Design of Some Identity- and Lattice-based Public Key Cryptosystem for Blockchain-Assisted IoT Applications

Thesis submitted in partial fulfilment
for the Award of Degree

Doctor of Philosophy

by

POOJA VERMA

RAJIV GANDHI INSTITUTE OF PETROLEUM TECHNOLOGY
JAIS - 229304

Roll No: 21CS0004

2026

CERTIFICATE

It is certified that the work contained in the thesis titled " **Design of Some Identity- and Lattice-based Public Key Cryptosystem for Blockchain-Assisted IoT Applications**" by "**Ms. Pooja Verma**" has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

It is further certified that the student has fulfilled all the requirements of Comprehensive, Candidacy, and SOTA.

Dr. Kalka Dubey
(Supervisor)

Dr. Daya Sagar Gupta
(External Co-Supervisor)

DECLARATION BY THE CANDIDATE

I, **"Pooja Verma"**, certify that the work embodied in this thesis is my own bona fide work and carried out by me under the supervision of **"Dr. Kalka Dubey"** and **"Dr. Daya Sagar Gupta"** from **"August 2021"** to **"January 2026"**, at the Rajiv Gandhi Institute of Petroleum Technology, Jais(India).

The matter embodied in this thesis has not been submitted for the award of any other degree. I declare that I have faithfully acknowledged and given credits to the research workers wherever their works have been cited in my work in this thesis. I further declare that I have not willfully copied any other's work, paragraphs, text, data, results, etc., reported in journals, books, magazines, reports, dissertations, theses, etc., or available at websites, and have not included them in this thesis and have not cited them as my own work.

Date: **(Pooja Verma)**

Place: Roll no. 21CS0004

CERTIFICATE BY THE SUPERVISOR

It is certified that the above statement made by the student is correct to the best of our knowledge.

(Dr. Kalka Dubey)
Supervisor
RGIPT, Jais

(Dr. Daya Sagar Gupta)
External Co-Supervisor
IIT Patna

Signature & Seal of Head of Department

CERTIFICATE

Certified that the work contained in the thesis titled **"Design of Some Identity- and Lattice-based Public Key Cryptosystem for Blockchain-Assisted IoT Applications "** by **"Ms. Pooja Verma"** has been carried out under our supervision. It is also certified that she fulfilled the mandatory requirement of two quality publications arose out of her thesis work.

It is further certified that the two publications (copies enclosed) of the aforesaid **"Ms. Pooja Verma"** have been published in the Journals indexed by –

(a) SCI

(b) SCI Extended

(c) SCOPUS

(Dr. Kalka Dubey)	(Dr. Daya Sagar Gupta)	(Dr. Vivek Singh Baghel)
Supervisor	External Co-Supervisor	Convener, DPGC
RGIPT, JAIS	IIT PATNA	RGIPT, JAIS

COPYRIGHT TRANSFER CERTIFICATE

Title of the Thesis: Design of Some Identity- and Lattice-Based Public Key Cryptosystem for Blockchain-Assisted IoT Applications.

Name of the Student: Ms. Pooja Verma

Copyright Transfer

The undersigned hereby assigns to the Rajiv Gandhi Institute of Petroleum Technology, Jais, all rights under copyright that may exist in and for the above thesis submitted for the award of the "DOCTOR OF PHILOSOPHY".

Date: Pooja Verma

Place: Roll no. 21CS0004

Note: However, the author may reproduce or authorize others to reproduce material extracted verbatim from the thesis or a derivative of the thesis for the author's personal use, provided that the source and the Institute's copyright notice are indicated.

Acknowledgment

First and foremost, I acknowledge Bholenath, my almighty, for his blessings and for giving me the strength, courage, wisdom, and moral support during the good and hard times of my Ph.D. work. This thesis has been the culmination of the research work done in the Department of Computer Science & Engineering, Rajiv Gandhi Institute of Petroleum Technology, Jais. I must acknowledge and express my sincere gratitude to the RGIPT Jais for allowing me to pursue Ph.D.

I am thankful to Dr. Kalka Dubey, Assistant Professor, RGIPT Jias, and Dr. Daya Sagar Gupta, Assistant Professor, IIT Patna, for giving me the opportunity to carry out my research work under their supervision and for their constant support and encouragement. Their scientific acumen has made them a continuous source of ideas that have inspired and enriched my growth as a researcher. I am indebted to them for their faith in me and for providing the right direction whenever I needed it the most. Their kind nature and affectionate souls have made me eligible to learn about their scholarly treasure. They have been supportive and have given me the freedom to pursue my ideas without objection. Their unstinting kindness, affectionate guidance, and noble generosity have created a lifelong impression on me.

