.

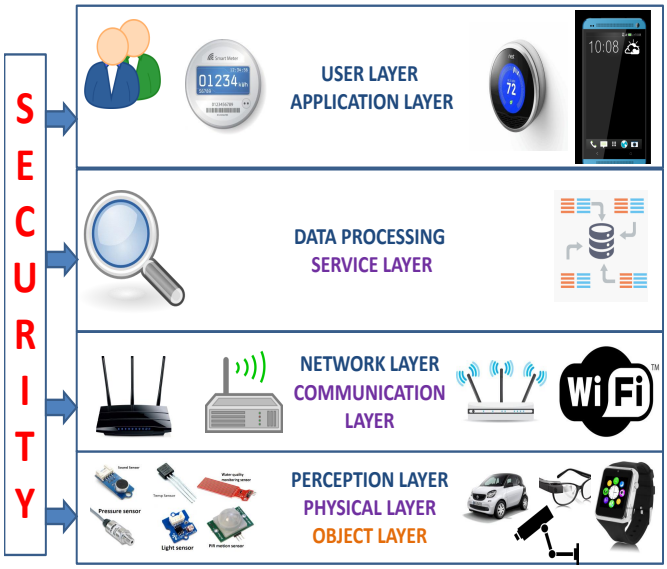

Figure 1.2. Four Layered Architecture [Patel & Doshi (2018)]

3. ***Data processing Layer / Service Layer*** Basic Functionalities of this layer

includes,

- Data mining and the data processing of the collected data over the cloud are a key functionalities of this layer. This layer works as a key enabler for the IoT applications and services in eco-system.
- Perform the data analytic, graph generation, solution generation, recommendation generation as per application requirement.
- Generating control signal, alerts, messages for the application layer users.
- Performing other service oriented functionalities for application.
- Availing on demand services on any time required by any one from the any place after performing successful authentication.
- **Security challenges :**
  - Data anonymity and data Privacy of the users and devices should be taken care.
  - Auditability of the sensed data.
  - Trust management and confidentiality of the stored data.

4. ***User Layer / Application Layer*** Basic Functionalities of this layer includes,
   - To create an ease of use for the users among IoT eco-system.
   - Preparing Many user oriented applications which receives data from the below layers.
   - Taking care of the heterogeneous private and public applications.

- Domain oriented application designing. Ex, Smart patient tracking system in Health care domains.
- **Security challenges :**
  - Confidentiality & Integrity of the data.
  - Signal transmission, authentication & access control between user applications and cloud storage.
  - Privacy of user and device data.

In IoT layered architecture, device at every layer need to authenticate it self with the other communicating devices. In, next section, we discusses basics about authentication and different authentication model followed in IoT.

1.3 Remote User Authentication

The Remote User Authentication (RUA) is a security technique that creates trust among two communicating parties over an insecure channel through secure and mutually authenticated session key generation. A reliable and efficient RUA scheme provides a secure session key to both the entities through which they can communicate over an insecure channel in the trusted environment. In 1981, Lamport introduced the first RUA scheme based on the hash chain and with the password table at the server-side [Lamport (1994)]. An authentication scheme lightened by Lamport provided use registration through identity (ID) and password (PW). Due to certain limitations of password-based schemes such as storing a password table leads to an insider attack and using a password as a uni-factor may lead to offline and online password guessing attack. Thus, in 1993, Chang et

al. [Chang & Hwang (1993)] introduced a first SC based authentication scheme. In the SC based authentication schemes, the user keeps a SC generated by the service provider as another security feature. The SC is a microprocessor device or chip that has secure memory and public memory, and it stores user data and secret parameters [Chang & Hwang (1993)].

In 1976, Diffie et al. introduced the discrete logarithm based information-sharing mechanism [Diffie & Hellman (1976)]. Rivest et al. [Rivest et al. (1978)] and Elgamal et al. [Elgamal (1985)] proposed a secured encryption mechanism in 1978 and 1985, respectively. Till today, many researchers have worked on that and proposed RUA schemes. But the critical challenge is designing an authentication scheme without any vulnerabilities. In 2004, Das et al. [Das et al. (2004)] proposed first dynamic identity based authentication scheme for the sensor network and generated path for other dimension of authentications. Due to several attacks such as SC stolen and power analysis, researchers introduced biometric-based authentication schemes. They created a path for the multi factor authentication, which was followed by OTP based authentication, token-based authentication, and so on.

Cryptography mathematics is the complex mathematics for computations, as well as implementations. With the run-up towards smart technology, the need for lightweight cryptography came in the picture. The reason behind this need is the use of numerous resource constraint devices in the deployment of sensor-based IoT application deployments. These resource constraint devices are short of computation memory and storage capability. For these devices, it is nearly impossible for them to perform sixteen rounds of DES and exponential computations of RSA promptly and without energy utilization. Thus, in 2006, Hankerson et al. [Hanker-

son et al. (2006)] highlighted *ECC*, which is much lighter than traditional crypto methods in computations and storage requirements [Hankerson et al. (2006)]. The ECC became a well-known cryptographic technique due to its attractive features like smaller key size, lower time-requirements, and fewer resource utilization. The *ECDHE* provides a lighter version of Diffie-Hellman with *ECDLP*.

A Authentication Models in IoT

The Authentication model in IoT can be divided in to two parts.

User - Gateway based Model

The User - Gateway based authentication model (Fig. 1.3) involves authentication between IoT application user and near by IoT device that receives data from the various sensors. We can consider following use-cases for this model. Authentication between:

- Smart home owner and home gateway
- Smart grid user and smart meter
- Doctor and healthcare gateway
- Industry owner and factory gateway
- so on ..

This model basically considers sensing devices as a very lightweight and resource constraint devices those are not capable to perform any crypto operation and have ability to just sense the data and transmit the data. In this model, the application

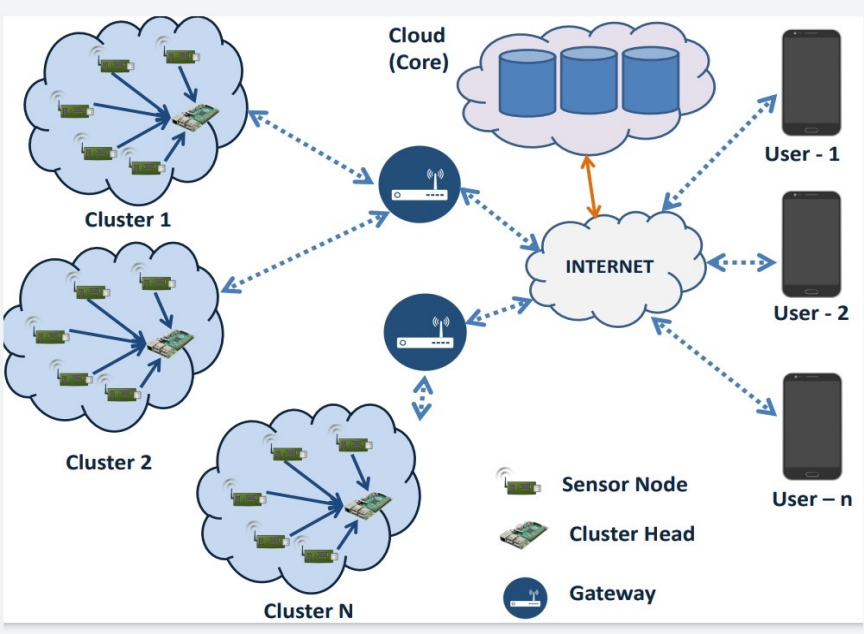

Figure 1.3. User - Gateway Authentication Model

user receives real-time live data after the successful authentication between it and the near by gateway device. The gateway device in this model is a resource capable device and can perform complex cryptography operations. The chapter 3 of this thesis, proposes lightweight and resource efficient authentication scheme compare to existing schemes based on this model.

User - Gateway - Sensor based Model

This model is highly used and adopted in IoT authentication in which all the three entities : the system user, intermediary gateway device and the sensors are involved in authentication. In this model, after the successful authentication and session key generation, the system user can directly receive a data from the sensor device using key. This authentication model further involves a cluster head device that is either raspberry pi or other little resource capable device compare to sens-

ing devices. The authentication between these entities can be designed based on

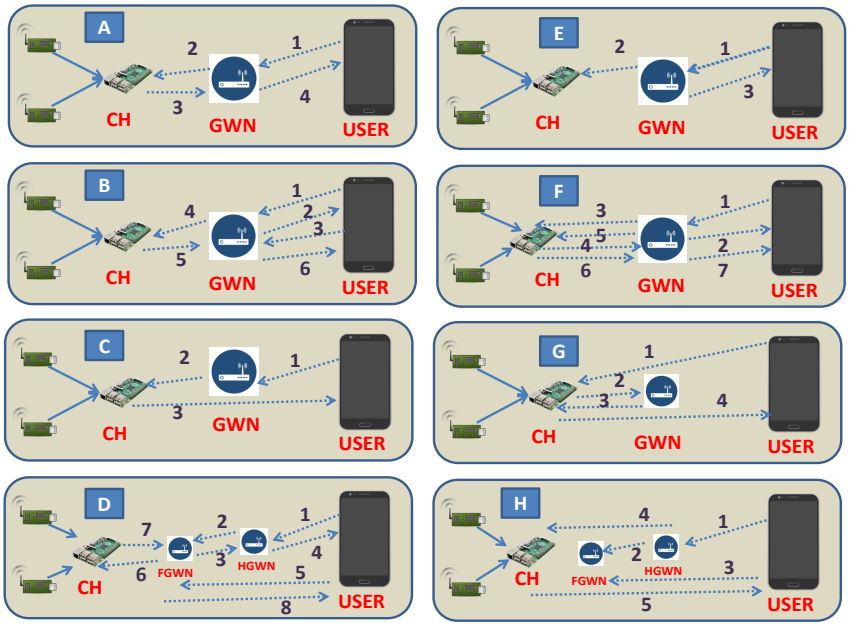

Figure 1.4. User - Gateway - Sensor Authentication Model

system requirements and design using one of the eight approaches shown in Fig. 1.4. There can be multiple gateway also possible in that scenario authentication steps involves home gateway as well as foreign gateway. This model considers only a gateway device as a resource capable device and fully trusted device while other two devices are considered as a lightweight devices and need resource optimization in terms of crypto operations. Chapters 4, 5 and 6 presents authentication scheme based on this network model.

IoT suffers from many other security issues parallel to authentication. In next section, we discuss other security issues available in IoT communication.

1.4 Security Issues

In this section, we discuss eight security parameters that must be satisfy during designing of any secured eco-system [Patel & Doshi (2019)]. The security octagon shown in Fig. 1.5 presents these eight parameters

1. **Confidentiality:**
    - Confidentiality can be defined as a "hiding of information from the person or device who is not authorized to read it".
    - In the IoT, devices collect the various sensitive information from the users. In the smart health care, medical devices collect the personal information such as heart bit-rate, pulse information, body temperature. In the smart home, person location, identity and other details such as availability can be tracked. Thus, all this information must be confidentially transmit from the sensing devices to the cloud storage / application.
    - Confidentiality can be achieved using various symmetric and asymmetric encryption algorithms such as RSA and DES. In the resource constraint environment, some lightweight algorithms such as ECC can be useful for authentication and encryption.
2. **Integrity:**
    - Integrity can be defined as a "preventing unauthorized user/device from modifying single bit of the data during transmission".

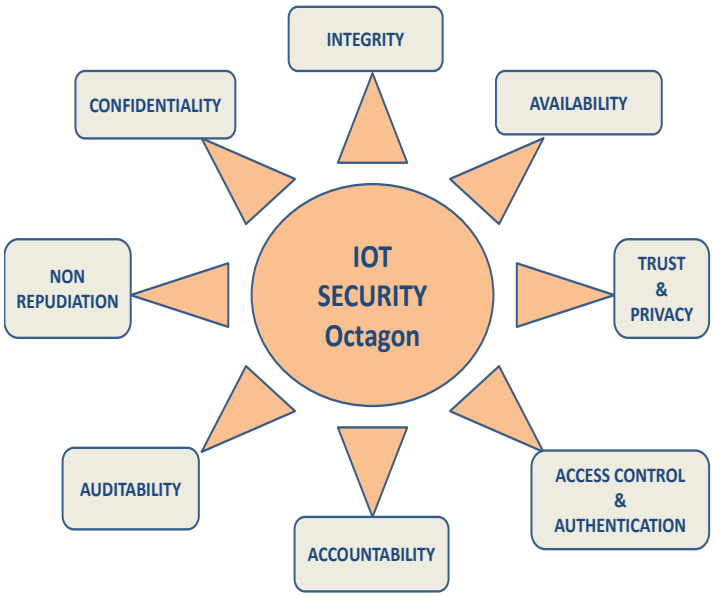

Figure 1.5. Security Heptagon [Patel & Doshi (2019)]

- In the IoT, integrity concerns becomes critical when it comes to modification of the information. For example, modification in SPO_2 level can damage functionalities of automated oxygen supply system. Thus, change in single bit can change the outcome of complete system who are working based on "True" or "False" and "Yes" or "No".
- Better integrity management can also support in smooth functioning of the financial transactions as well as industrial functioning.
- Integrity in the security can be assured using numerous verification methods such as checksum or one way hash function algorithms like SHA-1, SHA-256.

3. **Availability:**

- Availability can be defined as a "all the services running in the complete system must be available at any time to any authenticated entities".
- In the IoT, availability plays an important role in terms of resources availability or data availability when ever it is requires. The smart grid should control the power system in such a way that whenever user requires high power voltage, he/she gets it. Smart home user must get on time camera videos to verify the on gate guest. In the automated payment system, smart retails system must ensures the working of payment gateway when ever required.
- The biggest threat to availability is "Denial of Service attack" or "Distributed Denial of Service" attack.
- Availability in the security can be assured using parallel resource availability through the development of distributed system and authenticated accesses.

4. **Trust and Privacy:**

- Trust management in the IoT can be defined as a trust in the IoT ensures that peoples and devices involved in IoT system accepts the services and information with the full faith.
- In the IoT, Trust management involves reliable data collection, reliable data fusion and mining with enhanced user privacy. Successful trust creation in the IoT ensures identity of the users/devices and quality of the IoT Services.

- Trust management involves behavior based trust control, reputation based trust control, and fuzzy approach for trust based access control.
- Some of the parameters that are used for the calculation of trust are frequency of answers, consistency of answer, physical proximity, common goals, common eco-system, history of interaction, availability, and common communities.
- Privacy in the IoT can be defined as a "not a single bit of information collected from the person/device get relive with any one without the knowledge of the person/device". Privacy preserving can be assured using various attribute based and anonymity based encryption algorithms.
- In the IoT, Most of the sensors those are part of public services gathers huge personal information. Hence, to whom this personal information should be shared, must be decided/known by the person himself.

5. **Authentication and Access control:**

   - Authentication can be defined as a "both the communicating parties are sure that they are communicating with the authenticated valid counter part". Successful authentication mechanism supports many other properties such as confidentiality, integrity and the availability.
   - Ensuring authentication in the IoT becomes critical due to heterogeneity in the number of devices and types of devices involved in the IoT. Each device transmitting data or want access of the data from other device must pass through authentication. Each user who want an ac-

cess of data from the sensing devices via gateway need to authenticate him self with the sensing device.

- Authentication in the IoT requires lightweight algorithms due to resource constraint environment of eco-system. Lightweight cryptography based on ECC and other methods provides solution for this challenge.
- Access control means "only authenticated users/devices have access of other device or personal data and control".
- In the IoT eco-system, access control schemes can be divided based on arbitrary access control, mandatory access control, and role based access control.

6. **Accountability:**

   - Accountability in the IoT means "Who has generated data and who have processed those data must be clearly defined" and are accountable for those data.
   - Accountability assures that which device has generated which data and which device processed which data in the IoT eco-system. It is an ability to hold users or devices responsible for their actions.
   - Accountability can be assured using proper identification mechanism for devices. Most of the devices in the ground level communicate using RFID technology or NFC technology, while edge level devices get IPv6 based IP Address as an identity for tracing.

7. **Auditability:**

Paper	Security challenges
Atzori et al. (2010)	Authentication, Data integrity, MITM attack, Sensor data protection, Key agreement, Secure cloud computing, Privacy and protection
Miorandi et al. (2012)	Data confidentiality, Device identity management, User identity management, Integrity of data, Accountability, Authentication, Access control
Borgia (2014)	Secure boot strapping of object, Secure transmissions, IoT data security, Authentication and authorization, Auditability, Access control
Jing et al. (2014)	RFID protocol security, RFID authentication, Key management, Designing lightweight solutions, handling massive heterogeneity, Access control, Node trust management
Granjal et al. (2015)	Confidentiality, Integrity, Trust, Authentication, Access control
Gil et al. (2016)	Privacy, Integrity, Access control, Trust, Identification, Authentication
Maple (2017)	Confidentiality, Integrity, Availability, Authentication, Non-reputation

TABLE 1.1
IoT Security Related Surveys

- Auditability can be defined as a "we must be able to perform consistent monitoring for the occurrence of each event in the system".
- In the IoT, for the ground level device actions can be monitored by keeping various sniffing tools such as wireshark at fog devices or data collecting devices.

8. **Non repudiation:**

- Non repudiation in authentication can be defined as a "any participating entity can not deny that the message sent by it is not generated by it".
- Non repudiation assures integrity and origin of message.

In this section, we have gone through various security challenges faced by IoT

eco-systems. The Table 1.1 presents survey papers related to IoT security and challenges discussed in those papers. The Table 1.1 presents that *"authentication"* is a key security challenge in IoT eco-system. The IoT eco-system consists of numerous applications and every applications faces different security challenges based on its functionalities. The next section provides overview on different IoT applications and related challenges face by them.

1.5 Applications

There are numerous applications of an IoT eco-system. Fig. 1.6 presents major IoT applications.

A Smart Home

The SH connects various home appliances and objects with the central cloud server through the Smart Home Gateway. The home appliances such as refrigerator, air-conditioning, television, door-security system, washing machine, environment controller, music system, CCTV camera, computer systems, Mobile devices, tablet devices, smart meters can be considered as an data generator in smart home. The short range communication protocols such as radio waves, power lines, wifi, zigbee, bluetooth, RFID and NFC are the basic communication protocols used in home automation. The smart home gateway receives data from the appliances and forward those data for further processing and monitoring. The smart home network can be integrated with other IoT applications such as smart grid, intelligent water and gas supply and so on. The smart home system suffers from numerous security challenges. The location information or light usage data reveal can

leak the information about home owner presence inside or not. Thus, privacy and authentication between devices and user must be taken care appropriate access control mechanism. Chapter 4 presents an authentication scheme between user and smart meter through smart home gateway for smart home energy monitoring.

B Smart Grid

The SG is a network of devices those interconnects energy producer and energy consumer. The fundamental objective of the smart grid network is efficient energy generation, reliable energy and data transmission and sustainable energy consumption. The smart grid must provide smooth operational control, efficient fault tolerant and reliable load balancing based on two way energy generation. The smart meter is a device that collects the data related to energy consumption at every unit time and transmits these data to the billing and monitoring server for the further processing. The smart grid user (or energy consumer) can also control the load and do the real time monitoring for its energy consumption data. The smart grid system with its sustainable objectives tries to integrate traditional energy sources with the renewable energy sources to touch a united nation's sustainable development goals fixed to be achieve by 2030. The confidentiality and integrity of the consumption data, availability of the uninterrupted service are the major challenges in terms of security in the smart grid. If the polynomial time adversary can change the consumption data or the required energy data then it can damage the complete grid system. Hence, the authentication and the identification of devices and users becomes key security requirement for the smart grid network. Chapter 3 presents an authentication scheme between user and smart

gateway that can be used for secure communication between user and the smart meter or between smart meter and the near by gateway users.

C Smart Industry

The SI or Industrial Internet of Things (IIoT) or industry 4.0 are the common names those aims to revolutionize the industrial production system with the objective of reducing human intervention. The industry 4.0 aims to connect all the industrial aspects from row material mining to final product distribution and that requires reliable and sustainable automation, communication and monitoring. The industry 4.0 aims to connect all the industries such as agriculture, manufacturing, aviation, chemical, steel, cement and so many more. The industry 4.0 system suffers from many social, economical, political and organizational challenges. The major security challenges in industry 4.0 to secure the data sensed by numerous sensors from the unauthorised entities and maintaining integrity of the data with the taking care privacy of customers. Chapter 5 and 6 presents an authentication scheme between user, gateway and sensing devices which is applicable to most of the smart industry in which industry owner or employee want to receive live data from the sensing devices based on allowed accesses.

D Smart Healthcare

The SHe connects healthcare entities such as doctors, patients, nurses, remote doctors, ambulances, local hospitals, remote hospitals, mobile hospitals, medical suppliers, health insurance companies and health care researchers. The recent report published by a global firm Research and Market predicts that the healthcare

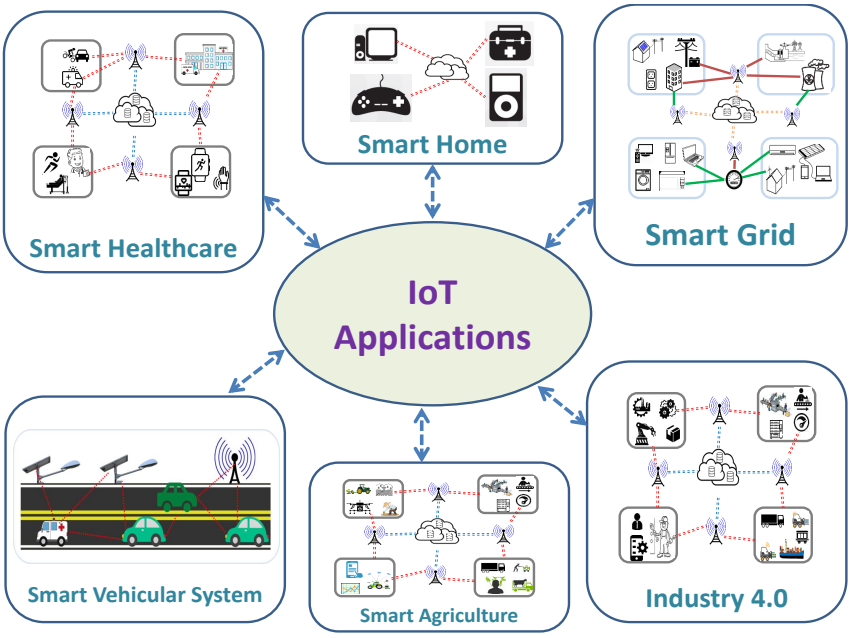

Figure 1.6. IoT Applications

market based on IT services and ICT infrastructure may reach from USD 187.6billion in 2019 to USD 390.7 billion by 2024. In 2021, still, there are many remote places in the world where achieving a good healthcare is a dream. In the smart healthcare system, doctors can do the realtime health monitoring of the patient and medicine intake through the wearable devices. The doctors can also get the information about past medical and medicine history of the patient through medical data processing. The major security challenge in the smart healthcare is to protect the health history of the data from the intruders. The data generated by the patient must be accessed by only authenticated doctors and hospitals. Hence, privacy and security are the key challenges those are required to handle in the smart healthcare. Chapter 5 and 6 presents an authentication scheme between user, gateway and sensing devices which is also applicable to healthcare industry in which doctor want to do realtime health monitoring of their patients.

E Smart Transportation System

The STS or Intelligent Transport System (ITS) is a set of road infrastructure, on road vehicles, air network, rail-way network, water transportation, planes, trains, and cruises. The basic objective of the smart transportation is to provide efficient, reliable and secure transport system with the dynamic navigation system to the people. The smart transportation includes GPS tracking, finding shortest route, identifying cheaper public transport, electric vehicles, accident monitoring and infrastructure for electric vehicles as a back-end tools. The smart transportation system handles vehicle to vehicle communication, vehicle to on road unit communication, vehicle to user communication, vehicle to road infrastructure communication and vehicle to cloud communication system. These all the communication need to be secured through better authentication between devices/units/users to achieve secure smart transportation system. Chapter 3 presents an authentication scheme between user and gateway devices that is applicable to smart transportation also where car owner sends data to near by STS gateway after successful authentication between them.

The fully secured "authentication" between devices and users in IoT ecosystem can solve many related security challenges. This thesis focuses on designing authentication protocols between user device - gateway device and user device - gateway device - sensor device. The next section discusses major research gaps, objectives and contributions followed by thesis preview.

1.6 Research gaps and contributions

Research gaps presents the existing problems available in the related field. Following research gaps were identified.

- Existing authentication schemes between user and gateway devices were not suitable for IoT environment because of high computation and communication costs.
- Exiting authentication schemes for IoT devices are not fully secured against well known security attacks and there is a need to protect an applications such as smart home and energy monitoring.
- In all, exiting authentication schemes, user need to register for each sensing devices that is ideally not applicable in realtime environment where thousands of devices are deployed. There is a need to design an authentication scheme that reduces registrations also.
- Existing authentication schemes are mostly based on simulation rather than realtime deployment of devices and run time authentication based implementations.

Next, we discuss major contributions of this thesis.

- Provide detailed literature review for existing authentication schemes for IoT communication and show the existing challenges.
- Propose novel and efficient authentication schemes for the U-GW (Chapter 3) and U-GW-S (Chapter 4) network model using ECC.

- Propose novel concept of level dependent authentication scheme (Chapter 5 and 6) using an ECC for the hierarchy oriented IoT communication.
- Provide security analysis for the proposed scheme using Dolev-Yao channel, ROR Model and BAN Logic (Chapter 3, 4, 5, 6).
- Provide comparative analysis for the proposed scheme with existing schemes based on communication cost and computation cost.
- Provide implementation using a realtime IoT test-beds.

1.7 Thesis Preview

In **Chapter 2**, we provide a conceptual understanding of the basic preliminaries used for designing and analysis of the proposed schemes. In this chapter, we provide a mathematical knowledge for ECC and Hash function. We also offer three security models that are used for security analysis of the proposed schemes. We discuss the Dolev-Yao model used for informal security analysis, and random oracle based ROR model for formal oracle security analysis and BAN logic for mutual authentication verification.

In **chapter 3**, we discuss the authentication scheme proposed for user-gateway based network model. This chapter provides a threat model, proposed scheme, security analysis using the Dolev-Yao model, ROR model and mutual authentication using BAN Logic. In this chapter, we also provide a comparative analysis of the proposed scheme with existing schemes based on communication cost, computation cost and networking parameters. The comparative analysis is followed by testbed design and implementation of the proposed scheme that is followed by a

conclusion and future work. In this chapter, we also provide applicability of the proposed scheme for the different IoT applications.

In **chapter 4**, we discuss the authentication scheme proposed for user-gateway-sensor based network model. This chapter provides a threat model, proposed scheme, security analysis using the Dolev-Yao model, ROR model and mutual authentication using BAN Logic. In this chapter, we also provide a comparative analysis of the proposed scheme with existing schemes based on communication cost, computation cost and networking parameters. The comparative analysis is followed by testbed design and implementation of the proposed scheme that is followed by a conclusion and future work. In this chapter, we also provide applicability of the proposed scheme for the different IoT applications with an example of smart home energy monitoring.

In **chapter 5**, we propose a novel concept of LDA. We start this chapter with an introduction to LDA and its conceptual discussion. We offer an LDA authentication scheme using ECC, password, and SC. We provide a security analysis using the Dolev-Yao model, ROR model and mutual authentication using BAN Logic. In this chapter, we also provide a comparative analysis of the proposed LDA scheme with existing schemes based on communication cost, computation cost and networking parameters. The comparative analysis is followed by testbed design and implementation of the proposed scheme that is followed by a summary of the chapter.

In **chapter 6**, we propose a novel concept of LDA using three factor based key exchange. We start this chapter with an introduction to LDA and its conceptual discussion. We offer an LDA authentication scheme using ECC, password, SC, and biometric. We provide a security analysis using the Dolev-Yao model, ROR

model and mutual authentication using BAN Logic. In this chapter, we also provide a comparative analysis of the proposed LDA scheme with existing schemes based on communication cost, computation cost and networking parameters. The comparative analysis is followed by testbed design and implementation of the proposed scheme that is followed by a summary of this chapter.

In **chapter 7**, we provide comprehensive conclusion for complete thesis and provide a future scope related to this thesis.

Chapter 2

Fundamental Preliminaries

This chapter discusses fundamental preliminaries used in this thesis. ***Section 2.1*** of this chapter provides basics about ECC. Next ***section 2.2*** provides overview on hash function and its properties. Furthermore, ***section 2.3*** presents fuzzy extractor that is used in designing of biometric based authentication schemes. Next ***Section 2.4*** provides discussion about security models which are used for security analysis of the proposed schemes in this thesis. ***Section 2.4*** provides introductory discussion about ROR model (***subsection A)*** and BAN Logic (***subsection B)***. At last, ***section 2.5*** summarise this chapter.

2.1 Elliptic Curve Cryptography

An ECC was proposed by Miller [Miller (1986)] and Koblitz [Koblitz (1987)] in 1986 and 1987 respectively. An ECC provides equal security to other public key crypto systems such as RSA with smaller key sizes. Following Table 2.1 shows that 160 bit key of an ECC can provide equal security to 1024 bit key of RSA

ECC Key Size (bits)	Equivalent RSA Key Size (bits)
160	1024
224	2048
256	3072
384	7680
521	15360

TABLE 2.1
RSA vs ECC

algorithm. The properties such as lower key size and low computational requirements made an ECC as a key player in designing of lightweight cryptography. We define an elliptic curve as an algebraic equation with non-repeatable roots:

$$Y^2 = (X^3 + aX + b) \; mod \; n \tag{2.1}$$

A curve can be defined over a finite field or binary field F of order n in the form E(F). In the above equation, two curve points are X and Y, which can be represented as P(X, Y). $a, b \in F_n$ are two constants their value must achieve,

$$4a^3 + 27b^2 \neq 0 \tag{2.2}$$

The elliptic curve performs two significant operations, point addition, and scalar point doubling. During implementation, we use the ECC's double and add method to perform a scalar multiplication between the sizeable random number and the curve point. We use a scalar point multiplication operation for the public key generation, that is computed as $n * P$ = P +...+ P for n times. An Elliptic curve point multiplication provides very high secrecy, and even if an adversary has $n * P$ and P, he can not extract the value of n in polynomial time. In ECC based authentica-

tion, devices generate the random number to use it as a secret key or private key that belongs to Z_p^*, where Z_p^* ranges between {1,2,3....,p-1}. We considered that selected value of random number is neither 1 nor p-1 but ranges between these two to avoid known value type attack from adversary.

A ECC Encryption and Decryption

ECC Encryption: The ECC encryption invokes an encoding for the message m in to the curve point P_m. For any random private key K_x generated by the user U_x, the relative public key $KP_x = K_x * G_p$, where G_p is any group point on the elliptic curve. To encrypt the P_m, user U_x selects the random number k and computes $C_m = (k*G_p, P_m + k * KP_y)$ where KP_y is a public key of the receiver U_y. User U_x sends C_m to U_y over a public channel.

ECC Decryption: The ECC decryption invokes a computation for the $P_m = P_m + k * (K_y * G_p) - (K_y*(k*G_p))$ where K_y is Y's private key. The advantage of adversary $\mathcal{A}$ in computing k from the $k*G_p$ can be defined as, $Adv_A^{Dec}(et) = Pr[Rand(k, G_p), \mathcal{A}: \chi_x]$ where $Adv_A^{Dec}(et) \leq \rho$, for any $\rho > 0$ and randomly generated pair (k, G_p) with execution time et in such a way that $\chi_x = k*G_p$.

B ECC Computational Problems

Elliptic Curve based Discrete Logarithm Problem (ECDLP) can be defined as: For given large prime number p and curve parameters a and b (Defined above), the generator point $G_p(X_p,Y_p) \in E_p(a,b)$ where $E_p(a,b)$ denotes the valid group of points on curve. For any generated random number R_i and computed $N_i = R_i * G_p \mod p$, an adversary advantage probability for $\mathcal{A}$ related to the ECDLP in finding

R_i based on given N_i and G_p defined by $Adv^E CDLP_A(et1) = \Pr[\text{Rand}(R_i)_{\mathcal{A}} : N_i = R_i * G_p]$ where $Adv^E CDLP_A(et1) \leq \chi$ for any $\chi > 0$ and Rand (R_i) implies that R_i is randomly generated by the adversary $\mathcal{A}$ within execution time *et1*. The (χ,et1) gives that an adversary $\mathcal{A}$ breaking the ECDLP such that $Adv^E CDLP_A(et1) \leq \chi$ with the maximum execution time *et1*.

Elliptic Curve based Diffie Hellman Problem (ECDHP) can be defined as : An ECDHP uses pair of public key and private key to generate a shared session key. Given a large prime number p and curve parameters a and b (Defined above), the generator point $G_p(X_p,Y_p) \in E_p(a,b)$ where $E_p(a,b)$ denotes the valid group of points on curve. For any generated random numbers R_i and R_j, compute $N_i = R_i * G_p \bmod p$ and $N_j = R_j * G_p$. For given value of N_i and N_j, it is computationally infeasible to compute $R_i{*}R_j{*}G_p$. An adversary's advantage probability for $\mathcal{A}$ related to the ECDHP in computing $R_i{*}R_j{*}G_p$ for given N_i and N_j defined by $Adv^E CDHP_A(et1) = \Pr[\mathcal{A} : = R_i{*}R_j{*}G_p]$ where $Adv^E CDHP_A(et1) \leq \chi$ for any $\chi > 0$ and Rand (R_i) and (R_j) implies that they are randomly generated by the adversary $\mathcal{A}$ within the execution time *et1*. The (χ,et1) gives that an adversary $\mathcal{A}$ breaking the ECDHP such that $Adv^E CDHP_A(et1) \leq \chi$ with the maximum execution time *et1*. Authors in [Patel & Doshi (2019)] have given detailed explanation about ECC including point multiplication, point addition, point doubling and so on. For implementation of our proposed schemes, we have used NIST recommended p-256 curve.

2.2 Hash Function

A cryptographic hash function H can be defined as $H : \{0,1\}^* \rightarrow \{0,1\}^n$. A function H takes and random length binary string $s \in \{0,1\}^*$ as input and generates a specific size binary string $B \in \{0,1\}^n$ as a product. H is a subordinate function used for the various cryptographic purposes like a key exchange, digital signature, authentication, random number generation, and many other cryptographic protocol developments. Every cryptographic hash function must satisfy properties such as pre-image resistant or one-way property, second pre-image resistant or weak collision property, fixed-size output, variable size input, efficiency, and randomness, collision-resistant or strong collision property [Stallings (2010)]:

1 **Variable Size Input:** Accepts variable size input.

2 **Efficiency and randomness:** It is near to impossible to generate same hash value or hash value in sequence.It will be completely random.

3 **Pre-image Resistant or one-way property:** If M is the input and hash value $h = H(M)$, then it is infeasible to find $h = h^*$ where $h^* = H(N)$.

4 **Fixed Size Output:** Out put size will be fix for same hash function and can be very for different hash functions.

5 **Second Pre-image Resistant or Weak Collision Property:** For any give M if $h = H(M)$ then it is computationally infeasible for third party attacker to find $N \neq M$ where $H(N) = H(M)$. This property protects against an attacker who knows input and its hash output, and wants to substitute other value as legitimate value in lieu of original. Hence, it achieves one degree of freedom.

6 **Collision Resistant or Strong Collision Property:** It is computationally infeasible for user itself to find pair of two values by M $\neq M^*$ such that it returns $H(M) = H(M^*)$. This property ensures that it is very difficult for user to find two input values with the same hash hence it achieves two degree of freedom. If hash is collision resistant then it is also second pre-image resistant.

For given n-bit hash value, it needs 2^n operations to get a weak collision and n $\geq$ 80 is sufficient size for weak collision. For strong collision, it requires $2^{n/2}$ operations to get a strong collision using the third birthday attack (birthday paradox) and it needs n $\geq$ 160 for to achieve strong collision resistance. There exists many cryptography hash functions. We can divide those functions based on either they are keyed hash function or unkeyed hash function. i.e. BLAKE2, BLAKE3, HMAC are examples of keyed hash functions while MD series, SHA series hash functions are examples of unkeyed hash function. In this thesis, SHA1 and SHA256 hash functions are used to compute hash values.

2.3 Fuzzy Extractor

A biometric-based security mechanism was adopted since long back due to the proven weak security mechanism of identity and password-based authentication. In biometric-based authentication, the user uses a finger-print or retina as a unique biometric input for verification. The security of stored biometric templates is a critical challenge in biometric-based security. A noisy nature of biometric does not suit the hash-based processing. Thus, in 2004, Dodis et al. Dodis et al. (2004) proposed a fuzzy extractor, which becomes a widely accepted biometric template protection mechanism. We consider following notations,

- $\mathcal{M}$ is a metric space with the distance function *Dis*: $\mathcal{M} * \mathcal{M} \rightarrow \mathbb{R}^+ =$ [0,∞]. The measurement of an "error" is occurred through hamming distance.
- $\mathcal{L}$ denotes length of biometric secret key.
- $\mathcal{T}$ denotes threshold value for error tolerance.
- Tuple (B_i,R_i,P_i) where B_i is a biometric for the i^{th} user, R_i is extracted random string defined as a $R_i \in 0,1^{\mathcal{L}}$, P_i is a public auxiliary string for the B_i.

Fuzzy Extractor is equipped with two algorithms called a a *Gen()* and *Rep()*. It can extract any random string from the given biometric template of a user i as input and produce the same random string even for the different biometric template from the same user i from the same trait.

- *Gen() :* Generation function is a uni-input and bio-output function that uses probabilistic generation procedure. It is defined as, $Gen(B_i) = (R_i,P_i)$. Over here R_i is a secret string while P_i is a public string.
- *Rep() :* Reproduction function is a bio-input and uni-output function that uses deterministic reproduction procedure. It is defined as, $Rep(B'_i, P_i) = (R_i)$. Over here, B'_i is a biometric template for same user i but obvious it will not be the same as B_i, thus it satisfies hamming distance $H_d(B_i,B'_i) \leq \mathcal{T}$.
- if (B_i,B'_i) belongs to the same user then the distance between B_i and B'_i will be low with high probability, $Pr[H_d(B_i,B'_i) < \mathcal{T}] \geq 1 - \varphi_{fn}$ where fn is "false negative".

- if (B_i,B_i') belongs to different user then the distance between B_i and B_i' will be high with low probability, $Pr[H_d(B_i,B_i') > \mathcal{T}'] \geq 1 - \varphi_{fp}$ where fp is "false positive" and $\mathcal{T}' > \mathcal{T}$.

2.4 Security Models

A ROR Model

In 2005, Abdalla et al. [Abdalla & Pointcheval (2005)] proposed the RealOr Random (ROR) model, which helps security designers to prove that the proposed password based scheme achieves polynomial-time security against an adversary $\mathcal{A}$'s advantage of breaking the security.

a. *Random Oracle:* The random oracle defined as a $H(.)$ also called as a hash function which takes message m_i as a input and computes the one-way irreversible output r_i. Whenever an adversary $\mathcal{A}$ generates a challenge with m_i, random oracle challenger $\mathcal{C}$ computes $r_i = H(m_i)$ and stores it in the list L initialized with NULL value as a pair of (m_i, r_i).

b. *Oracle Participants:* There are three participants in the proposed LDA-GIoT scheme, The user U_i, the gateway device GW, and the sensing device S_j.

c. *Oracles:* $\chi_{U_i}^p$, χ_{GW}^q, and $\chi_{S_j}^r$ are oracles with the instances p, q and r for the U_i, GW and S_j respectively, which are also called as a participants for the protocol $LDA-P$.

d. *Oracle Freshness* If using the reveal query $\mathcal{R}(\chi^x)$, an adversary $\mathcal{A}$ does not get success in receiving original session key $\mathcal{SK}$ then the oracles, $\chi_{U_i}^p$, χ_{GW}^q,

and $\chi^{r}_{S_j}$ are considered as a fresh oracles.

e. *Oracles Partnering:* Oracle instances χ^{x} and χ^{y} are called partner oracles if and only if they fulfill the following criteria simultaneously:

- Both instances χ^{x} and χ^{y} are in the acceptance state.
- Both χ^{x} and χ^{y} share the common session id *sid* and achieve the mutual authentication. "*sid*" is transcript of all the communicated messages between oracles.
- Both χ^{x} and χ^{y} satisfy the partner identification and vice-versa.
- No instance other than χ^{x} and χ^{y} accept with the partner identification equal to χ^{x} and χ^{y}.

f. *Adversary:* Let us assume that an adversary $\mathcal{A}$ is an eavesdropper who controls the complete communication channel defined over the Dolev-Yao model Dolev & Yao (1981). An adversary $\mathcal{A}$ can read, modify, inject, or fabricate the messages on the communication channel for the proposed network model. An adversary $\mathcal{A}$ has access for the following random oracle queries, which gives numerous capabilities to $\mathcal{A}$ for capturing and modifying the communicated messages and data.

1. $\mathcal{R}$ *(*χ^{x}*)* The *Reveal* query $\mathcal{R}$ provides current session key *SK* to the adversary $\mathcal{A}$ which is created by oracle instance χ^{x} and it's partnering instance.
2. $\mathcal{E}$ *(*χ^{x}*,*χ^{y}*)* The *Execute* query is formed as a passive attack on the communication between oracle participants χ^{x} and χ^{y}. This query provides all communicated messages to the adversary $\mathcal{A}$.