I would like to express my gratitude to my Doctoral RPEC Committee members (Dr. Gargi Srivastava, Dr. Alpesh Kumar, and Dr. Vijay Kumar Singh) for the constructive comments, guidance, and feedback they provided. I would especially mention Dr. Gargi Srivastava, who led my Doctoral Scrutiny Committee during my Ph.D. I am profoundly thankful to her for her assistance throughout the process.

I am thankful to the Director, Dean (R & D), and Head of the Department of Computer Science & Engineering for their support and for providing me with all the necessary facilities to fulfill my research work. I am also thankful to all my faculty colleagues for their moral support.

I am deeply grateful to Prof. Rakesh Kumar for his unwavering support and invaluable guidance during my M.Tech days. His mentorship during my M. Tech

thesis laid a strong foundation, shaping the research skills that helped me throughout my Ph.D.

I am deeply indebted to my laboratory colleagues Dr. Hema Shekhawat, Mr. Arpit Kumar, Mr. Hrituraj, Mr. Ankit, and Dr. Lachhita Soni for their unceasing encouragement, support, and attention.

My family has been an invaluable source of encouragement and support throughout the completion of this thesis. I am deeply grateful to my husband (Mr. Aris Chaudhary), my parents (Mr. Jagdish Prasad Verma and Mrs. Vimla Devi), my in-laws (Mr. Rajesh Chaudhary and Mrs. Shakuntala) and my sister (Shashi), and my brothers (Sandeep and Pradeep) for their constant belief in me and for allowing me to pursue my academic aspirations with unwavering patience and understanding. They have been steadfast pillars of strength during this journey.

I also extend my heartfelt gratitude to my daughter Avantika Chaudhary (Laddoo), the most precious blessing of my Ph.D. journey. Her presence has been a constant source of motivation and strength, inspiring me to persevere and strive beyond my limits. I sincerely hope to honor the many sacrifices made by my family in some meaningful way.

Finally, I would like to thank God for letting me through all the difficulties. I have experienced your guidance day by day. You are the one who let me finish my Ph.D. work. I will keep on trusting you for my future. I appreciate everyone whom I have thanklessly missed to remember and who has contributed towards the completion of the work.

Date: **(Pooja Verma)**

Place: Jais

Contents

List of Figures

List of Tables

List of Acronyms

CDH	Computational Diffie Hellman
IBC	Identity-Based Cryptography
IBS	Identity-Based Signature
ECC	Elliptic Curve Cryptography
CRL	Certificate Revocation List
LBC	Lattice Based Cryptography
CS	Cloud Server
DC	Distribution Center
DL	Discrete Logarithm
DLT	Distributed Ledger Technology
ECDDH	Elliptic Curve Decisional Diffie–Hellman
ECDH	Elliptic Curve Diffie–Hellman
ECDLP	Elliptic Curve Discrete Logarithm Algorithm
ICT	Information and Communication Technology
KGC	Key Generation Center
PBFT	Practical Byzantine Fault Tolerance
PKI	Public Key Infrastructure
PPT	Probabilistic Polynomial Time
TA	Trusted Authority
TPS	Transactions Per Second
IoT	Internet of Things
IHS	Intelligent Healthcare System
WBAN	Wireless Body Area Network
HIA	Health Information Accumulator
PHR	Patient Health Records
EHR	Electronic Health records
ITS	Intelligent Transportation System
VANET	Vehicular Adhoc Network

V2V	Vehicle-To-Vehicle
V2I	Vehicle-To-Infrastructure
RSU	Road Side Units
OBU	On Board Units
UAV	Unmanned Aerial Vehicle
KGC	Key Generation center
CL-PKC	Certificateless Public key Cryptography
RA	Registration Authority
ROM	Random Oracle Model
CVP	Closet Vector Problem
SVP	Shortest Vector Problem
SIS	Short Integer Solutions
ISIS	Inhomogeneous SIS
P2M	People-to-Machine
SOS	Shared Operational Systems
M2M	Machine-to-Machine
MITM	Man-in-the-Middle
BCT	Blockchain Technology
SC	Smart Contract
P2P	Peer-to-peer
SC	Smart Contract
BFT	Byzantine Fault Tolerance
DBFT	Delegated Byzantine Fault Tolerance
FBA	Federated Byzantine Agreement
DAG	Directed Acyclic Graph
POX	proof of Somethings
PoW	Proof of Work
PoI	Proof of Identity
PoS	Proof of Stake
DPoS	Delegated Proof of Stake
PoA	Proof of Activity
PoAu	Proof of Authority
PoB	Proof of Burn
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
POI	Proof of Importance
DDoS	Distributed Denial of Services

Abstract

The explosion of Internet of Things (IoT) devices has revolutionized our digital world, connecting countless resource-limited gadgets in key areas like smart and intelligent healthcare, intelligent transportation, and factory automation systems. These setups enable seamless real-time data gathering, automation, and smart decisions, but they also open the door to major security and privacy risks—from exposed wireless links and scattered networks to weak device power and highly sensitive information. Old-school centralized security setups just can't keep up, failing to deliver reliable trust, toughness, or privacy when data hops between edge gadgets, fog layers, clouds, and outside services. On top of that, quantum computers on the horizon could crack classic encryption like elliptic curves and pairings, so we need quantum-proof alternatives now.

This work tackles these issues head-on with a set of lightweight, secure, quantum-resistant crypto protocols built for blockchain-boosted IoT systems. We trace the shift in public-key cryptosystem: from identity-based schemes that ease key handling but risk escrow problems, to certificateless ones that ditch escrow and certs, and onward to lattice-based methods that stand firm against classical and quantum foes. Blockchain steps in to provide decentralized trust, unchangeable records, easy audits, and solid data origins, fixing the flaws of central IoT hubs and boosting overall strength.

Our main goals are to propose efficient schemes for key agreement, authentication, and privacy protection that lock down confidentiality, integrity, authenticity, anonymity, and unlinkability in varied IoT settings. We roll out four fresh schemes and put them through rigorous tests.

First, A pairing-free identity-based setup for healthcare data authentication and aggregation, letting patients, aggregators, and servers mutually verify each other securely with minimal computation and communication. It guards data wholeness, secrecy, and user privacy and blocks fakes like impersonation, replays, or MITM attacks for massive health data flows. Further, an upgraded certificateless protocol is introduced for mutual authentication and key sharing in cloud-linked wireless

body area networks, fixing flaws in prior work via formal proofs in the random oracle model. No escrow risks, no key swaps or impersonation, plus top-tier features like forward secrecy, key isolation, and two-way checks. Third, we introduce a quantum-safe blockchain authentication system using UAV for secure V2V and vehicle-to-roadside communication in intelligent transport, with lattices for post-quantum safety and blockchain for open trust. Security analysis is performed using the quantum random oracle model, which thwarts quantum hacks, fake signatures, impersonations, and privacy leaks. Next, a lightweight, quantum-hard blockchain authentication for smart healthcare, blending lattice key agreement with blind signatures for private, anonymous, checkable transactions. It nails session keys, unforgeable blinds, user hiding, and lasting logs, even against quantum threats.

Formal proofs using ROM and QROM, adversary models, other computational hard problems, and evaluations affirm robust security with minimal overheads in computation, communication, energy, and storage—tailored for constrained IoT. This framework synergizes IBC, CLC, LBC, and blockchain, yielding scalable, decentralized, quantum defenses for next-generation IoT applications. The implementations have been carried out using Hyperledger Fabric and cryptography libraries to evaluate the performance in terms of computational and communication costs and blockchain throughput and latency. The comparisons with other state-of-the-art schemes show the efficiency and feasibility of the proposed schemes.

Keywords: Internet of Things, Intelligent healthcare System, Intelligent Transportation System, Blockchain Technology, Identity-based Cryptography, Certificateless public key cryptography, Lattice-Based Cryptography, Confidentiality, Authentication, Integrity & Confidentiality, Authenticated Key Agreement, Quantum-safe.

CHAPTER 1

Introduction

1.1 Internet of Things

The Internet of Things (IoT) has developed into a widespread networking architecture wherein ordinary physical things are outfitted with sensors, embedded computers, and network connections in order to autonomously gather, exchange, and respond to data with minimal human intervention. The sensors and computing power built into IoT devices make widespread deployment possible in various environments. IoT implementations currently encompass intelligent healthcare, industrial automation, smart grids, smart cities, and intelligent transportation systems, wherein several heterogeneous devices collaborate to facilitate real-time monitoring, control, and data-driven decision-making.

According to Statista, an online platform specializing in market and consumer data, the number of IoT-connected devices is projected to reach 30.9 billion units worldwide by 2025, up from 13.8 billion in 2021. Furthermore, McKinsey Global Institute's forecasts indicate that the applications of IoT in various fields will generate an economic impact ranging from USD 3.9 trillion to 11.1 trillion per year on a global scale [1, 2]. The increasing use of IoT devices creates a lot of data, which requires a lot of computing power, storage space, and communication capacity. IoT devices are usually unable to complete all of the tasks on their own because they often have limited computing, storage, and networking capabilities and rely on batteries for power. Therefore, to make IoT devices work efficiently, they need help from more powerful resources. One common solution is to use Cloud Computing resources. However, the resources of cloud computing are usually found in only a