3. $\mathcal{S}$ (χ^x, m_i) The Send query is formed as an active attack performed by $\mathcal{A}$ on instance χ^x where χ^x can receive the message m_i as well as send the reply as a message m_i to $\mathcal{A}$.
4. *CorruptUserDevice*(χ^x) The *CorruptUserDevice* query models that the user U_i's device is available with $\mathcal{A}$ and $\mathcal{A}$ can capture all the data stored in it.
5. *CorruptSensingDevice* (χ^y) The *CorruptSensingDevice* query models that the sensing device S_j is available with $\mathcal{A}$ and $\mathcal{A}$ can capture all the data stored in it using power analysis or reverse engineering attack Messerges et al. (1999); Kocher et al. (1999).
6. *CorruptUserLevel* (χ^x) The *CorruptUserLevel* query models that the level of user U_i is available with an adversary $\mathcal{A}$.
7. *CorruptSensingLevel* (χ^y) The *CorruptSensingLevel* query models that the level of sensing device S_j is available with an adversary $\mathcal{A}$.
8. $\mathcal{T}$ (χ^x) Before starting of this oracle game, an unbiased coin b get tossed. The output of this toss decides the return value for the *Test* query $\mathcal{T}$. If the recently generated session key between the user U_i and the sensing device S_j is SK and an adversary $\mathcal{A}$ performs the test query on an instance χ^p which is the instance of U_i or its partner instance χ^r which is an instance of S_j then if the toss output is b = 1 then the participant instance χ^x returns an original session key. In contrast, if the output is b = 0, then the χ^x returns a random value of the session key SK's size to an adversary $\mathcal{A}$. If none of the condition matches, then an instance χ^x returns NULL. The semantic security of the session key is designed based on the *Test* query.

g. *Session key symmetric security:* The semantic security of the session key SK

generated between the user U_i and the sensing device S_j depends on an adversary $\mathcal{A}'s$ capability of indistinguishability between the actual session key and the random number. The output of a test query $\mathcal{T}$ depends on the value of *b'* guessed by an adversary $\mathcal{A}$. If the value of *b'* is similar to the value of *b* which is a hidden bit set by an oracle instance χ^x and used by $\mathcal{T}(\chi^x)$ to retrieve the original session key. Overall, the game depends on the correct guess by $\mathcal{A}$ for the bit *b*. If an adversary guesses the correct value of b, then it gets the correct session key.

Let $\mathcal{SC}$ define the position in which an adversary gets the success in this game. The advantage of an adversary $\mathcal{A}$ in capturing the correct session key *SK* for the proposed protocol LDA_P is defined as a Adv_p^{LDA}. Adv_p^{LDA} represents the success of an adversary, and if the Adv_p^{LDA} is negligible, then we can say that the proposed scheme is secured under the ROR model. Thus, we can define Adv_p^{LDA} as $Adv_p^{LDA}(\mathcal{A})$ = 2*Pr[$\mathcal{SC}$] - 1 which is similar to $Adv_p^{LDA}(\mathcal{A})$ = 2*Pr[b'= b] - 1. Where Pr[$\mathcal{SC}$] represents the probability for the success of an adversary $\mathcal{A}$. If we can prove that the Adv_p^{LDA} is negligible under the proposed scheme LDA_P, then we can say that the proposed scheme is secure.

Semantic Security for the Password based protocol: The semantic security for the password based protocol $LDA - P_{pw}$ defines an adversary $\mathcal{A}$'s capability of guessing the correct password. A password based protocol $LDA - P_{pw}$ is semantically secure if the advantage function $Adv_{LDA-P_{pw}}$ is negligible under the condition: $Adv_{LDA-P_{pw},|\mathcal{D}|}(\mathcal{A}) \geq \max(q_s, (\frac{1}{|\mathcal{DS}|}, \rho_{fp}))$. In this equation, q_s represents the number of send queries($\mathcal{S}$), $|\mathcal{DS}|$ shows the finite size of the password dictionary, ρ_{fp} shows probability of the false positive occurrence by

an adversary $\mathcal{A}$.

B BAN Logic

The BAN Logic is a tool that operates based on the proportional logic and is a widely accepted tool to prove the mutual authentication property of the authentication scheme [Burrows et al. (1989)].

Notations

In this subsection, we defines basic notations used during proof of authentication.

- $A \mid\equiv X$: Principal A believes that the statement X is true.
- $A \triangleleft X$: Principal A receives Message X. A can see this message and can operate on it. A can also send it to other principals.
- $A \mid\sim X$: Principal A once said message X, either now or sometime in the past, but it is true that A believes message X.
- $A \Rightarrow X$: Principal A has jurisdiction over statement X and all principals can trust A for the statement X.
- $\#(X)$: Message X is a recent message and the same message not sent in the past at any time of communication.
- $\langle X \rangle_Y$: The statement X is combined with the Y; thus, any principal say Y, then it will provide the identity of whoever says $\langle X \rangle_Y$. Therefore, Y can be used as proof of the origin of X.
- $\{X\}_k$: Statement X is encrypted using key k.

- $A \overset{k}{\longleftrightarrow} B$: Key k is shared between principals A and B.
- $\overset{K}{\longmapsto} A$: key K is public key of principal A.
- $A \overset{k_s}{\rightleftharpoons} B$: K_s is secret, which is only known by principals A and B. Either any one principal use this secret k_s to prove their identity to each other.

Inference Rules

In this subsection, we provide basic inference rules; using those rules, we derive certain properties like freshness and belief for authentication proof.

R_1. **Message Meaning Rule**, that says that if A receives message X that is encrypted by key K and the key is shared only with B then a message sent by B.

$$\frac{A|\equiv A \overset{K}{\longleftrightarrow} B, A \triangleleft \{X\}_K}{A|\equiv B|\sim X}$$

R_2. **Nonce verification rule** is used to prove that the message X is recent message and sender still believes on it.

$$\frac{A|\equiv \#(X), A|\equiv B|\sim X}{A|\equiv B|\equiv X}$$

R_3. **Jurisdiction rule** says that if A believes that B has jurisdiction over X then A trusts B on the correctness of message X.

$$\frac{A|\equiv Q \Rightarrow X, A|\equiv B|\equiv X}{A|\equiv X}$$

R_4. **Seeing rule** says that if message X is encrypted by shared key K then both, A and B can see X.

$$\frac{A|\equiv B\overset{K}{\longleftrightarrow}A, A\triangleleft\{X\}_K}{A\triangleleft X}$$

R_5. **Seeing rule** say that if A believes that K is a public key of B and X is encrypted by relative private key then A can see the message X.

$$\frac{A|\equiv\overset{K}{\longmapsto}B, A\triangleleft\{X\}_{K^{-1}}}{A\triangleleft X}$$

R_6. **Seeing rule** say that if A see (X,Y) then A sees X or A sees Y.

$$\frac{A\triangleleft(X,Y)}{A\triangleleft X}$$

R_7. **Fresh rule** says that if one part of formula (say X) is fresh, it means that entire formula is fresh.

$$\frac{A|\equiv\#(X)}{A|\equiv\#(X,Y)}$$

R_8. **Sensor belief rule** says that sensor believes on messages send by the user via gateway.

$$\frac{GWN|\equiv User|\equiv X, Sensor|\equiv GWN|\equiv X}{Sensor|\equiv User|\sim X}$$

R_9. **User belief rule** says that user believes on messages send by the sensor via a gateway device.

$$\frac{GWN|\equiv Sensor|\equiv X, User|\equiv GWN|\equiv X}{User|\equiv Sensor|\sim X}$$

2.5 Conclusion

In this chapter, basic preliminaries used for designing and analysis of the authentication schemes are discussed. This chapter discusses basics cryptography functions such as ECC, hash function, and fuzzy extractors as well as provides an overview on security analysis tools such as ROR model and BAN Logic.

Chapter 3

Secure Lightweight Key Exchange for User - Gateway Paradigm

This chapter proposes an authentication scheme between user device and the gateway device. ***Section 3.1*** presents introduction for the user-gateway model and proposed work. ***Section 3.2*** provides literature review related to proposed scheme. ***Section 3.3*** highlights threat model considered for designing authentication scheme. ***Section 3.4*** presents proposed authentication scheme with all the phases. ***Section 3.5*** put forward security analysis for the proposed scheme. ***section 3.6*** provides implementation aspects and use cases related to proposed work. ***section 3.7*** presents performance comparison of the proposed work with other existing schemes based on communication cost and computation cost. ***Section 3.8*** summarise this chapter with the limitations those are resolved in next chapter.

[1]**C. Patel and N. Doshi, "Secure Lightweight Key Exchange Using ECC for User-Gateway Paradigm", in IEEE Transactions on Computers, doi: 10.1109/TC.2020.3026027.**

3.1 Introduction

IoT communication includes three tier horizontal message transmission. The Tier one consists network of deployed tiny sensing devices, tier two consists intermediary devices like gateways, intelligent switches, servers, and so on. Tier three consists users of an IoT application. The sensor network provides a live data collection with the intelligent forwarding through edge devices to the cloud or to the server for storage and further processing. IoT also plays a significant role in industrial applications such as industrial monitoring, industrial data collection and analysis, product modeling, disaster prediction, medical applications, smart city design, and modeling.

In the user-gateway model (discussed in subsection A of section 1.3), the gateway device is lightweight (compared with the server) like raspberry pi and nodemcu. Thus, it is not elementary to get a quick response from the gateway device if we use conventional cryptographic approaches. In the proposed work, we consider generalized sensor-based IoT networks where sensors are tiny devices. The sensing devices in the proposed model collect data and publish those data to the nearest gateway device on the particular channel. The application user connects with the gateway device and subscribes to the channel on which the sensing device transmits data. We have implemented the proposed approach for the generalized IoT network using the MQTT protocol on the university campus.

3.2 Literature Review

ECC provides lightweight and secure computation with an equal level of security to the RSA and the other algorithms. In the recent past, authors in [X. Xu et al. (2013)], [Qiu et al. (2018)], [Chaudhry et al. (2015)] proposed the ECC based authentication schemes and proved that the proposed scheme provides security against most of the famous cryptographic attacks such as replay attack and MITM attack. In 2013, Xu et al. proposed an authentication scheme using ECC to secure key establishment between patient and health server. In this scheme, the authors shared user identity over an open channel, which makes it vulnerable to user impersonation attack and identity theft attack. In 2015, Chaudhry et al. [Chaudhry et al. (2015)] proposed an authentication scheme using ECC and temporary id and claimed that their scheme is secure against password guessing attack and user impersonation attack. But, in 2017, Qiu et al. [Qiu et al. (2018)] break the scheme proposed by Chaudhry et al. and proved that their scheme is vulnerable against both the attack. Furthermore, Qiu et al. also put forward a new authentication scheme. In 2015, Zhang et al. [Zhang & Zhu (2015)] and Qiu et al. [Qiu et al. (2018)] also proposed an authentication scheme using ECC and claimed novelty with security for their schemes.

3.3 Threat Model

The threat model considered for designing of the proposed scheme is referred from the model highlighted by Dolev and Yao in 1981 [Dolev & Yao (1981)], where abilities of an adversary $\mathcal{A}$ are discussed:

1 $\mathcal{A}$ can curb the communication channel and delete, modify, and study the data from the communication channel.

2 $\mathcal{A}$ can be either from the same network or from the different networks.

3 $\mathcal{A}$ can be an insider or can be an outsider of the system.

4 $\mathcal{A}$ can get the smart card as well as can read data from the smart card.

5 $\mathcal{A}$ can not extract a private key of the user or gateway.

6 $\mathcal{A}$ can generate pair of id and password using DIC_{ID} and DIC_{PW}.

7 $\mathcal{A}$ can either get the user's password through malicious card readers, or (ii) can read secret data of the smart card by side-channel attacks, but cannot get both simultaneously. Otherwise, it becomes a inconsequential case [Wang & Wang (2018)], [Dhillon & Kalra (2019)], [Sudhakar et al. (2020)].

3.4 Proposed Scheme

Following, Table 3.1 shows notations and abbreviations used in the proposed scheme:

A Initialization Phase

Step 1: The gateway device S_j generates the random number $k_s \in Z_p^*$. The gateway S_j computes public key parameter $PK_s = k_s * G$ and stores private key k_s and public key PK_s. The gateway device S_j generates the random number PR_{SC}. Here, PR_{SC} is the private key for the user device. Over here,

Symbols	Description
U	*User*
S	*Gateway*
r_i & n_p	*user random numbers*
$h(.)$	*One way Hash function*
T_S	*Time stamp*
k_s	*Gateway secret key*
ID	*user Identity*
SK	*Session key*
PW	*user Password*
PK_S	*Gateway Public key*
$Enc()$	*ECC Encryption operation*
$\Rightarrow$	*secure channel*
r_s & n_s	*Gateway random numbers*
$Dec()$	*ECC Decryption operation*
PU_{sc}	*User public key*
PR_{sc}	*User private key*
$\|\|$	*String concatenation*
$\mathscr{A}$	*Adversary*
$\rightarrow$	*Insecure channel*
SC	*Smart Card*

TABLE 3.1
Notation and Abbreviations

we assume that selected value of PR_{SC} is neither 1 nor p-1 but ranges between these two. The gateway S_j computes public key parameter PUB_{SC} = $PR_{SC} * G$ and stores private key PR_{SC} and public key PUB_{SC} in a secret memory of the user. The gateway S_j also makes PK_s as a publicly available parameter.

Step 2: Both the gateway S_j and the user U_i agrees on finite field p, elliptic curve $E_p()$, and generator point G.

B Registration Phase

USER PART 1 :

Step 1: U_i decides ID and PW_i. It also generates a random r_i such that $r_i \in Z_p^*$.

Step 2: U_i computes B = $r_i * G$, l_i = H($PW_i||B$) and forwards $\{l_i$,$ID\}$ to S_j via a secure channel.

GATEWAY PART 1 :

Step 1: S_j checks an identity ID and verifies its similarity in its record. If an identity ID is already registered then the gateway S_j asks for the different ID. Else, gateway device S_j continues computation.

Step 2: S_j computes A_i = H($H(ID) \oplus l_i$)

Step 3: S_j chooses random number r_s in such a way so that $r_s \in Z_p^*$ and computes T = H($(k_s + 1)||S_i$), $S_i = r_s$ * G, MID = $Enc_{PK_s}(ID||r_s)$, $O_i = T \oplus l_i$.

Step 4: S_j makes SC and keeps (S_i, MID ,A_i, O_i, H()) in it.

Step 5: S_j forwards generated SC to U_i through the protected channel.

USER PART 2 :

Step 1: U_i stores $Enc_{PUB_{SC}}$(B) in the SC.

Step 2: U_i stores SC = (O_i, S_i, MID ,A_i, H(), $Enc_{PUB_{SC}}$(B)).

C Login and Authentication Phase

This section presents login and authentication phase of the proposed scheme. Summary of authentication phase is given in Table 3.2.

USER PART 1 :

Step 1: U_i enter SC in SCR and also provides ID and PW_i. SCR computes l_i = H($PW_i||Dec_{PR_{SC}}(B)$), A_i^* = H(H(ID) $\oplus$ l_i). SCR verifies $A_i^* \stackrel{?}{=} A_i$, If, A_i is verified then SCR continues for further computations else SCR terminates computations.

Step 2: SCR generates random number n_p such that, $n_p \in Z_p^*$. SCR also computes $N_p = n_p * G$.

Step 3: SCR computes T = $O_i \oplus l_i$, L_i = H($N_p||ID$), PID = T $\oplus$ H($ID||L_i||TS$). Here, TS is the current time stamp of the user U_i.

Step 4: U_i sends {PID, N_p, TS, MID} to the gateway S_j over an insecure channel.

GATEWAY PART 1 :

Step 1: S_j gets current time-stamp TS^* and verifies $\Delta T \stackrel{?}{\leq} TS^*$ - TS. If this condition is satisfied then S_j continues computation else terminates system.

Step 2: S_j extracts an identity ID and the random number r_s by Dec_{k_s}(MID). S_j computes N_i = H($N_p||ID$), T = H($r_s||k_s+1$) and PID^* = T $\oplus$ H($ID||N_i||TS$).

Step 3: S_j verifies $PID^* \stackrel{?}{=} PID$. If this condition matches, then S_j continues with its computations else; if S_j gets the fake login requests for more than three times, then it concludes that this is modified replay attack and immediately blocks the SC for a day.

Step 4: Now, S_j erects new random n_s and enumerates NS = $n_s * G$. S_j computes a final session key, SK = H($T||n_s * N_p||N_s||ID||N_p$). It also computes SKV_i = H($SK||N_s||T||N_p||TS_{new}$) as a mutual authentication parameter.

Step 5: S_j sends N_s, SKV_i , TS_{new} to the user U_i over an open channel.

USER PART 2 :

Step 1: U_i gets current time-stamp TS^*_{new}, and verifies $\Delta T_{new} \stackrel{?}{\leq} TS^*_{new} - TS_{new}$. After the successful time-stamp verification, the user U_i computes the session key SK^* = H($ID||T||n_p * N_s||N_s||N_p$).

Step 2: U_i verifies the session key SK and verifies the gateway S_j

Step 3: U_i computes Q^*_i = H($N_s||T||N_p||TS_{new}||SK^*$) and checks Q^*_i.

Step 4: U_i verifies, $Q^*_i \stackrel{?}{=} SKV_i$. Successful verification of Q_i authenticates the gateway device S_j and validates the computed session key SK^*

D Password Update Phase

This section presents a password update phase for the proposed scheme.

USER PART 1:

Step 1: The U_i provides the Smart Card to the SCR and obtain password update step.

Step 2: The user provides ID, PW_i, PW_i^{new} to the SCR.

SCR PART 1:

User/SCR	**SCR**
Enter SC, ID and PW_i. SCR computes l_i = H(PW_i||$Dec_{PR_{SC}}(B)$), A_i^* = H(H(ID) $\oplus$ l_i). SCR verifies $A_i^* \stackrel{?}{=} A_i$	
Generate n_p,computes $N_p = n_p * G$, T = $O_i \oplus l_i$, L_i = H(N_p||ID), PID = T $\oplus$ H(ID||L_i||TS), Sends {PID, N_p, TS, MID} to GW.	Computes $\Delta T \stackrel{?}{\leq} TS^*$ - TS, Dec_{k_s}(MID), N_i = H(N_p||ID), T = H(r_s||$k_s + 1$) and PID^* = T $\oplus$ H(ID||N_i||TS), verifies $PID^* \stackrel{?}{=} PID$, Enumerates NS = $n_s * G$. S_j, SK = H(T||$n_s * N_p$||N_s||ID||N_p). It also computes SKV_i = H(SK||N_s||T||N_p||TS_{new}). sends N_s, SKV_i , TS_{new} to U_i
verifies $\Delta T_{new} \stackrel{?}{\leq} TS^*_{new} - TS_{new}$, computes SK^* = H(ID||T||$n_p * N_s$||N_s||N_p) and verifies. Q_i^* = H(N_s||T||N_p||TS_{new}||SK^*) and checks Q_i^*, $Q_i^* \stackrel{?}{=} SKV_i$.	

TABLE 3.2
Authentication and Key Exchange Phase

Step 1: The SCR gets B by the computation of $Dec_{PR_{SC}}$(B) and computes l_i = H(PW_i\|\|B),

Step 2: The SCR performs A_i^* = H($l_i \oplus H(ID)$) and verify $A_i^* \stackrel{?}{=} A_i$. If A_i is satisfied then the SCR continues,

Step 3: The SCR generates the random number $r_{i_{new}} \in Z_p^*$,

Step 4: Computes B_{new} = $r_{i_{new}}$ * G, l_i = H(PW_{new}\|\|B_{new}),

Step 5: Computes $O_{i_{new}} = l_i \oplus l_{i_{new}} \oplus O_i$,

Step 6: Computes $A_{i_{new}}$ = H($H(ID) \oplus l_{i_{new}}$),

Step 7: Now SCR updates A_i, O_i and generates updated SC = ($O_{i_{new}}$, S_i, MID ,$A_{i_{new}}$, H(), $Enc_{PUB_{SC}}(B_{new})$).

3.5 Security Analysis

A Informal Security Analysis

This section discusses informal security analysis for the proposed scheme. It highlight that the proposed scheme is secure against all well known attacks.

USER ANONYMITY

The proposed scheme do not communicate user identity over the public channel in the plain text format. U_i transmits the U_i's identity with the message PID. It generates PID using the hash function and the xor operation. Thus, it is infeasible for an intruder to retrieve an identity of the U_i. The proposed scheme achieves user anonymity from the adversary $\mathcal{A}$ as follow:

1. If an adversary $\mathcal{A}$ sniffs entire communication traffic then also $\mathcal{A}$ can not compute the ID because adversary $\mathcal{A}$ is utterly unaware about the parameter T. To computer the parameter T, an adversary $\mathcal{A}$ needs the secret key of S_j which is nearly impossible for the $\mathcal{A}$ to get it.

2. $\mathcal{A}$ needs correct random number n_p to compute parameter PID. It is almost impossible for the $\mathcal{A}$ to generate a same random number n_p in a polynomial time.

3. Even if an adversary $\mathcal{A}$ gets the value of A_i and O_i from a smart card, $\mathcal{A}$ can not compute the correct value of PID due to unawareness about a random parameter T and an encrypted parameter B.

Hence, the proposed scheme achieves user anonymity.

REPLAY ATTACKS

In the replay attack, an adversary $\mathcal{A}$ captures previous messages and replays those messages after some time. Thus, in this attack, an adversary $\mathcal{A}$ tries to pretend itself as a trusted registered entity. The proposed scheme achieves security from the replay attack by two means:

1. In the first, by using a random number n_p and n_s, which are non-repeated parameters.

2. In the second, it uses time-stamp TS for each message communication. The receiver of these messages verifies time-stamp using the pre-defined threshold.

OFF LINE PASSWORD GUESSING ATTACKS

The proposed scheme computes a smart card, SC = (O_i, S_i, MID ,A_i, H(), $Enc_{PUB_{SC}}$(B)). Overhear, B = r_i * G, that is encrypted using a public key of the user device. Thus, an adversary $\mathcal{A}$ can not get the value of B. Though, an adversary $\mathcal{A}$ presumes the correct identity ID and a password PW_i, $\mathcal{A}$ can not compute l_i = H($PW_i||Dec_{PR_{SC}}(B)$). Therefore, it is infeasible for the $\mathcal{A}$ to carry further session key computation. Similarly, if an adversary tries to go through the online password guessing attempts, then three consecutive wrong attempts will block the SC for a day. Therefore, the probability of winning the password guessing game by an adversary $\mathcal{A}$ becomes NULL. Thus, the proposed scheme is secured from the password guessing attack.

MUTUAL AUTHENTICATION AND MITM ATTACKS

The proposed scheme achieves mutual authentication through the gateway S_j's computation. The gateway S_j computes PID^* and compares PID^* with the received parameter PID during a login request from the user. The S_j verifies the identity and a password of the U_i. The U_i verifies the gateway by computing $Q_i^* \stackrel{?}{=} Q_i$, where Q_i, is computed using the session key, which is transmitted by the gateway S_j. Hence, the proposed scheme achieves mutual authentication. In the MITM attack, $\mathcal{A}$ tries to compute a session key using public parameters, public messages, and the stolen smart card. The proposed authentication scheme is secure against the MITM attack because $\mathcal{A}$ can not perform successful authentication with the correct session key generation due to ECC encryption and random numbers. Suppose an adversary $\mathcal{A}$ wants to authenticate itself with the user U_i, then $\mathcal{A}$ must require a secret key of the gateway as well as the generated random numbers. Similarly, when $\mathcal{A}$ tries to authenticate with the gateway, then $\mathcal{A}$ requires a user id, password file, and the value of B, which is impossible for an adversary $\mathcal{A}$ to get. Thus, it is quixotic for an adversary $\mathcal{A}$ to launch the MITM attack; hence, the proposed scheme is immune enough against the MITM attack.

PRIVILEGED INSIDER ATTACKS

Let us assume that the malicious system insider traces registration phase information and saves that data. Let us also presume that the inside attacker accesses all secret parameters such as ID, PW_i, and r_i. Then also, the insider can not compute the session key due to the run-time use of a random numbers such as n_p and n_s by both the communicating parties. The proposed scheme secures authentication

phase by using the hash function, ECC operations, and random numbers. Therefore, the proposed scheme achieves security from the privileged malicious insider attack.

PERFECT FORWARD SECRECY

In Perfect forward secrecy (PFS), even though the $\mathcal{A}$ receives all secrets and identities of the user, then also $\mathcal{A}$ can not compute the previous session key SK [Garg et al. (2020)]. The proposed model achieves PFS through the use of random numbers n_p and n_s. Even though an adversary $\mathcal{A}$ captures the gateway's secret key k_s and other public or private parameters, it is infeasible for an adversary $\mathcal{A}$ to guess the exact value of n_p, and n_s which were generated in the previous session and this proves inexpert for $\mathcal{A}$ to compute last session key. Therefore, the proposed scheme achieves perfect forward secrecy.

DENIAL OF SERVICE ATTACKS

By using DOS attack, the adversary $\mathcal{A}$ tries to block U_i and S_j from session key generation. The DoS attacker $\mathcal{A}$ may generate unnecessary traffic for any of the devices and create an undue delay in the authentication and key generation. In the realtime scenario, it is dishonesty to say that our scheme achieves complete security from the DoS attack, but the proposed scheme tries to protect it from the DoS attack through the blockage of SC and by using the time-stamp. Therefore, it somehow limits an adversary $\mathcal{A}$ to get control of the full system. The proposed scheme consider simulating it for the DDoS type attacks (like flooding) as a prospective aspect.

USER IMPERSONATION ATTACKS

In this attack, the adversary $\mathcal{A}$ tries to impersonate as a valid user and tries to generate a valid login request for the gateway device. To regenerate valid login request, $\mathcal{A}$ needs to generate a valid message {PID, N_p, TS, MID}. Over here, $\mathcal{A}$ can generate N_p (if aware of parameter G) and time-stamp TS. However, without knowledge of other secrets such as ID, PW_i and private key of the user device, an $\mathcal{A}$ can not generate parameter PID (where PID = T $\oplus$ H($ID||L_i||TS$)). To obtain the parameter MID, an $\mathcal{A}$ requires a physical smart card of the valid user U_i. It is a difficult task for an $\mathcal{A}$ to obtain all secrets of the user (like secretly shared private key of the user device and information stored in the smart card) at the same time. Thus, it is clear that the proposed scheme is secure enough against the user impersonation attack.

GATEWAY IMPERSONATION ATTACKS

In this attack, the adversary $\mathcal{A}$ tries to impersonate as a valid gateway device and tries to generate a valid reply on behalf of the authenticated gateway device. In the first reply, the gateway device sends { N_s, SKV_i , TS_{new}} to the user U_i. To regenerate the valid reply, an $\mathcal{A}$ generates time-stamp TS and parameter N_s (if aware about parameter G). However, an $\mathcal{A}$ also needs to compute valid SKV_i = H($SK||N_s||T||N_p||TS_{new}$). For this, an $\mathcal{A}$ requires knowledge about secret credentials such as user ID, server master secret k_s, and server random number r_s. It is highly infeasible for the $\mathcal{A}$ to capture all these parameters at the same time in the polynomial time. Therefore, it is clear that the proposed scheme is resistant enough against the gateway impersonation attack. Security comparison is

Scheme	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
[X. Xu et al. (2013)]	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
[Zhang & Zhu (2015)]	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
[Chaudhry et al. (2015)]	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
[Odelu et al. (2015)]	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
[Qiu et al. (2018)]	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 3.3
Security Comparison

shown in Table 3.3. Security attacks in Table 3.3 are notified as : **F1** = User anonymity, **F2**= Replay attack, **F3**= Offline Password guessing attack, **F4** = Mutual authentication and MITM attack,**F5** = Privilege insider attack,**F6** = Perfect forward secrecy,**F7** = Denial of Service attack, **F8** = User impersonation attack, **F9** = gateway impersonation attack, **F10** = Stolen smart card attack.

From the Table 3.3, it found that the schemes proposed by the Xu et al.'s [X. Xu et al. (2013)], Zhang et al.'s [Zhang & Zhu (2015)], Chaudhary et al.'s [Chaudhry et al. (2015)], Odelu et al.'s [Odelu et al. (2015)], and QIU et al.'s [Qiu et al. (2018)] suffer from the numerous attacks and vulnerabilities. An elliptic curve based on the scalar multiplication provides high security against the perfect forward secrecy and the related attacks due to its discrete logarithm properties. Thus, the proposed scheme provides a secure and computationally feasible scheme for the proposed network compares to other existing schemes.

STOLEN SMART CARD ATTACKS

In this attack, the adversary $\mathcal{A}$ captures the smart card of a valid user and tries to generate session key using smart card parameters and other public param-

eters. In the proposed scheme, the final SC available with the user is SC = (O_i, S_i, MID ,A_i, H(), $Enc_{PUB_{SC}}$(B)). To compute the final session key SK = H($T||n_s * N_p||N_s||ID||N_p$), an adversary $\mathcal{A}$ also requires secret credentials such as user ID, PW_i, server master secret k_s, server random number r_s, valid random numbers generated by user and gateway (n_p and n_s) and a private key of the user device. Thus, with the help of limited knowledge about the smart card and public parameter, an adversary $\mathcal{A}$ can not compute the final session key. Therefore, the proposed authentication scheme provides enough security against stolen smart card attacks.

B Mutual Authentication Using BAN Logic

The proposed scheme uses BAN logic for to prove that the proposed scheme achieves mutual authenticity property. It uses BAN Logic to verify the trust between principles involved in a communication. The BAN Logic focuses on the soundness and viability of the authentication process. The BAN Logic works based on several postulates, inference rules, and assumptions. The basic postulates and inference rules are defined in subsection B of section 2.4.

Assumptions

BAN Logic works based on a following assumptions:

- Secrets initially shared between each principal.
- Principals raise fresh nonces.
- Principals are faithful in specific ways.

- Principals can recognize their messages.
- If principal A assures that K is a public key, then A must be aware of the corresponding private key K^{-1}.

Goals

For any client-gateway architecture, followings are the fundamental goals those are necessary to achieve during the authentication process. Thus, with the help of individual assumptions, postulates, and inference rules, both client (C) and the gateway (S) try to attain specific goals, as shown in Table 3.4:

Goal No.	Goal
1	Client $C \mid\equiv C \overset{SessionKey}{\longleftrightarrow} S$
2	gateway $S \mid\equiv S \overset{SessionKey}{\longleftrightarrow} C$

TABLE 3.4
Goals

Theorem 1. *The proposed model achieves secure mutual authentication between the user U_i and the gateway S_j.*

Proof. The login and Authentication phase of a proposed scheme includes exchange of the messages those are rewritten in generic form as follow:

Message 1: $U_i \rightarrow S_j$: (H$(rd * G||(K_s + 1)) \oplus$ H($PW_i||Dec(ri * G)$) $\oplus$ H($PW_i||Dec(ri * G)$) $\oplus$ H($ID||H(n_p * G||ID)||TS_i$),$n_p * G$, TS_i,Enc($ID||r_s$))

Message 2: $S_j \rightarrow U_i$: ($n_s * G$, H(H$(ID||H(rs||K_s + 1)\ ||n_s * N_p||n_s * G||n_p * G)$ $||n_s * G||$ H$(rs||K_s + 1)\ ||n_p * G||TS_{new})$, TS_{new})

Idealized form: Above messages can be rewritten in idealized form as follow:

Message 1: $U_i \rightarrow S_j$: $\langle <((rs*G||(K_s+1)),(PW_i||Dec(ri*G))\,(PW_i||Dec(ri*G)),\,(ID||(n_p*G||ID)||TS_i),\,n_p*G,\,TS_i,\,\text{Enc}(ID||r_s)) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

Message 2: $S_j \rightarrow U_i$: $\langle <(n_s*G,\,((ID||(rs||K_s+1)\;||n_s*N_p||n_s*G||n_p*G)\;||n_s*G||\,(rs||K_s+1)\;||n_p*G||TS_{new}),\,TS_{new}) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

Goal: The Goals of a proposed scheme can be written as follow:

G1: $S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

G2: $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

Following assumptions are considered to prove the mutual authentication:

A1. $U_i \mid\equiv \#(TS_i)$

A2. $S_j \mid\equiv \#(TS_i)$

A3. $U_i \mid\equiv \#(TS_{new})$

A4. $S_j \mid\equiv \#(TS_{new})$

A5. $U_i \mid\equiv (U_i \overset{PK_s}{\longleftrightarrow} S_j)$

A6. $S_j \mid\equiv (U_i \overset{PK_s,D}{\longleftrightarrow} S_j)$

A7. $U_i \mid\equiv S_j \mid\sim (U_i \overset{SK}{\longleftrightarrow} S_j)$

A8. $S_j \mid\equiv \#(r_s,n_p,n_s)$

A9. $U_i \mid\equiv \#(r_s,n_p,n_s)$

A10. $U_i \mid\equiv \overset{PK_S}{\longmapsto} S_j$

Mutual authentication between user U_i and gateway GW_j is achieved as follows:

S1: from the message 1, it obtains,

$S_j \triangleleft \langle <((rd*G||(K_s+1)),(PW_i||Dec(ri*G))\,(PW_i||Dec(ri*G)),\,(ID||(n_p*G||ID)||TS_i),\,n_p*G,\,TS_i,\,\text{Enc}(ID||r_s)) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

S2: Using S1, R1 and A6, it procures,

$S_j \mid\equiv U_i \mid\sim \langle <((rs*G||(K_s+1)),(PW_i||Dec(ri*G))\,(PW_i||Dec(ri*G)),\,(ID||(n_p*G||ID)||TS_i),\,n_p*G,\,TS_i,\,\text{Enc}(ID||r_s)) > \rangle$

S3: Using S2, A2, R2, it comes by,

$S_j \mid\equiv U_i \mid\equiv \langle < ((rd*G||(K_s+1)), (PW_i||Dec(ri*G))\ (PW_i||Dec(ri*G)), (ID||(n_p*G||ID)||TS_i), n_p$*G, TS_i, Enc$(ID||r_s)) > \rangle$

S4: Using R7,R3 it obtains,

$GW_j \mid\equiv U_i \Rightarrow (rd*G||(K_s+1)), (PW_i||Dec(ri*G))\ (PW_i||Dec(ri*G)), (ID||(n_p*G||ID)||TS_i)$

S5: Using S3,S4, R8, it procures,

$GW_j \mid\equiv (\ rd*G||(K_s+1)\ )\ ,\ (\ PW_i||Dec(ri*G)\ )\ (\ PW_i||Dec(ri*G)\ ),\ (\ ID||(n_p*G||ID)||TS_i\ )$

S6: Using S1, S4, R2, A8, it comes by,

$S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ **[Goal 1]**

S7: using message 2, it obtains,

$U_i \lhd \langle < (n_s*G, ((ID||(rs||K_s+1)\ ||n_s*N_p||n_s*G||n_p*G)\ ||n_s*G||\ (rs||K_s+1)\ ||n_p*G||TS_{new}), TS_{new}) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

S8: Using S7, R1 and A6, it gets,

$U_i \mid\equiv S_j \mid\sim \langle < (n_s*G,\ ((ID||(rs||K_s+1)\ ||n_s*N_p||n_s*G||n_p*G)\ ||n_s*G||\ (rs||K_s+1)\ ||n_p*G||TS_{new}), TS_{new}) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

S9: Using S8, A2, R2, it gains,

$U_i \mid\equiv S_j \Rightarrow \langle < (n_s*G,\ ((ID||(rs||K_s+1)\ ||n_s*N_p||n_s*G||n_p*G)\ ||n_s*G||\ (rs||K_s+1)\ ||n_p*G||TS_{new}), TS_{new}) > \rangle_{U_i \overset{PK_s}{\longleftrightarrow} S_j}$

S10: Using S9, R7, R3, it gets,

$U_i \mid\equiv S_j \Rightarrow ((ID||(rs||K_s+1)\ ||n_s*N_p||n_s*G||n_p*G)\ ||n_s*G||\ (rs||K_s+1)\ ||n_p*G||TS_{new})$

$U_i \mid\equiv ((ID||(rs||K_s+1)\ ||n_s*N_p||n_s*G||n_p*G)\ ||n_s*G||\ (rs||K_s+1)\ ||n_p*G||TS_{new})$

S11: Using S10, S9, R8, it acquires,

$U_i \mid\equiv S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

S12: Using S8, S9, R2, A8, it earns,

$U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ **[Goal 2]**

Thus, the auspicious derivation of both the goals (goal one and goal two) proves that the proposed authentication scheme delivers mutual authentication property. □

C Formal security Analysis using ROR Model

The ROR model is one of the key formal and standard security models used to prove that an adversary $\mathcal{A}$ can not get confidence for the retrieved key using random oracle queries. An ultimate goal of the ROR model is to show that an adversary $\mathcal{A}$ can not distinguish between retrieved random value and the original session key.

Random oracle based ROR model

Random oracles and ROR model can be defined as follows:

Random Oracle: The proposed protocol P uses cryptographic public *hash* function that is formalized as a random oracle $\mathcal{H}(m)$. Thus, whenever any probabilistic polynomial adversary $\mathcal{A}$ communicates message m_i, then the oracle $\mathcal{H}(m_i)$ computes fix sized irreversible random value r_i. The oracle maintains a list L that is initialized with a "NULL" value. The oracle stores pair of (m_i,r_i) in L, for each i where i = 0 to n and returns value r_i to $\mathcal{A}$.

ROR Model: In the proposed RUA scheme, there are two participants U_i and

S_j. Thus, the ROR modeling for these participants and adversary $\mathcal{A}$ is as follow:

Oracles:→ $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ are oracles with instances m and n for U_i and S_j respectively. These oracles are also called participants for the proposed protocol P.

Oracles Freshness:→ $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ are called fresh oracles if the *reveal oracle query* $\mathcal{R}$ does not provided correct session key SK.

Oracles Partnering:→ The oracle instances $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ are called partners if they satisfy following conditions:

1. Both instances $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ are in the same acceptance state.
2. Both $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ share the common session id sid.
3. Both $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ satisfy the partner identification and viceversa.
4. No other instance other than $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$ is accepted with the same partner identification equal to $\mathcal{O}_{U_i}^m$ and $\mathcal{O}_{S_j}^n$.

Over here, sid is a transcript of all communicated messages between participants before achieving an acceptance state.

Oracle Accepted state:→ An instance $\mathcal{O}^x$ reaches to acceptance state after communicating the last message with the partner instance $\mathcal{O}^y$. The concatenation of all messages transmitted before reaching to acceptance state generates sid for a particular session.

Adversary Model:→ An adversary $\mathcal{A}$ can be derived using the famous *Dolev-Yao model* defined based on send (snd) and receive (rcv) channel. The adversary $\mathcal{A}$ can perform passive attacks as well as active attacks. The $\mathcal{A}$ can receive, read, update, delete, and add messages over an open communication channel.

Following random oracle queries are defined to provide access capabilities to a polynomial time adversary $\mathcal{A}$:

$\mathcal{R}(\mathcal{O}^m) :\rightarrow$ The *reveal* query $\mathcal{R}$ provides session key *SK* to the adversary $\mathcal{A}$ generated by an instance $\mathcal{O}^m$ and it's partner for the same *sid*.

$\mathcal{S}_{\rfloor}(\mathcal{O}^m_{U_i}) :\rightarrow$ The *send* query $\mathcal{S}_{\rfloor}$ can be formalized as an active attack by adversary $\mathcal{A}$. By using this query, an adversary $\mathcal{A}$ communicates with the user instance $\mathcal{O}^m$. The $\mathcal{A}$ sends messages to the user instance $\mathcal{O}^m$ as well as gets responce from it.

$\mathcal{S}_{\int}(\mathcal{O}^n_{S_j}) :\rightarrow$ The *send* query $\mathcal{S}_{\int}$ can be formalized as an active attack by the adversary $\mathcal{A}$. By using this query, an adversary $\mathcal{A}$ communicates with the gateway instance $\mathcal{O}^n$. $\mathcal{A}$ sends messages to the gateway instance $\mathcal{O}^n$ as well as gets responce from it.

$\mathcal{E}(\mathcal{O}^m, \mathcal{O}^n) :\rightarrow$ The *execute query* $\mathcal{E}$ can be formalized as a passive attack. By this query, an adversary $\mathcal{A}$ gets capacity to monitor the communication channel between two instances $\mathcal{O}^m$ and the $\mathcal{O}^n$. Thus, the *execute* query grants read permission to an adversary $\mathcal{A}$.

$\mathcal{T}(\mathcal{O}^m) :\rightarrow$ As a respond to this query, an $\mathcal{O}^m$ tosses unbiased coin and get value of bit *b*. The output of this toss decides return value of the *test* query $\mathcal{T}$. Let us consider *SK* as a newly generated session key between the user U_i and the gateway S_j. The adversary $\mathcal{A}$ performs the *test* query over either instance $\mathcal{O}^m$ or its partner instance $\mathcal{O}^n$. Then, if the output of toss is b = 1, then $\mathcal{O}^m$ returns the original session key else if b = 0, then the $\mathcal{O}^m$ returns a random value of the size equal to session key. If none of the condition matches then the instance $\mathcal{O}^m$ returns NULL.

$\mathcal{RSC}(\mathcal{O}^m_{U_i}) :\rightarrow$ The query *Read Smart Card (RSC)* enables an adversary $\mathcal{A}$

to extract records stored in the smart card SC. This query is formalized as an active attack. The Power analysis attack is the most famous attack using that an adversary $\mathcal{A}$ performs the $\mathcal{RSC}$ query. Therefore, this query returns all the parameters stored in the user U_i's smart card. This attack is considered as a stolen smart card attack.

Semantic Security in ROR Model: The ROR Model puts restrictions on the adversary $\mathcal{A}$ by limiting the attempts for performing the reveal query $\mathcal{R}$, $\mathcal{RSC}$. The semantic security also gives unlimited access for queries such as $\mathcal{T}$. The semantic security of the session key SK depends on an adversary $\mathcal{A}'s$ capability of indistinguishability between the random number and the actual session key.

The output of test query $\mathcal{T}$ depends on the value of b' guessed by an adversary $\mathcal{A}$. The bit b is hidden bit set by an oracle instance $\mathcal{O}^m$. The $\mathcal{T}(\mathcal{O}^m)$ uses it to retrieve the original session key. Overall game depends on the correct guess by an adversary $\mathcal{A}$ for the bit b'. The correct guess of b' (similar to b) provides the original session key to an adversary $\mathcal{A}$.

Let, the parameter $\mathcal{SUCC}$ defines the success position for an adversary $\mathcal{A}$. The advantage function for an adversary $\mathcal{A}$ can be defined as Adv_P. The Adv_P represents the success of an adversary $\mathcal{A}$ in the identification of the original session key. If it is proved that the parameter Adv_P is negligible under the proposed scheme, then it is considered that the proposed scheme is secured under the ROR model. The Adv_P is defined as follows:

$$Adv_P(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}] - 1 \tag{3.1}$$

The above equation can be rewritten as follow:

$$Adv_P(\mathcal{A}) = 2 * Pr[b' = b] - 1 \tag{3.2}$$

Here, Pr[$\mathcal{SUCC}$] represents the the success probability of an adversary $\mathcal{A}$.

Semantic Security for the Password based protocol The semantic security of the password based protocol P_{pw} defines an adversary $\mathcal{A}$'s capability of guessing the correct password. The password based protocol P_{pw} is semantically secure if the advantage function $Adv_{P_{pw}}$ is negligible under the following condition:

$$Adv_{P_{pw},|\mathcal{D}|}(\mathcal{A}) \geq max(q_s, (\frac{1}{|\mathcal{D}|}, \rho_{fp})) \tag{3.3}$$

In equation 6.3, q_s represents the number of send queries ($\mathcal{S}_s$), $|\mathcal{D}|$ shows finite size of the password dictionary, ρ_{fp} shows probability of the false positive occurrence by an adversary $\mathcal{A}$.