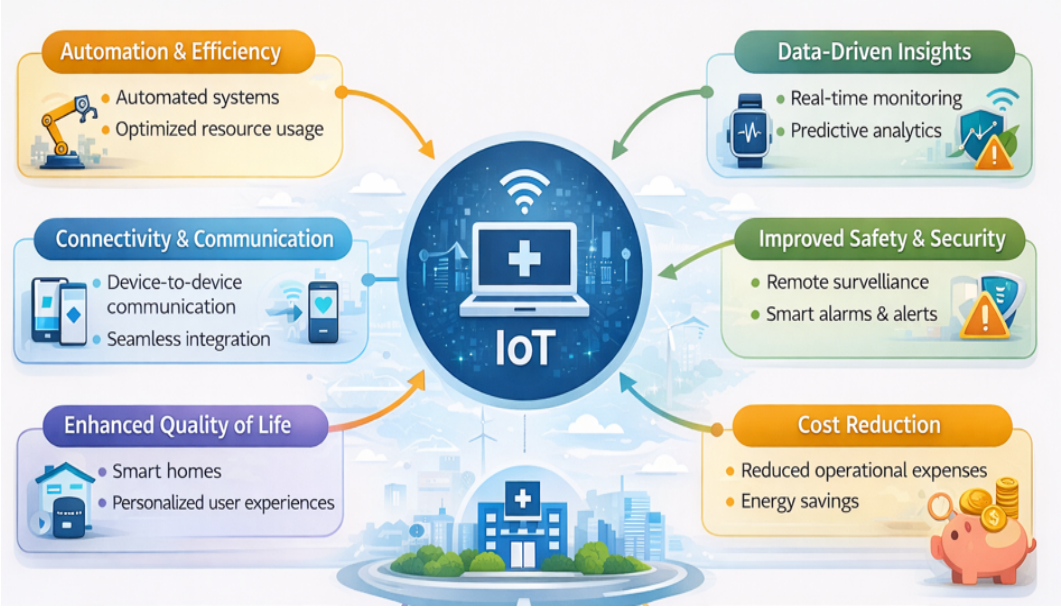


Figure 1.1: Benefits of IoT Technology.

few Data Centers (DCs), which are located far away from data producers and consumers. This significant distance between end devices and data centers can cause problems that cannot be tolerated for many applications and services. Therefore, cloud computing-based IoT system models are inefficient.

This widespread connectivity converts raw sensor data into useful knowledge while simultaneously raising the risk perimeter, particularly whenever sensitive information is transmitted over wireless connections, several administrative levels, and complex networks. In a standard IoT deployment, information typically moves through several distinct layers, beginning with highly constrained sensors and actuators at the network edge, passing through local gateways or fog/edge nodes, and finally reaching remote cloud platforms that handle large-scale analytics and long-term storage. Each layer relies on its own set of communication protocols, processing capabilities, and trust assumptions, which makes it difficult to maintain uniform security guarantees from the device up to the cloud. IoT has numerous benefits, such as automation, data-driven insights, improved safety, cost reduction, and improved connection, as well as enhanced quality of life, which are highlighted by Figure 1.1. It is challenging to preserve security assurances consistent from the device to the cloud since each tier has a unique set of communication techniques, computational potential, and reliability presumptions.

The aforementioned systems' devices have very inadequate resources, such as low power consumption, compact storage space dimensions, restricted space, petite battery life, and very limited channels for communication. These limits make it hard

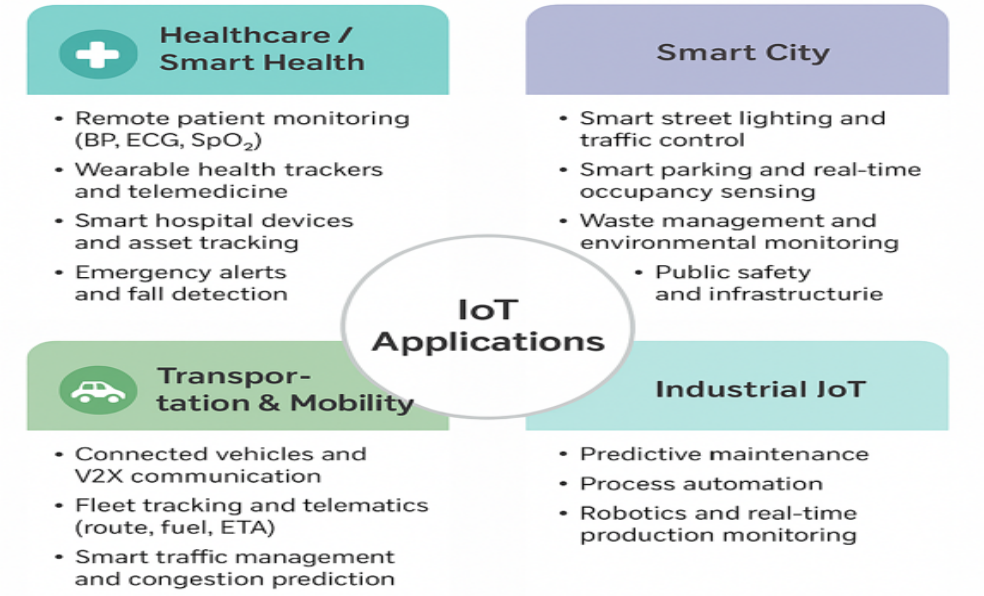


Figure 1.2: Classification of IoT-based Applications.

to use hefty cryptographic algorithms, complicated key-exchange methods, or multi-step security protocols without putting too much strain on devices or cutting their lifespan short. So, designers have to carefully identify and adjust security features so that they work with the resources of each device.

Numerous IoT applications, on the other hand, suffer from stringent criteria for latency and reliability. Healthcare monitoring platforms need to be able to quickly respond to unusual vital signs, and integrated or autonomous transportation systems are required to distribute warnings about hazards to people early enough to avoid accidents or traffic jams. In these time-sensitive situations, if every piece of data has to be transferred to a cloud server far away to undergo storage and assurance of security, the interruptions and possible transmission loss can quickly go beyond what is tolerable.

This condition puts robust defense and accurate execution at odds with each other. As an instance, the wide attack emerges, and the sensitive information that is being gathered means that every tier of the architecture needs strong guarantees of authentication, confidentiality, integrity, and availability. On the other hand, all security approaches must be lightweight enough to work on limited computing resources, integrate into a variety of networking techniques, and maintain according to rigorous latency and performance constraints. This means that scenarios involving IoT need to be designed with cryptographic techniques and lightweight protocols.

Figure 1.3: Benefits of Intelligent Healthcare System.

1.1.1 Classification of IoT-Based Applications

The applications of IoT can be classified into the following major areas based on the application area or industry and the type of devices or sensors used, which are represented by Figure 1.2:

1. **Intelligent Healthcare System:** The healthcare landscape is undergoing a transformative shift with the advent of the Internet of Things (IoT), introducing the Intelligent Healthcare System (IHS). The concept of the IoT in the healthcare sector encompasses validation, automated data compilation, and analysis. Intelligent healthcare systems enabled by IoT primarily handle sensitive patient information and medical findings. This system extends its services ubiquitously, offering accessibility to individuals anywhere and at any time through interconnected medical sensors [3]. The global embrace of medical sensors is manifest as their deployment surges in response to the escalating number of patients worldwide [4]. These sensors, strategically placed either on or within the body, assume distinct roles in collecting health monitoring data such as glucose levels, blood pressure, ECG readings, etc.

   Patients, leveraging these advancements, can seamlessly share their health histories and real-time records with healthcare professionals, eliminating the need for physical visits or hospital meetings. This facilitates more informed medical services and advice, enabling healthcare teams to administer precise treatments based on the received sensitive health records [5]. The Intelligent

Figure 1.4: Benefits of Intelligent Transportation System.

Healthcare System brings about a positive transformation in healthcare services, allowing professionals to intermittently monitor and diagnose a patient's health status, leading to timely and improved treatments. All the important benefits are highlighted by Figure1.3. In contrast to traditional healthcare systems, the adoption of Intelligent Healthcare Systems significantly reduces costs, expedites treatment delivery, and enhances overall quality [6]. This paradigm shift marks a profound evolution in healthcare, emphasizing efficiency, accessibility, and improved patient outcomes.

Protecting healthcare records and preventing their disclosure is essential to avoiding unethical or unpleasant occurrences [7]. The healthcare sector faced numerous challenges with specific requirements, including authentication, interoperability, secure data sharing, seamless transfer of medical records, and considerations for mobile health.

2. **Intelligent Transportation Systems:** The Intelligent Transportation System (ITS) refers to the application of advanced information and communication technologies to enhance the efficiency, safety, sustainability, and overall performance of transportation networks. As urban populations continue to grow and traffic congestion intensifies, traditional traffic management methods are becoming increasingly inadequate. ITS offers a transformative solution by integrating real-time data collection, analysis, and decision-making into the transportation infrastructure. One of the fundamental aspects of ITS is real-time traffic monitoring. All the important points of the ITS infrastructure are

represented by Figure 1.4.

Sensors embedded in roads, along with surveillance cameras and vehicle telematics, provide constant updates on traffic flow, congestion levels, and incidents. This data is processed in centralized traffic management centers, which use algorithms to optimize traffic signals, reroute traffic, and inform road users through dynamic message signs or mobile applications. As a result, travel times are reduced, fuel consumption is minimized, and emissions are decreased, contributing to both economic and environmental benefits.