Formal security proof

The formal security proof for the proposed protocol P can be proved by the following theorem:

Theorem 2. *Let the adversary $\mathcal{A}$ runs against the proposed protocol P for a limited time period t_A. Let an adversary $\mathcal{A}$ tries to attack the proposed protocol P using oracle queries. Here, q_h defines the number of hash ($\mathcal{H}$) queries, q_s defines the number of send ($\mathcal{S}$) queries, and q_e defines the number of execute ($\mathcal{E}$). The*

proposed protocol is secured against oracle queries of $\mathcal{A}$ *if,*

$$Adv_P(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{D}|}, \rho_{fp})) + (\frac{q_s + q_e}{2^{l_r}}) \tag{3.4}$$

In Equation 6.4, l_h *is the size of the return value of hash (*$\mathcal{H}$*) query generated by* $\mathcal{A}$ *in bits,* l_r *is the size of the random nonce generated by the protocol P.* $|\mathcal{D}|$ *shows the finite size of a password dictionary and* ρ_{fp} *shows the probability of a false positive occurrence by* $\mathcal{A}$*.*

Proof. For to prove that the proposed protocol P is secure against an adversary $\mathcal{A}$ and $Adv_P(\mathcal{A})$ is negligible under the ROR model, four security games are defined which are called as a Gm_0, Gm_1, Gm_2, Gm_3. The oracle game start with the game Gm_0 and terminates with the game Gm_3. Now, let define an event $SUCC_i$ that represents the correct guess of bit b for each game Gm_i via the test query $\mathcal{T}$ by an adversary $\mathcal{A}$. □

Game Gm_0**:** Gm_0 is the initial game in which a real protocol P is equal to an initial game :

$$Adv_P(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}_0] - 1 \tag{3.5}$$

Game Gm_1**:**

Definition: This game executes following query,

- The execute query $\mathcal{E}(\mathcal{O}^m, \mathcal{O}^n)$ eavesdrops communication between the user U_i and the gateway S_j.

After receiving all the messages, adversary $\mathcal{A}$ performs test query $\mathcal{T}$ to acquire the session key. If the information retrieved using the above query provides sufficient

erudition to the $\mathcal{A}$ for computing SK, then it can be said that the adversary $\mathcal{A}$ wins the game Gm_1. Else:

$$Pr[\mathcal{SUCC}_1] = Pr[\mathcal{SUCC}_0] \tag{3.6}$$

Simulation of Gm_1: Gm_1 defines passive attack. Let us assume that the adversary $\mathcal{A}$ gets messages :

Message 1: $U_i \rightarrow S_j$: (H($ri * G||(K_s + 1)$) $\oplus$ H($PW_i||Dec(ri * G)$) $\oplus$ H($PW_i||Dec(ri * G)$) $\oplus$ H($ID||H(n_p * G||ID_i)||TS_i$), $n_p * G$, TS_i, Enc($ID||r_s$))

Message 2: $S_j \rightarrow U_i$: (n_s*G, H(H($ID||H(rs||K_s + 1)$ $||n_s * N_p||n_s * G||n_p * G)$ $||n_s * G||$ H($rs||K_s + 1$) $||n_p * G||TS_{new}$), TS_{new})

The user computes final session key as SK_u = H($ID||T||n_p * N_s||N_s||N_p$) and the gateway computes final session key as SK_s = H($ID||T||n_s * N_p||N_s||N_p$). The computation of SK depends on a randomly generated unknown size numbers. The session key computation depends on parameters such as ID, n_s, n_p, and T. Therefore, eavesdropping of the first message and the second message does not improve the winning probability of an adversary $\mathcal{A}$ for game Gm_1. Thus, it is said that the equation 6.6 holds true.

Game Gm_2:

Definition: The game Gm_2 executes following queries,

- *Hash* query $\mathcal{H}(\mathcal{O}^m, m_i)$ to retrieve the hash output of message m_i.
- *Send* query $\mathcal{S}_{\mathcal{S}}(\mathcal{O}^n_{S_j})$ to get messages from the gateway device.
- *Send* query $\mathcal{S}_{\mathcal{C}}(\mathcal{O}^m_{U_i})$ to get messages from the User device.

In this game, the adversary $\mathcal{A}$ performs hash query $\mathcal{H}$, send query $\mathcal{S}_{\mathcal{S}}$, and the send query $\mathcal{S}_{\mathcal{C}}$ with the motive of creating a collision and creating a fake trust.

The birthday paradox defines a probability of collision for the hash query $\mathcal{H}$ as at most $\frac{q_h^2}{2^{l_h}}$. As shown in the first message and the second message, each message in the communication uses either the random numbers, time-stamps, or long-term secrets. None of the messages is similar to another one. Therefore, the adversary $\mathcal{A}$ can not create a collision between messages. The first message contains a random number of r_i, while the second message contains a random number of r_s. The maximum collision probability between messages with the random numbers is at most $\frac{q_s+q_e}{2^{l_r}}$. Consequently, after this game, following result is achieved:

$$Pr[\mathcal{SUCC}_2] - Pr[\mathcal{SUCC}_1] \leq \frac{q_h^2}{2^{l_h}} + \frac{q_s + q_e}{2^{l_r}} \tag{3.7}$$

Game Gm_3**:**

Definition: This game executes the following query,

- Query $\mathcal{RSC}$ ($\mathcal{O}_{U_i}^m$) performs Read Smart Card operation. Using this an adversary $\mathcal{A}$ gets data from the smart card.

In game Gm_3, the adversary $\mathcal{A}$ reads data from the U_i's SC. The user U_i's SC stores (O_i, S_i,MID, A_i, H(.), Enc(B)). By using $\mathcal{RSC}$ query, an adversary $\mathcal{A}$ reads all these parameters and tries to compute the password PW_i. But the password PW_i is protected through a random number, and the probability of guessing a correct random number is almost NULL. Now an adversary tries to perform the online password guessing attack, but for a particular session, the number of attempts allowed to execute is restricted. Thus, it is said that:

$$Pr[\mathcal{SUCC}_3] - Pr[\mathcal{SUCC}_2] \leq max(q_s, (\frac{1}{|\mathcal{D}|}, \rho_{fp})) \tag{3.8}$$

The adversary tries all possible attack to break the proposed protocol P before finishing all the games. But, the equations 6.5, 6.6, 6.7, and 6.8 shows that the adversary does not achieve any success till the last game. Now, the only option left with the adversary $\mathcal{A}$ is to guess the correct value of b and to perform the test ($\mathcal{T}$) query on either the user U_i or the gateway S_j. The success probability for this guess is defined as follow:

$$Pr[\mathcal{SUCC}_3] = \frac{1}{2} \tag{3.9}$$

From Equations 6.5 and 6.6,

$$Adv_P(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}_1] - 1 \tag{3.10}$$

Using Equations 6.6, 6.7, 6.8, 6.9,

$$\begin{aligned} Pr[\mathcal{SUCC}_3] - Pr[\mathcal{SUCC}_1] \leq [Pr[\mathcal{SUCC}_2] - Pr[\mathcal{SUCC}_1]] \\ + [Pr[\mathcal{SUCC}_3] - Pr[\mathcal{SUCC}_2]] \end{aligned} \tag{3.11}$$

this is followed by,

$$\frac{Adv_P(\mathcal{A})}{2} \leq \frac{\frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{D}|}, \rho_{fp})) + (\frac{q_s+q_e}{2^{l_r}})}{2} \tag{3.12}$$

Thus,

$$Adv_P(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{D}|}, \rho_{fp})) + (\frac{q_s + q_e}{2^{l_r}}) \tag{3.13}$$

The equation 6.12 shows that the proposed protocol P is secure against all the oracles of an adversary $\mathcal{A}$ as well as secure under the ROR model. The game Gm_1 proves that even though an adversary $\mathcal{A}$ tracks all the publicly communicated

messages, he can not compute the original session key. The Gm_2 focuses on the hash collision and the security against the fake messages. The adversary $\mathcal{A}$ does not achieve success in the game Gm_2 due to frequent use of the random numbers, timestamps, and long term secrets in each message. The Gm_3 enables an adversary $\mathcal{A}$ to perform the active attack over the user's SC and allow him/her to retrieve the correct password PW_i using those data. The proposed protocol protects the password using the hash function and the random numbers. Thus, the adversary $\mathcal{A}$ does not get any success after Gm_3 also. Therefore,it is said that the proposed protocol is secured under the ROR formal security model.

3.6 Implementation and Testbeds

This section discusses the aspects related to the realtime implementation of the proposed scheme. The similar assumptions and the scenario discussed by authors in [Chaturvedi et al. (2017)], and [Radhakrishnan & Karuppiah (2018)] is considered. The system computes throughput, round trip delay, and the other results for the proposed scheme using an MQTT protocol. It uses the following parameters and scenarios for the implementation of a proposed scheme. The raspberry-pi as a gateway device and nodemcu as a sensing device are used to create a testbeds. To implement the MQTT protocol, the mosquitto broker was installed in the Linux and raspbian OS. A secure pipeline was designed using a Transport Layer Security (TLS) over MQTT protocol for secure communications.

In the proposed scheme, the JavaScript file is used as a *SC* for the authentication purpose. In the registration phase, the system user publishes his/her secret credential to the gateway device over an MQTT secure channel. The gateway de-

vice performs registration steps and creates a JavaScript (UID_sc.js) file for the user device. The gateway device publishes this file to the user device over an MQTT secure channel. The user device receives file and stores it into the device's secret memory (like secret memory of secure element). In the login and authentication phase, the user device reads stored JavaScript file whenever it needs to use SC.

Network Model	User - gateway
Protocol	Using MQTT
Broker	Mosquitto
Secure channel	By Enabling TLS communication in Mosquitto
ECC Curve	NIST P-256 Curve
ECC Multiplication	Using double and Add method
Message format	JSON Type
Language	Python
System	Architecture = x86-64, Processor: Intel (R) Core (TM) i5-7500 CPU with 3.40 GHz.

TABLE 3.5
Implementation Environment

The NodeMCU is used to connect with sensors as a publisher who publishes data to the directly connected gateway device. Users of the smart university application subscribe with the gateway device and try to receive the data from a gateway device. For better implementation, a laptop with the Linux OS is used as a user and raspberry-pi as a gateway device. The mobility of the three meters per second to 20 meters per second is considered for the users. The message size is computed in bits. For communication size computation, The MQTT packet header size is not considered. The size of the data part is only considered for the computation. The Table 3.5 provides configuration for the setup that is used for implementation of the proposed scheme. The MySQL server was setup to store

parameters and data. The Raspberry Pi 3 Model B+ was used as an intermediary gateway device as well as some user devices with the laptop. The wireshark tool was installed for the data collection of the MQTT packets. The Wireshark tool was used to trace the MQTT protocol in the .pcap extension file. The python program was used to perform the analysis and filtering of these messages. The Fig.3.1. shows derived session key between the user device (laptop) and a gateway device (raspberry pi) using the proposed protocol. For implementations we have used SHA-256 as a hash function and NIST recommended P-256 Curve. The security level achieved through implementation of proposed scheme is 128 bit.

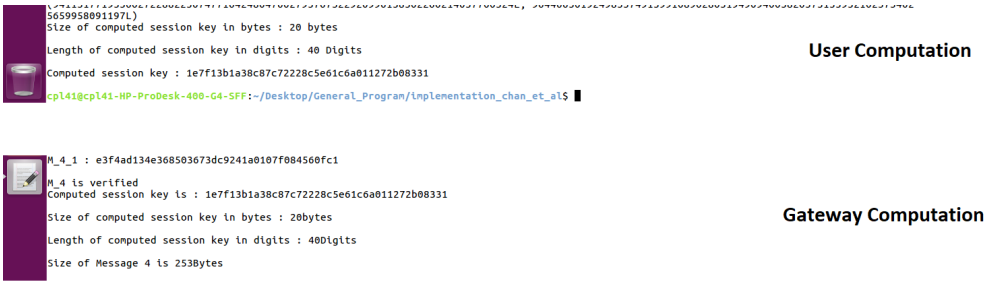

Figure 3.1. Computed Session Key

3.7 Use cases

This chapter proposes RUA scheme for the user-gateway model. The proposed scheme aims to provide a lightweight communication between the user device and the gateway device. The following are some of the significant use-cases of the proposed work.

- The proposed scheme can be used for secure and authenticated data exchange in the smart home area network.

- it can be also useful in smart factory, smart mall, smart agriculture and many more applications where it needs efficient and lightweight authentication mechanism between user (mobile/laptop) and the IoT device.

- If a case scenario of the *smart family health monitoring* application is considered in which the user (a doctor) wants to monitor the data published by a family member over the home gateway device. Then using the proposed RUA scheme, the registered doctor can login with the home gateway device and can monitor the realtime patient data.

- The proposed RUA scheme can be also useful for automated payment system in the shopping mall.

The novelty of the proposed work lies in its adaptive environment, lightweight computations and realtime implementations. The realtime implementation of the proposed ECC based scheme using MQTT protocol (that itself is a lightweight communication protocol used by the IoT industry) makes it novel. The realtime implementation using MQTT protocol has advantages like the lightweight header and reliable communication over other conventional protocols.

3.8 Comparative Analysis

In this section, the performance analysis for the proposed scheme is given by comparing it with the other existing schemes of Qiu et al's [Qiu et al. (2018)], Xu et al.'s [X. Xu et al. (2013)], Zhang et al.'s [Zhang & Zhu (2015)], Chaudhary et al.'s [Chaudhry et al. (2015)], and Odelu et al.'s [Odelu et al. (2015)].

A Network Parameter Analysis

As per shown in the Table 3.5, the widely adopted IoT protocol MQTT is used for our implementation. The MQTT protocol uses the publish-subscribe model for its communication in which the publisher publishes a sensing data to the broker, and the subscriber receives those data from the broker. For the computation of networking parameters, experiments were performed where the user device publishes an authentication request to the gateway device over an insecure channel.

1. **Average round-trip delay:** The round trip delay computed with the help of four major delays. These delays include queuing delay, processing delay, transmission delay, and the propagation delay. Thus, the overall round trip delay is the average time required by the user and the gateway to communicate the single message. Therefore, for one packet, it is (T_{rec} - T_{send}), and subsequently, for n packet, it is the sum of all. For the proposed setup, initially fifty requests were communicated from the ten users for the login and authentication using the MQTT protocol. As an outcome of this setup, it founds 0.2232s as an average RTT for the proposed scheme. If the communication cost is considered then the RTT for Xu et al. [X. Xu et al. (2013)], Zhang et al. [Zhang & Zhu (2015)], Chaudhary et al. [Chaudhry et al. (2015)], Odelu et al. [Odelu et al. (2015)] and Qiu et al. [Qiu et al. (2018)] will be approx 0.35s, 0.4s, 0.32s, 0.32s, 0.5s respectively. Thus, the average RTT for the proposed scheme is minimum compared to other existing schemes.

2. **Average throughput:** Throughput in the networking indicates an average

number of bits communicated per unit time. Thus, it is (number of transmitted packets * size of each delivered packet)/total delay. The computed throughput value for the proposed scheme, Xu et al., Zhang et al., Chaudhary et al., Odelu et al. and, Qiu et al. is 103 bits per second (bps), 123 bps, 123 bps, 133 bps, 143 bps, 126 bps respectively. Thus, the throughput of the proposed scheme is minimum compare to other existing schemes due to the more modest communication cost of the MQTT packets.

3. **Average packet loss:** The packet loss is computed based on the number of lost packets during the communication for a specific time. The mosquitto broker was used to set up an MQTT communication. Thus, it founds that the packet loss is rare. But if the global brokers such as Amazon AWS or hivemq are used, then the average packet loss of around 0.019 in total 1800 seconds for the proposed scenario. It is also computed for the other existing schemes. Then, it founds that there were 0.022, 0.022, 0.032, and 0.041, and 0.036 for Xu et al., Zhang et al., Chaudhary et al., Odelu et al., and Qiu et al. respectively. Sometimes, the inefficient implementation of the cryptographic algorithms increases packet loss, but more or less, the packet loss highly depends on the communication protocol and network infrastructure.

B Communication Cost

The communication cost in a security mechanism represents the number of bits communicated prior to the session key establishment. To perform the communication cost analysis, first, individual parameter size is considered in bits. The

Scheme	cost
[X. Xu et al. (2013)]	1184 bits
[Zhang & Zhu (2015)]	1184 bits
[Chaudhry et al. (2015)]	1344 bits
[Odelu et al. (2015)]	1600 bits
[Qiu et al. (2018)]	1280 bits
Ours	800 bits

TABLE 3.6
Comparison of the Communication Cost

computed size of identity and password is 160 bits; the hash output is 256 bits for SHA-256. The size of the randomly generated nonce is 128 bits, and the timestamp size is 32 bits. The ECC based implementations is used for the computation of the key parameters. Each point in the ECC has two coordinates, and each coordinate size is 160 bits. Thus, the size of one curve point P(X_p, Y_p) is 160 bits + 160 = 320 bits. Therefore, the public key size (PK_s) is 320 bits, while the private key(K_s) size is 160 bits. Table 3.6 shows comparative communication cost.

C Computation Cost

The computation cost comparison is performed based on the time consumed by various operations during the implementation of the proposed scheme. For the comparison, some operations such as xor and string concatenation operations are neglected due to the lower computation time. Major operations used for the comparison are as follows:

- T_h : Time required for the hash operation.
- T_{Pa} : Time required for the elliptic curve point addition.
- T_{Pm} : Time required for the elliptic curve point multiplication.

Scheme	User computation	Gateway computations	Total
[X. Xu et al. (2013)]	$6T_H + 3T_{Pm} \approx 6.6918$ ms	$5T_H + 3T_{Pm} \approx 6.6895$ ms	$11T_H + 6T_{Pm} \approx 13.3813$ ms
[Zhang & Zhu (2015)]	$7T_H + 3T_{Pm} + 1T_{Sym} \approx 6.6987$ ms	$7T_H + 3T_{Pm} + 1T_{Sym} \approx 6.6987$ ms	$14T_H + 6T_{Pm} + 2T_{Sym} \approx 13.3974$ ms
[Chaudhry et al. (2015)]	$5T_H + 4T_{Pm} \approx 8.9155$ ms	$4T_H + 3T_{Pm} \approx 6.6872$ ms	$9T_H + 7T_{Pm} \approx 15.6083$ ms
[Odelu et al. (2015)]	$7T_H + 3T_{Pm} + 1T_{Sym} \approx 6.6987$ ms	$6T_H + 2T_{Pm} + 2T_{Sym} \approx 4.475$ ms	$13T_H + 5T_{Pm} + 3T_{Sym} \approx 11.1737$ ms
[Qiu et al. (2018)]	$8T_h + 2T_{Pm} \approx 4.4704$ ms	$5T_h + 2T_{Pm} \approx 4.4635$ ms	$13T_h + 4T_{Pm} \approx 8.9339$ ms
Proposed	$6T_H + 2T_{Pm} + 1T_{Sym} \approx 4.4704$ ms	$5T_H + 2T_{Pm} + 1T_{Sym} \approx 4.4681$ ms	$11T_H + 4T_{Pm} + 2T_{Sym} \approx 8.9385$ ms

TABLE 3.7
Comparison of the Computation Cost

- T_{Sym} : Time required for the symmetric encryption / decryption operation.

For the scenario of Table 3.5, the time required for the above four operations is 0.0023 ms, 0.028 ms, 2.226 ms, 0.0046 ms, respectively. The Table 3.7 highlights computation cost comparison for the proposed scheme with the other existing scheme.

As per shown in the Table 3.7, the total time required by the Xu et al.'s [X. Xu et al. (2013)], Zhang et al's [Zhang & Zhu (2015)], Chaudhary et al.'s [Chaudhry et al. (2015)], Odelu et al.'s [Odelu et al. (2015)], QIU et al.'s [Qiu et al. (2018)] and the proposed scheme is 13.3813 ms, 13.3974 ms, 15.6083 ms, 11.1737 ms, 8.9393 ms, 8.9385 ms respectively. It shows that the computation time required by the proposed scheme is minimum compared to other existing schemes. Following

Fig.3.2. gives performance comparison of communication cost and computation cost of the proposed scheme with the existing schemes.

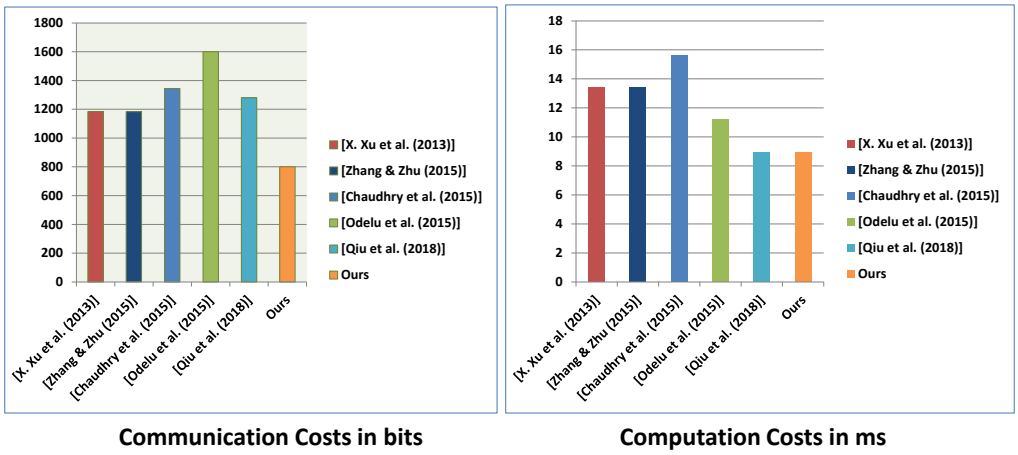

Figure 3.2. Performance Comparison Chart

3.9 Summary

Authentication is one of the critical challenges in the IoT environment. To address these challenges, many authors proposed authentication schemes. This chapter proposes a novel authentication scheme for the User-Gateway based communication model. This model is widely used in the IoT and the sensor networks. The proposed authentication scheme uses identity, password, and the smart card as a factors. The ECC is used for the key generation. After successful registration, the application user performs login and authentication to establish a secure session key. After the successful mutual authentication, the application user determines a secure session key with the cluster gateway device. The security is proved for the proposed key exchange scheme using the formal security analysis as well as informal security analysis. A random oracle based ROR model

and the BAN Logic are used for the formal analysis and the Dolev-Yao channel for informal analysis of the proposed scheme. The proposed scheme was implemented for a realtime scenario generated on the university campus. The session key between the users (laptop and raspberry pi devices) and the gateway device (raspberry-pi) was generated for the further communication. In this chapter, the comparison of the proposed scheme with the other existing schemes is also provided in terms of the communication cost, computation cost, throughput, RTT, and the packet loss. Overall, this chapter proposes reliable and lightweight RUA scheme for user-gateway communication. The proposed scheme in this chapter considers sensing devices as a tiny devices which are not capable to do cryptography operations but now a days we find many sensing devices those are capable to perform cryptography operations and can play an important role in session key derivation. Hence, next chapter 4 proposes an authentication scheme between the user device, gateway device and the sensing device.

Chapter 4

Secure Lightweight Key Exchange for User - Gateway - Sensor Paradigm

This chapter proposes an authentication scheme between user device, gateway device and the sensing device. ***Section 4.1*** presents introduction for the user-gateway model and proposed work. ***Section 4.2*** provides literature review related to proposed scheme. ***Section 4.3*** highlights threat model considered for designing authentication scheme. ***Section 4.4*** presents proposed authentication scheme with all the phases.

Section 4.5 put forward security analysis for the proposed scheme. ***Section 4.6*** provides implementation aspects and use cases related to proposed work. ***Section 4.7*** presents performance comparison of the proposed work with other existing schemes based on communication cost and computation cost. ***Section 4.8*** sum-

[1]**Patel C., Doshi N. (2020) A Novel Lightweight Authentication for Intelligent Energy Monitoring in Smart Home. Advances in Intelligent Systems and Computing, vol 1148. Springer, Singapore. https://doi.org/10.1007/978-981-15-3914-5**$_{1}$7

marise this chapter with the limitations those are resolved in next chapter.

4.1 Introduction

The Sensing devices are very tiny devices that are not capable of storing a large amount of data, not competent to perform a higher level of computations, and not capable of transmitting the data for a longer distance. The major wireless communication protocols which are used by the sensing devices are BLE, Z-Wave, Zigbee (make use of IEEE 802.15.4), NFC, RFID and so on. These protocols can transmit the data for a short-range (100 meter), so the sensing devices are connected with micro-controllers or micro-processors (like Arduino, Intel Galileo, Raspberry-Pi, NodeMCU) through either a wireless channel or a wired channel. These micro-controllers and micro-processors communicate through the Internet using a WiFi or a Wired LAN. The Internet Protocol (IP) or 6LoWPAN which is adaption of IPv6 for low power networks is the primary communication protocols that are used by these controllers and processors.

The application users can use the protocols like a HTTP, MQTT, CoAP and XMPP for receiving the data from a cloud or directly from the sensing devices. The meaningful communication in IoT based application occurs through insecure channels or open channels (like normal MQTT or CoAP) that connect sensing devices with the users via a gateway and the cloud servers [Patel & Doshi (2019)]. This chapter considers network model discussed in subsection A of section 1.3 . For designing of the authentication scheme, this chapter considers use case of smart home energy monitoring. The Smart Home is an intelligent staying location that is built up through an interconnection of appliances like a Smart light, Smart

Fridge, Smart TV, Smart Door and Window system, Smart locking system, and so on. These devices are called smart sensing devices. These sensing devices perform their tasks by consuming electricity. These all Internet enabled devices share their electricity consumption with the SM (Ex. Smart bulb with motion sensor which get on only when motion is there else go to sleep mode and reduce power consumption).

The SM collects data related to electricity consumption from each sensing device at a particular time interval and forwards those data through the Internet to smart home users via a gateway device. Let us denote the smart home user devices as a SHU_i in which i represents the i^{th} user, the sensing devices as SD_j in which j represents the j^{th} sensing device, the smart meter as a SM and the gateway device as GW. The communication between SD_j to SM is considered as a secure communication because many times, this communication also occurred in a wired environment. The remaining communication of SM to GW and GW to SHU_i is considered as an insecure communication. This chapter presents a lightweight and secure authenticated key exchange mechanism between the User SHU_i, the gateway GW, and the smart meter SM by considering SM as a resource constraint device like sensing devices.

4.2 Literature Review

Sensing devices (SHD_j where j ranges between 1 to number of smart devices in home), communicates data with the near by gateway (SH_{gw_k} where k ranges between 1 to number of smart home gateway). Ex. Smart home gateway SH_{gw_k} some time works as a fog device which processes a messed up data received from SH_{gw_k}

or some time works as just a forwarder of data. The SH_{gw_k} transmits this data to central service provider which is also called as cloud service provider (SH_{csp}). User SHU_i (Where i ranges between 1 to number of home users) communicates via GSM (4G/5G) with SH_{csp} for receiving data from the SHD_j. Secure as well as privacy preserving authentication and key-exchange is necessary between all these entity to achieve the basic security goals. An authentication protocol designer need to keep in mind that here, SHD_j is a resource constraint device and is not capable to perform complected cryptography operations that much. Most of the authentication protocol designed for models such as smart home aims to generate secure SK through SHU_i registration phase, SHD_j registration phase, SH_{gw_k} initialize as well as registration phase. secure SHU_i login and authentication phase.

Kumar et al. [Kumar et al. (2015)] proposed lightweight authentication scheme for smart home model. Authors in [Kumar et al. (2015)] designed a authentication scheme between smart device and gateway using service provider server. Service provider server generates symmetric key parameters for further computations. Authors provided simulation of proposed protocol using TinyOS and formal security analysis using AVISPA tool.

Kang et al. in [Kang et al. (2016)] proposed zero-knowledge based authentication scheme. Kang et al. also followed similar network model to kumar et al. [Kang et al. (2016)] in which smart home gateway initiates communication with service provider server followed by user request to service provider. Author in [Kang et al. (2016)] claims that their proposed scheme is secured against SA_1, SA_11 and satisfy the SG_1,SG_2, SG_3 and SG_4. During study of this literature, it is observed that author had not discussed any simulation, implementation and formal

security analysis for proposed scheme.

Authors in [Wazid et al. (2017)] proposed authentication scheme between user, gateway and smart device using registration authority. During proposed scheme designing Wazid et al. assumed that gateway node and registration authority are the secured and trusted device. Authors used AVISPA for security, ROR for formal security model and NS2 for simulation.

Farash et al. in [Farash et al. (2016)] proposed an authentication scheme for the IoT network using only XOR and Hash functions. Proposed scheme in this paper is having four phases: i.e. pre-deployment phase, registration phase, login and authentication phase and password deployment phase. Through informal security analysis, authors in this paper proved that their scheme is secured against attacks such as MITM, replay, stolen verifier, and impersonation attack. Authors also proved mutual authentication using BAN logic and AVISPA tool. The proposed scheme in this chapter is compared with this paper and proved that is more secure and cost-efficient compare to scheme proposed by Farash et al.in [Farash et al. (2016)]. Wu et al. in [Wu et al. (2017)], presented authentication scheme for the WSN based IoT deployment and provided NS2 based simulation for the proposed work. Authors of this paper proved formal security analysis using ProVerif tool. Through informal security analysis, author proved that their scheme is secured against forgery attacks, tracing attacks and guessing attacks.

Amin et al. in [Amin et al. (2018)], proposed an authentication scheme for the U-GW-sensor based network model used in healthcare system. The proposed scheme consists of setup phase, registration phase, login and authentication phase and password change phase. Author provided simulation for the proposed scheme using AVISPA and proved mutual authentication using BAN Logic. Through in-

formal security analysis, author proved security of the proposed scheme against anonymity, stolen attack, untraceability, and guessing attacks. Authors in [Amin et al. (2018)] proved efficiency of the proposed scheme by comparing it with other existing schemes based on computation cost and communication cost. Through comparison, it is proved that the proposed scheme of this chapter is more computationally feasible compare to [Amin et al. (2018)].

In 2019, Zhou et al. [Zhou et al. (2019)] proposed a key agreement scheme for the IoT environment that also supported privacy preservation and proved formal security analysis using random oracle and mutual authentication using BAN Logic. Through the comparative analysis based on communication cost and computation cost, it is proved that proposed scheme of this chapter is more computationally feasible compare to the scheme proposed in [Zhou et al. (2019)].

In 2019, Poh et al. [Poh et al. (2019)] proposed an authentication scheme for the smart home environment that is based on U-GW-Sensor model. Authors in this paper done an assumption that the gateway device is a semi-trusted device with that they also considered that the secret key of a gateway device is stored securely. In this paper, setup phase and authentication phases are discussed for key setup. Authors in this paper proved confidentiality, authentication and privacy preservation of their scheme and provided comparative analysis with existing schemes to prove computational efficiency.

4.3 Threat Model

The adversary model describes the capability of an adversary $\mathcal{A}$. The adversary $\mathcal{A}$ makes use of these capabilities to break the security of the proposed scheme. The

design of a strong adversary model helps a security researcher to focus on designing of foolproof security mechanism. For the proposed authentication scheme, following adversary model is considered.

1 An adversary $\mathcal{A}$ has full access on the communication channel between User - Gateway and Gateway - Smart Meter.

2 An adversary $\mathcal{A}$ can compute a valid pair of $identity * password$ offline in polynomial time.

3 An adversary $\mathcal{A}$ may get a previously calculated session key.

4 An adversary $\mathcal{A}$ can get either information from the smart card or can get a password of the user. He can't get both info at the same time.

5 An adversary A can compromise a sensor node physically and can get a piece of stored information.

4.4 Proposed Scheme

This section discusses the proposed scheme which includes the system initialization phase by gateway GW, the registration phase by the user SHU_i and the smart meter SM, and at last the authentication and key establishment phase by an user SHU_i and an intelligent meter SM through a gateway GW. The proposed scheme makes use of the symbols and notations highlighted in Table 4.1.

Symbol	*Description*	*Symbol*	*Description*
UID	User ID	GWID	Gateway ID
SMID	Smart Meter ID	r_i	Random number
T_i	Time stamp	P	Curve Point
$\bigoplus$	XOR	SK_{usm}	Session key

TABLE 4.1
Notation and Abbreviations

A Initialization Phase

The system initialization phase is performed offline by the gateway GW. The gateway device GW generates a random master key M_k of 160 bit size. The gateway GW computes public key $P_k = M_k$*P where P is a generator point of elliptic curve. The gateway computes $\chi_i = \mathrm{H}(GWID||UID_i||M_k)$ and $\rho_j = \mathrm{H}(GWID||SMID||M_k)$ where $GWID$ is the gateway identity, UID_i is the identity of the i^{th} user and $SMID$ is the identity of the smart meter. These identities are generated randomly by the gateway GW. The GW stores the UID_i, $GWID$ and χ_i in SHU_is temper proof memory and $SMID$, $GWID$ and ρ_j in the SMs temper proof memory. The GW stores (UID_i,χ_i) and ($SMID$,ρ_j) in its own secret memory.

B Registration Phase

In the registration phase, the user SHU_i and the smart meter SM register with the gateway GW through their identities.

User Registration

The user device sends the UID_i and $Temp = \mathrm{H}(UID_i||\chi_i)$ to the gateway through the secure channel. The gateway device verifies the $Temp^* = \mathrm{H}(UID_i||H(GWID||$

$UID_i||M_k)) \stackrel{?}{=} Temp$. Now the gateway computes $U_{i_{pub}} = (\chi_i + M_k)$*P and $U_{i_{Pri}} = \frac{1}{\chi_i + M_k}$ as a public parameter and private parameter for the i^{th} user respectively. After the successful computation, the gateway GW provides (F_p, P, E, n, H(.),$U_{i_{pub}}$,$U_{i_{Pri}}$, UID_i, $SMID$, $GWID$) to the user. The User SHU_i stores these parameters with χ_i in its temper proof memory.

Smart Meter Registration

The smart meter sends the $SMID$ and $Temp_1 = \text{H}(SMID||\rho_j)$ to the gateway through the secure channel. The gateway device verifies the $Temp_1^* = \text{H}(SMID||$ $H(GWID||SMID||M_k)) \stackrel{?}{=} Temp_1$. Now the smart meter computes $SM_{pub} = (\rho +$ $M_k)$*P and $SM_{Pri} = \frac{1}{\rho_i + M_k}$ as a public parameter and a private parameter for the smart meter SM respectively. After the successful computation, the gateway GW provides (F_p, P, E, n, $SMID_t$, ID_{SM_t} H(.),SM_{pub},SM_{Pri}, UID_i, $SMID$, $GWID$) to the smart meter. The smart meter SM stores these parameters with ρ_j in its temper proof memory. Here the $SMID_t$ is the secret token for the smart meter and the ID_{SM_t} is the token identifier for SM.

C Authentication and Key Exchange Phase

In the authentication and key establishment phase, the user SHU_i establishes a shared session key SK_{usm} to receive the data from the smart meter SM. The authentication and key-establishment phase is performed through an unconfident channel, and the whole communication of this phase is available to an intruder $\mathcal{A}$. Summary of authentication phase is given in Table 4.2.

User $SHU_i \xrightarrow{Message1}$ **Gateway** GW**:**

The user SHU_i selects the random number r_1 and computes the $X_1 = r_1$*P and $X_2 = X_1 * U_{i_{Pri}}$. The SHU_i computes the parameter $M_1 = \mathrm{H}(UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1)$, $M_2 = UID_i \bigoplus \chi_i$, $M_3 = SMID \bigoplus \chi_i$. The user SHU_i creates a first request as a *message 1* = (X_1, M_1,M_2,M_3,T_1) and sends the *message 1* to the gateway GW. Here T_1 is the current time stamp. It is assumed that the clock between SHU_i, GW and SM is synchronized.

Gateway $GW \xrightarrow{message2}$ **Smart Meter** SM**:**

The gateway GW verifies the time-stamp T_1 by verifying $\Delta T \leq T_1^*$-T_1 where ΔT is the pre-defined timing threshold and T_1^* is the current time-stamp at gateway. The gateway GW retrieves UID_i till it successfully performs $UID_i^* \stackrel{?}{=} M_2 \bigoplus \chi_i^*$ where UID_i^* and χ_i^* is the stored pair. The gateway retrieves $SMID$ in similar way by computation of $SMID^* \stackrel{?}{=} M_3 \bigoplus \chi_i^*$ where $SMID^*$ is the retrieved identity of the Smart Meter. Gateway GW finds the suitable ρ_j for the $SMID^*$ from its secret memory. The gateway verifies $M_1^* = \mathrm{H}(X_1||X_2||UID_i^*||SMID^*||GWID||\chi_i^*||T_1) \stackrel{?}{=} M_1$ by computing $X_2^* = \frac{1}{M_k+\chi_i^*}$. After the successful verification of the user, the gateway GW computes the parameters for the smart meter SM. The gateway GW selects the random number r_2 and computes $Z_1 = r_2$*P and $Z_2 = \frac{1}{M_k+\rho_j}$. The gateway computes $M_4 = \mathrm{H}(UID_i^*||SMID||GWID||Z_1||Z_2||\rho_j||T_2)$, $M_5 = \mathrm{H}(SMID||GWID||M_k) \bigoplus GWID$ and M_6= $\mathrm{H}(SMID||GWID||M_k) \bigoplus UID_i^*$, $M_7 = X_2^* \bigoplus \rho_j$ where T_2 is the current time-stamp. The gateway GW creates the second message as a *message 2* = (Z_1, M_4, M_5, M_6, T_2, M_7) and sends the *message 2* to the smart meter SM with an identity as a $SMID$.

Smart Meter $SM \xrightarrow{message3}$ **Gateway** GW**:**

The smart meter SM verifies the time-stamp T_2 by verifying $\Delta T \leq T_2^*$-T_2 where ΔT is the pre-defined timing threshold and T_2^* is the current time-stamp at smart meter. The Smart meter computes $GWID* = \rho_j \bigoplus M_5$, $UID_i^{**} = \rho_j \bigoplus M_6$, $Z_2^* = Z_1$ * SM_{Pri} and performs the verification, $M_4^* = \mathrm{H}(UID_i^{**}||GWID^*||Z_1||Z_2^*||T_2||\rho_j) \stackrel{?}{=} M_4$. After the successful verification of M_4, the smart meter performs the further computation. The smart meter SM generates the random number r_3, computes $Y_1 = r_3$*P, Y_2=H($Y_1||UID_i^{**}$), $X_2^{**} = M_7 \bigoplus rho_j$, $Y_3 = r_3 * P_k$, $M_7 = \mathrm{H}(UID_i^{**}||GWID^*||Y_2 ||SMID||Y_3||\rho_j||T_3)$ and $M_8 = \mathrm{H}(Z_2^*||X_1||P_k)$. Here T_3 is the current time-stamp at smart meter. The smart meter creates a third message as a *message 3* = (Y_1,Y_2,M_7,M_8,T_3,T_2) and sends *message 3* to the gateway GW. The smart meter SM computes the session key,

$$SK_{usm} = \mathrm{H}(UID_i^{**}||SMID||GWID^*||T_2||T_3||X_2^{**}||Y_3^*).$$

Gateway $GW \xrightarrow{message4}$ **User** SHU_i**:**

The gateway GW verifies the time-stamp T_3 by verifying $\Delta T \leq T_3^*$-T_3 where ΔT is the pre-defined timing threshold and T_3^* is the current time-stamp at gateway. The gateway performs $Y_2^* = \mathrm{H}(Y_1||UID_i^*) \stackrel{?}{=} Y_2$, Y_3^*=Y_1 * M_k, $M_7 = \mathrm{H}(UID_i^*||GWID|| SMID||Y_2^*||Y_3^*||T_3||\rho_j) \stackrel{?}{=} M_7$, $M_8^* = \mathrm{H}(Z_2||X_1||P_k) \stackrel{?}{=} M_8$. After the successful verification of all the messages, the gateway computes $Y_4 = \chi_i \bigoplus Y_3^*$, $K_1 = \mathrm{H}(Y_3^*||UID_i ||SMID||X_2^*||GWID||T_4)$, $K_2 = \mathrm{H}(K_1||T_4||GWID||T_3||T_2||T_1)$, $K_3 = \mathrm{H}(GWID||\chi_i^* ||T_4||UID_i||SMID)$. The gateway creates a fourth message as, *message 4* = (K_2, K_3, Y_4, T_4) and sends *message 4* to the user SHU_i with identity as UID_i.

User SHU_i:

The user verifies the time-stamp T_4 by verifying $\Delta T \leq T_4^*$-T_4 where ΔT is the pre-defined timing threshold and T_4^* is the current time-stamp at SHU_i. The user verifies $K_3 = \mathrm{H}(UID_i||GWID||T_4||SMID||\chi_i) \stackrel{?}{=} K_3$ and computes $Y_3^*= Y_4 \bigoplus \rho_i$, $K_1^* = \mathrm{H}(Y_3^*||UID_i||SMID||GWID||T_4||X_2)$, $K_2^* = \mathrm{H}(K_1^*||T_1||T_2||T_3||T_4) \stackrel{?}{=} K_2$, and computes the session key,

$$SK_{usm} = \mathrm{H}(UID_i||SMID||GWID||T_2||T_3||X_2||Y_3^*).$$

4.5 Security Analysis

A Informal Security Analysis

This subsection discusses the informal security analysis for the proposed scheme. It discusses how the proposed protocol is secured against various attacks for the Dolev-Yao channel. The Dolev-Yao channel is a *send* and *receive* insecure channel which is widely accepted in security protocols [Dolev & Yao (1981)].