Additionally, ITS is essential for facilitating autonomous driving as well as intelligent vehicles. Vehicles are capable of sharing data regarding position, speeds, and traffic conditions with one another, utilizing Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) networking. As connected and autonomous vehicles (CAVs) become increasingly prevalent, the need for fast, reliable, and secure communication frameworks has grown substantially. Data sharing regarding the traffic networks among the vehicles is possible with the help of On Board Unit (OBU) and Road Side Units (RSU), which are mainly used to collect, analyze and share the data as per requests made by vehicles. To enable V2V and V2I communication, RSUs are placed beside roads.

A robust transportation system is essential for creating infrastructure and making the economy stronger. ITS is the use of modern information, communication, sensing, and control technologies in transportation infrastructure and vehicles to make transportation systems more secure, effective, and trustworthy, as well as environmentally conscious.

1.1.2 Why Security is Essential in the IoT ?

One of the primary explanations for why security is vital with IoT is that data transfer can be public and distributed widely. The majority of IoT devices use mobile communications over public or semi-public internet connections, which makes whatever information they provide vulnerable to threats, including spying, spoofing, replay attacks, and intrusion incidents. Adversaries may retrieve or alter highly confidential information involving personal health data and location-related data and execute operations if they are not properly encrypted and authenticated. An additional major concern is that IoT devices don't have an adequate quantity of resources. Most IoT-connected devices do not possess a lot of computational power, storage space or energy to operate, which makes it hard to use standard comprehensive safety measures.

Figure 1.5: Several Security Attacks in IoT.

Security in IoT is also important for making sure that data is reliable and secure. Continuous information is very important for IoT systems to make decisions and run autonomously. If bad people change or add to details, it might cause error-prone choices, system breakdowns, or worse, personal harm. This is especially pertinent in highly hazardous areas like health care surveillance and transport systems, where improper information may turn catastrophic.

Although IoT presents numerous advantages to humans through various application areas, such as Intelligent Transportation Systems (ITS), Smart Healthcare, Smart Homes, etc. However, as IoT-based environments rely on wireless technologies for data transfer, this poses significant security challenges, mainly due to the vulnerability of wireless channels to data interception and alteration.

In recent years, IoT applications have faced many security attacks. In December 2013, security experts from a company called Proofpoint found the first-ever IoT botnet. They discovered that more than one-fourth of the botnet included appliances like smart TVs and baby monitors, not just computers. More recently, a company called Dyn, which helps with domain names, had trouble with its services because of a coordinated cyberattack.

Also, protecting individual privacy is an important part of IoT security. IoT devices constantly gather a lot of sensitive and relevant facts, which is a big privacy concern. To stop unwanted access as well as improper use of information about users, there must be robust access control, authentication, and security-preserving systems.

Table 1.1: Mapping of IoT Challenges to Security Goals and Solutions

IoT Challenge	**Security Goal**	**Possible Solutions**
Device Security	Authentication, Integrity	Lightweight authentication protocols, ECC-based schemes, lattice-based authentication, secure boot mechanisms
Data Privacy	Confidentiality, Privacy	End-to-end encryption, lightweight cryptography, data anonymization, access control mechanisms
Scalability	Availability, Secure Key Management	Hierarchical key management, certificateless cryptography, group key management schemes
Energy Constraints	Lightweight Security, Efficiency	Lightweight cryptographic primitives, symmetric-key encryption, optimized key exchange protocols
Network Vulnerabilities	Integrity, Availability	Secure routing protocols, intrusion detection systems (IDS), anti-jamming techniques
Data Management and Storage	Integrity, Confidentiality	Encrypted cloud storage, access control policies, secure data aggregation
Latency and Reliability	Availability, Timeliness	Edge/fog computing, low-latency secure communication protocols
Physical Security	Integrity, Confidentiality	Tamper-resistant hardware, secure key storage, physical access control

The major attacks on IoT can be categorized in the following way: (a) Physical Attacks (tampering, side-channel attack), (b) Network Attacks (Eavesdropping, traffic analysis), (c) Malware Attacks (botnets, malicious firmware), (d) Spoofing Attacks (Fake id, FPS signal, Address spoofing), (e) MITM Attacks (Data alteration, DNS spoofing), (f) Social Engineering Attacks (Phishing, Deceptive Pretexting). This is demonstrated by Figure 1.5. Therefore, the privacy and security of IoT data and their communications remain prime concerns for researchers. All the challenges related to IoT are described in Table 1.1 along with targeting security goals and respective solutions.

1.1.3 Introduction to Blockchain Technology and Its Integration with IoT

Blockchain is a series of data blocks maintained by a distributed network of peers. These data blocks are connected through a hash in such a way that a single bit change in one block reflects in other subsequent blocks. Moreover, data blocks contain a list of transactions, and any new transaction can be added to the block only if it is validated by all the participating peers. The blocks are added to the chain continuously at certain intervals, and this chain is replicated at other participating nodes on the network. The blocks may contain a nonce, timestamp, Merkle hash tree, smart contract, etc. The integrity is ensured by a hash and a Merkle tree. The Merkle tree allows quick search and verification of data. Therefore, it is suitable for lightweight devices. Blockchain is categorized into four types based on access requirements, which are demonstrated by Figure 1.6.

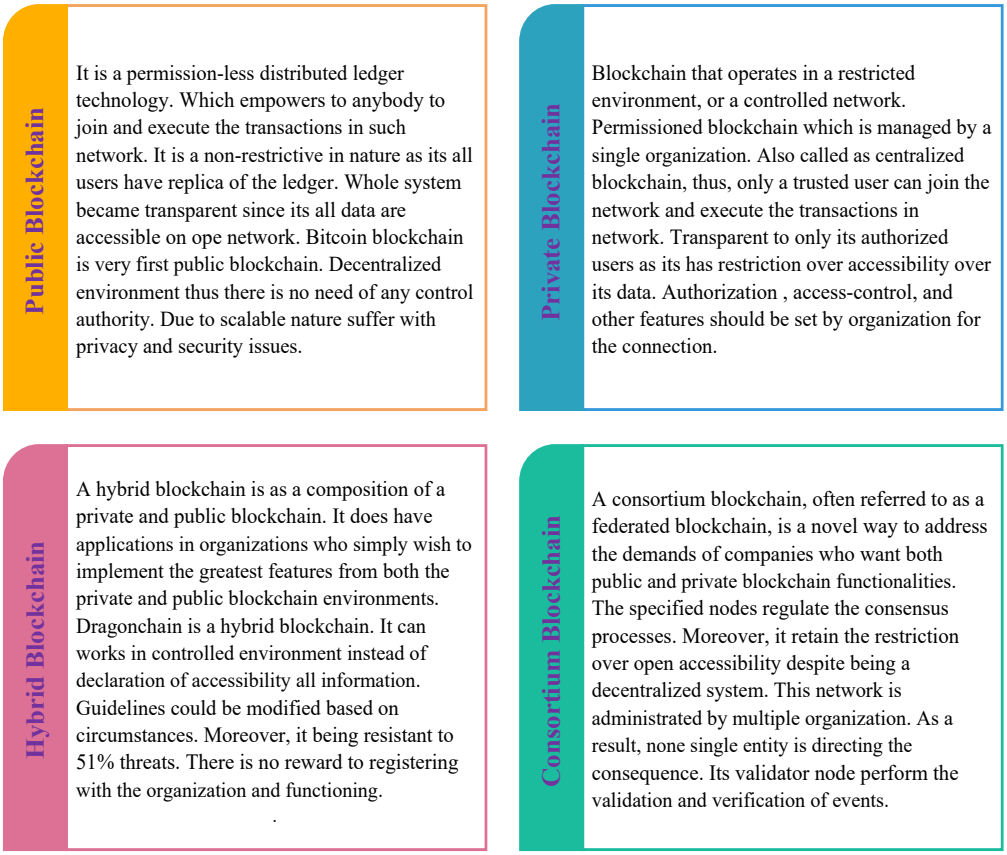


Figure 1.6: Categories of Blockchain.

Figure 1.7 highlights the challenges associated with IoT infrastructure and how these are resolved by combining the concept of blockchain. Thus, this integrated system improved the data privacy and security concerns in internet-connected communication units. The integration of blockchain with IoT brings some additional features:

i. **Improved Trust and Security**: The distributed and immutable nature of blockchain would eliminate a single point of failure/vulnerability for attackers/hackers. All transactions are cryptographically signed using unforgeable signatures, making them non-repudiable and resistant to attacks.

ii. **More Robust**: A distributed environment will make IoT more accessible, and damage costs from hacks can be more easily prevented or avoided altogether. Intermediaries that operate for centralized IoT systems will be eliminated through decentralizing IoT, thereby reducing the associated costs.

iii. **Autonomy**: Blockchains enable smart devices to act independently according to the pre-determined logic (using smart contracts). This would completely

Figure 1.7: Importance of Integrating IoT with Blockchain Technology.

remove intermediary players and central authority.

iv. **High Transparency**: The use of efficient smart contracts for communication in an IoT environment, along with the decentralization offered by blockchain, makes the entire system more trustworthy.

v. **Data provenance**: Since all transactions are recorded on the ledger and signed by the devices/entities generating data, data provenance can be achieved.

However, IoT devices have limited processing and storage power, which makes it hard for them to operate the blockchain ledger on their own. This means that IoT devices can't be nodes or peers in the blockchain network. Cloud servers can be used as blockchain nodes to handle the blockchain ledger. This enables the use of blockchain in the IoT context. From an operational perspective, blockchain makes systems more fault-tolerant and available while eliminating the need for centralized servers. Blockchain-based IoT solutions may organize data securely, efficiently, and in a way that can grow once utilized together with cloud computing and lightweight cryptography. In general, blockchain is a basic technology that can be used to make IoT ecosystems that are safe, reliable, and ready for the future.