MUTUAL AUTHENTICATION

The mutual authentication property in security protocol assures that both the communicating entities have trust on each other and both are assured about the identity of each other. In the proposed protocol, the user SHU_i communicates with the smart meter SM via the trusted entity gateway GW. So whenever the user SHU_i and the smart meter SM establish the session key SK_{usm}, they both verify the identity of each other. In the *message 1*, SHU_i sends $M_2 = UID_i \bigoplus \chi_i$ in which UID_i is

User	Gateway	Smart Meter
Selects r_1, computes X_1 = r_1*P and $X_2 = X_1 * U_{i_{Pri}}$. , M_1 = H(UID_i||$SMID$||$GWID$||X_1||X_2|| χ_i||T_1), $M_2 = UID_i \bigoplus \chi_i$, $M_3 = SMID \bigoplus \chi_i$. Sends *message 1* = ($X_1$, M_1,M_2,M_3,T_1) to *GW*		
	Verify $\Delta T \leq T_1^*$-T_1, retrieves UID_i by $UID_i^* \stackrel{?}{=} M_2 \bigoplus \chi_i^*$, retrieve $SMID$ by $SMID^* \stackrel{?}{=} M_3 \bigoplus \chi_i^*$. verifies M_1^* = H(X_1||X_2||UID_i^*||$SMID^*$||$GWID$ ||χ_i^*||T_1) $\stackrel{?}{=} M_1$ by computing $X_2^* = \frac{1}{M_k+\chi_i^*}$. selects r_2 and computes Z_1 = r_2*P and $Z_2 = \frac{1}{M_k+\rho_j}$, M_4 = H(UID_i^*||$SMID$||$GWID$||Z_1||Z_2 ||ρ_j||T_2), M_5 = H($SMID$||$GWID$||M_k) $\bigoplus$ $GWID$ and M_6= H($SMID$||$GWID$||M_k) $\bigoplus$ UID_i^*, $M_7 = X_2^* \bigoplus \rho_j$. Send *message 2* = ($Z_1$, M_4, M_5, M_6, T_2, M_7) to *SM*	Verify $\Delta T \leq T_2^*$-T_2, Computes $GWID* = \rho_j \bigoplus M_5$, $UID_i^{**} = \rho_j \bigoplus M_6$, $Z_2^* = Z_1$ * SM_{Pri} and performs M_4^* = H(UID_i^{**}||$GWID^*$||Z_1||Z_2^* ||T_2||ρ_j) $\stackrel{?}{=} M_4$. Generates r_3, computes Y_1 = r_3*P, Y_2=H(Y_1||UID_i^{**}), X_2^{**} = $M_7 \bigoplus rho_j$, Y_3 = r_3 * P_k, M_7 = H(UID_i^{**}||$GWID^*$||Y_2||$SMID$ ||Y_3||ρ_j||T_3) and M_8 = H(Z_2^*||X_1||P_k). Sends *message 3* = (Y_1,Y_2,M_7,M_8,T_3,T_2) to *GW*
Verifies $\Delta T \leq T_4^*$-T_4, verifies K_3 = H(UID_i||$GWID$||T_4||$SMID$||χ_i) $\stackrel{?}{=} K_3$ and computes Y_3^*= $Y_4 \bigoplus \rho_i$, K_1^* = H(Y_3^*||UID_i||$SMID$||$GWID$||T_4 ||X_2), K_2^* = H(K_1^*||T_1||T_2||T_3||T_4) $\stackrel{?}{=} K_2$.	Verify $\Delta T \leq T_3^*$-T_3. performs Y_2^* = H(Y_1||UID_i^*) $\stackrel{?}{=} Y_2$, Y_3^*=Y_1 * M_k, M_7 = H(UID_i^*||$GWID$||$SMID$|| Y_2^*||Y_3^*||T_3||ρ_j) $\stackrel{?}{=} M_7$, M_8^* = H(Z_2||X_1||P_k) $\stackrel{?}{=} M_8$. Computes $Y_4 = \chi_i \bigoplus Y_3^*$, K_1 = H(Y_3^*||UID_i||$SMID$||X_2^* ||$GWID$||T_4), K_2 = H(K_1||T_4||$GWID$||T_3 ||T_2||T_1), K_3 = H($GWID$||χ_i^*||T_4||UID_i||$SMID$). Send *message 4* = (K_2, K_3, Y_4, T_4) to user	
SK_{usm} = H(UID_i|| $SMID$|| $GWID$|| T_2|| T_3|| X_2||Y_3^*)		SK_{usm} = H(UID_i^{**}|| $SMID$|| $GWID^*$|| T_2||T_3|| X_2^{**} ||Y_3^*).

TABLE 4.2
Authentication and Key Exchange

an identity of the user and χ_i is a secret parameter which is computed using the secret master of the gateway device. The gateway device can achieve the authentication of the user device by performing $M_1^* = \mathrm{H}(X_1||X_2||UID_i^*||SMID^*||GWID||\chi_i^*||T_1) \stackrel{?}{=} M_1$. Similarly the smart meter confirms the authenticity of user and gateway device by performing $M_4^* = \mathrm{H}(UID_i^{**}||GWID^*||Z_1||Z_2^*||T_2||\rho_j) \stackrel{?}{=} M_4$ and the user performs authenticity verification of gateway and sensing device by computing $K_3 = \mathrm{H}(UID_i||GWID||T_4||SMID||\rho_i) \stackrel{?}{=} K_3$. The whole authenticity verification is secured against the one-way hash function so the integrity verification is also achieved over here.

USER NON-TRACEABILITY

The non-traceability property in the IoT based applications assures that even though the adversary $\mathcal{A}$ captures the identity of user SHU_i, $\mathcal{A}$ must not be able to trace the other communication of user SHU_i. Now let us assume that an adversary $\mathcal{A}$ capture UID_i as an identity for i^{th} user. Now SHU_i communicates the *message 1* = (X_1, M_1,M_2,M_3,T_1) in which M_1 and M_2 includes an identity of user SHU_i but an adversary does not know the other parameter χ_i which is required to generate M_1. Similarly $\mathcal{A}$ doesn't know the parameters like X_2 and $GWID$ so for an adversary $\mathcal{A}$ can't compute M_2 also by achieving UID_i. Similarly all the messages which includes an identity of user are either secured through hash function or XOR operation with the secure parameter. So the proposed protocol achieves the user non-traceability property.

ANONYMITY AND UNSINKABLE

Let us assume that an adversary $\mathcal{A}$ captures all the communicated messages through an insecure channel. The user communicated *message 1* with the gateway *GW* in which the user sends computed parameters to the gateway but doesn't send its identity in plain text. The gateway device communicates the *message 2* with the smart meter and the *message 4* with the user. In none of the message, gateway device doesn't communicate it's identity *GWID* through a public channel. Similarly, the smart meter communicates the *message 3* with the gateway device but the *SM* doesn't send *SMID* in plain text. So all the three identities UID_i, *GWID* and *SMID* are secured through a hash function and the XOR with secret parameters. So the anonymity property is achieved in the proposed protocol. All the three entities SHU_i, *GW* and *SM* generates r_1, r_2 and r_3, the random numbers before the computation of other parameters. The random number assures that the *message 1* and *message 1** by user SHU_i will not be similar. Similarly *message 2*, *message 4* by gateway *GW* will not be similar to other messages *message 2**,*message 4** by same gateway *GW* due to the random numbers . So an adversary $\mathcal{A}$ can't link the two messages of the same entity. Thus the proposed protocol also achieves unlinkability property.

Scheme	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
[Farash et al. (2016)]	✓	✗	✓	✓	✗	✓	✗	✓	✓	✓
[Amin et al. (2018)]	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
[Zhou et al. (2019)]	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓
[Poh et al. (2019)]	✓	✓	✗	✓	✓	✗	✓	✗	✓	✓
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 4.3
Security Comparison

Security attacks in Table 4.3 are notified as : **F1** = User anonymity, **F2**= Replay attack, **F3**= Offline Password guessing attack, **F4** = Mutual authentication and MITM attack,**F5** = Privilege insider attack,**F6** = Perfect forward secrecy,**F7** = Denial of Service attack, **F8** = User impersonation attack, **F9** = gateway impersonation attack, **F10** = Stolen smart card attack.

SECURE AGAINST REPLAY ATTACK

In the replay attack, an adversary $\mathcal{A}$ try to create the trust by re-communicating the previously communicated messages by valid entities. Let us assume that an adversary $\mathcal{A}$ eavesdrops the *message 1* = (X_1, M_1,M_2,M_3,T_1), *message 2* = (Z_1, M_4, M_5, M_6, T_2, M_7), *message 3* = (Y_1,Y_2,M_7,M_8,T_3,T_2), *message 4* = (K_2, K_3, Y_4, T_4). Now let us assume that an adversary $\mathcal{A}$ replays *message 1* after sometime to gateway GW. The gateway verifies $\Delta T \leq T_1^*$-T_1 where ΔT is the pre-defined timing threshold and T_1^* is the current time-stamp at gateway. The verification will fail because the value of ΔT will exceed the predefined threshold. Similarly whenever the adversary $\mathcal{A}$ try to re-communicate the other messages *message 2,message 3,message 4*, the gateway GW, the user SHU_i and the smart meter SM verify the time-stamp threshold ΔT. None of the replayed message will satisfy this threshold, thus the proposed protocol is secured against the replay-attack.

SECURE AGAINST MITM ATTACK

In the MITM attack, an adversary capture the message, modify the message and send the message to destination. The adversary $\mathcal{A}$ modifies the message in such a way that destination accept the modified message. Now let us assume that an adversary $\mathcal{A}$ captures *message 1* = (X_1, M_1,M_2,M_3,T_1) and try to modify any of the

value. Let us assume that $\mathcal{A}$ modifies X_1 then at the gateway $M_1^* = \mathrm{H}(X_1||X_2||UID_i^*||SMID^*||GWID||\chi_i^*||T_1) \stackrel{?}{=} M_1$ verification will fail. The adversary $\mathcal{A}$ can't generate X_2 and χ_i from the public parameters. Similarly any modification to other messages will fail the $UID_i^* \stackrel{?}{=} M_2 \bigoplus \chi_i^*$ and $SMID^* \stackrel{?}{=} M_3 \bigoplus \chi_i^*$ verification. Similarly when ever an adversary $\mathcal{A}$ modifies the *message 2*, the verification $M_4^* = \mathrm{H}(UID_i^{**}||GWID^*||Z_1||Z_2^*||T_2||\rho_j) \stackrel{?}{=} M_4$ will not be get success. The modification in *message 3* and *message 4* will fail the $M_7 = \mathrm{H}(UID_i^*||GWID||SMID||Y_2^*||Y_3^*||T_3||\rho_j) \stackrel{?}{=} M_7$ and $K_3 = \mathrm{H}(UID_i||GWID||T_4||SMID||\rho_i) \stackrel{?}{=} K_3$ verification respectively. So any modification or new message in the proposed protocol doesn't give success to an adversary $\mathcal{A}$ and thus the proposed protocol is secured against the Man-In-The-Middle attack.

RESISTANT TO SMART METER COMPROMISED ATTACK

The physical attack on smart devices is one of the most challenging attack in the IoT network. In the smart home environment, the smart devices are physically secured and it is difficult for an adversary to capture the smart meter *SM* in physical. Let us assume that an adversary $\mathcal{A}$ performs the physical attack on smart meter *SM* and try to perform the power analysis attack [Kocher et al. (1999)] to get the secret parameters stored in side the smart meters. Now let us assume that an adversary $\mathcal{A}$ captures (F_p, P, E, n, $SMID_t$, ID_{SM_t} H(.),SM_{pub},SM_{Pri}, ρ_j ,UID_i, $SMID$, $GWID$) from the smart meter's memory. Now the session key computation, $SK_{usm} = \mathrm{H}(UID_i||SMID||GWID||T_2||T_3||X_2||Y_3^*)$ which includes X_2 and Y_3 which are computed based on randomly generated parameters r_1 and r_3 respectively. The messages X_2 and Y_3 are never communicated through public channel. The Y_3 is secured through the private master of the gateway *GW*. Thus

even through the adversary physically captures (which is very difficult in smart home environment) the smart meter, the adversary $\mathcal{A}$ can't compute the session key SK_{usm}.

RESISTANT TO IMPERSONATION ATTACK

The adversary $\mathcal{A}$ willing to impersonate the user SHU_i, may try to capture the *message 1* = (X_1, M_1,M_2,M_3,T_1) in the first communication.However the adversary $\mathcal{A}$ can not know the real identity of the user SHU_i which is UID_i. The identity of SHU_i is confidential through the secret χ_i and the hash function H(.). Since the secret key $SK_{usm} = \mathrm{H}(UID_i||SMID||GWID||T_2||T_3||X_2||Y_3^*)$ is computed using the identity of all the 3 entity and none of the identity is communicated in the plain text. Similarly let us assume that an adversary $\mathcal{A}$ try to impersonate smart meter SM and capture the *message 3* = (Y_1,Y_2,M_7,M_8,T_3,T_2). The identity of smart meter is secured through the secret ρ_j and smart gateway master M_k. The secrets χ_i and ρ_j is secured through the master secret of the gateway GW. Thus, the proposed protocol is secured against the impersonation attack.

SECURED AGAINST GATEWAY NODE BYPASSING ATTACK

In the gateway node by passing attack, an adversary $\mathcal{A}$ try to replace the gateway node and works as a trusted entity between the user SHU_i and the smart meter SM. Let us assume adversary $\mathcal{A}$ receives the *message 1* = (X_1, M_1,M_2,M_3,T_1) and compute the parameters for the smart meter SM. Let us assume that an adversary $\mathcal{A}$ creates $message2^* = (Z_1^*, M_4^*, M_5^*, M_6^*, T_2^*, M_7^*)$ and sends it to the smart meter SM. Now when the smart meter performs $M_4^* = \mathrm{H}(UID_i^{**}||GWID^*||Z_1||Z_2^*||T_2||\rho_j) \stackrel{?}{=} M_4$, this verification doesn't get success because the adversary doesn't know the

correct value of ρ_j which is computed by the real gateway device for the smart meter SM by using it's own master secret M_k. Similarly at user side, the verification $K_3 = \mathrm{H}(UID_i||GWID||T_4||SMID||\chi_i) \stackrel{?}{=} K_3$ doesn't get success due to the secret parameter χ_i computed by the real gateway GW. Thus for any adversary $\mathcal{A}$, it's impossible to bypass the gateway device GW and so that the proposed scheme is secured against the gateway node by passing attack.

B Mutual Authentication using BAN Logic

This section discusses the mutual authentication verification between the user SHU_i and the lightweight device SM using the BAN Logic. This proof mainly consists of the following steps: (I) Initial assumption, (II) Goal declaration, (III) Message formation, and (IV) Formal verification.

1. ***Initial assumptions:***

   J_1. $SHU_i \mid\equiv \#(T_i), SM_j \mid\equiv \#(T_i), GW \mid\equiv \#(T_i)$;

   J_2. $SHU_i \mid\equiv \#(r_1), GW \mid\equiv \#(r_2), SM_j \mid\equiv \#(r_2), SHU_i \mid\equiv \#(r_3)$;

   J_3. $GW \mid\equiv SHU_i \Rightarrow X, GW \mid\equiv SM_j \Rightarrow X, SM_j \mid\equiv GW \Rightarrow X$.

   J_4. $SHU_i \mid\equiv (\stackrel{GWID}{\longmapsto} GW), SM_j \mid\equiv (\stackrel{RIDU_i}{\longmapsto} U_i), GW \mid\equiv (\stackrel{SID_j}{\longmapsto} SM_j)$

   J_5. $SHU_i \mid\equiv SM_j \Rightarrow (SHU_i \stackrel{SK}{\longleftrightarrow} SM_j)$

2. ***Goal Declaration:*** The expected goals in the proposed LDA scheme includes trust in shared key and freshness of communicated messages. In LDA scheme, expected goals are as follow:

   G_1. $SHU_i \mid\equiv SHU_i \stackrel{SK}{\longleftrightarrow} SM_j$

G_2. $SM_j \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$

G_3. $SHU_i \mid\equiv SM_j \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$

G_4. $SM_j \mid\equiv SHU_i \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$

G_5. $SHU_i \mid\equiv GW \mid\sim \#(X)$

G_6. $GW \mid\equiv SM_j \mid\sim \#(X)$

G_7. $SM_j \mid\equiv GW \mid\sim \#(X)$

3. ***Message Formation:*** The login and Authentication phase of the proposed LDA scheme includes exchanges of the following messages which can be written in the generic form as follow:

Message 1: $SHU_i \rightarrow GW$: (X_1, M_1,M_2,M_3,T_1)

Message 2: $GW \rightarrow SM_j$: (Z_1, M_4, M_5, M_6, T_2, M_7)

Message 3: $SM_j \rightarrow GW$: (Y_1,Y_2,M_7,M_8,T_3,T_2)

Message 4: $GW \rightarrow SHU_i$: (K_2, K_3, Y_4, T_4)

Idealized form: The ideal forms for the above messages can be written as follows:

Message 1: $SHU_i \rightarrow GW$: $GW \triangleleft \langle < ((r_1 * \mathrm{P}), (\mathrm{H}(UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1)), (UID_i \bigoplus \chi_i), (SMID \bigoplus \chi_i), T_1 > \rangle_{GW \mid\equiv (\overset{SID_j}{\longmapsto} SM_j)}$

Message 2: $GW \rightarrow SM_j$: $SM_j \triangleleft \langle < ((r_2 * \mathrm{P}), (\mathrm{H}(UID_i^*||SMID||GWID||Z_1||Z_2||\rho_j||T_2)), (\mathrm{H}(SMID||GWID||M_k) \bigoplus GWID), (\mathrm{H}(SMID||GWID||M_k) \bigoplus UID_i^*), (X_2^* \bigoplus \rho_j), T_2) > \rangle_{SM_j \mid\equiv (\overset{GWID}{\longmapsto} GW)}$

Message 3: $SM_j \rightarrow GW$: $GW \triangleleft \langle < ((r_3 * \mathrm{P}), (\mathrm{H}(Y_1||UID_i^{**})), (\mathrm{H}(UID_i^{**}||GWID^*||Y_2||SMID||Y_3||\rho_j|$ $(\mathrm{H}(Z_2^*||X_1||P_k)), T_3, T_2) > \rangle_{GW|\equiv(\overset{SID_j}{\longmapsto} SM_j)}$

Message 4: $GW \rightarrow SHU_i$: $SHU_i \triangleleft \langle < ((\mathrm{H}(K_1||T_4||GWID||T_3||T_2||T_1)), (\mathrm{H}(GWID||\chi_i^*$ $||T_4||UID_i||SMID)), (\chi_i \bigoplus Y_3^*), T_4) > \rangle$

4. ***Formal Verification***

Theorem 1. *The proposed scheme achieves the secure mutual authentication between the user SHU_i and the sensing device SM_j, and it achieves expected goals.*

Proof. Expected goals are [G_1 - G_7] achieved as follow:

S_1**:** from the message 1,

$GW \triangleleft \langle < ((r_1 * \mathrm{P}), (\mathrm{H}(UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1)), (UID_i \bigoplus \chi_i), (SMID \bigoplus \chi_i),$ $T_1 > \rangle_{GW|\equiv(\overset{SID_j}{\longmapsto} SM_j)}$

S_2**:** Using S_1, R_1 and J_1,

$GW \mid\equiv SHU_i \mid\sim \langle < ((UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1)), (UID_i \bigoplus \chi_i),$ $T_1) > \rangle$

S_3**:** Using S_2, J_2, R_2,

$GW \mid\equiv SHU_i \mid\equiv \langle < ((UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1), (SMID \bigoplus \chi_i),$ $T_1) > \rangle$

S_4**:** from the message 2,

$SM_j \triangleleft \langle < ((r_2 * \mathrm{P}), (\mathrm{H}(UID_i^*||SMID||GWID||Z_1||Z_2||\rho_j||T_2)), (\mathrm{H}(SMID||GWID||M_k)$ $\bigoplus GWID), (\mathrm{H}(SMID||GWID||M_k) \bigoplus UID_i^*), (X_2^* \bigoplus \rho_j), T_2) > \rangle_{SM_j|\equiv(\overset{GWID}{\longmapsto} GW)}$

S_5**:** Using S_4, R_2, R_5 and J_2,

$SM_j \mid\equiv GW \mid\sim \langle < ((r_2 * \mathrm{P}), (UID_i^* || SMID || GWID || Z_1 || Z_2 || \rho_j || T_2), ((SMID || GWID || M_k) \oplus GWID), T_2) > \rangle$

S_6**:** Using S_4, R_4, R_6 and J_3,

$SM_j \mid\equiv GW \mid\equiv \langle < ((r_2 * \mathrm{P}), (UID_i^* || SMID || GWID || Z_1 || Z_2 || \rho_j || T_2), ((SMID || GWID || M_k) \oplus UID_i^*), (X_2^* \oplus \rho_j), T_2), T_2) > \rangle$

S_7**:** Using S_4, S_5, R_1, R_7 and J_1,

$SM_j \mid\equiv (SMID || GWID || M_k) \oplus UID_i^*), SM_j \mid\equiv (UID_i^* || SMID || GWID || Z_1 || Z_2 || \rho_j || T_2),$

S_8**:** from the message 3,

$GW \triangleleft \langle < ((r_3 * \mathrm{P}), (\mathrm{H}(Y_1 || UID_i^{**})), (\mathrm{H}(UID_i^{**} || GWID^* || Y_2 || SMID || Y_3 || \rho_j || T_3)), (\mathrm{H}(Z_2^* || X_1 || P_k)), T_3, T_2) > \rangle_{GW \mid\equiv (\overset{SID_j}{\longmapsto} SM_j)}$

S_9**:** Using S_8, R_3, R_4 and J_4,

$GW \mid\equiv SM_j \mid\sim \langle < ((r_3 * \mathrm{P}), (Y_1 || UID_i^{**}), (UID_i^{**} || GWID^* || Y_2 || SMID || Y_3 || \rho_j || T_3), T_2, T_3) > \rangle$

S_{10}**:** Using S_8, S_9, R_8,

$GW \mid\equiv SM_j \mid\equiv \langle < ((r_3 * \mathrm{P}), (UID_i^{**} || GWID^* || Y_2 || SMID || Y_3 || \rho_j || T_3), (Z_2^* || X_1 || P_k), T_3, T_2) > \rangle$

S_{11}**:** Using S_8, S_9, R_{10}, R_8 and J_1,

$GW \mid\equiv (Y_1 || UID_i^{**}), SM_j \mid\equiv (UID_i^{**} || GWID^* || Y_2 || SMID || Y_3 || \rho_j || T_3,$

S_{12}**:** from the message 4,

$SHU_i \triangleleft \langle < (((K_1 || T_4 || GWID || T_3 || T_2 || T_1)), ((GWID || \chi_i^* || T_4 || UID_i || SMID)), (\chi_i \oplus Y_3^*), T_4) > \rangle$

S_{13}**:** Using S_{12}, R_3, R_8 and J_1,,

$SHU_i \mid\equiv GW \mid\sim \langle < ((K_1||T_4||GWID||T_3||T_2||T_1), (GWID||\chi_i^*||T_4||UID_i||SMID)$

$T_4) > \rangle$

S_{14}**:** Using S_{12}, S_{13}, R_3, R_8 and J_1,

$SHU_i \mid\equiv GW \mid\sim \langle < (\chi_i \bigoplus Y_3^*), T_4) > \rangle$

S_{15}**:** Using S_{13}, S_{14}, R_3, R_8 and J_1,,

$SHU_i \mid\equiv (GWID||\chi_i^*||T_4||UID_i||SMID)$,

S_{16}**:** Using S_1, S_7, S_8, R_3, R_8 and J_1,,

$SM_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SM_j$ $[G_2]$

S_{17}**:** Using S_4, S_{10}, S_{12}, R_3, R_8 and J_1,,

$SHU_i \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$ $[G_1]$

S_{18}**:** Using S_7, S_{11}, S_{13}, S_{14}, S_{16} J_4, R_4, and R_7,

$SHU_i \mid\equiv SM_j \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$ $[G_3]$

S_{19}**:** Using S_9, S_{12}, S_{14}, J_4, R_1, R_6 and R_7,

$SM_j \mid\equiv SHU_i \mid\equiv SHU_i \overset{SK}{\longleftrightarrow} SM_j$ $[G_4]$

S_{20}**:** Using S_3, S_7, S_9, R_4, R_6, R_7, J_2 and J_4,

$SM_j \mid\equiv GW \mid\sim \#(Message1)$

S_{21}**:** Using S_8, S_{12}, S_{14}, S_{18}, R_6, R_9, J_3 and J_5,

$SM_j \mid\equiv GW \mid\sim \#(Message1, Message3)$ $[G_7]$

S_{22}**:** Using S_9, S_{11}, S_{15}, R_4, R_7, J_3 and J_5,

$GW \mid\equiv SM_j \mid\sim \#(Message2)$ $[G_6]$

S_{23}**:** Using S_{19}, S_{20}, S_{21}, R_3, R_5, J_2, J_4,

$SHU_i \mid\equiv GW \mid\sim \#(Message4)$ $[G_5]$

Thus, the above verification clearly shows that the proposed authentication scheme achieves all defined goals (G_1 - G_7). □

C Formal Security Analysis Using ROR

The ROR model [Abdalla & Pointcheval (2005)] is widely accepted security model to preform the formal security analysis. The ROR model is based on random oracles, which are oracle queries performed by an adversary $\mathcal{A}$ to perform the various attacks on the proposed scheme. The basic objective of an adversary $\mathcal{A}$ is to distinguish the real session key and retrieved random values successfully. A similar approach to formal security analysis is also applied in [Roy et al. (2018)].

Random Oracles: The random oracles are hash function H(.) which outputs non-reversible random value r_i for any message m_i.

Participants and oracle instances: In the proposed scheme, there are three participants, the user SHU_i, the gateway GW, and the smart meter SM. Let us assume that $\Omega_{U_i}^{x}$, Ω_{GW}^{y}, and Ω_{SM}^{z} are oracles with the instances x, y and z for the SHU_i, GW and S_j respectively.

Partnering: The two oracle instances Ω^{t_1}, Ω^{t_2} are the partnering instances if both share the same *sid* (transcript of all messages) and both are in the acceptance state.

Freshness: The oracle Ω^{t_1} is fresh if no reveal query is performed by an adversary $\mathcal{A}$ on the instance t_1.

Adversary: The adversary $\mathcal{A}$ is an intruder who controls the complete Dolev-Yao channel and performs the oracle queries.

$\mathcal{R}(\Omega^{t_1})$: The reveal query provides a session key computed by an instance t_1 with its partnering instances t_2.

$\mathcal{T}(\Omega^{t_1})$:. The test query provides a random key size value to an adversary based on the value of bit b decided by adversary $\mathcal{A}$ using an unbiased coin. This query is performed for a limited number of times for one session.

$\mathcal{S}(\Omega^{t_1}, m_i)$ The send query enables an adversary $\mathcal{A}$ to perform the communication using message m_i with an instance t_1.

$\mathcal{E}(\Omega^{t_1}, \Omega^{t_2})$ The execute query enables an adversary $\mathcal{A}$ to perform the passive monitoring of communicated messages between legal instances t_1 and t_2.

CorruptSmartMeter The query *CorruptSmartMeter* enables an adversary $\mathcal{A}$ to capture the stored information inside the secret memory of the smart meter. This query is performed for limited number of times for one session.

CorruptUserDevice The query *CorruptSmartMeter* enables an adversary $\mathcal{A}$ to capture the stored information inside the secret memory of the user device. This query is performed for a limited number of times for one session.

If the $\mathcal{SUCC}$ defines the success of an adversary $\mathcal{A}$ then the advantage function Adv_P defines the success of an adversary $\mathcal{A}$ in capturing SK_{usm}. And if Adv_P is negligible, then the proposed scheme is secured. Thus Adv_P can be defined as follows:

$$Adv_P(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}] - 1 \tag{4.1}$$

The Formal Security Proof:

The formal security proof for the proposed protocol can be given as follows:

Theorem 2. *The security of the proposed protocol from the polynomial time ad-*

versary $\mathcal{A}$ is defined as,

$$Adv_P(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{1}{2^{l_r+1}} \tag{4.2}$$

The l_h, l_r are the size of hash output and random number in bits respectively. Here the q_h , q_s and q_e define the number of times in which hash query, send query and execute query are performed by an Adversary $\mathcal{A}$.

Proof. To prove the security of the proposed protocol P, Four security games (G_0, G_1, G_2 and G_3) are defined. In the security games Pr[$\mathcal{SUCC}$] define the success probability for an adversary to successfully distinguish computed or retrieved value and real session key SK_{usm}.

Game G_0***:*** The game G_0 is an identical game to real protocol P. So the game G_0 can be defined as per shown in Equation. 5.2,

$$Adv_P(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}_0] - 1 \tag{4.3}$$

Game G_1***:*** The game G_1 enables an adversary to perform a passive attack through the execute $\mathcal{E}$ query. Let us assume that an adversary $\mathcal{A}$ performs $\mathcal{E}(\Omega^x, \Omega^y, \Omega^z)$ query to track the communicated messages between all the three participants. The Pr[$\mathcal{SUCC}$] in this game for $\mathcal{A}$ depends on the computation of the session key SK_{usm} from the tracked messages. There are four messages communicated by the three participants. Let us assume that through the $\mathcal{E}$ query, an adversary receives *message 1* = (X_1, M_1,M_2,M_3,T_1), *message 2* = (Z_1, M_4, M_5, M_6, T_2, M_7), *message 3* = (Y_1,Y_2,M_7,M_8,T_3,T_2), *message 4* = (K_2, K_3, Y_4, T_4). Now the computation of SK_{usm} = H($UID_i||SMID||GWID||T_2||T_3||X_2||Y_3^*$) which includes secured identi-

ties of all the three participants, also includes the parameters like X_2 and Y_3 which are secured through the master secret of trusted entity gateway device GW. None of the parameters required for the session key is received by $\mathcal{A}$ in the plain text. So after the game G_1,

$$Pr[\mathcal{SUCC}_1] = Pr[\mathcal{SUCC}_0] \tag{4.4}$$

The equation 5.3 shows that Pr[$\mathcal{SUCC}$] for $\mathcal{A}$ is negligible under G_1.

Game G_2***:*** The game G_2 enables an adversary $\mathcal{A}$ to perform the active attack through the simulation of Hash query and send query. The basic objective of an adversary $\mathcal{A}$ is to modify the messages and convince the communicated parties to accept the modified messages. Now let us assume that $\mathcal{A}$ receives *message 1* = (X_1, M_1,M_2,M_3,T_1) from SHU_i and try to modify *message 1* in such a way that gateway accept the modified message but the computation of M_1 = H($UID_i||SMID||GWID||X_1||X_2||\chi_i||T_1$), $M_2 = UID_i \bigoplus \chi_i$, $M_3 = SMID \bigoplus \chi_i$ includes identities, random number based X_1 and X_2 and time stamp so there is no probability of occurrence of the collision. The collision probability of hash query is at most $\frac{q_h^2}{2^{l_h+1}}$ [Chatterjee et al. (2018)] and collision probability for the random numbers is $\frac{(q_s+q_e)^2}{2^{l_r+1}}$ [Chatterjee et al. (2018)]. So, through the birthday paradox:

$$Pr[\mathcal{SUCC}_2] - Pr[\mathcal{SUCC}_1] \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s+q_e)^2}{2^{l_r+1}} \tag{4.5}$$

Game G_3***:*** The game G_3 considers the other attacks performed by an adversary $\mathcal{A}$ using oracle queries. The $\mathcal{A}$ makes use of *CorruptSmartMeter* and *CorruptUserDevice* and retrieves (F_p, P, E, n, $SMID_t$, ID_{SM_t} H(.),SM_{pub},SM_{Pri}, UID_i, $SMID$, $GWID$) and (F_p, P, E, n, H(.),$U_{i_{pub}}$,$U_{i_{Pri}}$, UID_i, $SMID$,χ_i , $GWID$) respectively. Now an adversary tries to compute the session key SK_{usm} = H($UID_i||SMID$

$||GWID||T_2||T_3||X_2||Y_3^*)$ from the received values. Through these queries, $\mathcal{A}$ can get UID_i, $SMID$, $GWID$ but can't compute the value X_2 and Y_3 because both the values depend on random numbers r_1 and r_3. The guessing probability of a random number for an $\mathcal{A}$ is $\frac{1}{2^{l_r+1}}$. Hence,

$$Pr[\mathcal{SUCC}_3] - Pr[\mathcal{SUCC}_2] \leq \frac{1}{2^{l_r+1}} \tag{4.6}$$

So now, after the completion of all the games, an option left with an adversary $\mathcal{A}$ is to toss the unbiased coin to guess the correct value of bit b whose probability is almost $\frac{1}{2}$. So, from the Equation 5.2, 5.3, 6.3,

$$\frac{1}{2}Adv_P(\mathcal{A}) = Pr[\mathcal{SUCC}_1] - Pr[\mathcal{SUCC}_3] \tag{4.7}$$

With the help of Equation 6.3,6.4,6.5,

$$Adv_P(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h+1}} + \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{1}{2^{l_r+1}}$$

□

4.6 Implementation and Testbeds

The proposed protocol is implemented using MQTT as an application layer protocol and IP as a network layer protocol. An MQTT broker was installed in the gateway device. The user device make use of a 1GB RAM Mobile device. A laptop with Intel (R) Core(TM) i5-7500CPU with 3.40 GHz with 8GB RAM was used as a gateway device. As a smart meter, a raspberry pi 3.0 with 1 GB RAM

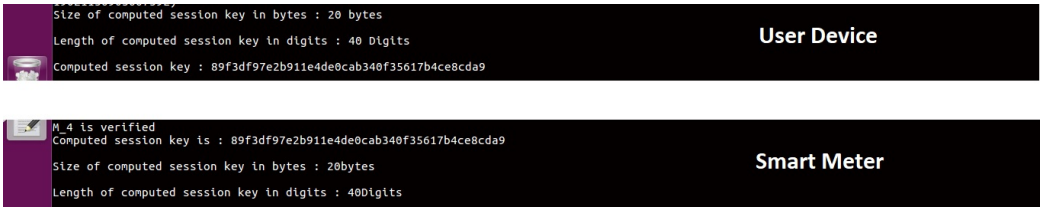

Figure 4.1. Session Key

was used. The following Fig. 4.1 shows the computed session key between User SHU_i and Smart Meter SM using the proposed protocol. For implementations we have used SHA256 as a hash function and NIST recommended P-256 Curve. The security level achieved through implementation of proposed scheme is 128 bit.

4.7 Use cases

This chapter proposes RUA scheme for the user-gateway-sensor model. The proposed scheme aims to provide a lightweight communication between the user device and the lightweight sensing device. The following are some of the significant use-cases of the proposed work.

- The proposed scheme can be used for secure and authenticated data exchange in IoT applications such as smart home where home owner want to get realtime data from the sensing devices.
- it can be also useful in ST, SI, SH and many more applications where it needs efficient and lightweight authentication mechanism between user (mobile/laptop) and the sensing devices.
- The proposed RUA scheme can be also useful for all the environment where user want to receive live data from the sensing devices.

The novelty of the proposed work lies in its adaptive environment, lightweight computations and real-time implementations. Real-time implementation of the proposed ECC based scheme using MQTT protocol (that itself is a lightweight communication protocol used by the IoT industry) makes it novel. The real-time implementation using MQTT protocol has advantages like the lightweight header and reliable communication over other conventional protocols.

4.8 Comparative Analysis

In this section, the performance analysis for the proposed scheme is given by comparing it with the other existing schemes of Farash et al. [Farash et al. (2016)], Amin et al. [Amin et al. (2018)], Zhou et al. [Zhou et al. (2019)] and Poh et al. [Poh et al. (2019)].

A Network Parameter Analysis

In 2016, Farash et al. [Farash et al. (2016)] proposed an authentication scheme for the IoT environment in which the user device sends the request to the sensing device, and they both mutually authenticate with the gateway device. The communication cost in Farash et al. scheme was more than 3200 bits, and the computation cost was 0.6s.In 2018, Amin et al. [Amin et al. (2018)] proposed a scheme for patient monitoring by the doctor in which the doctor receives live patient data through the gateway device. The communication cost incurred by the scheme of Amin et al. was 3500 bits and 1.1s. Recently, In 2019, Zhou et al. [Zhou et al. (2019)] and Poh et al. [Poh et al. (2019)] proposed a scheme for a smart home where user device establishes a session key with a smart device

Scheme	*User*	*Gateway*	*Sensor*	*Total Cost*
[Farash et al. (2016)]	632	792	2048	3472
[Amin et al. (2018)]	832	1120	320	3072
[Zhou et al. (2019)]	832	2048	672	3552
[Poh et al. (2019)]	640	2240	800	3680
Proposed	832	1248	864	2944

TABLE 4.4
Comparison of Communication Cost

through a gateway device. The overall communication cost and computation cost up to session key establishment by Zhou et al.'s scheme and Poh et al.'s scheme were 3600bits, 3000bits, and 0.65s and 0.7s. The overall communication cost and computation cost in the proposed scheme are 2800 bits and 0.6s, which is lesser compared to the many recent schemes. The computation of key indexed is done through the various cryptanalysis performed on the [Farash et al. (2016)], [Amin et al. (2018)], [Zhou et al. (2019)] and [Poh et al. (2019)] or their discussion regarding the security of proposed scheme in manuscript.

B Communication Cost

The communication represents the number of bits communicated over insecure channels. To compute the communication cost, 160 bits output size of the hash function and 160 bits identity size of each entity is considered. The size of a random number is 128 bits, and one of the timestamps is 28 bits. The size of the elliptic curve point is 320 bit, with each point size being 160 bits. Following Table 4.4 presents comparison of communication cost between proposed scheme and existing schemes.

Scheme	User	Gateway	Lightweight Device	Total
[Farash et al. (2016)]	11*T_h	14*T_h	7*T_h	0.6934
[Amin et al. (2018)]	12*T_h	16*T_h	6*T_h	0.9576
[Zhou et al. (2019)]	4*T_{ecm}+5*T_h	3*T_{ecm}+7*T_h	4*T_{ecm}+6*T_h	1.19933
[Poh et al. (2019)]	6*T_h	2*T_h+2*T_e	6*T_h	0.803
Proposed	4*T_h	8*T_h	4*T_h	0.689

TABLE 4.5
Comparison of the Computation Cost

C Computation Cost

The computation cost gives information about the number of operations used in the proposed protocol. Let T_h represent the hash operation; T_{ecm} represents the elliptic curve multiplication and division operation. The computation cost of the XOR operation is not considered due to very little time required by it. The average value of T_h at the user device, gateway, and the smart meter is 0.00041 ms, 0.00034 ms, 0.0607 ms. The average value of T_{ecm} at the user device, gateway, and the smart meter is 0.0607 ms, 0.0589 ms, 0.0703 ms. Above Table 4.5 presents comparison of computation cost between proposed scheme and existing schemes. Following Fig.4.2. gives performance comparison of communication cost and computation cost of the proposed scheme with the existing schemes.

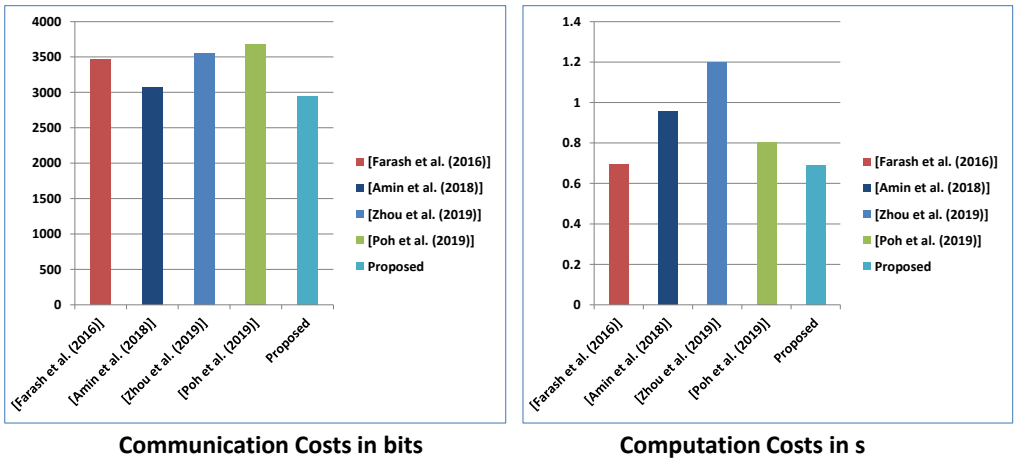

Figure 4.2. Performance Comparison Chart

4.9 Summary

The smart home and the smart grid are two widely accepted IoT Applications. This chapter proposes a secure authentication scheme for the user - gateway - sensor model with an example of smart home energy monitoring. This chapter presents an authentication scheme between the smart home user, the gateway, and the smart meter using elliptic curve cryptography. In the proposed authentication scheme, the gateway device works as an intermediary between the user and the smart meter. The formal security analysis of the proposed protocol is given using a random oracle based ROR Model and BAN Logic.The implementation of the proposed protocol is presented using an MQTT protocol. At last, performance comparison of the proposed protocol with the recently proposed other existing schemes for the same environment is given. The IoT based Smart grid is the future of the electricity transmission system, which will not only enhance the electricity distribution, but it will also reduce energy consumption. Similarly, the smart home

system also improves the living standard of the people. The primary concern is the security and privacy of their data. In proposed scheme, every user have to register for each sensing device. Now let us take scenario where user needs to receive data from hundreds of sensing device and it becomes difficult to register each user for each sensor. Hence, to solve this problem, next chapter 5 proposes a level dependent authentication scheme using two factor based approach. It provides a novel approach though which number of registration in the IoT authentication can be reduced significantly.

Chapter 5

Level Dependent Authentication using Two Factor

This chapter proposes a level dependent key exchange scheme between the user device, gateway device and the sensing device using two factor authentication. ***Section 3.1*** presents introduction for the user-gateway model and proposed work. ***Section 5.2*** provides literature review related to proposed scheme. ***Section 5.3*** highlights threat model considered for designing authentication scheme. ***Section 5.4*** presents proposed authentication scheme with all the phases. ***Section 5.5*** put forward security analysis for the proposed scheme. ***section 5.6*** provides implementation aspects and use cases related to proposed work. ***section 5.7*** presents performance comparison of the proposed work with other existing schemes based on communication cost and computation cost. ***section 5.8*** summarise this chapter

[1]**C. Patel, N. Doshi, 'Cryptanalysis and Improvement of Barman et al.s Secure Remote User Authentication Scheme', international journal of circuits, systems and signal processing, pp.604-610, Sept 2019**

with the limitations those are resolved in next chapter.

5.1 Introduction

The sensing devices deployed on "ground" collect data from the environment and transmit those data to the nearby home agent (gateway). The neighboring home agents can be a micro-controller, micro-processor, mobile towers, routers, or any data receiver device which integrates data from the sensing devices and forwards those data to the users via other Internet devices. The recent study shows that the latest home agents also work like fog devices or edge computing devices capable of performing the local data processing and converting those unorganized data into organized raw data. In traditional IoT network, deployed sensing devices create a local cluster and communicate pieces of information with the cluster heads (CHs) thorough the short-range protocols like Zigbee, Z-Wave, Beacon or Bluetooth Low Energy (BLE). In some of the IoT model, the sensing devices communicate with the cluster head through a wired medium. The CH connects the gateway devices (GW) with the sensing devices. The gateway devices are resource capable devices that can perform complex security operations and can forward the received data to the IoT application users through a long-range Internet protocol like IP or 6LoWPAN. In IoT, users can access stored data as well as realtime live data. Thus, the gateway devices transmit data to the cloud server for storage and processing or to the user for realtime monitoring. Secure storage and processing of the data in the cloud lead toward intelligent decision making through machine learning. In numerous recently proposed key agreement schemes, the application users register with the gateway devices for each sensing device [Zhou et

al. (2019)], but let us take the realtime scenario in which there are thousands of sensing devices deployed on the ground. Users like the company owner or the cluster manager want to receive the data from each sensing device in realtime. Then, they need to register for each sensing device individually, which is not a practical and feasible solution. During deployment of the realtime scenario, it is observed that in the recently proposed schemes, the gateway needs to create a separate smart card for each of the sensing devices for each of the application users who require $N_u * N_{sd}$ registrations and $N_u * N_{sd}$ time gateway initial computations. Here N_u shows the number of users, and N_{sd} indicates the number of sensing devices. Thus, this chapter presents a novel solution for the problems mentioned earlier, using a bizarre concept of the LDA.

A Level Dependent Authentication

To achieve successful access control in the IoT scenario is also one of the principal challenges. Essential factors that affect the liberty of access control is the availability of the tonnes of sensing devices and heterogeneity of their technical capabilities. Thus, for every user, if separate access control list is maintained or perform the registration, then it will be an unvaried task. As per the current literature, the user registers for individual sensors. If the user is eligible for hundreds of sensing devices, then he/she needs to maintain hundreds of smart cards. This is a significant challenge to reduce the space requirement with less complexity and wipe out the user's multiple registrations.

Rather than the traditional approach, a novel concept of the LDA is presented to tackle the above said challenges using a less computation cost, low energy

consumption, fewer operations, and little memory requirement. The working of the LDA concept is highlighted in Fig. 5.1. The following algorithm provides a working mechanism of the LDA concept.

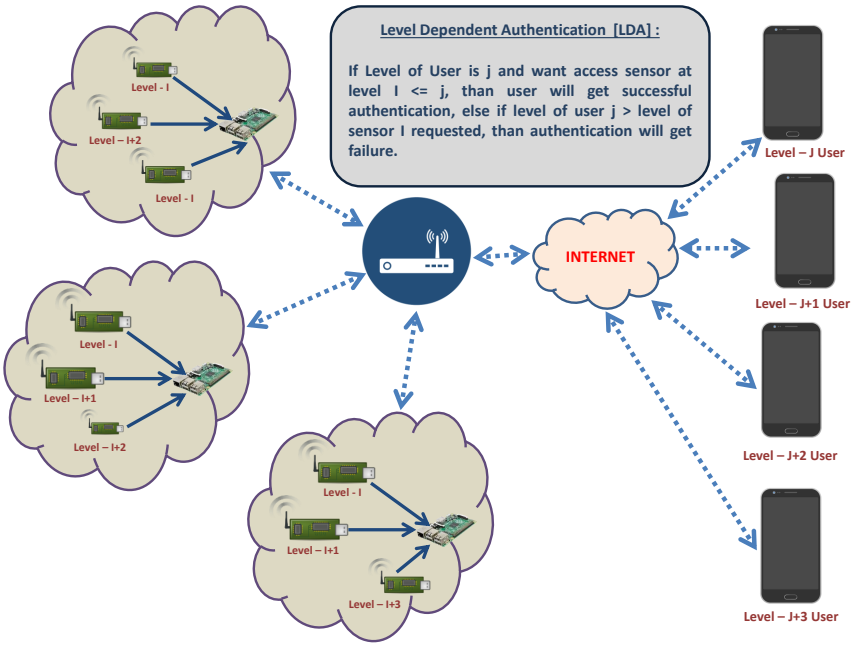

Figure 5.1. Network Model and Level Dependent Authentication

Result: Access of Sensing Device to User
User-level = i;
Sensor-level = j;
while *Gateway received request from user* **do**
  if $j \leq i$ **then**
    Access-Allowed;
  else
    Access-Not-Allowed;
  end
end

Algorithm 1: Level Dependent Authentication

Therefore, the major advantages of using LDA can be listed as follows:

- Reduction in the access control complexity.

- Number of registration phases and initialize phases will be reduced to the number of users rather than the number of sensing devices.
- Smooth replacement of the user device, sensing device, and gateway device compare to the existing traditional approach.
- Reduction of the computation cost, energy consumption, and memory utilization at user devices, gateway devices, and sensing devices.

The major challenges and future research directions related to LDA concept can be listed as follows:

- Little increase in the computation cost at the gateway device.
- In the proposed scheme, the decision for users and sensors' level will be taken by gateway. The proposed scheme can get further extension where separate authority like registration authority can be created for the purpose of initial parameter computation as well as level decision.

5.2 Literature Survey

Recently, Shuai et al. published an authentication scheme for the smart home using an ECC [Shuai et al. (2019)]. In this chapter, the Registration Authority (RA) is a trusted entity that performs an initialization step and generates secret credentials for the sensing device SD_j and the gateway node GW. The scheme proposed in [Shuai et al. (2019)] is a two-factor authentication scheme in which the user makes use of the password and smart card to perform the login and authentication. The other authentication scheme was recently proposed by Lyu et

al. [Lyu et al. (2019)] for the intelligent home using ECC. Authors in [Lyu et al. (2019)] put forward an authentication scheme which provides security against the traceability and useful for the uncertain Internet services and environment like "If This Then That (IFTTT)". In the same paper, authors give a formal security analysis using a practical scyther tool. In 2018, Chifor et al. [Chifor et al. (2018)] proposed a unique authentication scheme for the "Fast IDntity Online (FIDO)" model. In the FIDO model, the user does not use any authentication factors like a password. Still, it uses ECC generated parameters as keys stored by the trusted party and biometric-based access for those keys. The other authentication protocol for the smart home using a password was proposed by Naoui et al. in [Naoui et al. (2019)]. Authors in [Naoui et al. (2019)] proposed a lightweight and secure password-based authentication scheme called "LSP-SHAP" for the smart home monitoring and management. In 2016, Jo et al. proposed an authentication mechanism for the smart grid using ECC. Jo et al. [Jo et al. (2016)] proposed an authentication scheme between a smart meter (SM), Data Collection Unit (DCU), Advanced Metering Infrastructure (AMI) using ECC based key pair generation. In 2017, Vahedi et al. proposed an ECC based authentication scheme for the grid in [Vahedi et al. (2017)].

Authors in [Vahedi et al. (2017)] proposed an authentication mechanism between a smart meter (which collects an energy consumption from smart appliances), a gateway (which aggregates the data from all smart meter) and operation center (which works as a bill generating location) using a Trusted Third Party (TTP). In 2018, Wazid et al. [Wazid et al. (2018)] proposed a User Authenticated Key Management Protocol (UAKMP) for the smart home IoT network. Authors in [Wazid et al. (2018)] followed the user-gateway-sensor based network model

and the random oracle based ROR model for the formal security analysis. In 2019, Das et al. [Das et al. (2019)] proposed a lightweight access control and key agreement protocol for the IoT environment (LACKA-IoT) using ECC. Recently, in 2019, Gope et al. [Gope & Sikdar (2019)] proposed a privacy-preserving authentication scheme for the IoT devices using a Physical unclonable Function (PUF). The PUF provides a lightweight hardware implementation of the random number generator. In [Das et al. (2019)], the ROR based security model was followed, and the security simulation is produced using the AVISPA tool. They simulated the proposed protocol using a widely used simulator Network Simulator 2 (NS2).

A publish-subscribe based MQTT protocol is widely accepted for IoT based applications. The Lohachab et al. [Lohachab & Karambir (2019)], in 2019, proposed an ECC based authentication and access control scheme for the MQTT based communications. The Machine to Machine communication through the MQTT protocol plays a significant role in automated service developments. The authors in [Esfahani et al. (2019)] proposed an authentication scheme between the sensing device and routing device (device-device) using lightweight operations like hash function and XOR operation. The Internet of Drones (IoD) is a network of uncrewed areal vehicles called Drones. In IoD based interface, the secure live streaming and reliable access control of the drone devices are essential security aspects. The authors in [Wazid et al. (2019)] set forth a crucial lightweight agreement scheme for the drone deployment in which the ground user securely communicates with the Drone Data Transmitter (DDT) through the server as a trusted entity.

5.3 Threat Model

The threat model used in this chapter is contemplated from the homogeneous model discussed in [Dolev & Yao (1981)]. An adversary $\mathcal{A}$ is an eavesdropper who controls the complete public communication channel. In the IoT based network model, it is possible to define an adversary $\mathcal{A}$ with the robust capabilities for improvement in the designing of the reliable protocol and also to perform the better security validation for the proposed authentication concept. the following adversarial model is followed in this chapter:

G_1. An Adversary $\mathcal{A}$ can compute valid pair of the $identity * password$ offline in polynomial time using dictionary [Wazid et al. (2017)], [Gope & Sikdar (2019)].

G_2. An Adversary $\mathcal{A}$ can extract the data from the user's smart card after receiving smart card in either ways [Gope & Sikdar (2019)], [Shuai et al. (2019)].

G_3. An Adversary $\mathcal{A}$ have full access on the communication channel between a User - Gateway, Sensor node - Gateway, and User - Sensor node [Wazid et al. (2018)], [Gope & Sikdar (2019)], [Shuai et al. (2019)].

G_4. An Adversary $\mathcal{A}$ can get the previously computed session key between the user and sensor. $\mathcal{A}$ can use this key to compute the next session key [Kumar et al. (2015)].

G_5. An Adversary $\mathcal{A}$ can have the level information of the user device or the sensing device at a time but can't have the level of both at a time [Shuai et al. (2019)].

G_6. An Adversary $\mathcal{A}$ can have the secrets of a gateway node during the system failure situations. $\mathcal{A}$ can use this old secrets to break the newly established system after failure.

G_7. An Adversary A can perform the physical attacks on sensor nodes and can retrieve the information stored into it [Kumar et al. (2015)].

G_8. An Adversary $\mathcal{A}$ can generate bot nodes and can send the simultaneous ping messages to the sensor node with the aim to perform DoS attacks [Kumar et al. (2015)].

5.4 Proposed Scheme

In this section, the proposed Level Dependent Authentication Scheme for Generic IoT (LDA-GIoT) is discussed. An LDA-GIoT is proposed between the user device and the sensing device through the intermediary gateway node. As earlier said, it is assumed the gateway device as a trusted and secure node. The proposed vital agreement scheme consists of three phases: system initialization phase, user registration phase, and login and key-agreement phase. The gateway device is considered as a master device, and the key-agreement is also going to carry through the gateway device. The necessary notations used for designing of the proposed scheme are highlighted in Table 5.1.

Symbols	Description
R_x	Random Number
T_x	Time-stamp
l_i	User Level
l_j	Sensor Level
U_i	User Device
S_j	Sensor Device
GW	Gateway Node
SID_j	Sensor Identity
UID_i	User Identity
$GWID$	Gateway Identity
G_p	Elliptic Curve Generator
ΔT	Time-stamp Threshold
K_s	Gateway Node Master Secret
H(.)	One-way Hash Function
Enc(.)/Dec(.)	ECC Encryption/Decryption
$\bigoplus$, $\|\|$	XOR and Concatenation Respectively

TABLE 5.1
Notation and Abbreviations

A System Initialize Phase

In this subsection, the system's initialization phase is discussed. All steps in the initialization phase of the system are carried out by the gateway node in an offline manner. Thus, message generation and message communication in this phase occur in a secure environment. The gateway device computes parameters for the user devices and sensing devices. The gateway device decides level for the user device based on the position of the user in an organizational hierarchy and the level of sensing device based on its location of deployment in the environment. It is necessary to observe that none of the devices store their levels in any format.

Gateway Initialize Phase

The gateway initialize phase occurs as follow,

- Generates random private key $RGWN_k$ from the range of 1 to n where n is the large prime order of the elliptic curve.
- Generates a gateway random master key K_s.
- Computes gateway node public key as a $PUB_{GW_k} = RGWN_k$ * P, where P is the curve point.

User Device Initialize Phase

The user device initialize phase occurs as follow,

- Generates a random private key for each i^th user as RU_i from the range of 1 to n where n is a large prime order of the curve, and i ranges from 1 to the number of users in the IoT network.
- Computes public key for the user U_i as a $PUB_{U_i} = RU_i$ * P, where P is curve point.
- Gateway node will compute $B_1 = \mathrm{H}(PUB_{U_i}||K_s)$ and $B_2 = \mathrm{H}(l_i||K_s||H(PUB_{U_i}))$ where l_i is level of ith user based on its role in organization and k_s is master secret of gateway node. GWN will store B_1 and B_2 also in user device. GWN will it self not store.

Sensor Device Initialize Phase

The sensor device initialize phase occurs as follow,

- Generates a random number as a private key for each sensor node S_j called as a RSN_j.
- Computes public key for the sensing device S_j as a $PUB_{S_j} = RSN_j * P$, where P is the curve point.
- Generates random identity for each sensor node S_j as a SID_j.
- Gateway node will compute $D_1 = \mathrm{H}(PUB_{S_j}||K_s)$ and $D_2 = \mathrm{H}(l_j||K_s||H(PUB_{S_j}))$ where l_j is level of jth sensor based on its role in organization and k_s is master secret of gateway node. GWN will store D_1 and D_2 in sensor device. GWN will it self not store.

Gateway node fly parameters PUB_{GW_k}, PUB_{U_i}, PUB_{S_j} as a public parameters.

B User Registration Phase

In this section, the user registration process is discussed that is carried out in a secured manner between user device and the gateway device. The user registration phase follows following steps:

1. $U_i \xrightarrow{Request} GW$**:** The user U_i selects the password UPW_i, generates the random numbers R_a, R_b, computes the $TPW_i = \mathrm{H}(UPW_i||R_a) \bigoplus R_b$ and sends $Request = \{UID_i\}$ to the gateway GW.
2. $GW \xrightarrow{SmartCard} U_i$**:** The gateway computes, $Reg_i = \mathrm{H}(UID_i||K_s)$, computes $B_i = \mathrm{H}(l_i||K_s||UID_i)$ where l_i is the level of ith user based on its role in the organization. Generate smart card $SC = \{Reg_i, B_i, \mathrm{H}(.), E_p(a,b)\}$ and sends to the user U_i.

3. The user computes $L_1 = H(UID_i) \bigoplus R_a$, $TPW_i' = TPW_i \bigoplus R_b$, $L_2 = \mathrm{H}(UID_i||TPW_i')$, $Reg_i* = Reg_i \bigoplus R_b$, replaces Reg_i by Reg_i* in SC and creates final SC = $\{Reg_i*, L_1, L_2, B_i, \mathrm{H}(.), E_p(a,b)\}$

C Login and Session Key Agreement Phase

In this subsection,the login phase and the session key agreement phase is presented in which the user device U_i wants to access the data from the sensing device S_j, and for that, it tries to establish a session key with the S_j. In the login phase, the user provides UID_i, UPW_i and SC to the Smart Card Reader (SCR), the SCR verifies all the parameter and computes new parameters for the key agreement phase. All the steps of the session key agreement phase perform through the public channel. In this phase, U_i sends a request to the GW. The GW verifies level and other parameters of the U_i and prove it's access capabilities. Later on, through the GW device, U_i and S_j generates a mutually authenticated session key SK. Summary of authentication phase is given in Table 5.2. The login phase and the session key agreement consist of the following steps:

1. $U_i \xrightarrow{Request} SCR$**:** The user provides UID_i and UPW_i and SC to the SCR. The SCR computes $R_a* = L_1 \bigoplus \mathrm{H}(UID_i*)$, $TPW_i* = \mathrm{H}(UPW_i||R_a*)$, $L_2* = \mathrm{H}(UID_i||TPW_i*)$ and verifies $L_2* = L_2$. If verification gets success, SCR allows U_i for the further key agreement else abort the procedure.

2. $U_i \xrightarrow{Message1} GW$**:** Generate random number $r_1 \in F_p$. Compute $M_1 = \mathrm{H}(r_1||B_1)$, Get current time stamp T_1, Compute $M_2 = \mathrm{H}(M_1||PUB_{U_i}||T_1)$, Compute $M_3 = M_2 \oplus PUB_{U_i}$, Compute $M_4 = M_1 \oplus M_3$, Compute $M_5 = EncPUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j})$, Send $Message\ 1 = (M_3, M_4, M_5, T_1)$.

3. $GW \xrightarrow{Message2} S_j$**:** Verify time stamp $\Delta T = T_1* - T_1$, Compute $M_1^* = M_3 \oplus M_4$, Get r_1^*, PUB_{U_i}, $PUB_{S_j} = Dec_{RGWN_k}(M_5)$, Verify $M_1^{**} = \mathrm{H}(r_1^* || H(PUB_{U_i} || K_S))$, Compute $Tmp_1 = \mathrm{H}(PUB_{GW_k} || H(PUB_{S_j} || K_s) || T_2)$, Send *Message 2* = ($Tmp_1^*$, T_2) to S_j.

4. $S_j \xrightarrow{Message3} GW$**:** Verify time stamp $\Delta T = T_2* - T_2$, Verify $Tmp_1^* = \mathrm{H}(PUB_{GW_k} || D_1 || T_2) \stackrel{?}{=} Tmp_1$, If yes, send *Message 3* = ($M_6 = \mathrm{H}(PUB_{GW_k} || D_1 || T_3)$, T_3, D_2) to GW,

5. $GW \xrightarrow{Message4} S_j$**:** Verify time stamp $\Delta T = T_3* - T_3$. Verify $M_6^* = \mathrm{H}(PUB_{GW_k} || H(PUB_{S_j} || K_s) || T_3) \stackrel{?}{=} M_6$. Get l_i and l_j from B_2 and D_2. if $l_i \leq l_j$, then allow else deny. Compute current time stamp T_4. Generate random number k_1. Compute $M_7 = \mathrm{H}(k_1 || PUB_{GW_k} || PUB_{S_j} || PUB_{U_i} || r_1^* || T_4)$. Compute $M_8 = PUB_{GW_k} \oplus PUB_{U_i}$. Compute $M_9 = Enc_P UB_{S_j}(k_1, r_1^*)$. Compute $M_{10} = Enc_P UB_{U_i}(k_1)$. Send *Message 4* = ($M_7$,$M_8$,$M_9$,$T_4$) to SN_j and Send *Message 5* = (M_{10}, T_4) to U_i.

6. U_i ***computes:*** Verifies $\Delta \mathrm{T} \leq T_4* - T_4$. Get $k_1^* = Dec_{RU_i}(M_{10})$. Generate r_2. $M_{11} = Enc_P UB_{S_j}(r_2)$. Send M_{11}, T_5. Compute session key SK = $\mathrm{H}((r_1 || PUB_{U_i}), \mathrm{H}(k_1^* || PUB_{GW_k} || PUB_{S_j} || PUB_{U_i} || r_1), r_2, T_5)$.

7. S_j ***computes:*** Verifies $\Delta \mathrm{T} \leq T_4* - T_4$. Get k_1^*,$r_1^{**} = DeC_{RSN_j}(M_9)$. Get $PUB_{U_i} = M_8 \oplus PUB_{GW_k}$. Compute $M_7^* = \mathrm{H}(k_1^* || PUB_{GW_k} || PUB_{S_j} || PUB_{U_i} || r_1^{**} || T_3) \stackrel{?}{=} M_7$. $X_1 = \mathrm{H}(k_1^* || PUB_{GW_k} || PUB_{S_j} || PUB_{U_i} || r_1^{**})$. $X_2 = \mathrm{H}(r_1 || RU_i)$ and sleep. Wake up and Verify $\Delta \mathrm{T} \leq T_5* - T_5$. Get $r_2^* = Dec_{RSN_j}(M_{11})$. Verify $X_2* = \mathrm{H}(r_1 || RU_i^*) \stackrel{?}{=} X_2$. Verify $X_2* = \mathrm{H}(r_1 || RU_i^*) \stackrel{?}{=} X_2$. Compute Session key SK = $\mathrm{H}(X_2 || X_1 || r_2^* || T_5)$.

User/SCR	Gateway	Sensor
UID_i and UPW_i and SC to SCR. Computes R_a* = $L_1 \bigoplus$ H(UID_i*), TPW_i* = H($UPW_i||R_a*$), L_2* = H($UID_i||TPW_i*$) and verifies $L_2* = L_2$. Generate $r_1 \in F_p$. Compute M_1 = H($r_1||B_1$), M_2 = H($M_1||PUB_{U_i}||T_1$), M_3 = $M_2 \oplus PUB_{U_i}$, M_4 = $M_1 \oplus M_3$, M_5 = $Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j})$, Send *Message 1* = ($M_3$, M_4, M_5, T_1). to GW	Verify $\Delta T = T_1* - T_1$, Compute $M_1^* = M_3 \oplus M_4$, Get r_1^*, PUB_{U_i}, PUB_{S_j} = $Dec_{RGWN_k}(M_5)$, M_1^{**} = H($r_1^*||H(PUB_{U_i}||K_S)$), Tmp_1 = H($PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_2$), Send *Message 2* = ($Tmp_1^*$, T_2) to S_j.	Verify $\Delta T = T_2* - T_2$, Verify Tmp_1^* = H($PUB_{GW_k}||D_1||T_2$)$\stackrel{?}{=}$ Tmp_1, If yes, send *Message 3* = (M_6 = H($PUB_{GW_k}||D_1||T_3$), T_3, D_2) to GW,
Verifies $\Delta T \leq T_4* - T_4$. Get k_1^* = $Dec_{RU_i}(M_{10})$. Generate r_2. M_{11} = $Enc_PUB_{S_j}(r_2)$. Send M_{11}, T_5. Compute session key SK = H(($r_1||PUB_{U_i}$), H($k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1$)), r_2, T_5).	Verify $\Delta T = T_3* - T_3$, M_6^* = H($PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_3$)$\stackrel{?}{=}$ M_6. Get l_i and l_j from B_2 and D_2. if $l_i \leq l_j$, then allow else deny. Compute T_4. Generate random k_1. Compute M_7 = H($k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}$ $||r_1^*||T_4$), M_8 = $PUB_{GW_k} \oplus PUB_{U_i}$, M_9 = $Enc_PUB_{S_j}(k_1, r_1^*)$, M_{10} = $Enc_PUB_{U_i}(k_1)$. Send *Message 4* = (M_7,M_8,M_9,T_4) to SN_j and Send *Message 5* = (M_{10}, T_4) to U_i.	Verifies $\Delta T \leq T_4*$ - T_4. Get k_1^*,r_1^{**} = $DeC_{RSN_j}(M_9)$. Get PUB_{U_i} = $M_8 \oplus PUB_{GW_k}$. Compute M_7^* = H($k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}$ $||r_1^{**}||T_3$)$\stackrel{?}{=}$ M_7. X_1 = H($k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}$ $||r_1^{**}$). X_2 = H($r_1||RU_i$) and sleep. Wake up and Verify $\Delta T \leq T_5* - T_5$. Get r_2^* = $Dec_{RSN_j}(M_{11})$. Verify X_2* = H($r_1||RU_i^*$)$\stackrel{?}{=}$ X_2. Verify X_2* = H($r_1||RU_i^*$)$\stackrel{?}{=}$ X_2. Compute Session key SK = H($X_2||X_1||r_2^*||T_5$).

TABLE 5.2
Authentication and Key Exchange

5.5 Security Analysis

A Informal Security Analysis

The Dolev-Yao channel [Dolev & Yao (1981)] is a communication model based on *snd* and *rcv* operations. In this subsection, the informal security analysis for the proposed protocol based on a Dolev-Yao channel is presented. The polynomial time adversary $\mathcal{A}$ can access and control the Dolev-Yao channel. In the proposed scheme, the initialize phase implemented over the secure channel, and the gateway device is a trusted secure device. Table 5.3 presents security comparison between proposed schemes and existing schemes.

ANONYMITY AND TRACEBILITY

The anonymity for the security algorithm assures that an identity of the user is secured against the adversary's knowledge. In the initialize phase of the proposed scheme, the trusted GW generates an identity of the i^th user as UID_i and j^th sensing device as SID_j. Later on GW computes $B_1 = \mathrm{H}(PUB_{U_i}||K_s)$ and $B_2 = \mathrm{H}(l_i||K_s||H(PUB_{U_i}))$ for each U_i and $L_2 = \mathrm{H}(UID_i||TPW_i')$ in user registration phase. During the login and key-exchange phase, user communicates message $M_5 = EncPUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j})$ which is secured through the public-key of gateway. Now, let us assume that an adversary $\mathcal{A}$ intercepts other messages. All the intercepted messages are either protected through the one-way hash function H(.) or the encryption. Thus, no vulnerability exists which helps an adversary $\mathcal{A}$ to achieve the UID_i. In many realtime application, it is expected that an adversary $\mathcal{A}$ must not be able to trace the user and messages communicated by him/her. The

$\mathcal{A}$ can trace U_i if and only if an identity of the U_i is revealed. Thus, the proposed LDA based scheme achieves anonymity and tracebility.

ACHIEVES MUTUAL AUTHENTICATION AND SESSION KEY AGREEMENT

The mutual authentication property assures each party that the message is received from the valid source. After receiving of the first message from GW, the S_j device retrieves it's identity and performs verification of $Tmp_1^* = \mathrm{H}(PUB_{GW_k}||D_1||T_2) \stackrel{?}{=} Tmp_1$. Any adversary $\mathcal{A}$ uses PUB_{GW_k} to prove himself/herself as a valid user, $\mathcal{A}$ does not get success due to presence of parameters like D_1 in the verification which are not available with $\mathcal{A}$. Similarly, verification $M_6^* = \mathrm{H}(PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_3) \stackrel{?}{=} M_6$ assure about the authenticity of S_j to the GW. The verification $X_2* = \mathrm{H}(r_1||RU_i^*) \stackrel{?}{=} X_2$ helps user device U_i to authenticate the GW and the verification $M_7^* = \mathrm{H}(k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^{**}||T_3) \stackrel{?}{=} M_7$ helps S_j to authenticate the GW. The computed session key $\mathrm{SK} = \mathrm{H}(X_2||X_1||r_2^*||T_5)$ also includes identities of each entity in indirect manner and random numbers thus the proposed protocol achieves mutual authentication and session key agreement.

SECURE AGAINST REPLAY ATTACK

In the replay attack, an adversary $\mathcal{A}$ replays previously communicated messages after some time or in the next session. To provide security against the replay attack, the proposed scheme uses random parameters and timestamps. Each communicated message contains time-stamp T_i which is validated by the receiving entity through $\Delta\mathrm{T} \leq T_i*$ - T_i verification where T_i* is the current time at receiver side and ΔT predefined maximum threshold time. Even though $\mathcal{A}$ replays any

message, the LDA receiver will catch that the received message is replayed. Thus, the proposed scheme is secure against the replay attack.

SECURE AGAINST USER/SENSOR LEVEL SIDE CHANNEL ATTACK

The level l_i defines the level of user U_i in the hierarchy and l_j defines the level of sensing device S_j in the deployment. If an adversary $\mathcal{A}$ gets the level l_i then he/she can guess the role of U_i in the organization, similarly if l_j is available to the $\mathcal{A}$ then he/she can guess the sensing device deployment location. Thus, it is important to secure l_i and l_j. In the proposed protocol, none of the entity (not even GW) stores l_i and l_j. The U_i stores l_i in parameter $B_2 = \mathrm{H}(l_i||K_s||H(PUB_{U_i}))$ and the S_j stores l_j in parameter $D_2 = \mathrm{H}(l_j||K_s||H(PUB_{S_j}))$ which are protected by one-way hash function and the gateway master K_s. Hence, the proposed LDA-GIOT scheme is secured against a level side channel attack.

KEY ESTABLISHMENT WITH PERFECT FORWARD SECRECY

In perfect forward secrecy, it is assumed that the adversary $\mathcal{A}$ somehow obtains the user secret key RU_i and sensing device secret key RSN_j, then the adversary $\mathcal{A}$ can retrieve the the messages through the knowledge of RSN_j. The hash function protects the messages M_i, and r_i is an unknown random number that does not provide any useful information. Similarly, through the RU_i, an adversary $\mathcal{A}$ can obtain the messages. These all the parameters are secured through the one-way hash function and do not provide any useful information. The session key computed using $\mathrm{SK} = \mathrm{H}((r_1||PUB_{U_i}), \mathrm{H}(k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1), r_2, T_5)$. Where parameter r_2 is not available to $\mathcal{A}$ and $\mathcal{A}$ can not get k_1 or r_2 by just knowing a RU_i and RSN_j. It is assumed that the user device's physical capturing and

the user's secret key relieve will not coincide, and this assumption is valid because one is a physical attack while the other is a guessing attack. Thus the proposed LDA scheme achieves the perfect forward secrecy.

GATEWAY DEVICE BYPASS ATTACK

In the gateway device bypass attack, an adversary $\mathcal{A}$ tries to behave as a GW or any one of the device U_i or S_j try to behave as a GW. In the proposed scheme, during the initialize phase, the GW computes $B_1 = \mathrm{H}(PUB_{U_i}||K_s)$ and $B_2 = \mathrm{H}(l_i||K_s||H(PUB_{U_i}))$ while $D_1 = \mathrm{H}(PUB_{S_j}||K_s)$ and $D_2 = \mathrm{H}(l_j||K_s||H(PUB_{S_j}))$ for S_j. All these computation involves gateway master secret K_s. Thus, neither $\mathcal{A}$ nor the U_i or S_j can compute the above parameters. Hence, the proposed LDA scheme is secured against gateway device bypass attack.

STOLEN USER DEVICE ATTACK

In this attack, an adversary $\mathcal{A}$ gets physical user device and retrieves stored parameters B_1 and B_2. Now the session key is computed as $\mathrm{SK} = \mathrm{H}(X_2||X_1||r_2^*||T_5)$ where r_2 is the random parameters. If an adversary $\mathcal{A}$ gets the user secret RU_i then also he can not guess random r_t. By capturing the user device, an adversary $\mathcal{A}$ can not capture the user identity also. Thus, it is computationally non-feasible for an adversary to compute the session key SK in polynomial time.

SENSING DEVICE CAPTURE ATTACK

In this attack, an adversary $\mathcal{A}$ gets the physical user device and gets stored parameters $\{D_1, D_2\}$. Now if an adversary $\mathcal{A}$ tries to compute the session key $\mathrm{SK} = \mathrm{H}((r_1||PUB_{U_i}), \mathrm{H}(k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1), r_2, T_5)$ then it requires three

random numbers r_1, r_2 and as well as the timestamp T_5. Thus, even though an adversary $\mathcal{A}$ physically attacks the sensing device as well as track the messages, he can not obtain the r_1, r_2 and r_t from it. The sensing device does not store the sensing device identity; thus, through the sensing device attack, $\mathcal{A}$ can not track the sensing device also; thus the proposed scheme is secured against the sensing device capture attack.

USER DEVICE IMPERSONATION ATTACK

In this attack, an adversary $\mathcal{A}$ intercepts all the messages send by user U_i and tries to replace those messages by other manually generated messages. Let $\mathcal{A}$ intercepts *Message 1* = $\{M_3, M_4, M_5, T_1\}$. The *Message 1* is secured through the hash functions, still let us assume that $\mathcal{A}$ creates *Message 1** = $\{M_3*, M_4*, M_5*, T_1*\}$ and forwards it to gateway. Now the gateway device *GW* extracts data from the message *Message 1** and performs the verification $M_1^{**} = \mathrm{H}(r_1^* || H(PUB_{U_i} || K_S))$ which contains the fresh random number generated by U_i and the gateway master secret K_s which is computationally unfeasible to generate same $K_s* = K_s$ in a polynomial time for an adversary $\mathcal{A}$.

Scheme	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
[Farash et al. (2016)]	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
[Wazid et al. (2018)]	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗	✗
[Zhou et al. (2019)]	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗
[Shin & Kwon (2020)]	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗
[Jangirala et al. (2020)]	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗
LDA-GIoT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Legends: S_1: Tracebility, S_2: Anonymity, S_3: Mutual authentication and Integrity, S_4: Replay attack , S_5: Man-in-The-Middle Attack, S_6: Forward secrecy, S_7: Gateway by pass attack, S_8: Gateway impersonation attack , S_9: Sensing device capture attack, S_{10}: Privilege insider attack, S_{11}: Level Dependent Authentication, ✓: the protocol supports this feature, ✗: the protocol doesn't support this feature.

TABLE 5.3
Security Comparison

Sensing Device Impersonation Attack

In this attack, an adversary $\mathcal{A}$ intercepts all the messages send by the sensing device S_j to the gateway device GW. Let $\mathcal{A}$ intercepts *Message 3* = (M_6 = H($PUB_{GW_k}||D_1||T_3$), T_3, D_2) and try to replace these messages by *Message 3** = $\{(M_6* = \text{H}(PUB_{GW_k} * ||D_1 * ||T_3*), T_3*, D_2*)\}$. The message M_6 is encrypted through the public-key of GW. At the other side after receiving this message, GW performs $M_6^* = \text{H}(PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_3) \stackrel{?}{=} M_6$ which includes secure master secret K_s. Thus, it is infeasible to generate a *Message 3** which is similar to *Message 3*. Thus, the proposed LDA scheme is secured against a sensing device impersonation attack.

Gateway Device Impersonation Attack

In this attack, an adversary $\mathcal{A}$ intercepts all the messages send by the gateway device GW and trie to impersonate as a gateway device. Now let an adversary

$\mathcal{A}$ captures *Message 4* = (M_7,M_8,M_9,T_4) to SN and Send *Message 5* = (M_{10}, T_4) and generates new messages *Message 4** = (M_7*,M_8*,M_9*,T_4*) , *Message 5** = (M_{10}*, T_4*). and forwards *Message 4** and *Message 5** to sensing device U_i and S_j respectively. These messages are either encrypted by the device secrets respectively as well as hash functions. Thus it is infeasible to get these both the secrets in polynomial time for an adversary $\mathcal{A}$. Hence, the proposed LDA scheme is secured against the gateway device impersonation attack.

B Mutual Authentication Using BAN Logic

In this section,the mutual authentication verification phase is discussed between the user U_i and the sensing device SD_j using the BAN Logic. This proof mainly consists of the following steps: (I) Initial assumption, (II) Goal declaration, (III) Message formation, and (IV) Formal verification.

1. ***Initial assumptions:***

   J_1. $U_i \mid\equiv \#(T_i)$, $SN_j \mid\equiv \#(T_i)$,$GW \mid\equiv \#(T_i)$;

   J_2. $U_i \mid\equiv \#(r_1)$, $GW \mid\equiv \#(r_2)$, $SD_j \mid\equiv \#(r_2)$, $U_i \mid\equiv \#(r_3)$;

   J_3. $GW \mid\equiv U_i \Rightarrow X$, $GW \mid\equiv SN_j \Rightarrow X$, $SN_j \mid\equiv GW \Rightarrow X$.

   J_4. $U_i \mid\equiv (\overset{GWID}{\longmapsto} GW)$, $SN_j \mid\equiv (\overset{RIDU_i}{\longmapsto} U_i)$, $GW \mid\equiv (\overset{RSN_j}{\longmapsto} SN_j)$

   J_5. $U_i \mid\equiv SN_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} SN_j)$

2. ***Goal Declaration:*** The expected goals in the proposed LDA scheme includes trust in shared key and freshness of communicated messages. In LDA scheme, expected goals are as follow:

G_1. $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SN_j$

G_2. $SN_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SN_j$

G_3. $U_i \mid\equiv SN_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SN_j$

G_4. $SN_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SN_j$

G_5. $U_i \mid\equiv GW \mid\sim \#(X)$

G_6. $GW \mid\equiv SN_j \mid\sim \#(X)$

G_7. $SN_j \mid\equiv GW \mid\sim \#(X)$

3. ***Message Formation:*** The login and Authentication phase of the proposed LDA scheme includes exchanges of the following messages which can be written in the generic form as follow:

Message 1: $U_i \rightarrow GW$: $((M_2 \oplus PUB_{U_i}), (M_1 \oplus M_3), (Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j}), T_1)$

Message 2: $GW \rightarrow SN_j$: $(\mathrm{H}(PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_2), T_2)$

Message 3: $GW \rightarrow U_i$: $(\mathrm{H}(PUB_{GW_k}||D_1||T_3), T_3, D_2)$

Message 4: $GW \rightarrow SN_j$: $((\mathrm{H}(k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)), (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_PUB_{S_j}(k_1,r_1^*)), T_4)$

Message 5: $GW \rightarrow U_i$: $(Enc_PUB_{U_i}(k_1), T_4)$

Idealized form: The ideal forms for the above messages can be written as follows:

Message 1: $U_i \rightarrow GW$: $GW \triangleleft \langle < ((M_2 \oplus PUB_{U_i}), (M_1 \oplus M_3), (Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j}), T_1) > \rangle_{GW \mid\equiv (\overset{S_j}{\longmapsto} S_j)}$

Message 2: $GW \rightarrow SN_j$: $SN_j \triangleleft \langle < ((PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_2), T_2) > \rangle$

$S_j|\equiv(\overset{U_i}{\vdash\!\!\!\longrightarrow}U_i)$

Message 3: $GW \rightarrow U_i$: $U_i \triangleleft \langle < ((PUB_{GW_k}||D_1||T_3), T_3, D_2) > \rangle$ $GW|\equiv(\overset{U_i}{\vdash\!\!\!\longrightarrow}U_i)$

Message 4: $GW \rightarrow S_j$: $S_j \triangleleft \langle < (((k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)), (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_PUB_{S_j}(k_1,r_1^*)), T_4) > \rangle$

Message 5: $GW \rightarrow U_i$: $U_i \triangleleft \langle < (Enc_PUB_{U_i}(k_1), T_4) > \rangle$

4. ***Formal Verification***

Theorem 3. *The proposed scheme achieves the secure mutual authentication between the user U_i and the sensing device SD_j, and it achieves expected goals.*

Proof. Expected goals [G_1 - G_7] are achieved as follow:

S_1**:** from the message 1,

$GW \triangleleft \langle < ((M_2 \oplus PUB_{U_i}), (M_1 \oplus M_3), (Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j}), T_1)) > \rangle$ $GW|\equiv(\overset{U_i}{\vdash\!\!\!\longrightarrow}U_i)$

S_2**:** Using S_1, R_1, R_2 and J_2,

$GW \mid\equiv U_i \mid\sim \langle < ((Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j}), T_1) > \rangle$

S_3**:** Using S_1, S_2, J_1, R_3,

$GW \mid\equiv U_i \mid\equiv \langle < ((M_2 \oplus PUB_{U_i}), (M_1 \oplus M_3), (Enc_PUB_{GW_k}(r_1, PUB_{U_i}, PUB_{S_j}), T_1)) > \rangle$

S_4**:** from the message 2,

$S_j \triangleleft \langle < ((PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_2), T_2) > \rangle$ $S_j|\equiv(\overset{GW_k}{\vdash\!\!\!\longrightarrow}GW_k)$

S_5**:** Using S_4, R_1, R_2 and J_3,

$S_j \mid\equiv GW \mid\sim \langle < (((PUB_{GW_k}||H(PUB_{S_j}||K_s)||T_2), T_2) > \rangle$

S_6**:** Using S_4, J_2, R_2,

$GW \mid\equiv SD_j \mid\equiv \langle < ((r_3), (GWID||SID_j||r_3||T_8), (GWID||SID_j||r_2||T_8), T_8) > \rangle$

S_7**:** Using J_3, J_4, J_6, R_8, R_2,

$S_j \mid\equiv (GWID||SID_j||r_3||T_8)$, $S_j \mid\equiv (GWID||SID_j||r_2||T_8)$,

S_8**:** from the message 3,

$U_i \triangleleft \langle < ((PUB_{GW_k}||D_1||T_3), T_3, D_2) > \rangle_{U_i \mid\equiv (\overset{GW_k}{\longmapsto} GW_k)}$

S_9**:** Using S_8, R_2, J_2, J_3,

$U_i \mid\equiv GW \mid\sim \langle < (PUB_{GW_k}||D_1||T_3), T_3, D_2) > \rangle$

S_{10}**:** Using S_9, J_3, J_4, R_4,

$U_i \mid\equiv GW \mid\equiv \langle < (PUB_{GW_k}||D_1||T_3), T_3, D_2) > \rangle$

S_{11}**:** from the message 4,

$S_j \triangleleft \langle < (((k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)), (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_PUB_{S_j}(k_1,r_1^*)), T_4) > \rangle$

S_{12}**:** Using S_{11}, R_2, R_3 and J_1,

$S_j \mid\equiv GW \mid\sim \langle < (((k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)), (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_PUB_{S_j}(k_1,r_1^*)), T_4) > \rangle$

S_{13}**:** Using S_{13}, J_2, R_2,

$S_j \mid\equiv GW \mid\sim \langle < (((k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)), (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_PUB_{S_j}(k_1,r_1^*)), T_4) > \rangle$

S_{14}: Using S_{13}, S_{14}, J_3,J_4 R_7,R_3,

$S_j \mid\equiv (PUB_{GW_k} \oplus PUB_{U_i}), (Enc_P UB_{S_j}(k_1, r_1^*)), T_4$,

S_{15}: Using S_{11}, S_{13}, S_4, S_{14}, R_2, R_7, J_3,

$S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ [G_2]

S_{16}: Using S_8, S_9, S_{10}, R_4, R_8, J_3,

$U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ [G_1]

S_{17}: from the message 5,

$U_i \triangleleft \langle < (Enc_P UB_{U_i}(k_1), T_4) > \rangle$

S_{18}: Using S_{17}, R_8, R_9 and J_1,

$U_i \mid\equiv GW \mid\sim \langle < (Enc_P UB_{U_i}(k_1), T_4) > \rangle$

S_{19}: Using S_{17}, S_{18}, J_1, R_8,

$U_i \mid\equiv GW \mid\sim \langle < (Enc_P UB_{U_i}(k_1), T_4) > \rangle$

S_{20}: Using S_3,S_4, S_{14}, S_{15}, J_5, R_3, R_7 and R_8,

$S_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ [G_3]

S_{21}: Using S_9,S_{10}, S_{11}, J_5, R_3, R_7 and R9,

$U_j \mid\equiv S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$ [G_4]

S_{22}: Using S_{20}, S_{21}, R_5, R_8, R_6,

$S_j \mid\equiv GW \mid\sim \#(Message1)$

S_{23}: Using S_4, S_5, S_{12}, S_{22}, R_3, R_6, R_7, J_2 and J_3,

$S_j \mid\equiv GW \mid\sim \#(Message1, Message3)$ [G_7]

S_{24}: Using S_6, S_{12}, S_{13}, R_3, R_6, R_8, J_2 and J_3,

$GW \mid\equiv S_j \mid\sim \#(Message2)$ $[G_6]$

S_{25}**:** Using S_{10}, R_3, R_7, R_9, J_2 and J_4,

$U_i \mid\equiv GW \mid\sim \#(Message4)$ $[G_5]$

Thus, the above verification clearly shows that the proposed authentication scheme achieves all defined goals (G_1 - G_7). □

C Formal Security Analysis using ROR Model

In this section, the formal security analysis for the proposed scheme using a widely accepted and proved secure random oracle based model proposed by Abdalla et al. [Abdalla & Pointcheval (2005)] is discussed. The authors in [Abdalla & Pointcheval (2005)] proposed the Real-Or-Random (ROR) model, which helps security designers to prove that the proposed scheme achieves polynomial-time security against an adversary $\mathcal{A}$'s advantage of breaking the security. *Security Proof:* The security model discussed in chapter 2 is discussed for to prove formal security of the proposed scheme.

Theorem 4. *If $\mathcal{A}$ is a polynomial time attacker running against the proposed protocol $LDA-P$ within a limited time t. Let q_h determines the range space of hash ($\mathcal{H}$) queries, q_s denotes the number of send ($\mathcal{S}$) queries, q_e represents the number of execute ($\mathcal{E}$) query, the uniformly distributed password dictionary is defined as DC either against the user U_i or the sensing device S_j and Adv_{ρ}^{ECDLP} defines the advantage of $\mathcal{A}$ of breaking the discrete logarithm problem of $\mathcal{A}$ then*

the proposed protocol is considered secured if,

$$Adv_p^{LDA}(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}|}, \rho_{fp})) + Adv_\rho^{ECDLP} + (\frac{1}{2^{l_j}}) \tag{5.1}$$

In equation 5.1, l_h is the size of the return value of a hash ($\mathcal{H}$) query generated by an adversary $\mathcal{A}$ in bits, l_r is the size of the random nonce generated by the protocol $LDA-P$. $|\mathcal{DC}|$ shows the finite size of a password dictionary, and ρ_{fp} shows the probability of a false positive occurrence by $\mathcal{A}$.

Proof. The proposed protocol is secured if the $Adv_p^{LDA}(\mathcal{A})$ is negligible using the ROR model.To prove the security of the proposed scheme, five games say Gm_0, to Gm_4 are defined. Now, let us define an event SC_i which represents the correct guess for the bit b in each game Gm_i via the test query $\mathcal{T}$ by an adversary $\mathcal{A}$. □

Gm_0: The first game Gm_0 is the original security game which is corresponding to an original attack performed by an adversary $\mathcal{A}$ on the *LDA-P*. At the beginning of the game, adversary $\mathcal{A}$ chooses bit b. Hence it follows that,

$$Adv_p^{LDA}(\mathcal{A}) = 2 * Pr[\mathcal{SC}_0] - 1. \tag{5.2}$$

Gm_1: The Gm_1 is modelled as a passive attack in which $\mathcal{A}$ performs execute query $\mathcal{E}(\chi_{U_i}^p, \chi_{GW}^q, \chi_{S_j}^r)$ and captures all communicated messages (*Message 1* to *Message 7*). Based on all these messages $\mathcal{A}$ tries to determine the session key *SK* and after completion of the game $\mathcal{A}$ performs a test query $\mathcal{T}$. The output of $\mathcal{T}$ determines weather it is veritable session key or the random number. The session key is computed by the user U_i and the sensing device S_j as **SK** = H($(r_1||PUB_{U_i})$,

$H(k_1^*||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1)$, r_2, T_5) and $\mathbf{SK} = H(X_2||X_1||r_2^*||T_5)$ respectively. The session key computation involves random numbers which are secured through the private keys of user and sensing devices. Since, interception of the messages *Message 1* to *Message 7* does not lead to compromise of the session key *SK* or any other secret credentials. Thus, the winning probability of the adversary $\mathcal{A}$ does not increase in Gm_1.

$$Pr[\mathcal{SC}_0] = Pr[\mathcal{SC}_1]. \tag{5.3}$$

Gm_2*:* The Gm_2 involves two more queries in the Gm_1. The Gm_2 executes *Send* query and Hash *H(.)* through which an adversary $\mathcal{A}$ communicates with the user U_i and the sensor S_j. Through the several *H(.)* queries, $\mathcal{A}$ verifies hash digest. Thus, Gm_2 is an active attack in which $\mathcal{A}$ tries to convince the U_i and S_j to accept the forged messages. The messages $M_7 = H(k_1||PUB_{GW_k}||PUB_{S_j}||PUB_{U_i}||r_1^*||T_4)$. , $M_8 = PUB_{GW_k} \oplus PUB_{U_i}$, $M_9 = Enc_PUB_{S_j}(k_1,r_1^*)$, $M_{10} = Enc_PUB_{U_i}(k_1)$, $M_{11} = Enc_PUB_{S_j}(r_2)$ involves throughout the use of random numbers, time-stamps, sensing device identity, gateway master secret, user identity which will not provide any success to an adversary $\mathcal{A}$ in collusion verification of the generated message digest. Thus, through the birthday paradox, it follows that,

$$Pr[\mathcal{SC}_1] - Pr[\mathcal{SC}_2] \leq \frac{q_h^2}{2^{l_h}}. \tag{5.4}$$

Gm_3*:* The Gm_3 translated from Gm_2. The Gm_3 performs all the *Corrupt* queries. Through the query *CorruptUserDevice*, an adversary $\mathcal{A}$ receives all the stored parameters like X_1, X_2, K_1, B_1 and the other curve parameters. Now, $\mathcal{A}$ tries to guess the correct user ID and password PW for the user U_i through the dictionary

attack. To guess the correct password, $\mathcal{A}$ needs UPW and R_a* to validate the $TPW_i* = \mathrm{H}(UPW_i||R_a*)$, $L_2* = \mathrm{H}(UID_i||TPW_i*)$ and verifies $L_2* = L_2$. The value of R_a* is a random value and it's correct guess depends on the correct guess for an identity UID_i. Thus, due to these limitations for the *Send* query access in a polynomial time, it is infeasible to guess the correct pair of (UID_i, UPW_i) in a polynomial time. In similar way, Thus,

$$Pr[\mathcal{SC}_3] - Pr[\mathcal{SC}_2] \leq max(q_s, (\frac{1}{|\mathcal{DS}|}, \rho_{fp})) \tag{5.5}$$

Gm_4: The Gm_4 is translated from the Gm_3. In this game an adversary $\mathcal{A}$ performs *CorruptUserLevel*(χ^x), *CorruptSensingLevel*(χ^y), *CorruptSensingDevice*. Through these queries, $\mathcal{A}$ tries to get the level of the user device or the sensor device. Now, let us assume that the probability of guessing the correct level is $\frac{1}{2^{l_j}}$ where 2^{l_j} represents the number of bits used for the level. Thus, after guessing the level of user device or sensing device, $\mathcal{A}$ tries to validate it's guess. To validate the user level l_i, an adversary $\mathcal{A}$ requires $B_1 = \mathrm{H}(PUB_{U_i}||K_s)$ and $B_2 = \mathrm{H}(l_i||K_s||H(PUB_{U_i}))$ and to validate the sensing device level l_j, an adversary $\mathcal{A}$ needs $D_1 = \mathrm{H}(PUB_{S_j}||K_s)$ and $D_2 = \mathrm{H}(l_j||K_s||H(PUB_{S_j}))$. To get these parameters, $\mathcal{A}$ must need the secret key of the user device (RU_i) or sensing device (RSN_j) which is computationally infeasible for an adversary to get in polynomial time. Thus,

$$Pr[\mathcal{SC}_4] - Pr[\mathcal{SC}_3] \leq \frac{1}{2^{l_j}} + Adv_{\rho}^{ECDLP} \tag{5.6}$$

Now, after completion of all the games, $\mathcal{A}$ doesn't get success. Now $\mathcal{A}$ have only one option left in which $\mathcal{A}$ try to guess the correct value of bit "b" and perform the $\mathcal{T}$ query. The success probability of this query is $\frac{1}{2}$. So after all the

games, it is clear that,

$$Pr[\mathcal{SC}_4] = \frac{1}{2} \tag{5.7}$$

Now, from equation 5.2, $\frac{1}{2} * Adv_{LDA-P}$ = $[Pr[SC_0]$ - $\frac{1}{2}]$. So by using the triangular inequality, $[Pr[\mathcal{SC}_1]$ - $[Pr[\mathcal{SC}_4] \leq [Pr[\mathcal{SC}_1]$ - $[Pr[\mathcal{SC}_2]$ + $[Pr[\mathcal{SC}_2]$ - $[Pr[\mathcal{SC}_4]$ $\leq [Pr[\mathcal{SC}_1]$ - $[Pr[\mathcal{SC}_2]$ + $[Pr[\mathcal{SC}_2]$ - $[Pr[\mathcal{SC}_3] \leq \frac{q_h^2}{2^{l_h}}$ + max$(q_s, (\frac{1}{|\mathcal{DC}|}, \rho_{fp}))$ + (Adv_{ρ}^{ECDLP}) + $(\frac{1}{2^{l_j}})$. Using equations 6.4-6.6,

$$\begin{aligned} |Pr[\mathcal{SC}_0] - \frac{1}{2}| \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}|}, \rho_{fp})) \\ + Adv_{\rho}^{ECDLP} + (\frac{1}{2^{l_j}}) \end{aligned} \tag{5.8}$$

So finally, from the equation 5.2 and 6.7,

$$\begin{aligned} Adv_{LDA-P}(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}|}, \rho_{fp})) \\ + Adv_{\rho}^{ECDLP} + (\frac{1}{2^{l_j}}) \end{aligned} \tag{5.9}$$

5.6 Implementations and Testbeds

An environment for the implementation of the proposed LDA scheme is highlighted in the following Table 5.4. As a user device, laptop and desktop is used, the raspberry pi as a gateway device and the NodeMCU connected with the sensors as a sensing device.

Network Model	Generic IoT Model
Broker	Mosquitto
Protocol	Using MQTT
Language	Python
ECC Curve	NIST P-256 Curve
Secure channel	By Enabling TLS communication in Mosquitto
ECC Multiplication	Using double and Add method
Message format	JSON Type
User Device	Intel (R) Core (TM) i3-7500 CPU with 2.80 GHz.
Gateway System	Raspberry Pi 3 Model B, 1 GB RAM.
Sensing device	NodeMCU + Raspberry Pi

TABLE 5.4
Implementation Environment

The following Fig. 5.2 shows the computed session key between the user device and the sensing device.

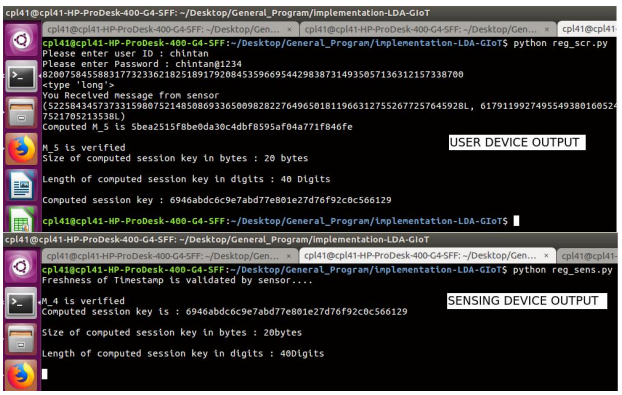

Figure 5.2. LDA-GIoT Session Key

For implementations we have used SHA256 as a hash function and NIST recommended P-256 Curve. The security level achieved through implementation of proposed scheme is 128 bit.

5.7 Use cases

This chapter proposes two factor based LDA scheme for the user-gateway-sensor model. The proposed scheme aims to provide a lightweight communication between the multiple user devices and the numerous amount of lightweight sensing devices deployed on the ground. The following are some of the significant use-cases of the proposed work.

- The proposed scheme presents significantly improved authentication scheme for the IoT application where the quantity of deployed sensing devices is large and have hierarchical user base.
- The proposed scheme can be highly useful in smart industry, smart education campus, smart transportation and so on.
- The significant use case of the proposed scheme mostly lies in environment where user doesn't have option to provide biometric trait as an input and still wants to receive a data from the sensing devices.

The novelty of the proposed work lies in its dynamic environment, lightweight computations and real-time implementations. Real-time implementation of the proposed ECC based LDA scheme using MQTT protocol (that itself is a lightweight communication protocol used by the IoT industry) makes it novel. The real-time implementation using MQTT protocol has advantages like the lightweight header and reliable communication over other conventional protocols.

5.8 Comparative Analysis

A Network Parameter Analysis

Round-trip Delay

A Round-Trip Delay (RTD) is computed as an average time required by a communicated packet to arrive at the destination from the source. The round-trip delay involves queuing delay, processing delay, transmission delay, and the propagation delay. The processing delay includes cryptographic operations, while the propagation delay includes travel time required by a packet. For the experimental purpose, through our scenario of numerous users, uni gateway, and multiple sensing devices, simultaneous requests was generated to the gateway device from the user devices for accessing the sensors at different levels. Then, The average RTD at the sensing device, which includes the time between the sensor's reply to the gateway and gateway's response to the sensor, is 0.4825 second. The average RTD at a user device, which includes the time required between sending a request to receiving a reply from the gateway via sensing devices, is 0.5282 seconds. If some requests are sent in which the user is not eligible to access the sensor at a particular level, then the RTD gets a little hike due to the gateway node taking little more verification time. If the gateway device does not find a valid user, subsequently, it communicates 0 signal to both the user and sensor to indicate invalid requests received.

Throughput

Throughput can be defined in either way. The first way is based on the number of bits communicated in unit-time, and the second way is the number of packets transmitted in unit-time. During the implementation of the proposed LDA-GIoT, data for numerous static users was gathered, uni gateway, and numerous static sensors. The throughput is 162 bps, 233 bps, and 91bps at the user, gateway, and sensor. Thus, if computation cost is required for the proposed scheme, then the average transmission time as per the throughput will be 4.28 seconds, 5.16 seconds, and 8.69 seconds needed for the user, gateway, and sensor, respectively. the "MOSQUITTO" [Light et al. (2017)] broker at the gateway was installed for the implementation, and data from the gateway device was gathered. Now, if the number of packets transmitted per unit time is considered, then the throughput can be computed as $\frac{totalpacketreceived*packetsize}{totaltime}$. Thus, by computed using this formula, the average number of MQTT packet received at the user is 7, the sensor is 12 and gateway is 42 where packet size communicated from the user to gateway is 7 byte, gateway to user and sensor is 9 byte and sensor to the gateway is 5 byte through MQTT. Thus, the average throughput for the proposed scheme is 19.48 bps.

B Communication Cost

The communication cost defines the total number of bits transmitted on the public channel. During the implementation of the proposed LDA-GIoT, a python-based programming approach was used. Table 5.5 shows the total number of bits communicated in the cited schemes over the public channel. The computation of the communication cost is done as follows: to compute the communication cost, the

output size for each parameter was considered in the unit of "bits" using python. In our implementation, the size of the generated identity and password is 160 bits. A SHA-256 was used as a hash function; thus, the size of the hash output is 256 bits. The timestamp size is 32 bits, and the size of the generated random number is 128 bits. Table 5.5 summarizes communication cost comparison between the proposed scheme and other existing scheme.

Scheme	***User***	***Gateway***	***Sensor***	***Total Cost***
[Farash et al. (2016)]	632	792	2048	3472
[Wazid et al. (2018)]	736	1344	512	2592
[Zhou et al. (2019)]	832	2048	672	3552
[Shin & Kwon (2020)]	1158	1560	678	3552
[Jangirala et al. (2020)]	1012	1127	517	2656
LDA-GIoT	512	1344	704	2560

TABLE 5.5
Comparison of Communication Costs

C Computation Cost

The computation cost highlights the number of cryptographic operations used in the proposed scheme during the login and authentication stage. It also gives the total time required by those operations at each participant's devices. Let T_h, T_E/T_D, T_P and T_{fe} represent the computation cost of one-way hash function H(.), ECC encryption/decryption operation, ECC Point multiplication and fuzzy extractor respectively. the computation cost of bitwise XOR operation is not considered because it takes very little time (almost 0 ms) compare to other operations. During our implementation, it is observed that,

- For user device, T_E/T_D operation takes 0.07083 seconds, T_h operation takes 0.00041 seconds, the T_P operation takes 0.0607 seconds and T_{fe} operation

takes 0.0503 seconds.

- For sensing device, T_E/T_D operation takes 0.08883 seconds, T_h operation takes 0.00084 seconds and the T_P operation takes 0.0703 seconds.
- For gateway device, T_E/T_D operation takes 0.06783 seconds, T_h operation takes 0.00034 seconds and the T_P operation takes 0.0589 seconds.

Above all the costs are an average of 100 times verified outputs. Table 5.6 summarizes computation cost comparison between the proposed scheme and other existing scheme.

Scheme	*User*	*Gateway*	*Sensor*	*Time(ms)*
[Farash et al. (2016)]	$11*T_h$	$14*T_h$	$7*T_h$	10.4341
[Wazid et al. (2018)]	T_{fe} + $13*T_h$ + $2*T_e$	$5*T_h + 4*T_e$	$4*T_h + 2*T_e$	8.99
[Zhou et al. (2019)]	$4*T_P + 5*T_h$	$3*T_P + 7*T_h$	$4*T_P + 6*T_h$	10.693
[Shin & Kwon (2020)]	$3*T_P + 14*T_h$	$T_P + 12*T_h$	$2*T_P + 5*T_h$	8.66
[Jangirala et al. (2020)]	$5*T_P + 13*T_h$	$3*T_P$ + $23*T_h$	$2*T_P + 9*T_h$	22.5
LDA-GIoT	$6*T_h + 2*T_e$	$13*T_h$ + $6*T_e$	$6*T_h + 3*T_e$	7.92

Legends: T_h: One-way hash function cost, T_E/T_D: ECC Encryption/Decryption cost, T_P: ECC point multiplication cost

TABLE 5.6
Comparison of Computation Cost

Following Fig.5.3. gives performance comparison of communication cost and computation cost of the proposed scheme with the existing schemes.

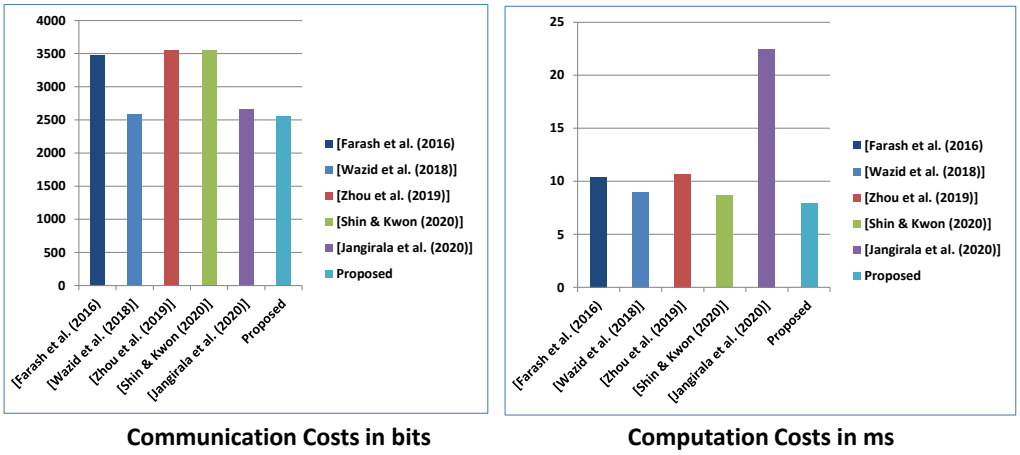

Figure 5.3. Performance Comparison Chart

5.9 Summary

In this chapter, a novel IoT authentication approach was proposed using an ECC as a Level Dependent Authentication for Generic IoT (LDA-GIoT). The LDA-GIoT reduces the number of user registrations and smooths the access control mechanism of the IoT system. The informal security analysis of the proposed scheme is given through the Dolev-Yao channel. The formal security analysis of the proposed scheme is given using a widely accepted AVISPA tool and random oracle based ROR Model. The comparative analysis of the LDA-GIoT with the other existing systems shows a little increase in computation and communication costs in the authentication phase. Still, it drastically decreases the efforts in multiple user registration and maintenance of the access control list. The implementation of a proposed LDA-GIoT is done through the MQTT protocol as an application layer protocol and raspberry-pi as a sensing device. Overall the proposed LDA-GIoT opens the new door for the researchers to study access control free, only

authentication dependent security systems. This chapter presents an authentication scheme using two factor authentication and does not use any biometric trait for authentication. Even, two factor authentication provides secure key generation environment, it has certain limitations in terms of security and it requires to design an authentication scheme that use biometric trait as a third factor. Next chapter 6 proposes three factor based LDA scheme to improve the security of the proposed scheme in this chapter.

Chapter 6

Level Dependent Authentication using Multi Factor

This chapter proposes a level dependent key exchange scheme between the user device, gateway device and the sensing device using three factor authentication. ***Section 6.1*** presents introduction for the user-gateway model and proposed work. ***Section 6.2*** provides literature review related to proposed scheme. Furthermore, ***Section 6.3*** highlights threat model considered for designing authentication scheme. ***Section 6.4*** presents proposed authentication scheme with all the phases. ***Section 6.5*** put forward security analysis for the proposed scheme. ***Section 6.6*** provides implementation aspects and use cases related to proposed work. ***Section 6.7*** presents performance comparison of the proposed work with other existing schemes based on communication cost and computation cost. ***Section 6.8*** sum-

[1] **Chintan Patel Nishant Doshi (2021) LDA-IoT : a level dependent authentication for IoT paradigm, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2021.1931573**

marise this chapter.

6.1 Introduction

The Internet of Things (IoT) based services are getting a widespread expansion in all the directions and dimensions of the 21st century. The IoT based deployment involves an Internet connected sensor, mobiles, laptops, and other networking and computing devices. In most IoT based applications, the sensor collects the data and communicates it to the end-user via gateway device or fog device over a precarious Internet channel. The attacker can use this open channel to capture the sensing device or the gateway device to collect the IoT data or control the IoT system. For a long time, numerous researchers are working towards designing the authentication mechanism for the sensor network to achieve reliable and computationally feasible security. For the resource constraint environment of the IoT, it is essential to design reliable, efficient, and secure authentication protocol. In this chapter, a novel approach of authentication in the IoT paradigm called a LDA is proposed. In the LDA protocol, reliable and resource efficient key sharing mechanism is proposed in which users at level l_i can communicate with the sensor at level l_j if and only if the level of user in the organizational hierarchy is lower or equal to the level of sensor deployment. Next subsection presents an overview on LDA approach.

A Level Dependent Authentication

An access control (AC) mechanism assures that only a valid user can access the sensing device based on their level in a hierarchy. Handling an access control for

the small number of devices is not a significant challenge and can be implemented using the Access Control List (ACL). In the IoT, the scalability and heterogeneity of devices are two significant challenges against smooth access control implementation. For performing AC, every intermediary device like gateway needs to maintain a list of valid users for each sensor. The physical attack is also one of the critical challenges, and it is infeasible to keep the static list for this highly dynamic environment. Rather than this approach, a novel and realtime implemented approach is proposed to tackle these challenges using a single user registration and lightweight authentication mechanism.

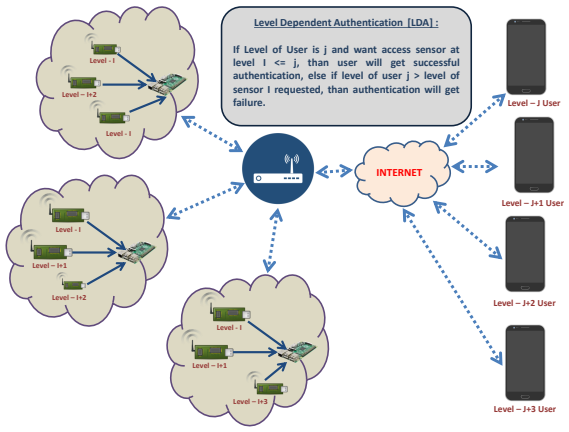

Figure 6.1. Level Dependent Authentication

As shown in Fig.6.1., every user in the IoT application will be assigned a particular level by the gateway device (or by registration authority in the multi-gateway scenario) during the user registration. Assigning this level will be based on the role of the user in the organization or industry. In parallel, during the sensor initialization phase, every sensor also gets a level based on its data generation mechanism. Thus, the overall idea of the LDA approach is articulated in the following algorithm: So the major advantages of using level dependent authenti-

Result: Access of Sensing Device to User
User-level = i;
Sensor-level = j;
while *Gateway received request from user* **do**
  if $j \leq i$ **then**
    Access-Allowed;
  else
    Access-Not-Allowed;
  end
end

Algorithm 2: Level Dependent Authentication

cation can be listed as follow:

- No need to maintain ACL in the system. This characteristic differentiates the proposed approach from other access control mechanisms.
- No chance for any unauthorized user to get access to any sensing device.
- Lighter implementation of intermediary devices.
- Very less registration by the user device (near to one only).
- Reduction of the computational complexity of the gateway device and sensing device for unauthorized requests.

The LDA approach's noticed challenge is that it increases the short time (in millisecond) for the gateway device for level verification. In most of the IoT implementations, gateway devices are powerful routing devices or servers which are resource capable of performing this operation. Thus, this is a negligible challenge. The proposed LDA approach does not. require to store any separate access control list or the list of user's roles. Various other Role-based Access Control or

Attribute-Based Access Control methods available are used only for access control verification; these methods do not provide any mechanism for authentication or the secure key setup. These methods consume space, memory, and list verification time, which increases unnecessary space and time complexity. At the same time, the proposed LDA scheme provides access control with authentication in a resource-constrained environment with highly optimized resource utilization.

6.2 Literature Review

In 2014, Turkanovic et al. [Turkanović et al. (2014)] proposed RUA scheme using a smart card for IoT notations and pointed out that their scheme provides secure key agreement and lower computation cost compare to existing schemes. In 2016, Farash et al. [Farash et al. (2016)] pointed out certain limitations in Turkanovic et al.'s scheme. They highlighted that Turkanovic et al.'s scheme is vulnerable against the session key and secret parameter disclosure attack, stolen smart card attack, Man-in-the-Middle (MITM) attack, and user traceability. In the same paper, Farash et al. [Farash et al. (2016)] proposed a new authentication scheme for the IoT environment and claimed that their scheme is highly secured and computationally feasible compare to analyzed schemes and other existing schemes. Unfortunately, in the same year, Amin et al. [Amin et al. (2018)] highlighted that the scheme proposed by Farash et al. is vulnerable against session-specific temporary attack, user impersonation, and offline password guessing attack. In the same paper, Amin et al. proposed a new smart card based three-factor authentication scheme using XOR and hash operations and claimed security and feasibility of the scheme. In 2017, Jiang et al. [Jiang et al. (2017)] pointed out that the scheme

proposed by Amin et al. is vulnerable from tracking attack and session-specific temporary credential attack. Jiang et al. [Jiang et al. (2017)] also proposed a secure, lightweight, and efficient authentication scheme based on bio-hacking and rubin cryptosystem for wearables.

The *smart Home (SH)* is one of the most promising IoT application which consists of heterogeneous intelligent home appliances like smart tv, smart meter, smart refrigerator, smart washing machine, smart door lock, smart window viper and so on. Due to the overloaded adoption of smart homes, it becomes essential for the researcher to focus on security and privacy issues also. In 2017, Wazid et al. [Wazid et al. (2017)] proposed a secure key establishment scheme using a hash function with symmetric encryption and provided the security analysis using random oracle based Real-Or-Random (ROR) model. In the same year, Kumar et al. [Kumar et al. (2015)] also proposed a new authentication scheme for the smart home environment using XOR, hash, and symmetric encryption. By security analysis using AVISPA and performance comparison based on computation cost and communication cost, Kumar et al. proved the feasibility and security of their scheme in resource constraint environment. By considering privacy as a significant challenge, Poh et al. [Poh et al. (2019)]proposed an authentication scheme in two phases. In the first phase, the author proposed a lightweight key establishment protocol, and in the second phase, the author used token-based searchable encryption-based queries to achieve privacy.

The *smart Grid (SG)* is a network of all renewable and non-renewable energy generators, distributors, and consumers. The smart grid's fundamental objective is to reduce non-renewable energy consumption in the world and forecast renewable energy to let consumers also become a generator. The smart meter

is an intelligent device that collects the information regarding energy consumption and needs from the user and passes that information at a regular interval to the energy control station. Information regarding energy usage must be securely communicated with the energy control station and other smart grid entities because leakage of this information can create a loophole in the user's privacy. In 2017, Mahmood et al. [Mahmood et al. (2018)] proposed an ECC based authentication scheme without making use of the third party like a registration server. In 2018, Gope et al. [Gope & Sikdar (2019)] introduced an authentication scheme between the smart meter and energy service provider using a Physically Unclonable Function (PUF). In 2018, Abbasinezhad et al. [Abbasinezhad-Mood & Nikooghadam (2018)] reviewed an authentication scheme proposed in [Mahmood et al. (2018)] and pointed out that their scheme is vulnerable against perfect forward secrecy, private key leakage, and known session-specific temporary information attack. Abbasinezhad et al. proposed a new authentication scheme using ECC and claimed reliable security and computational feasibility by comparing it with the previously proposed scheme. The smart grid will play a significant role in the successful deployment of Electric Vehicle (EV) in the market. Communication between EV and SG networks must ensure that an intruder can not trace the location of EV and can not play with the user's privacy. Roman et al. [Roman et al. (2019)], in 2019, proposed a paring based authentication scheme between EV and authentication servers of the smart grid.

Industrial Internet of Things (IIoT) is one of the prominent applications of IoT. IIoT makes use of sensing technology for trusted observation of industrial production, reliable data management, intelligent decision making, and efficient human resource utilization. An authentication between IIoT user and sensing de-

vices plays a vital role in managing the security of the complete hierarchy of IIoT. Li et al. [Li et al. (2017)] in 2018 proposed a privacy-preserving authentication scheme for the IIoT environment and proved security for the proposed scheme using random oracle based adversary capabilities. Karate et al. [Karati et al. (2018)] proposed a signature-based authentication scheme for a cloud-centric IIoT environment. In 2018, karati et al. [Karati et al. (2018)] proposed an efficient scheme between the key generator, cloud server, data owner, and data consumer and prove the security using widely accepted game theory-based analysis. An authors in [Wang & Wang (2018)] proposed a secure authentication protocol for industrial IoT. The limitation of these protocols is that they consider all sensors at a similar level, but in the realtime industrial deployment, it is not the real scenario.

Overall, the authentication in the IoT environment consists of many independent entities, complicated technologies, and resource constraint devices. Thus, it is essential to propose an authentication scheme that is reliable, efficient, and suitable for IoT's resource constraint environment.

6.3 Threat Model

An adversary model used in this chapter is considered from a similar model discussed in [Dolev & Yao (1981)]. The adversary is an intruder who tries to capture the communicated messages as well as tries to gain physical access to devices belongs to network topology. In the IoT based network model, it is possible to define an adversary with the robust capabilities for improvement in the designing of reliable protocols and also to perform better security validation for the proposed authentication protocol.

C_1. An Adversary $\mathcal{A}$ can compute valid pair of $identity * password$ offline in polynomial time using the available dictionary.

C_2. An Adversary $\mathcal{A}$ can capture a smart card and can retrieve the information from it.

C_3. An Adversary $\mathcal{A}$ have full access on the communication channel between user - gateway, sensor node - gateway, and user - sensor node.

C_4. An Adversary $\mathcal{A}$ may get previously computed session key.

C_5. An Adversary $\mathcal{A}$ may get the secret of the gateway node during failure situations.

C_6. An Adversary A can compromise sensor node physically and can get stored information.

C_7. An Adversary $\mathcal{A}$ can capture the level of user or sensing device.

C_8. An Adversary $\mathcal{A}$ can generate bot nodes and can send simultaneous ping messages to the user device and sensing device.

6.4 Proposed Scheme

In this section, the proposed authentication and key exchange scheme is set forth for the U-GW-Sensor based network model discussed in chapter 1 . The proposed scheme consists of the initialization phase in which the gateway device works as a master device and performs offline and secure initialization for the user device, sensing device, and gateway itself. The initialization phase is followed by the registration phase and the login and authentication phase.

A Initialization Phase

This phase call on offline by the gateway device with trusted platform module: Table 6.1 presents notations and symbols used in the articulation of the proposed scheme. We use the random number and timestamp to protect the proposed scheme from the replay attack. We assume that all the entities have a synchronized clock.

Symbols	Description
R_x	Random Number generated during initialize phase
r_x	Random Number generated during authentication phase
T_x	Time-stamp
l_i	User Level
l_j	Sensor Level
SID_j	Sensor identity
UID_i	User identity
$GWID$	Gateway identity
G_p	Elliptic curve generator
ΔT	Time-stamp threshold
K_s	Gateway node master secret
Bio_i	User biometric template
M_x	Message Number
TID	Temporary Identity
Gen(.)	Generation procedure in fuzzy extractor
Rep(.)	Reproduction procedure in fuzzy extractor
H(.)	One-way Hash function
Enc(.)/Dec(.)	ECC encryption/decryption
$\oplus$, $\|\|$	XOR and Concatenation respectively
B,Z	Computed Parameters

TABLE 6.1
Notation and Abbreviations

User Initialization

Let R_i be a secret random number generated by the gateway GW for the user device U_i as a password. The GW computes $UID_i = R_i$*P where P is a point on elliptic curve. The GW generates random master secret K_s and computes $X_1 = \mathrm{H}(UID_i||K_s)$, $Z_1 = \mathrm{H}(l_i||K_s||H(UID_i))$. Here l_i is the level for the user U_i. GW stores R_i, UID_i, X_1 and Z_1 in the hidden memory of U_i.

Sensing Device initialization

Let R_j be a secret random number generated by the gateway GW for a sensing device SD_j. The GW computes $SID_j = R_j$*P where P is a point on elliptic curve. The GW computes $X_2 = \mathrm{H}(SID_j||K_s)$, $Z_2 = \mathrm{H}(l_j||K_s||H(SID_j))$. Here l_j is the level for the sensor SD_j. GW stores R_j, SID_j, X_2 and Z_2 in hidden memory of SD_j.

Gateway Node Initialization

The gateway GW generates random number R_k as a secret parameter and compute $GWID = R_k$*P as a public identity.

B Registration Phase

User Registration

User registration is a phase which is carried out through secure channel.

1. $U_i \rightarrow GW$: User U_i sends X_1, UID_i to GW.
2. $GW \rightarrow U_i$: Gateway node verifies $X_1* = \mathrm{H}(UID_i||K_s) \stackrel{?}{=} X_1$. The GW device generates r_1 and computes $TID_i = \mathrm{H}(r_1||X_1)$, $M_1 = \mathrm{H}(TID_i||GWID||R_i)$,

$B_1 = X_1 \oplus r_1$ and sends smart sard (SC) = (TID_i, M_1, B_1).

3. User U_i generates biometric parameters using fuzzy extractor as BiO = Gen(B_i) = (δ_i, γ_i) in which δ_i is a secret key parameter which is used for verification and γ_i is a public parameter which is also an input during reproduction stage. U_i computes M_2 = H($TID_i||\delta_i||R_i$), user stores SC = (TID_i, M_1, B_1, M_2).

C Authentication and Key Exchange Phase

The authentication and key establishment between User U_i and SD_j involves authentication of U_i with GW and SD_j with GW. After an authentication, U_i and SD_j computes session key. An ECC based encryption/decryption was used during key establishment phase. Summary of authentication phase is given in Table 6.2.

1. $U_i \rightarrow GW$: User provides R_i,UID_i, X_1, Z_1 and $Bio*$ to Smart card reader (SCR) with SC. SCR computes δ_i* = Rep($Bio*$, γ_i) and verifies following:

$$TID_i* = \mathrm{H}(r_1||X_1) \stackrel{?}{=} TID_i,$$

$$r_1 = X_1 \oplus B_1,$$

$$M_1{}^* = \mathrm{H}(TID_i{}^*||GWID||R_i) \stackrel{?}{=} M_1,$$

$$M_2{}^* = \mathrm{H}(TID_i{}^*||\delta_i{}^*||R_i) \stackrel{?}{=} M_2,$$

If all conditions successfully verified, then SCR generates random number r_2, and computes following:

$$B_2 = TID_i{}^* \oplus UID_i,$$

$$M_4 = Enc_{GWID}(r_2, UID_i, SID_j),$$

$$M_5 = \mathrm{H}(TID_i^* || SID_j || GWID || T_1),$$

$$TID_i^{**} = \mathrm{H}(TID_i^* || r_2),$$

sends authentication *Message 1* = (TID_i^{**}, M_4, M_5, T_1, Z_1 ,B_2) to GW.

2. $GW \rightarrow SD_j$: Gateway Node GW computes current time T_2 and verifies $\Delta T \leq T_2$ - T_1, if yes then performs following:

$$(r_2, SID_j, UID_i) = Dec_{R_k}(M_4),$$

$$TID_i = \mathrm{H}(r_1 || H(UID_i || K_s))$$

$$TID_i^{'} = \mathrm{H}(TID_i || r_2) \stackrel{?}{=} TID_i^{**}, \text{ if yes}$$

$$M_6 = \mathrm{H}(\mathrm{H}(SID_j || K_s) || T_3)),$$

sends *message 2* = (M_6, T_3) to SD_j.

3. $SD_j \rightarrow GW$: SD_j takes current time T_4 and verifies $\Delta T \leq T_4$ - T_3, if verified then computes following:

$$M_6^* = \mathrm{H}(\mathrm{H}(X_2) || T_1) \stackrel{?}{=} M_6,$$

$$M_7 = \mathrm{H}(X_2 || GWNID_k || T_5),$$

sends *message 3* = (M_7, Z_2, T_5) to GW.

4. $GW \rightarrow SD_j$: Gateway Node GW takes current time T_6 and verifies $\Delta T \leq$ T_6 - T_5, if yes then performs following:

$$M_7^* = \mathrm{H}(\mathrm{H}(SNID_j||K_s)||GWNID_k||T_5) \stackrel{?}{=} M_7,$$

Gets l_i and l_j from Z_1 and Z_2 respectively by computing:

$$Z_1*=\mathrm{H}(l_i||K_s||H(UID_i)) \text{ till } Z_1* \stackrel{?}{=} Z_1 \text{ satisfies for valid } l_i,$$

$$Z_2*=\mathrm{H}(l_j||K_s||H(SID_j)) \text{ till } Z_2* \stackrel{?}{=} Z_2 \text{ satisfies for valid } l_j,$$

Verifies if $l_i \leq l_j$, then continue else transmits 0 signal to U_i and sensing device SD_j,

$$\text{Verifies } M_5^* = \mathrm{H}(TID_i^*||SID_j||GWID||T_1) \stackrel{?}{=} M_5,$$

$$M_8 = \mathrm{H}(UID_i||r_2||SID_j||GWID||T_7),$$

$$M_9 = Enc_{SID_j}(UID_i, r_2),$$

sends *message 4* = (M_8, M_9, T_7) to sensing device SD_j.

5. $SD_j \rightarrow GW$: SD_j takes current time T_8 and verifies $\Delta\mathrm{T} \leq T_8$ - T_7,if verified then computes following:

$$Dec_{R_j}(M_9) = (UID_i, r_2)),$$

$$M_8^* = \mathrm{H}(UID_i||r_2||SID_j||GWID||T_7) \stackrel{?}{=} M_8),$$

generates random number r_3,

$$M_{10} = Enc_{RIDU_i}(r_3),$$

$$M_{11} = \mathrm{H}(GWID||SID_j||r_3||T_8),$$

$$M_{12} = \mathrm{H}(GWID||SID_j||r_2||T_8),$$

send *message 5* = (M_{10},M_{11},M_{12},T_8) to GW.

6. $GW \rightarrow U_i$: Gateway node GW takes current time T_9 and verifies $\Delta\mathrm{T} \leq T_9$ - T_8, if yes then performs following:

$$M_{12} = \mathrm{H}(GWID_k||SID_j||r_2||T_8) \stackrel{?}{=} M_{12},$$

gets current time stamp, T_9,

$$M_k = \mathrm{H}(Z_1||Z_2||K_s||T_9)$$

sends *message 6* = (M_{10},M_{11},M_k,T_8,T_9) to U_i and *message 7* = (M_k, T_9) to SD_j

7. *User* U_i*:* takes current time T_{10} and verifies $\Delta T \leq T_{10}$ - T_9, if yes then performs following:

$$Dec_{R_i}(M_{10}) = (r_3),$$

$$M_{11}^{*} = \mathrm{H}(GWID_k||SID_j||r_3||T_8) \stackrel{?}{=} M_{11}, \text{ if yes.}$$

$$\mathbf{SK} = \mathrm{H}(r_2||r_3||UID_i||SID_j||GWID_k||T_8||M_k||T_9)$$

8. *Sensor* SD_j*:* takes current timestamp T_{11} and verifies $\Delta T \leq T_{11}$ - T_9, if yes then performs following:

$\mathbf{SK} = \mathrm{H}(r_2||r_3||UID_i||SID_j||GWID_k||T_8||M_k||T_9)$

6.5 Security Analysis

A Informal Security Analysis

In this subsection, an informal security analysis for the proposed LDA Protocol was discussed.Table 6.3 presents summary of comparison of the proposed scheme with the other existing schemes based on secure against attacks.

User/SCR	Gateway	Sensor
SCR computes : δ_i* = Rep($Bio*$, γ_i), TID_i* = H(r_1||X_1) $\stackrel{?}{=}$ TID_i, r_1 = $X_1 \oplus B_1$, $M_1{}^*$ = H($TID_i{}^*$||$GWID$||R_i) $\stackrel{?}{=}$ M_1, $M_2{}^*$ = H($TID_i{}^*$||$\delta_i{}^*$||R_i) $\stackrel{?}{=}$ M_2		
Generate r_2, B_2 = $TID_i{}^* \oplus UID_i$, M_4 = $Enc_{GWID}(r_2, UID_i, SID_j)$, M_5 = H($TID_i{}^*$||SID_j||$GWID$||T_1), $TID_i{}^{**}$ = H($TID_i{}^*$||r_2)	verifies $\Delta T \leq T_2$ - T_1, (r_2, SID_j, UID_i) = $Dec_{R_k}(M_4)$, TID_i = H(r_1||$H(UID_i$||$K_s)$), $TID_i{}'$ = H(TID_i||r_2) $\stackrel{?}{=}$ $TID_i{}^{**}$, if yes M_6 = H(H(SID_j||K_s)||T_3))	verifies $\Delta T \leq T_4$ - T_3, M_6^* = H(H(X_2)||T_1)$\stackrel{?}{=}$ M_6, M_7 = H(X_2||$GWNID_k$||T_5)
Message 1 = ($TID_i{}^{**}$, M_4, M_5, T_1, Z_1 ,B_2) to GW	sends *message 2* = (M_6, T_3) to SD_j	sends *message 3* = (M_7, Z_2, T_5) to GW.
	verifies $\Delta T \leq T_6$ - T_5, M_7^* = H(H($SNID_j$||K_s)||$GWNID_k$||T_5)$\stackrel{?}{=}$ M_7, Z_1*=H(l_i||K_s||$H(UID_i)$) till $Z_1* \stackrel{?}{=} Z_1$, Z_2*=H(l_j||K_s||$H(SID_j)$) till $Z_2* \stackrel{?}{=} Z_2$, M_5^* = H($TID_i{}^*$||SID_j||$GWID$||T_1) $\stackrel{?}{=}$ M_5, M_8 = H(UID_i||r_2||SID_j||$GWID$||T_7), M_9 = $Enc_{SID_j}(UID_i, r_2)$, sends *message 4* = (M_8, M_9, T_7)	verifies $\Delta T \leq T_8$ - T_7, $Dec_{R_j}(M_9)$ = (UID_i, r_2)), M_8^* = H(UID_i||r_2||SID_j||$GWID$||T_7)$\stackrel{?}{=}$ M_8), generates r_3, M_{10} = $Enc_{RIDU_i}(r_3)$, M_{11} = H($GWID$||SID_j||r_3||T_8), M_{12} = H($GWID$||SID_j||r_2||T_8), send *message 5* = (M_{10},M_{11},M_{12},T_8) to GW
verifies $\Delta T \leq T_{10}$ - T_9, $Dec_{R_i}(M_{10})$ = (r_3), M_{11}^* = H($GWID_k$||SID_j||r_3||T_8) $\stackrel{?}{=}$ M_{11}, if yes. **SK** = H(r_2||r_3||UID_i||SID_j||$GWID_k$ ||T_8||M_k||T_9)	verifies $\Delta T \leq T_9$ - T_8, M_{12} = H($GWID_k$||SID_j||r_2||T_8) $\stackrel{?}{=}$ M_{12}, M_k = H(Z_1||Z_2||K_s||T_9), sends *message 6* = (M_{10}, M_{11}, M_k, T_8, T_9) to U_i and *message 7* = (M_k, T_9) to SD_j	verifies $\Delta T \leq T_{11}$ - T_9, if yes, **SK** = H(r_2 || r_3|| UID_i|| SID_j|| $GWID_k$|| T_8|| M_k|| T_9)

TABLE 6.2
Authentication and Key Exchange

SECURED AGAINST TRACEABILITY

In most of the IoT application, it is expected that an original identity of the user or device must not relieve. If an adversary $\mathcal{A}$ can capture the identity of the user then it creates a problem for user's privacy preservation. The adversary $\mathcal{A}$ can easily trace the user and it's functionalities. In the proposed LDA scheme, the first public authentication request *message 1* by user U_i to GW is $\langle TID_i^{**}, M_4, M_5, T_1, Z_1, B_2 \rangle$. The *message 1* is distinct for each new request due to following reason. The author computes $M_4 = Enc_{GWID}(r_2, TID_i^*, SID_j)$, $M_5 = \mathrm{H}(TID_i^* || SID_j || GWID || T_1)$, in which r_2 is a random number and T_1 is a latest timestamp, Thus, the inclusion of r_2 and T_1 ensures that M_4 and M_5 are different for each session. Similarly other communicated messages (*message 2, message 3, message 4, message 5, message 6, message 7*) by user U_i and sensing device SD_j are also distinct for each session.

ANONYMITY

An anonymity preservation can be achieved by securing an identity of the user U_i and the sensor SD_j from adversary A. Identity of user U_i is UID_i which is secured by one-way hash function in computations of $X_1 = \mathrm{H}(UID_i || K_s)$ and $Z_1 = \mathrm{H}(l_i || K_s || H(UID_i))$. During authentication messages, user uses temporary id $TID_i = \mathrm{H}(r_1 || X_1)$ which is protected using one-way hash function and ECC based encryption. Identity of the sensing device (SID_j) is also protected by computed parameters $X_2 = \mathrm{H}(SID_j || K_s)$, $Z_2 = \mathrm{H}(l_j || K_s || H(SID_j))$ which are protected by one-way hash function. Gateway node communicates following messages with the user and the sensor device, $\langle M_6, T_3 \rangle$, $\langle M_8, M_9, T_7 \rangle$, and $\langle M_{10}, M_{11}, T_8, T_9 \rangle$. Each message communicated by the gateway device uses timestamp and is protected

by one-way hash function and encryption. Therefore, anonymity of the sensing device and the user device is achieved in the proposed LDA scheme.

MUTUAL AUTHENTICATION AND INTEGRITY

The mutual authentication is an important property that assures security from unauthorized access to any unknown adversary $\mathcal{A}$. The proposed LDA scheme achieves mutual authentication for the participating entities. The gateway device GW authenticates user U_i by performing the following operations. The GW computes $TID_i^{'} = \mathrm{H}(TID_i||r_2) \stackrel{?}{=} TID_i^{**}$ to ensure the temporary id, which is received from the user U_i. Computation of $TID_i = \mathrm{H}(r_1||H(UID_i||K_s))$ includes secret parameter K_s which an adversary can never get. The gateway GW authenticates sensing device by computing $M_7^* = \mathrm{H}(\mathrm{H}(SNID_j||K_s)||GWID||T_5) \stackrel{?}{=} M_7$, which includes master secret K_s and gateway node identity $GWID$. Similarly, the user authenticates gateway and sensing device during verification of the parameter M_{11}, and the sensing device validates user and gateway during verification of the parameter M_8. In the proposed scheme, all the communicated messages are verified by a one-way hash function, which ensures integrity for the proposed scheme.

PROTECTED AGAINST REPLAY ATTACK

In the replay attacks, an adversary $\mathcal{A}$ replays old messages which are captured by $\mathcal{A}$ by tracking of communications between U_i, GW and SD_j. Let us assume, an adversary $\mathcal{A}$ captures *message 1* = $\langle TID_i^{**}, M_4, M_5, T_1, Z_1, B_2 \rangle$ and communicate *message 1* to start a new session with the GW device. After receiving a message from $\mathcal{A}$, gateway GW performs validation of time interval between current time and message generation time by performing $\Delta \mathrm{T} \leq T_2$ - T_1. Here T_2 is current

time or receiving time at gateway GW. This validation will fail because interval will not be lesser then the previously decided ΔT. Similarly, every communication between user U_i, gateway GW and sensor SD_j is protected by threshold ΔT. Thus, the proposed LDA scheme is secured against a replay attack.

SECURED AGAINST MAN-IN-THE-MIDDLE ATTACK (MITM)

In this attack, an adversary $\mathcal{A}$ captures the messages communicated by valid participants of the system and modifies messages. it is assumed that $\mathcal{A}$ captures the *message 1* = $\langle TID_i^{**}, M_4, M_5, T_1, Z_1, B_2 \rangle$ communicated by user U_i to GW. Let $\mathcal{A}$ updates the *message 1* by *message 1'* = $\langle TID_i^{**'}, M_4', M_5', T_1', Z_1', B_2' \rangle$ and send to GW. Now the gateway verifies $TID_i^{**'}$ with TID_i by capturing r_2 from the decryption of M_4. Thus, the modified $TID_i^{**'}$ does not match with the TID_i. Therefore, the gateway devices stops further communication. Similarly, all the communications in the proposed protocol is secured using encryptions and one-way hash functions. Hence, the proposed scheme is immune enough against the MITM attack.

KEY ESTABLISHMENT WITH PERFECT FORWARD SECRECY

The protection against perfect forward secrecy assures that even through an adversary $\mathcal{A}$ captures all messages of the previous session and final session key, then also $\mathcal{A}$ must not get the ability to compute key for the ongoing session. the session key in the proposed scheme is computed as follows. The session key SK = H($r_2||r_3||UID_i||SID_j||GWID_k||T_8$), that includes two random numbers and one timestamp. It is nearly impossible for an adversary $\mathcal{A}$ to generate the random numbers, which will match with the same r_2 and r_3. Now let us assume that an

adversary gets the long term secret K_s from the gateway. Then also, the adversary could not break the previous session messages due to hash protected random numbers. Therefore, the proposed LDA scheme achieves perfect forward secrecy, and it is nearly impossible for an adversary to link the current session with the previous session.

RESISTANCE AGAINST GATEWAY NODE BYPASS ATTACK

In the proposed authentication scheme, it is impossible for any valid user U_i and sensing device SD_j to bypass the gateway device GW because the proposed scheme does not provide any direct communication between the user device and sensing device. In our MQTT implementation, U_i and SD_j can publish data with the gateway GW device's topic only and can not publish on each other's topic. In the proposed LDA scheme, when gateway node receives first authentication request from the user U_i, it computes $M_6 = \mathrm{H}(\mathrm{H}(SID_j||K_s)||T_3))$ which is not possible for the user U_i to compute because U_i is not aware of K_s. Similarly when gateway node receives message from user, it computes $M_k = \mathrm{H}(Z_1||Z_2||K_s)$ before key generation which is not possible for SD_j to compute because SD_j is also not aware of K_s and Z_1. As a result, due to master key K_s and time stamp validation and random numbers, it is impossible for U_i, SD_j or $\mathcal{A}$ to bypass the gateway GW.

RESISTANCE AGAINST GATEWAY NODE IMPERSONATION ATTACK

Let us assume that an adversary $\mathcal{A}$ intercepts *message 1, message 2, message 3, message 4, message 5, message 6, message 7* and try to create other valid messages *message 1', message 2', message 3', message 4', message 5', message 6', message 7'* on behalf of the gateway GW where *message 1* $= \langle TID_i{}^{**}, M_4, M_5, T_1, Z_1, B_2 \rangle$,

message 2 = $\langle M_6, T_3 \rangle$, *message 3* = $\langle M_7, Z_2, T_5 \rangle$, *message 4* = $\langle M_8, M_9, T_7 \rangle$,*message 5* = $\langle M_{10}, M_{11}, M_{12}, T_8 \rangle$,*message 6* = $\langle M_{10}, M_{11}, M_k, T_8, T_9 \rangle$ and *message 7* = $\langle M_k \rangle$. Now let us assume that an adversary $\mathcal{A}$ generates random number $\{r_2', r_3'\}$ and time-stamps $\{T_1', T_2', T_3', T_4', T_5', T_6', T_7'\}$. Still to compute the message M_6 and M_k, adversary requires K_S. To calculate M_8, M_{11} and M_{12}, the adversary $\mathcal{A}$ needs $\{UID_i, SID_j\}$. These are secured identities. Thus, an adversary $\mathcal{A}$ can not compute valid messages *message 1', message 2', message 3', message 4', message 5', message 6', message 7'* on behalf of the gateway *GW*. Therefore, the proposed LDA scheme is protected against the gateway node impersonation attack.

Scheme	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
[Challa et al. (2017)]	✓	✗	✓	✓	✗	✓	✓	✓	✗	✓	✗
[Shin & Kwon (2020)]	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗
[Zhou et al. (2019)]	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗
[Wu et al. (2020)]	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
[Shuai et al. (2020)]	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗	✗
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: S_1: Tracebility, S_2: Anonymity, S_3: Mutual authentication and Integrity, S_4: Replay attack , S_5: Man-in-The-Middle Attack, S_6: Forward secrecy, S_7: Gateway by pass attack, S_8: Gateway impersonation attack , S_9: Sensing device capture attack, S_{10}: Privilege insider attack, S_{11}: Level Dependent Authentication, ✓: the protocol supports this feature, ✗: the protocol doesn't support this feature.

TABLE 6.3
Security Comparison

SECURED AGAINST SENSING DEVICE CAPTURE ATTACK

Now, let us assume that an adversary $\mathcal{A}$ physically captures sensing device SD_j and uses a power analysis attack [Messerges et al. (1999); Kocher et al. (1999)] retrieves a piece of information stored in sensing device SD_j. Thus, let us assume that $\mathcal{A}$ successfully retrieves $\langle R_j, SID_j, Z_2, X_2 \rangle$ from sensor memory. For com-

puting SID_j, the gateway node GW uses a Random Number Generator (RNG). For the computation of Z_2 and X_2, the gateway GW uses master secret K_s. Thus, RNG helps gateway GW in unique identity generation for each sensor. Thus, the compromise of session key between U_i and SD_j does not compromise the other session keys. Now let us assume that $\mathcal{A}$ tries to compute session key by using extracted data. Still, $\mathcal{A}$ could not get success due to non-availability of the master key K_S and other parameters like r_2, Z_1 and X_1. Thus, the proposed LDA scheme achieves safety against the sensing device capture attack.

SECURED AGAINST SMART-CARD AND PASSWORD CAPTURE ATTACK

Now, let us assume that an adversary $\mathcal{A}$ gets the user's smart card and try to generate the password from the password list. Thus, $\mathcal{A}$ has $\{TID_i, M_1, B_1, M_2\}$ as a smart card parameters and R_i' as a guessed password. If $\mathcal{A}$ tries to log in, then he must provide the $Bio*$ with the SC and other parameters. It is nearly impossible for $\mathcal{A}$ to come up with the same biometric parameter $Bio*$ based on which SCR performs further computation. Thus, adversary $\mathcal{A}$ could not satisfy the condition $M_2{}^* = \mathrm{H}(TID_i{}^* || \delta_i{}^* || R_i) \stackrel{?}{=} M_2$. Hence, even though an adversary $\mathcal{A}$ captures the smart card and generates a valid password, he/she could not access the proposed system.

SECURED AGAINST PRIVILEGE INSIDER ATTACK

Let us assume that $\mathcal{A}$ is a malicious insider of the gateway node GW. He/she is aware of $\{UID_i, SID_j, GWID\}$ and the gateway master secret K_s. Let us assume that $\mathcal{A}$ also tracks messages received by the gateway node GW, and thus,

he/she receives Z_1 and Z_2 from the valid user U_i and the sensor device SD_j respectively. Let us assume that $\mathcal{A}$ also gets ECC parameters, biometric generator function, and stored data in the user smart card. In the proposed LDA scheme, the session key computation occurs as follows. The session key $SK = \mathrm{H}(r_2||r_3||UID_i||SID_j||GWID_k||T_8||M_k)$. After tracking all the above-said information on the gateway node, malicious insider $\mathcal{A}$ obtains T_8 from the *message 5*. Still, $\mathcal{A}$ needs random number r_3 to compute the session key, which is generated by the sensing device SD_j for the valid U_i. The r_3 is encrypted with parameter M_{10} and is not possible to decrypt for $\mathcal{A}$ or even valid gateway GW to get it. Thus, it is nonproductive efforts for any malicious insider $\mathcal{A}$ to compute the final session key or next session key.

B Mutual Authentication using BAN Logic

In this section, mutual authentication, freshness, and key-establishment capacity of the proposed scheme is presented using the BAN Logic. The BAN Logic is a tool that operates based on the proportional logic and is a widely accepted tool to prove the mutual authentication property of the authentication scheme [Burrows et al. (1989)].

Mutual Authentication using BAN Logic

This proof mainly consists of the following steps: (I) Initial assumption, (II) Goal declaration, (III) Message formation, and (IV) Formal verification.

1. ***Initial assumptions:*** This proof mainly consists of the following steps: (I) Initial assumption, (II) Goal declaration, (III) Message formation, and (IV)

Formal verification.

J_1. $U_i \mid\equiv \#(T_i)$, $SD_j \mid\equiv \#(T_i)$,$GW \mid\equiv \#(T_i)$;

J_2. $U_i \mid\equiv \#(r_1)$, $GW \mid\equiv \#(r_2)$, $SD_j \mid\equiv \#(r_2)$, $U_i \mid\equiv \#(r_3)$;

J_3. $GW \mid\equiv U_i \Rightarrow X$, $GW \mid\equiv SD_j \Rightarrow X$, $SD_j \mid\equiv GW \Rightarrow X$.

J_4. $U_i \mid\equiv (\overset{GWID}{\longmapsto} GW)$, $SD_j \mid\equiv (\overset{RIDU_i}{\longmapsto} U_i)$, $GW \mid\equiv (\overset{SID_j}{\longmapsto} SD_j)$

J_5. $U_i \mid\equiv SD_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} SD_j)$

2. ***Goal Declaration:*** The expected goals in the proposed LDA scheme includes trust in shared key and freshness of communicated messages. In LDA scheme, expected goals are as follow:

G_1. $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

G_2. $SD_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

G_3. $U_i \mid\equiv SD_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

G_4. $SD_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$

G_5. $U_i \mid\equiv GW \mid\sim \#(X)$

G_6. $GW \mid\equiv SD_j \mid\sim \#(X)$

G_7. $SD_j \mid\equiv GW \mid\sim \#(X)$

3. ***Message Formation:*** The login and Authentication phase of the proposed LDA scheme includes exchanges of the following messages which can be written in the generic form as follow:

Message 1: $GW \rightarrow SD_j$: (H($UID_i||r_2||SID_j||GWID||T_7$), Enc($UID_i$, r_2), T_7)

Message 2: $SD_j \rightarrow GW$: (Enc(r_3), H($GWID||SID_j||r_3||T_8$), H($GWID||SID_j||r_2||T_8$), T_8)

Message 3: $GW \rightarrow U_i$: (Enc(r_3), H($GWID||SID_j||r_3||T_8$), H($Z_1||Z_2||K_s||T_9$), T_8, T_9)

Message 4: $GW \rightarrow SD_j$: (H($Z_1||Z_2||K_s||T_9$), T_9)

Idealized form: The ideal forms for the above messages can be written as follows:

Message 1: $GW \rightarrow SD_j$: $SD_j \triangleleft \langle < ((UID_i||r_2||SID_j||GWID||T_7), (UID_i, r_2), T_7) > \rangle_{GW|\equiv(\overset{SID_j}{\longmapsto} SD_j)}$

Message 2: $SD_j \rightarrow GW$: $GW \triangleleft \langle < ((r_3), (GWID||SID_j||r_3||T_8), (GWID||SID_j||r_2||T_8), T_8) > \rangle_{SD_j|\equiv(\overset{RIDU_i}{\longmapsto} U_i)}$

Message 3: $GW \rightarrow U_i$: $U_i \triangleleft \langle < ((r_3), (GWID||SID_j||r_3||T_8), (Z_1||Z_2||K_s||T_9), T_8, T_9) > \rangle_{GW|\equiv(\overset{RIDU_i}{\longmapsto} U_i)}$

Message 4: $GW \rightarrow SD_j$: $SD_j \triangleleft \langle < ((Z_1||Z_2||K_s||T_9), T_9) > \rangle$

4. ***Formal Verification***

Theorem 5. *The proposed scheme achieves the secure mutual authentication between the user U_i and the sensing device SD_j, and it achieves expected goals.*

Proof. Expected goals [G_1 - G_7] are achieved as follow:

S_1**:** from the message 1,

$SD_j \triangleleft \langle < ((UID_i||r_2||SID_j||GWID||T_7), (UID_i, r_2), T_7) > \rangle_{GW|\equiv(\overset{SID_j}{\longmapsto} SD_j)}$

S_2**:** Using S_1, R_1 and J_1,

$SD_j \mid\equiv GW \mid\sim \langle < ((UID_i||r_2||SID_j||GWID||T_7), (UID_i, r_2), T_7) > \rangle$

S_3**:** Using S_2, J_2, R_2,

$SD_j \mid\equiv GW \mid\equiv \langle < ((UID_i||r_2||SID_j||GWID||T_7), (UID_i, r_2), T_7) > \rangle$

S_4**:** from the message 2,

$GW \triangleleft \langle < ((r_3), (GWID||SID_j||r_3||T_8), (GWID||SID_j||r_2||T_8), T_8) > \rangle_{SD_j|\equiv(\overset{RIDU_i}{\longmapsto} U_i)}$

S_5**:** Using S_4, R_1 and J_1,

$GW \mid\equiv SD_j \mid\sim \langle < ((r_3), (GWID||SID_j||r_3||T_8), (GWID||SID_j||r_2||T_8), T_8) > \rangle$

S_6**:** Using S_4, J_2, R_2,

$GW \mid\equiv SD_j \mid\equiv \langle < ((r_3), (GWID||SID_j||r_3||T_8), (GWID||SID_j||r_2||T_8), T_8) > \rangle$

S_7**:** Using J_3,J_4 R_7,R_3,

$SD_j \mid\equiv (GWID||SID_j||r_3||T_8)$, $SD_j \mid\equiv (GWID||SID_j||r_2||T_8)$,

S_8**:** from the message 3,

$U_i \triangleleft \langle < ((r_3), (GWID||SID_j||r_3||T_8), (Z_1||Z_2||K_s||T_9), T_8, T_9) > \rangle_{GW|\equiv(\overset{RIDU_i}{\longmapsto} U_i)}$

S_9**:** Using S_8, R_1 and J_3,J_4,

$U_i \mid\equiv GW \mid\sim \langle < ((r_3), (GWID||SID_j||r_3||T_8), (Z_1||Z_2||K_s||T_9), T_8, T_9) > \rangle$

S_{10}**:** Using S_9, J_3, R_2,

$U_i \mid\equiv GW \mid\equiv \langle < ((r_3), (GWID||SID_j||r_3||T_8), (Z_1||Z_2||K_s||T_9), T_8, T_9) > \rangle$

S_{11}: Using J_4, R_7,R_3,

$U_i \mid\equiv (GWID||SID_j||r_3||T_8)$, $U_i \mid\equiv (Z_1||Z_2||K_s||T_9)$,

S_{12}: from the message 4,

$SD_j \triangleleft \langle < ((Z_1||Z_2||K_s||T_9), T_9) > \rangle$

S_{13}: Using S_1, R_1 and J_1,

$SD_j \mid\equiv GW \mid\sim \langle < ((Z_1||Z_2||K_s||T_9), T_9) > \rangle$

S_{14}: Using S_4, J_2, R_2,

$SD_j \mid\equiv GW \mid\sim \langle < ((Z_1||Z_2||K_s||T_9), T_9) > \rangle$

S_{15}: Using J_3,J_4 R_7,R_3,

$SD_j \mid\equiv (Z_1||Z_2||K_s||T_9)$,

S_{16}: Using S_3,S_4, S14, S15, R_3, R_8,J_5,

$SD_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$ $[G_2]$

S_{17}: Using S_9,S_{10}, S_{11}, R_3, R9, J_5,

$U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$ $[G_1]$

S_{18}: Using S_3,S_4, S_{14}, S_{15}, J_5, R_3, R_7 and R_8,

$SD_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$ $[G_3]$

S_{19}: Using S_9,S_{10}, S_{11}, J_5, R_3, R_7 and R9,

$U_j \mid\equiv SD_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SD_j$ $[G_4]$

S_{20}: Using S_1, R_2, R_5, R_7, J_1 and J_2,

$SD_j \mid\equiv GW \mid\sim \#(Message1)$

S_{21}: Using S_4, S_{12}, R_2, R_5, R_7, J_1 and J_2,

$SD_j \mid\equiv GW \mid\sim \#(Message1, Message3)$ $[G_7]$

S_{22}**:** Using S_4, R_2, R_5, R_7, J_1 and J_2,

$GW \mid\equiv SD_j \mid\sim \#(Message2)$ $[G_6]$

S_{23}**:** Using S_8, R_2, R_5, R_7, J_1 and J_2,

$U_i \mid\equiv GW \mid\sim \#(Message4)$ $[G_5]$

Thus, the above verification clearly shows that the proposed authentication scheme achieves all defined goals (G_1 - G_7). □

C Formal Security Model using ROR

To perform the formal security analysis of the proposed LDA scheme, a widely accepted random oracle based Real-Or-Random (ROR) model is used. The ROR model [Abdalla & Pointcheval (2005)] is one of the key formal and conventional security models used by security researchers to prove that an adversary $\mathcal{A}$ with the random oracle queries can not achieve self-confidence for the retrieved key. The supreme target of the ROR model is that an adversary $\mathcal{A}$ must not be able to distinguish the retrieved random value and the real session key.

Random Oracle: The proposed LDA protocol LDA_P uses cryptographic public *Hash* function which is formalized as a random oracle call $\mathcal{H}(m)$. Thus, whenever a probabilistic polynomial adversary $\mathcal{A}$ communicates with the message m_i then the oracle $\mathcal{H}(m_i)$ computes fix size irreversible random value r_i for the message. The oracle maintains a list L initialized with the NULL value in which it stores pair of (m_i,r_i) for each i where i = 0 to n and the returns value r_i to an adversary $\mathcal{A}$.

Participants: In the proposed LDA scheme, there are three participants, user U_i, gateway GW, and sensing device SD_j. The ROR modeling for these participants and adversary $\mathcal{A}$ is as follow:

Oracles:$\rightarrow$ $\Omega_{U_i}^m$, Ω_{GW}^n, and $\Omega_{SD_j}^o$ are oracles with the instances m, n and o for the U_i, GW and SD_j respectively. These are also called as a participants for the protocol LDA_P.

Oracles Freshness:$\rightarrow$ $\Omega_{U_i}^m$, Ω_{GW}^n and $\Omega_{SD_j}^o$ consider as a fresh oracles if the *reveal* oracle query $\mathcal{R}$ does not provide correct session key SK to $\mathcal{A}$.

Oracles Partnering:$\rightarrow$ Oracle instances Ω^x and Ω^y are partners if following conditions are fulfilled simultaneously:

- Both instances Ω^x and Ω^y are in the acceptance state.
- Both Ω^x and Ω^y share the common session-id sid and achieve the mutual authentication.
- Both Ω^x and Ω^y satisfy the partner identification and vice-versa.
- No other instance other than Ω^x and Ω^y accept with partner identification equal to Ω^x and Ω^y.

sid is a session identification. It represents the transcript of all the communicated messages between the participants before acceptance state.

Oracle Accepted state:$\rightarrow$ An instance Ω^x reaches to accept state after communicating the last message with the partner instance Ω^y. The concatenation of all delivered messages by both participants before reaching the acceptance state makes sid for a particular session.

Adversary:→ Let us assume that an adversary $\mathcal{A}$ controls complete communication channel defined over Dolev-Yao model. An adversary $\mathcal{A}$ can read, modify, inject, or fabricate the messages over this channel. An adversary $\mathcal{A}$ accesses random oracle queries that give the capability to $\mathcal{A}$ for capturing and modifying the communication.

Freshness:→ The instances $\Omega^m_{U_i}$ or $\Omega^o_{SD_j}$ are fresh instances if SK computed between $\Omega^m_{U_i}$ and $\Omega^o_{SD_j}$ is not revealed to an adversary $\mathcal{A}$ by using *reveal* query on any instance Ω^x.

Adversary Model:→ The modeling of an adversary $\mathcal{A}$ is done using the famous Dolev-Yao model. It defines send (*snd*) and receive (*rcv*) channels. Adversary $\mathcal{A}$ can perform the passive attack as well as an active attack. The adversary $\mathcal{A}$ can receive, read, update, and delete the communicated messages as well as add the new messages on the communication channel. The following random oracle queries give the capabilities to an adversary $\mathcal{A}$:

$\mathcal{R}(\Omega^x)$:→ The **reveal** query $\mathcal{R}$ provides session key *SK* to the adversary $\mathcal{A}$ generated by Ω^x and it's partner for the same *sid*.

$\mathcal{S}(\Omega^x, msg)$:→ The adversary $\mathcal{A}$ formalizes *send* query as an active attack. Using the *send* query, an adversary $\mathcal{A}$ communicates with an instance Ω^x. An adversary $\mathcal{A}$ sends message to Ω^x as well as gets response from the participant instance Ω^x. Thus, using this query, adversary $\mathcal{A}$ communicates with the user participant with instance Ω^x.

$\mathcal{E}(\Omega^x, \Omega^y)$:→ The *execute* query $\mathcal{E}$ is formalized as a passive attack through which an adversary $\mathcal{A}$ gets the capacity to monitor the communication between instances Ω^x and Ω^y.Thus, an execute query grants the read permission to an adversary $\mathcal{A}$.

CorruptUserDevice$(\Omega_{U_i}^m)$**:**$\rightarrow$ The *CorruptUserDevice* query enables an adversary $\mathcal{A}$ to capture the stored data inside the user U_i 's device.

CorruptUserSc$(\Omega_{U_i}^m)$**:**$\rightarrow$ The ***CorruptUserSc*** query enables an adversary $\mathcal{A}$ to capture the data from the user U_i's smart card (SC).

CorruptSensingDevice$(\Omega_{SD_j}^o)$**:**$\rightarrow$ The ***CorruptSensingDevice*** query enables an adversary $\mathcal{A}$ to capture the stored data inside the sensing device SD_j. This query also enables an adversary $\mathcal{A}$ to receive the secret credential (R_j) of the sensing device SD_j . Above three queries (*CorruptUserDevice*, *CorruptUserSc*, *CorruptSensingDevice* achieve the weak-corruption model in which the participant's internal data and temporary key does not corrupt.

CorruptLevelSensingDevice$(\Omega_{SD_j}^o)$**:**$\rightarrow$ The ***CorruptLevelSensingDevice*** enables an adversary $\mathcal{A}$ to get the deployment level (l_j) for the sensing device(SD_j) with an objective of the performing an active attack.

$\mathcal{T}(\Omega^x) :\rightarrow$ Before starting of this oracle game, an unbiased coin b get tossed. The output of this toss decides the return value for the *test* query $\mathcal{T}$. If the recently generated session key between the user U_i and the sensor device SD_j is SK, and an adversary $\mathcal{A}$ performs the test query on an instance Ω^m which is the instance of U_i or its partner instance Ω^o which is an instance of SD_j then if the toss output is b = 1, participant instance Ω^m returns an original session key. In contrast, if b = 0, then Ω^m returns a random value of the session key SK's size to an adversary $\mathcal{A}$. If none of the condition matches, then an instance Ω^m returns NULL. According to an adversarial model discussed earlier in the chapter, the gateway device GW is a trusted device; thus, an adversary $\mathcal{A}$ can not apply these oracles on the gateway device. An adversary $\mathcal{A}$ has limited access to *CorruptUserDevice*, *CorruptUserSc*, *CorruptSensingDevice*, *CorruptLevelSensingDevice* queries while $\mathcal{A}$

can perform test query ($\mathcal{T}(\Omega^x)$) as many time as it wish.

Semantic Security of session key in ROR Model: The semantic security of the session key SK depends on an adversary $\mathcal{A}'s$ capability of indistinguishability between the random number and the actual session key. The output of a *test* query $\mathcal{T}$ depends on the value of b' guessed by $\mathcal{A}$. If the value of b' is similar to the value of b, which is a hidden bit set by an oracle instance Ω^m and used by $\mathcal{T}(\Omega^m)$ to retrieve the original session key. Thus, the overall game depends on the correct guess by $\mathcal{A}$ for bit b. If an adversary guesses the correct value of b, then it gets the correct session key.

Let $\mathcal{SUCC}$ define success position for an adversary, then the advantage of an adversary $\mathcal{A}$ in capturing the correct session key SK for the proposed protocol LDA_P is Adv_{LDA-P}. The Adv_{LDA-P} represents the success of an adversary $\mathcal{A}$, and if the Adv_{LDA-P} is negligible, then it is said that the proposed scheme achieves security under the ROR model. Thus, Adv_{LDA-P} as follow:

$$Adv_{LDA-P}(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}] - 1 \tag{6.1}$$

Same equation can be written as follow:

$$Adv_{LDA-P}(\mathcal{A}) = 2 * Pr[b' = b] - 1 \tag{6.2}$$

Where Pr[$\mathcal{SUCC}$] represents the probability for the success of an adversary $\mathcal{A}$, if it is proved that the Adv_{LDA-P} is negligible under the proposed protocol LDA_P, then the proposed protocol is secure.

Semantic Security for the Password based protocol The semantic security

of the password base protocol $LDA - P_{pw}$ defines an adversary $\mathcal{A}$'s capability of guessing the correct password. A password based protocol $LDA - P_{pw}$ is semantically secure if the advantage function $Adv_{LDA-P_{pw}}$ is negligible under the following condition:

$$Adv_{LDA-P_{pw},|\mathcal{D}|}(\mathcal{A}) \geq max(q_s, (\frac{1}{|\mathcal{DS}|}, \rho_{fp})) \tag{6.3}$$

In equation 6.3, q_s represents number of send queries ($\mathcal{S}$), $|\mathcal{DS}|$ shows the finite size of the password dictionary, ρ_{fp} shows probability of the false positive occurrence by an adversary $\mathcal{A}$.

Formal Security Proof

The Security proof for the proposed protocol $LDA - P$ is represented in the following theorem:

Theorem 6. *Let $\mathcal{A}$ is a polynomial-time attacker running against the proposed protocol $LDA - P$ within a limited time t_A. Let an adversary $\mathcal{A}$ tries to attack the proposed protocol $LDA - P$ then it needs to perform the oracle queries. Let q_h defines the number of hash ($\mathcal{H}$) queries, q_s defines the number of send ($\mathcal{S}$) queries, q_e establishes the number of execute ($\mathcal{E}$) query, the number of bits for the biometric key is defined as ρ_i, the uniformly distributed password dictionary is defined as DC either against the user U_i or the sensor device SD_j then the proposed protocol is secured if,*

$$Adv_{LDA-P}(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}| * 2^l}, \rho_{fp})) + (\frac{q_s + q_e}{2^{l_r}}) + (\frac{1}{2^{l_j}}) \tag{6.4}$$

In Equation 6.4, l_h is the size of the return value of Hash($\mathcal{H}$) query generated by $\mathcal{A}$ in bits, l_r is the size of the random nonce generated by the protocol $LDA-P$. $|\mathcal{DC}|$ shows the finite size of a password dictionary, and ρ_{fp} shows the probability of a false positive occurrence by $\mathcal{A}$.

Proof. four security games are defined : $\{Gm_0, Gm_1, Gm_2, Gm_3, Gm_4\}$ for to prove that the proposed protocol P is secured against an adversary $\mathcal{A}$ and $Adv_{LDA-P}(\mathcal{A})$ under defined ROR model. The Game starts with the Gm_0 and terminates at the Gm_4. Now let us define an event $SUCC_i$ which represents the correct guess for the bit b in each game Gm_i via the *test* query $\mathcal{T}$ by an adversary $\mathcal{A}$. □

Game Gm_0:

The Gm_0 is the initial game in which a real protocol $LDA-P$ is equal to an initial game so that:

$$Adv_{LDA-P}(\mathcal{A}) = 2 * Pr[\mathcal{SUCC}_0] - 1 \tag{6.5}$$

Game Gm_1:

In the Gm_1, an *Execute query* $\mathcal{E}$ is performed by single intruder or multiple intruder $\mathcal{A}$. An adversary $\mathcal{A}$ performs $\mathcal{E}(\Omega^m, \Omega^n, \Omega^o)$ query to spoof communication between valid participants. An Adversary $\mathcal{A}$ performs the test query $\mathcal{T}$ to achieve the session key. An adversary $\mathcal{A}$ must be capable enough to differentiate between a real session key and a random number. If the information retrieved using above queries provide sufficient information to $\mathcal{A}$ for the computation of SK,

then the Gm_1 is won by adversary $\mathcal{A}$ else :

$$Pr[\mathcal{SUCC}_0] = Pr[\mathcal{SUCC}_1] \tag{6.6}$$

Simulation of Gm_1: The game Gm_1 defines passive attack. Let us assume that adversary gets messages :

Message 1: $GW \rightarrow SD_j$: (H($UID_i||r_2||SID_j||GWID||T_7$), Enc($UID_i$, r_2), T_7)

Message 2: $SD_j \rightarrow GW$: (Enc(r_3), H($GWID||SID_j||r_3||T_8$), H($GWID||SID_j||r_2||T_8$), T_8)

Message 3: $GW \rightarrow U_i$: (Enc(r_3), H($GWID||SID_j||r_3||T_8$), H($Z_1||Z_2||K_s||T_9$), T_8, T_9)

Message 4: $GW \rightarrow SD_j$: (H($Z_1||Z_2||K_s||T_9$), T_9)

Note that the session key computation at user side U_i and Sensing device side SD_j occurs as SK = H($r_2||r_3||UID_i||SID_j||GWID_k||T_8||M_k||T_9$). where M_k = H($Z_1||Z_2||K_s||T_9$). To Compute the session key an adversary $\mathcal{A}$ needs $\{r_2, r_3, UID_i, SID_j$ and $GWID_k\}$ which involves random number and digested identities which are unknown to him/her. Thus, the winning probability for an adversary $\mathcal{A}$ does not increase. And hence, it is obvious that an Equation 6.6 satisfies for an adversary $\mathcal{A}$ under the game Gm_1.

Game Gm_2:

Definition: This game executes following queries,

- Hash query $\mathcal{H}(\Omega^m, m_i)$ to retrieve hash output of message m_i.
- Send query $\mathcal{S}(\Omega^n_{SD_j})$ to get messages from Sensing device.
- Send query $\mathcal{S}(\Omega^m_{U_i})$ to get messages from User device.

In this game an adversary $\mathcal{A}$ convinces legal participants for receiving a modified messages. Thus in this game $\mathcal{A}$ can also perform hash operation to validate the modification. An adversary $\mathcal{A}$ validates collision and as per birthday paradox defines, the probability of a collision for the oracle $\mathcal{H}$ is at most $\frac{q_h^2}{2^{l_h}}$. In the proposed protocol (*LDA-P*), the first message communicated between GW and SD_j is *message 1* = (H($UID_i||r_2||SID_j||GWID||T_7$), Enc($UID_i$, r_2), T_7) which includes secret identity of the user device and sensing device, random number r_2 and time stamp T_7. Thus, it is not possible to generate collision after performing the send query $\mathcal{S}$. Similarly every other communicated messages (*message 2*, *message 3*, *message 4*) contains random nonces, time-stamps, gateway master secrets and so on. maximum collision possibility for the random numbers is at most $\frac{q_s+q_e}{2^{l_r}}$. Thus after this game:

$$Pr[\mathcal{SUCC}_2] - Pr[\mathcal{SUCC}_1] \leq \frac{q_h^2}{2^{l_h}} + \frac{q_s + q_e}{2^{l_r}} \tag{6.7}$$

Game Gm_3:

Definition: This game executes after the Gm_0, Gm_1, Gm_2 games. The game Gm_3 performs following query,

- Query *CorruptUserSc*($\Omega_{U_i}^m$) provides user U_i's smart card data to an adversary $\mathcal{A}$.
- Query *CorruptUserDevice*($\Omega_{U_i}^m$) provides data stored in user U_i's device to an adversary $\mathcal{A}$.
- An adversary $\mathcal{A}$ tries to guess the password used by the user.

In this game Gm_3, an adversary tries to read the smart card data for the user U_i.

The user U_i stores (TID_i, M_1, B_1, M_2) in the smart card where TID_i = H($r_1||X_1$), M_1= H($TID_i||GWID||R_i$), $B_1 = X_1 \oplus r_1$, M_2 = H($TID_i||\delta_i||R_i$). Using *CorruptUserSc* query, an adversary $\mathcal{A}$ reads these all the parameters and tries to compute password PW_i. But the password PW_i is protected through a random number by gateway and the probability of guessing a correct random number is almost NULL. Now an adversary tries to acquire the biometric ρ_i from the user's data by performing the query *CorruptUserDevice*. In the proposed *LDA-P* protocol, strong fuzzy extractor is used which retrieves at most l bits and guessing probability for $\rho_i \in o, 1^l$ is approximately $\frac{1}{2^l}$. Now an adversary tries to perform the online password guessing attack but for a particular session, the number of attempts allowed to execute is limited Thus,

$$Pr[\mathcal{SUCC}_3] - Pr[\mathcal{SUCC}_2] \leq max(q_s, (\frac{1}{|\mathcal{DS}| * 2^l}, \rho_{fp})) \tag{6.8}$$

Game Gm_4**:**

Definition: This game executes after the completion of Gm_0 to Gm_3 games. The game Gm_4 performs following query,

- Query *CorruptSensingDevice*($\Omega^o_{SD_j}$) enables an adversary $\mathcal{A}$ to capture stored parameter inside the sensing device SD_j.
- Query *CorruptLevelSensingDevice*($\Omega^o_{SD_j}$) enables an adversary $\mathcal{A}$ for capability of guessing the level l_j for the sensing device SD_j

Now let us assume an adversary $\mathcal{A}$ performs above query on the proposed protocol *LDA-P* then it receives Z_2, X_2, and sensing device long term secret R_j from the memory of sensing device. The session key computed between user U_i and sens-

ing device SD_j does not use either Z_2 or X_2. Now let us assume that $\mathcal{A}$ uses R_j to encrypt the receiving messages then he/she can get r_2 parameter of the session key SK but can not capture the r_3 and other parameters due to their ECC based encryption, randomness and hashed outputs. By using *CorruptLevelSensingDevice* query, an adversary $\mathcal{A}$ guesses the level l_j for the sensing device. During the session key computation, neither the user U_i nor the sensing device SD_j uses l_j directly for the session key computation. Now let us assume that the probability of guessing the correct level is $\frac{1}{2^{l_j}}$ where 2^{l_j} represents the number of bits used for the level. Thus, after this game,

$$Pr[\mathcal{SUCC}_4] - Pr[\mathcal{SUCC}_3] \leq \frac{1}{2^{l_j}} \tag{6.9}$$

Thus, after completion of all games Gm_0 to Gm_4, an adversary $\mathcal{A}$ tried all the random oracle queries but he/she does not achieve success in session key generation. Now the only option left with an adversary is to guess the correct bit b for to win this game. Thus, an adversary $\mathcal{A}$ performs *test* query $\mathcal{T}$. Hence overall,

$$Pr[\mathcal{SUCC}_4] = \frac{1}{2 * 2^{l_j}} \tag{6.10}$$

Now, from equation 6.6, $\frac{1}{2}*Adv_{LDA-P}$ = $[Pr[Succ_0]$ - $\frac{1}{2}]$. So by using the triangular inequality, / $[Pr[\mathcal{SUCC}_1]$ - $[Pr[\mathcal{SUCC}_4] \leq [Pr[\mathcal{SUCC}_1]$ - $[Pr[\mathcal{SUCC}_2]$ + $[Pr[\mathcal{SUCC}_2]$ - $[Pr[\mathcal{SUCC}_4] \leq [Pr[\mathcal{SUCC}_1]$ - $[Pr[\mathcal{SUCC}_2] + [Pr[\mathcal{SUCC}_2]$ - $[Pr[\mathcal{SUCC}_3] \leq \frac{q_h^2}{2^{l_h}}$ + $\max(q_s, (\frac{1}{|\mathcal{DC}|*2^l}, \rho_{fp}))$ + $(\frac{q_s+q_e}{2^{l_r}})$ + $(\frac{1}{2^{l_j}})$. Using equation

6.6-6.10,

$$|Pr[\mathcal{SUCC}_0] - \frac{1}{2}| \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}| * 2^l}, \rho_{fp})) + (\frac{q_s + q_e}{2^{l_r}}) + (\frac{1}{2^{l_j}}) \quad (6.11)$$

So finally, from the equation 6.4 and 6.11,

$$Adv_{LDA-P}(\mathcal{A}) \leq \frac{q_h^2}{2^{l_h}} + max(q_s, (\frac{1}{|\mathcal{DC}| * 2^l}, \rho_{fp})) + (\frac{q_s + q_e}{2^{l_r}}) + (\frac{1}{2^{l_j}}) \quad (6.12)$$

6.6 Implementation and Testbeds

In this section, the implementation approach for the proposed LDA scheme is discussed. The proposed authentication scheme is implemented using the Message Queuing Telemetry Transport (MQTT) protocol is an application layer protocol and IPv6 as a network layer protocol. In this section, a comparison of the proposed authentication scheme is also presented with existing schemes in terms of communication cost (in bits), computation cost (based on operations), and networking parameters like round-trip delay, packet loss, and throughput.

Implementation Scenario An environment for the implementation of the proposed LDA scheme is highlighted in the following Table 6.4.

Network Model	Generic IoT Model
Protocol	Using MQTT
Broker	Mosquitto
Secure channel	By Enabling TLS communication in Mosquitto
ECC Curve	NIST P-256 Curve
ECC Multiplication	Using double and Add method
Message format	JSON Type
Language	Python
User Device	Intel (R) Core (TM) i3-7500 CPU with 2.80 GHz.
Sensing device	1 GB RAM, Raspberry PI 3.0
Gateway System	Architecture = x86-64, Processor: Intel (R) Core (TM) i5-7500 CPU with 3.40 GHz.

TABLE 6.4
Implementation Environment

For the implementation of the proposed LDA Protocol, ten user devices, ten sensing devices, and one intermediate gateway was taken. The configuration for all the devices is highlighted in Table 6.4. A publish-subscribe model was created for communication using the MQTT protocol in which the user device and the sensing device subscribes to the gateway topics and communicate with each other via the gateway device. For the initialization and registration phase, a secure channel was created using the TLS connection of the MQTT protocol. To achieve higher accuracy in MQTT communication is a big challenge. The lightweight and secure MQTT communication is also a futuristic research area for the IoT security paradigm. For implementations we have used SHA256 as a hash function and NIST recommended P-256 Curve. The security level achieved through implementation of proposed scheme is 128 bit.

The above Fig.6.2. shows the implementation model, and the following Fig.6.3 shows the generated session key between the user device and the sensing device.

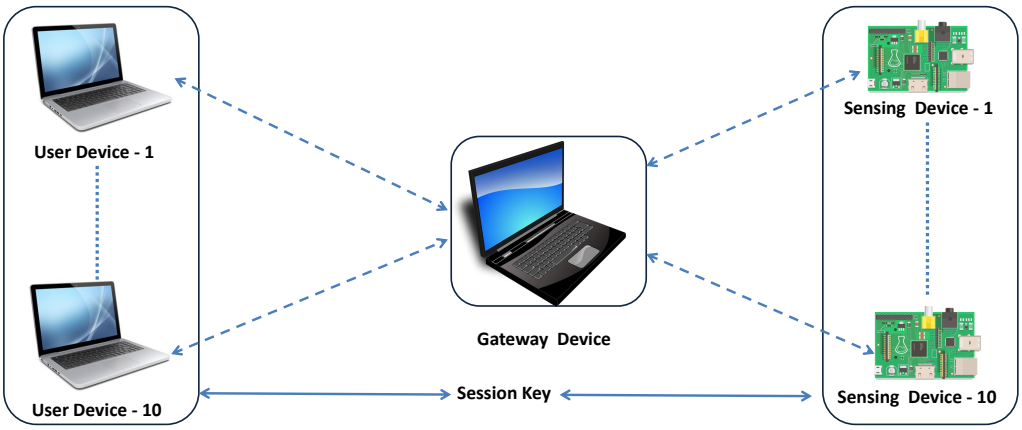

Figure 6.2. IoT Implementation Model

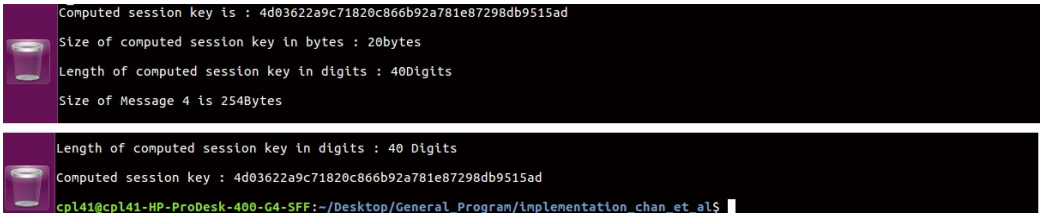

Figure 6.3. Computed Session Key

6.7 Use cases

This chapter proposes three factor based LDA scheme for the user-gateway-sensor model. The proposed scheme aims to provide a lightweight communication between the multiple user devices and the numerous amount of lightweight sensing devices deployed on the ground. The following are some of the significant use-cases of the proposed work.

- The proposed scheme presents significantly improved authentication scheme for the IoT application where the quantity of deployed sensing devices is large and have hierarchical user base.
- The proposed scheme can be highly useful where user is having mobile

devices capable to sense the biometric traits such as finger print.

- The significant use case of the proposed scheme mostly lies in environment where user have option to provide biometric trait as an input and wants to receive a data from the sensing devices.

The novelty of the proposed work lies in its dynamic environment, lightweight computations and real-time implementations. Real-time implementation of the proposed ECC based LDA scheme using MQTT protocol (that itself is a lightweight communication protocol used by the IoT industry) makes it novel. The real-time implementation using MQTT protocol has advantages like the lightweight header and reliable communication over other conventional protocols.

6.8 Comparative Analysis

A Network Parameter Analysis

In this subsection, networking related parameters like throughput, round trip delay (RTD), and packet loss are discussed. The "WIRESHARK" tool was setup to monitor MQTT packets and collect the data regarding packet loss, the number of packets transmitted per unit time, and how much time each packet takes to reach the destination. Rather than simulation through any simulator, a raspberry pi, and laptop devices were deployed to implement the proposed scheme and collected real-time data. Thus, the outcome of throughput, RTD, and packet loss for the proposed scheme is as follows.

Throughput

The throughput can be computed in two ways, one based on the number of bits transmitted per unit time and the second based on the number of packets transmitted per unit time. Our implementation collected data for ten static users, one gateway, and ten static sensors. The throughput is 167 bps, 237 bps, and 93bps at the user, gateway, and the sensor device, respectively. Thus, the computation cost for the proposed scheme is 4.98 seconds, 5.26 seconds, and 9.29 seconds at the user, gateway, and sensor device, Now, if the number of packets transmitted per unit time is considered, then the throughput can be computed as $\frac{total\,packet\,received * packet\,size}{total\,time}$. Thus, by this formula, the average number of packet received at the user is 9, the sensor is 11 and gateway is 45 where packet size communicated from user to gateway is 7 byte, gateway to user and sensor is 9 byte and sensor to the gateway is 5 byte through MQTT. Therefore, the average throughput for the proposed scheme is 22.58 bps.

Round Trip Delay (RTD)

The round trip delay is computed as an average time required by a communicated packet to arrive at the destination from the source. For the experimental purpose, through our scenario of ten users, one gateway, and ten sensing device, a simultaneous requests to gateway device were generated from each user for accessing sensors of different levels. Then the average RTD at the user device, which includes the time required between sending a request to receiving a reply from the gateway via a sensing device, is 0.4836 seconds. The average RTD at the sensing device, which includes the time between the sensor's reply to the gateway and

gateway's response to the sensor, is 0.4625 second. If some requests are sent in which the user is not eligible to get access to the sensor of a particular level, then the RTD increases due to gateway node take little more verification time. If the gateway device does not found a valid user, subsequently, it communicates zero signal to both the user and sensor device to terminate communication.

B Communication Cost

The communication cost represents the number of bits communicated before establishing the session key for the proposed protocol. A communication cost in the unit of "bits" is taken. In the proposed LDA protocol, the size of a randomly generated identity and a password is 160 bits. Due to resource constraints, as a biometric parameter, The 160 bits random binary string was used. The output size of a hash operation is 256 bits due to the use of SHA-256. The size of the randomly generated nonce is 128 bits, and the size of the timestamp is 32 bits. An ECC is used for the key generation. Each point in the ECC has two coordinates, and each coordinate size is 160 bits; thus, one point (X_p, Y_p) represents a total of 160 + 160 = 320 bits. Therefore, the public key size (PK_s) is 320 bits, while the private key (K_s) size is 160 bits as per ECC computation of the proposed LDA scheme. Table 6.5 presents communication cost comparison for the proposed scheme and existing schemes.

Scheme	*User*	*Gateway*	*Sensor*	*Total Cost*
[Challa et al. (2017)]	992	1024	512	2528
[Shin & Kwon (2020)]	1158	1560	678	3552
[Zhou et al. (2019)]	832	2048	672	3552
[Wu et al. (2020)]	813	2110	789	3712
[Shuai et al. (2020)]	1260	1696	1344	4256
Proposed	832	1248	864	2944

TABLE 6.5
Communication Costs Comparison

C Computation Cost

The computation cost represents the number of cryptographic operations used in the proposed scheme during the login and authentication stage. It also gives the total time required by those operations at each participant's devices. Let T_E/T_D, T_h, T_{ecm}, T_{fe} represent the computation cost of ECC encryption/decryption operation, one-way hash function H(.), ECC Point multiplication and fuzzy extra traction operation respectively. The computation cost of the bitwise XOR operation is not considered because it takes very little time (almost 0 ms) compare to other operations. Following are the realtime observations through deployment of devices in university campus,

- For the gateway device, T_E/T_D operation takes 0.06783 seconds, T_h operation takes 0.00034 seconds and the T_ecm operation takes 0.0589 seconds.
- For the user device, T_E/T_D operation takes 0.07083 seconds, T_h operation takes 0.00041 seconds, the T_ecm operation takes 0.0607 seconds and T_{fe} operation takes 0.0503 seconds.
- For the sensing device, T_E/T_D operation takes 0.08883 seconds, T_h operation

takes 0.00084 seconds and the T_ecm operation takes 0.0703 seconds.

Above all the costs are an average of 100 times verified outputs. Table 6.6 presents computation cost comparison for the proposed scheme and existing schemes.

Scheme	***User***	***Gateway***	***Sensor***	***Time(S)***
[Challa et al. (2017)]	T_{fe}+5*T_{ecm}+5*T_h	5*T_{ecm}+4*T_h	4*T_{ecm}+3*T_h	1.11501
[Shin & Kwon (2020)]	3*T_P + 14*T_h	T_P + 12*T_h	2*T_P + 5*T_h	0.866
[Zhou et al. (2019)]	4*T_{ecm}+5*T_h	3*T_{ecm}+7*T_h	4*T_{ecm}+6*T_h	1.069
[Wu et al. (2020)]	13*T_h + T_{BKG}	15*T_h	4*T_h	1.503
[Shuai et al. (2020)]	7*T_h + T_{ecm} + T_{QR}	7*T_h + T_{QR}	5*T_h + T_{ecm}	2.150
Proposed	7*T_h + 2*T_e + T_{fe}	11*T_h+2*T_e	5*T_h+1*T_e	0.72656

TABLE 6.6
Computation Cost Comparison

Following Fig.6.4. gives performance comparison of communication cost and computation cost of the proposed scheme with the existing schemes. The compar-

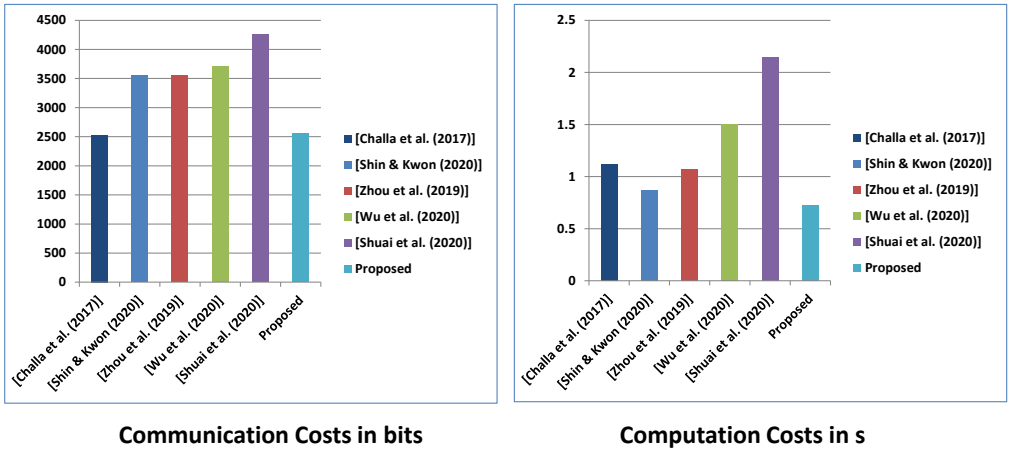

Figure 6.4. Performance Comparison Chart

ative analysis shows that the proposed scheme has little higher computation and communication costs than other existing schemes. Still, the security index for the proposed scheme is higher compared to other existing schemes. In the proposed scheme, a novel idea of the LDA is presented, which fulfills the requirement of real-time deployment of the IoT application. The proposed LDA scheme is used in IoT applications with a hierarchical access control model. The proposed LDA reduces the number of user registration steps, reduces memory utilization and search operations and discard the need of separate access control mechanism.

6.9 Summary

This chapter highlights a novel concept of Level Dependent Authentication (LDA) in which the gateway defines user and sensor levels during initialization. Later on, only valid users can access the sensors deployed at a particular level. The proposed scheme has three important phases. In the initialization phase, the gateway creates a credential for the user and sensor. Later on, the user does the registration, which is followed by login and authentication. The security analysis for the proposed scheme is provided through the random oracles and an AVISPA tool. The BAN Logic is used to prove the mutual authentication property. The implementation of the proposed scheme was performed through the raspberry-pi and laptops as a static device and the MQTT as an application layer protocol. Overall, the proposed scheme successfully achieves a security objective for the real-time deployment of any generic IoT application. This chapter also provides a comparative analysis of the proposed scheme with the other existing schemes. Even though, this chapter presents strong enough security for the LDA scheme using

three factor authentication, there is a scope of further development and have broad application oriented future work. Next chapter 7 provides conclusion and future work related to this thesis.

Chapter 7

Conclusions and Future Scope

This chapter presents summary of major contributions of this thesis and also provide light on road-map for future research directions related to the user and device authentication in IoT. ***Section 7.1*** presents conclusion of this thesis. ***Section 7.2*** provides future work related to this thesis

7.1 Conclusions

The major contributions of this thesis can be outlined as follows. In this thesis, we have proposed four remote user authentication schemes using ECC for the IoT environment in the following manners.

- Authentication between user - gateway where sensing devices are not capable to perform computations.
- Authentication between user - gateway - sensing devices in which user have to register for each sensing device.

- Level dependent authentication using two factors that tries to reduce number of registrations for users and provide light weight key exchange.
- Level dependent authentication using three factors that provides better security compare to two factor based authentication with light weight computations.

In the *first* contribution (***Chapter 3***), we have proposed a novel authentication scheme between user device and the gateway device. In this contribution, we have considered sensing devices as a very tiny devices which are not capable to perform complex cryptography operations. In this contribution, there are three phases. The first phase provides initialization step that is performed in secure environment. Next phase is login and authentication phase that is performed over open channel and as an outcome of this phase, a session key will be generated between user and gateway device. We validate the security of this proposed work by formal security models such as BAN Logic and ROR model as well as informal security analysis over dolev-yao channel. The proposed scheme of this contribution was also implemented in realtime environment and was compared with other existing schemes. This contribution was enhanced in next chapter in which we have also used smart lightweight device also for session key generation with other devices.

In the *second* contribution (***Chapter 4***), we have proposed a novel authentication scheme between user device, gateway device and the smart meter which is considered as a lightweight device capable to perform cryptography operations. In this contribution, we have considered real time IoT environment with the application of energy monitoring in smart home. In this contribution, there are three phases. The first phase provides initialization step that is performed in secure en-

vironment. Next phase is registration phase, and last phase is authentication and key exchange phase that is performed over open channel and as an outcome of this phase, a session key will be generated between user and the smart meter. We validate the security of this proposed work by formal security models such as BAN Logic and ROR model as well as informal security analysis over dolev-yao channel. The proposed scheme of this contribution was also implemented in realtime environment by considering raspberry pi as a smart meter device and was compared with other existing schemes. The major drawback of this work we found was that user need to register for each sensing device. Now, in the IoT environment where thousands sensing devices are deployed then it becomes tedious task to do that. Thus, this contribution was enhanced in next chapter in which we have come up with novel ideal of level dependent authentication.

In the *third* contribution (***Chapter 5***), we have proposed a novel authentication scheme called as a level dependent authentication between user device, gateway device and the sensing device using two factors. In this contribution, we have considered real time IoT environment where thousands of sensors are deployed on ground and there is a need to provide data to system users on time. In this contribution, user registers with some level and he/she will have access of sensing devices which are deployed below that level. In this contribution, there are three phases. The first phase provides initialization step that is performed in secure environment by the gateway device. Next phase is registration phase and last phase is authentication and key exchange phase that is performed over open channel and as an outcome of this phase, a session key will be generated between user and the sensing devices. User will have access of all those sensors which are below the user level. So onetime registration of the user will enable access to all eligible

sensing devices. We validate the security of this proposed work by formal security models such as BAN Logic and ROR model as well as informal security analysis over dolev-yao channel. The proposed scheme of this contribution was also implemented in realtime environment by considering raspberry pis as a sensing devices and was compared with the other existing schemes. The major drawback of this work we found was that this in this scheme, user does not use any biometric trait to perform authentication and that sometime leads toward system breakout. Thus, this contribution was enhanced in next chapter in which we have used three factor based authentication with LDA for to improve the security.

In the *last* contribution (***Chapter 6***), we have proposed a novel authentication scheme called as a level dependent authentication between user device, gateway device and the sensing device using three factors. In this contribution, we have considered real time IoT environment where thousands of sensors are deployed on ground and there is a need to provide data to system users on time. In this contribution, user registers with some level and he/she will have access of sensing devices which are deployed below that level. In this contribution, there are three phases. The first phase provides initialization step that is performed in secure environment by the gateway device. Next phase is registration phase and last phase is authentication and key exchange phase that is performed over open channel and as an outcome of this phase, a session key will be generated between user and the sensing devices. User will have access of all those sensors which are below the user level. So onetime registration of the user will enable access to all eligible sensing devices. We validate the security of this proposed work by formal security models such as BAN Logic and ROR model as well as informal security analysis over dolev-yao channel. The proposed scheme of this contribution was also

implemented in realtime environment by considering raspberry pis as a sensing devices and was compared with the other existing schemes. Still, there is a scope of development in the RUA for IoT environment using new lightweight cryptography operations and lightweight hardware. In next section 7.2, we provide an overview on tentative future work.

7.2 Future Work

In this section, we put forward some possible directions for future work. One can investigate following research directions :

A Deployment of LDA approach in other realtime IoT applications

In this thesis, we proposed LDA approach that we implemented in university campus environment but it still needs to verify in other IoT applications such as SH, SI and so on. Now a days people uses authentication and access control as a separate security mechanism but we believe that LDA can work as a joint approach for both these security parameters. Hence, there is a wide scope of further development in LDA approach. An IoT authentication can have multiple other phases such as dynamic device addition, run time user registration, user revocation or re-registration phases. This thesis doesn't provide this phases for LDA approach that needs to design to improve the services in IoT eco-system. Hence, in future, we will implement proposed LDA with other realtime IoT applications and improve the security and services by designing new phases.

B M2M or D2D Authentication

An IoT eco-system is a network of billions of interconnected heterogeneous devices. In this network, interconnected devices communicates with each other as well as communicates back-end infrastructures (such as databases, cloud) to transmit the data and store the data. Similarly M2M communication connects numerous machine with IoT network. Thus, authentication between (or "of") these devices and machines must be done to secure the whole eco-system. Hence only authorized devices and machines can connect and communicate inside the IoT eco-system.

C Authentication in the Multi gateway Environment

In this thesis, we considered only single gateway environment where user had only one intermediary option to communicate with the sensing device and this gateway device can be always a targeted point for the attacker to perform DoS and DDoS type attacks. Hence, there can be multiple gateway deployed to provide uninterrupted services to the IoT eco-system. There can be multi gateway environment to provide faster services to the user (i.e. many road side units to provide faster services to ongoing vehicles) and in that circumstances, authentication must occur faster in very lighter mode without damage to the security.

D Authentication thorough Physical Unclonable Functions or Secure Elements

The Secure Element (SE) aims to store the secret credential in the tamper-resistant environment. The SE is like micro controller which is having two type of memory. In first memory, it stores public elements and in second memory it stores secret elements which are accessible by only SE controller. Hence, designing an authentication schemes for IoT devices using SE can improve the IoT eco-system security. The Physical Unclonable Function (PUF) is also another option that is adopted by many researchers who are working on hardware based authentication for IoT devices [Garg et al. (2020)]. Any PUF is unclonable and it's output is non predictable as well as dependent on physical system.

E Learning Approach for Authentication

Machine Learning (ML) and its other sub parts such as deep learning and federated learning are the growing fields. Authors in presents an authentication scheme based on deep neural network framework that allows real-time authentication of the wireless nodes, using the effects of inherent process variation on Radio Frequency (RF) properties of the wireless transmitters (Tx), detected through in-situ machine learning at the receiver (Rx) end. In this paper, author uses DNN techniques to authenticate the devices [Chatterjee et al. (2019)]. Similarly other authors used random forest classification, generative adversarial network and other classification based methods to develop authentication approach for IoT devices. Though there are challenges such as proper dataset and run time modeling exists in learning based techniques, there is a wide scope of research in this direction.

List of Publications

Book

1. Chintan Patel, and Nishant Doshi, Internet of things security: challenges, advances, and analytics. CRC Press (Taylor and Francis Group), Auerbach Publications, ISBN: 9780429454448, 2018.
   DoI : `https://doi.org/10.1201/9780429454448`. [Chapter 1-2-3]

Book Chapter

1. Chintan Patel, and Nishant Doshi, Patel C.,(2019) Security Challenges in IoT Cyber World. In: Hassanien A., Elhoseny M., Ahmed S., Singh A. (eds) Security in Smart Cities: Models, Applications, and Challenges. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham.
   DoI : `https://doi.org/10.1007/978-3-030-01560-2_8`[Chapter 1-2-3]

Journal Publications

1. Chintan Patel, and Nishant Doshi, "Secure Lightweight Key Exchange Using ECC for User-Gateway Paradigm "**Published** in IEEE Transactions on Computers (SCI Indexed, H-Index = 118),
   DoI : `10.1109/TC.2020.3026027` [Chapter 3].

2. Chintan Patel, and Nishant Doshi,"Cryptanalysis and Improvement of Barman et al.'s Secure Remote User Authentication Scheme "**published** in International journal of Circuits, Systems And Signal Processing (Scopus), NAUN Publication, Vol 13 No. 1, pp. 604-610, 2019.[Chapter 5]

3. Chintan Patel & Nishant Doshi (2021) LDA-IoT : a level dependent authentication for IoT paradigm, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2021.1931573. [Chapter 6]

Conference Publications

1. Chintan Patel, and Nishant Doshi,"Internet of Things: A Review on Major Challenges and Applications,"Proceedings of INCRS 2018, IIITDM Jabalpur, Gupta V., Varde P., Kankar P., Joshi N. (eds) Reliability and Risk Assessment in Engineering. Lecture Notes in Mechanical Engineering. Springer, Singapore. [Chapter 1-2-3]
   DoI : `https://doi.org/10.1007/978-981-15-3746-2_40`

2. Chintan Patel, and Nishant Doshi,"Cryptanalysis of ecc-based key agreement scheme for generic IoT network model,"

Proceedings of 10th IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT), IIT Kanpur, India, 2019. [Chapter 1,2,3]

DoI : 10.1109/ICCCNT45670.2019.8944674.

3. Chintan Patel, and Nishant Doshi"A Novel Lightweight Authentication for Intelligent Energy Monitoring in Smart Home,"Proceedings of ISTA 2019, International Conference on Applied Soft computing and Communication Networks (ACN'19), December 18-21, 2019, IIITM, Trivandrum, India. in Thampi S. et al. (eds) Intelligent Systems, Technologies and Applications. Advances in Intelligent Systems and Computing, vol 1148. Springer, Singapore. [Chapter 4]

   DoI : https://doi.org/10.1007/978-981-15-3914-5_17.

4. Chintan Patel, and Nishant Doshi, "A Novel MQTT Security framework In Generic IoT Model, CoCoNet-2019, IIITM Trivandrum, in Procedia Computer Science, Elsevier, Vol 171, pp. 1399-1408, 2020. [Chapter 1-2-3]

   DoI : https://doi.org/10.1016/j.procs.2020.04.150.

Bibliography

Abbasinezhad-Mood, D., & Nikooghadam, M. (2018). Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. *Future Generation Computer Systems*, *84*, 47 - 57. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0167739X17315376` doi: https://doi.org/10.1016/j.future.2018.02.034

Abdalla, M., & Pointcheval, D. (2005). Simple password-based encrypted key exchange protocols. In *Cryptographers' track at the rsa conference* (pp. 191–208).

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, *17*(4), 2347-2376. doi: 10.1109/COMST.2015.2444095

Amin, R., Islam, S. H., Biswas, G., Khan, M. K., & Kumar, N. (2018). A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Generation Computer Systems*, *80*, 483 - 495.

Retrieved from http://www.sciencedirect.com/science/article/pii/S0167739X16301509 doi: https://doi.org/10.1016/j.future.2016.05.032

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787–2805.

Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications,Elsevier*, *54*, 1–31. Retrieved from http://dx.doi.org/10.1016/j.comcom.2014.09.008 doi: 10.1016/j.comcom.2014.09.008

Burrows, M., Abadi, M., & Needham, R. M. (1989). A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, *426*(1871), 233–271.

Challa, S., Wazid, M., Das, A. K., Kumar, N., Goutham Reddy, A., Yoon, E., & Yoo, K. (2017). Secure signature-based authenticated key establishment scheme for future iot applications. *IEEE Access*, *5*, 3028-3043. doi: 10.1109/ACCESS.2017.2676119

Chang, C.-C., & Hwang, S.-J. (1993). Using smart cards to authenticate remote passwords. *Computers & Mathematics with Applications*, *26*(7), 19–27.

Chatterjee, B., Das, D., Maity, S., & Sen, S. (2019). Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning. *IEEE Internet of Things Journal*, *6*(1), 388-398. doi: 10.1109/JIOT.2018.2849324

Chatterjee, S., Roy, S., Das, A. K., Chattopadhyay, S., Kumar, N., & Vasilakos, A. V. (2018, Sep.). Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. *IEEE Transactions on Dependable and Secure Computing*, *15*(5), 824-839. doi: 10.1109/TDSC.2016.2616876

Chaturvedi, A., Mishra, D., & Mukhopadhyay, S. (2017). An enhanced dynamic id-based authentication scheme for telecare medical information systems. *Journal of King Saud University - Computer and Information Sciences*, *29*(1), 54 - 62. Retrieved from http://www.sciencedirect.com/science/article/pii/S1319157815000865 doi: https://doi.org/10.1016/j.jksuci.2014.12.007

Chaudhry, S. A., Naqvi, H., Shon, T., Sher, M., & Farash, M. S. (2015, Apr 26). Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. *Journal of Medical Systems*, *39*(6), 66. Retrieved from https://doi.org/10.1007/s10916-015-0244-0 doi: 10.1007/s10916-015-0244-0

Chifor, B.-C., Bica, I., Patriciu, V.-V., & Pop, F. (2018). A security authorization scheme for smart home internet of things devices. *Future Generation Computer Systems*, *86*, 740–749.

Cisco. (2014). The internet of things reference model. Retrieved from http://cdn.iotwf.com/resources/71/IoT-Reference-Model-White-Paper-June-4-2014.pdf

Das, A. K., Wazid, M., Yannam, A. R., Rodrigues, J. J. P. C., & Park, Y. (2019). Provably secure ecc-based device access control and key agreement protocol

for iot environment. *IEEE Access*, *7*, 55382-55397. doi: 10.1109/ACCESS .2019.2912998

Das, M. L., Saxena, A., & Gulati, V. P. (2004). A dynamic id-based remote user authentication scheme. *IEEE transactions on Consumer Electronics*, *50*(2), 629–631.

Dhillon, P. K., & Kalra, S. (2019). Secure and efficient ecc based sip authentication scheme for voip communications in internet of things. *Multimedia Tools and Applications*, *78*(16), 22199–22222.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, *22*(6), 644–654.

Dodis, Y., Reyzin, L., & Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques* (pp. 523–540).

Dolev, D., & Yao, A. C. (1981). On the security of public key protocols. In *Proceedings of the 22nd annual symposium on foundations of computer science* (pp. 350–357). Washington, DC, USA: IEEE Computer Society. Retrieved from `https://doi.org/10.1109/SFCS.1981.32` doi: 10.1109/SFCS.1981 .32

Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, *31*(4), 469-472. doi: 10.1109/TIT.1985.1057074

Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F. B., Rodriguez, J., Bicaku, A., ... Bastos, J. (2019, Feb). A lightweight authentication mechanism for m2m communications in industrial iot environment. *IEEE Internet of Things Journal*, *6*(1), 288-296. doi: 10.1109/JIOT.2017.2737630

Farash, M. S., Turkanović, M., Kumari, S., & Hölbl, M. (2016). An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. *Ad Hoc Networks*, *36*, 152–176.

Garg, S., Kaur, K., Kaddoum, G., & Choo, K.-K. R. (2020). Toward secure and provable authentication for internet of things: Realizing industry 4.0. *IEEE Internet of Things Journal*, *7*(5), 4598-4606. doi: 10.1109/JIOT.2019.2942271

Garg, S., Kaur, K., Kaddoum, G., & Choo, K. R. (2020). Toward secure and provable authentication for internet of things: Realizing industry 4.0. *IEEE Internet of Things Journal*, *7*(5), 4598-4606.

Gil, D., Ferrández, A., Mora-Mora, H., & Peral, J. (2016). Internet of things: A review of surveys based on context aware intelligent services. *Sensors*, *16*(7). Retrieved from `http://www.mdpi.com/1424-8220/16/7/1069` doi: 10.3390/s16071069

Gope, P., & Sikdar, B. (2019, Feb). Lightweight and privacy-preserving two-factor authentication scheme for iot devices. *IEEE Internet of Things Journal*, *6*(1), 580-589. doi: 10.1109/JIOT.2018.2846299

Granjal, J., Monteiro, E., & Silva, J. S. (2015, thirdquarter). Security for the internet of things: A survey of existing protocols and open research issues.

IEEE Communications Surveys Tutorials, *17*(3), 1294-1312. doi: 10.1109/COMST.2015.2388550

Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer Science & Business Media.

Hatton, M. (2020). The iot in 2030: 24 billion connected things generating $1.5 trillion. *IoT Business news*.

Jangirala, S., Das, A. K., Wazid, M., & Vasilakos, A. V. (2020). Designing secure user authentication protocol for big data collection in iot-based intelligent transportation system. *IEEE Internet of Things Journal*, 1-1. doi: 10.1109/JIOT.2020.3040938

Jiang, Q., Zeadally, S., Ma, J., & He, D. (2017). Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access*, *5*, 3376–3392.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014, November). Security of the internet of things: Perspectives and challenges. *Wirel. Netw.*, *20*(8), 2481–2501. Retrieved from `http://dx.doi.org/10.1007/s11276-014-0761-7` doi: 10.1007/s11276-014-0761-7

Jo, H. J., Kim, I. S., & Lee, D. H. (2016, May). Efficient and privacy-preserving metering protocols for smart grid systems. *IEEE Transactions on Smart Grid*, *7*(3), 1732-1742. doi: 10.1109/TSG.2015.2449278

Kang, J., Park, G., & Park, J. H. (2016). Design of secure authentication scheme between devices based on zero-knowledge proofs in home automation service environments. *The Journal of Supercomputing*, *72*(11), 4319–4336.

Karati, A., Islam, S. H., & Karuppiah, M. (2018). Provably secure and lightweight certificateless signature scheme for iiot environments. *IEEE Transactions on Industrial Informatics*, *14*(8), 3701–3711.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, *48*(177), 203–209.

Kocher, P. C., Jaffe, J., & Jun, B. (1999). Differential power analysis. In *Proceedings of the 19th annual international cryptology conference on advances in cryptology* (pp. 388–397). Berlin, Heidelberg: Springer-Verlag. Retrieved from `http://dl.acm.org/citation.cfm?id=646764.703989`

Kumar, P., Gurtov, A., Iinatti, J., Ylianttila, M., & Sain, M. (2015). Lightweight and secure session-key establishment scheme in smart home environments. *IEEE Sensors Journal*, *16*(1), 254–264.

Lamport, L. (1994). *Latex: a document preparation system: user's guide and reference manual*. Addison-wesley.

Li, X., Niu, J., Bhuiyan, M. Z. A., Wu, F., Karuppiah, M., & Kumari, S. (2017). A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things. *IEEE Transactions on Industrial Informatics*, *14*(8), 3599–3609.

Light, R. A., et al. (2017). Mosquitto: server and client implementation of the mqtt protocol. *J. Open Source Software*, *2*(13), 265.

Lohachab, A., & Karambir. (2019). Ecc based inter-device authentication and authorization scheme using mqtt for iot networks. *Journal of Information Security and Applications*, *46*, 1 - 12. doi: https://doi.org/10.1016/j.jisa.2019.02.005

Lyu, Q., Zheng, N., Liu, H., Gao, C., Chen, S., & Liu, J. (2019). Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme. *IEEE Access*, *7*, 41835-41851. doi: 10.1109/ACCESS.2019.2907602

Mahmood, K., Chaudhry, S. A., Naqvi, H., Kumari, S., Li, X., & Sangaiah, A. K. (2018). An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Generation Computer Systems*, *81*, 557 - 565. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0167739X17309263` doi: https://doi.org/10.1016/j.future.2017.05.002

Maple, C. (2017). Security and privacy in the internet of things. *Journal of Cyber Policy*, *2*(2), 155–184. Retrieved from `https://www.tandfonline.com/doi/full/10.1080/23738871.2017.1366536` doi: 10.1080/23738871.2017.1366536

Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (1999). Power analysis attacks of modular exponentiation in smartcards. In Ç. K. Koç & C. Paar (Eds.), *Cryptographic hardware and embedded systems* (pp. 144–157). Berlin, Heidelberg: Springer Berlin Heidelberg.

Miller, V. S. (1986). Use of elliptic curves in cryptography. In H. C. Williams (Ed.), *Advances in cryptology — crypto '85 proceedings* (pp. 417–426). Berlin, Heidelberg: Springer Berlin Heidelberg.

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, *10*(7), 1497–1516.

Naoui, S., Elhdhili, M. E., & Saidane, L. A. (2019, Mar 09). Lightweight and secure password based smart home authentication protocol: Lsp-shap. *Journal of Network and Systems Management*. Retrieved from `https://doi.org/10.1007/s10922-019-09496-x` doi: 10.1007/s10922-019-09496-x

Odelu, V., Das, A. K., & Goswami, A. (2015, Sept). A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Transactions on Information Forensics and Security*, *10*(9), 1953-1966. doi: 10.1109/TIFS.2015.2439964

Patel, C., & Doshi, N. (2018). *Internet of things security: challenges, advances, and analytics*. CRC Press.

Patel, C., & Doshi, N. (2019). Security challenges in iot cyber world. In A. E. Hassanien, M. Elhoseny, S. H. Ahmed, & A. K. Singh (Eds.), *Security in smart cities: Models, applications, and challenges* (pp. 171–191). Cham: Springer International Publishing. Retrieved from `https://doi.org/10.1007/978-3-030-01560-2_8` doi: 10.1007/978-3-030-01560-2_8

Poh, G. S., Gope, P., & Ning, J. (2019). Privhome: Privacy-preserving authenticated communication in smart home environment. *IEEE Transactions on Dependable and Secure Computing*.

Qiu, S., Xu, G., Ahmad, H., & Wang, L. (2018). A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems. *IEEE Access*, *6*, 7452-7463. doi: 10.1109/ACCESS.2017.2780124

Radhakrishnan, N., & Karuppiah, M. (2018). An efficient and secure remote user mutual authentication scheme using smart cards for telecare medical information systems. *Informatics in Medicine Unlocked*. Retrieved from `http://www.sciencedirect.com/science/article/pii/S2352914817301697` doi: https://doi.org/10.1016/j.imu.2018.02.003

Rivest, R. L., Shamir, A., & Adleman, L. (1978, February). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, *21*(2), 120–126. Retrieved from `https://doi.org/10.1145/359340.359342` doi: 10.1145/359340.359342

Roman, L. F., Gondim, P. R., & Lloret, J. (2019). Pairing-based authentication protocol for v2g networks in smart grid. *Ad Hoc Networks*, *90*, 101745. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1570870518305997` (Recent advances on security and privacy in Intelligent Transportation Systems) doi: https://doi.org/10.1016/j.adhoc.2018.08.015

Roy, S., Chatterjee, S., Das, A. K., Chattopadhyay, S., Kumari, S., & Jo, M. (2018, Aug). Chaotic map-based anonymous user authentication scheme with

user biometrics and fuzzy extractor for crowdsourcing internet of things. *IEEE Internet of Things Journal*, *5*(4), 2884-2895. doi: 10.1109/JIOT.2017.2714179

Shin, S., & Kwon, T. (2020). A privacy-preserving authentication, authorization, and key agreement scheme for wireless sensor networks in 5g-integrated internet of things. *IEEE Access*, *8*, 67555-67571. doi: 10.1109/ACCESS.2020.2985719

Shuai, M., Xiong, L., Wang, C., & Yu, N. (2020). A secure authentication scheme with forward secrecy for industrial internet of things using rabin cryptosystem. *Computer Communications*, *160*, 215 - 227. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0140366420300736` doi: https://doi.org/10.1016/j.comcom.2020.06.012

Shuai, M., Yu, N., Wang, H., & Xiong, L. (2019). Anonymous authentication scheme for smart home environment with provable security. *Computers and Security*, *86*, 132 - 146. doi: https://doi.org/10.1016/j.cose.2019.06.002

Stallings, W. (2010). *Cryptography and network security: Principles and practice* (5th ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.

Sudhakar, T., Natarajan, V., Gopinath, M., & Saranyadevi, J. (2020, May 26). An enhanced authentication protocol for multi-server environment using password and smart card. *Wireless Personal Communications*. Retrieved from `https://doi.org/10.1007/s11277-020-07462-4` doi: 10.1007/s11277-020-07462-4

Turkanović, M., Brumen, B., & Hölbl, M. (2014). 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*, *20*, 96–112.

Vahedi, E., Bayat, M., Pakravan, M. R., & Aref, M. R. (2017). A secure ecc-based privacy preserving data aggregation scheme for smart grids. *Computer Networks*, *129*, 28–36.

Wang, D., & Wang, P. (2018). Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Transactions on Dependable and Secure Computing*, *15*(4), 708-722.

Wazid, M., Das, A. K., Kumar, N., Vasilakos, A. V., & Rodrigues, J. J. P. C. (2019, April). Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment. *IEEE Internet of Things Journal*, *6*(2), 3572-3584. doi: 10.1109/JIOT.2018.2888821

Wazid, M., Das, A. K., Odelu, V., Kumar, N., Conti, M., & Jo, M. (2018, Feb). Design of secure user authenticated key management protocol for generic iot networks. *IEEE Internet of Things Journal*, *5*(1), 269-282. doi: 10.1109/JIOT .2017.2780232

Wazid, M., Das, A. K., Odelu, V., Kumar, N., & Susilo, W. (2017). Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Transactions on Dependable and Secure Computing*.

Wu, F., Li, X., Xu, L., Vijayakumar, P., & Kumar, N. (2020). A novel three-factor authentication protocol for wireless sensor networks with iot notion. *IEEE Systems Journal*, 1-10.

Wu, F., Xu, L., Kumari, S., Li, X., Shen, J., Choo, K.-K. R., ... Das, A. K. (2017). An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in iot deployment. *Journal of Network and Computer Applications*, *89*, 72 - 85. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1084804516303150` (Emerging Services for Internet of Things (IoT)) doi: https://doi.org/10.1016/j.jnca.2016.12.008

Xu, L. D., He, W., & Li, S. (2014, Nov). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, *10*(4), 2233-2243. doi: 10.1109/TII.2014.2300753

Xu, X., Zhu, P., Wen, Q., Jin, Z., Zhang, H., & He, L. (2013, Nov 21). A secure and efficient authentication and key agreement scheme based on ecc for telecare medicine information systems. *Journal of Medical Systems*, *38*(1), 9994. Retrieved from `https://doi.org/10.1007/s10916-013-9994-8` doi: 10.1007/s10916-013-9994-8

Zhang, L., & Zhu, S. (2015, Mar 03). Robust ecc-based authenticated key agreement scheme with privacy protection for telecare medicine information systems. *Journal of Medical Systems*, *39*(5), 49. Retrieved from `https://doi.org/10.1007/s10916-015-0233-3` doi: 10.1007/s10916-015-0233-3

Zhou, Y., Liu, T., Tang, F., Wang, F., & Tinashe, M. (2019). A privacy-preserving authentication and key agreement scheme with deniability for iot. *Electronics*, *8*(4). Retrieved from `https://www.mdpi.com/2079-9292/8/4/450` doi: 10.3390/electronics8040450