1.2 Motivation and Objectives

Cryptographic security mechanisms are essential for ensuring the security of Internet of Things (IoT) devices and communications. Numerous cryptographic security

schemes have been presented in the literature to achieve authentication, privacy, integrity, and confidentiality. Authentication can be achieved by authenticated key agreement schemes, signature schemes, and encryption schemes. Integrity can be ensured by signature schemes and encryption schemes. Confidentiality can be achieved in signature, encryption, and key agreement schemes.

- Traditional security schemes proposed in the literature face many challenges;

    1. Conventional Public Key Infrastructure (PKI) relies heavily on digital certificates and trusted Certificate Authorities (CAs).
    2. large-scale IoT environments, certificate generation, verification, storage, and revocation introduce significant computational, communication, and management overhead.
    3. To eliminate certificate overhead, Identity-Based Cryptography (IBC) was introduced, where public keys are derived directly from user identities and managed by a Private Key Generator (PKG).
    4. Certificateless Cryptography (CLC) emerges as a balanced solution by removing certificates while avoiding key escrow.
    5. With the advent of quantum computing, IoT systems face long-term security risks. Lattice-Based Cryptography (LBC), grounded, offers quantum-resistant security, efficient key operations, and suitability for lightweight implementations.

    Despite strong cryptographic foundations, IoT systems still suffer from issues such as centralized trust, data tampering, lack of transparency, and insecure access control. Cryptographic security schemes need to rely on a trusted third party, such as a certificate authorization center, authentication server, key generation center, and so on.

- This dependency causes the following drawbacks:

    1. The over-dependence on a single trusted authority creates a central point of failure, which can cause the entire system to become paralyzed if that authority fails.
    2. The single trusted authority may not be efficient in handling massive amounts of data generated by some IoT applications, such as IoV, IoMT, etc.

3. The traceability and accountability of data stored in centralized servers may not be guaranteed due to the possibility of tampering or deletion of the data.
4. A centralized server can impede the growth of IoT due to its inability to efficiently handle a vast amount of end-to-end communications. This inefficiency can lead to delays, performance bottlenecks, and even system crashes, ultimately limiting the scalability and reliability of the IoT network.

Blockchain eliminates single points of failure, enables trustless authentication, supports secure key updates and access control, and ensures data integrity across distributed IoT nodes. Blockchain, as a new distributed system technology, coincides with the distributed characteristics of the IoT, which provides a new way to solve the security problems of the IoT. It eliminates single points of failure, enables trustless authentication, supports secure key updates and access control, and ensures data integrity across distributed IoT nodes.

- To address the security and efficiency challenges discussed above, the following objectives are defined:

1. **Objective 1:** Designing secure and efficient cryptographic security schemes based on cryptographic primitives such as authenticated key agreement, signature, and homomorphic encryption to ensure the authenticity, confidentiality, and integrity in the IoT applications.
2. **Objective 2:** Designing suitable cryptographic security schemes according to the needs and architecture of the underlying IoT applications.
3. **Objective 3:** Design a distributed architecture by integrating blockchain technology into the underlying IoT application. To address the challenges of centralized architectures of IoT-based applications, such as intelligent healthcare and intelligent transportation systems.
4. **Objective 4:** Present the concrete mathematical proofs for the soundness, completeness, and computational intractability of the proposed schemes against some severe attacks.
5. **Objective 5:** Analyzing the feasibility of the blockchain operations in the proposed schemes for the underlying IoT applications using the consortium blockchain platform Hyperledger Fabric.

1.3 Research Contributions

This thesis concentrates on tackling significant security issues in IoT-based application areas, specifically highlighting Intelligent Transportation Systems (ITS) and Intelligent Healthcare Systems (IHS). The research seeks to develop robust, effective, and privacy-preserving procedures that guarantee authentication, data integrity, and confidentiality in extensive, diverse internet of things (IoT) systems.

The thesis suggests safe ways for vehicles to talk to each other (V2V) and to infrastructure (V2I) as part of Intelligent Transportation Systems. These methods are meant to keep users' privacy safe while also protecting against impersonation, message tampering, and replay assaults. The proposed methods improve trust and dependability in safety-critical transportation services by allowing vehicles and roadside equipment to use strong authentication and secure key establishment.

This thesis presents a safe and distributed healthcare architecture that employs blockchain technology inside interconnected hospital systems, specifically throughout the framework of Intelligent Healthcare Systems. The suggested structure guarantees secure preservation and clear communication of confidential medical information among those with authorization. To make data security even stronger, a decentralized signcryption technique is being developed to protect patient data transmission by providing secrecy, authentication, and integrity all at the same time. This method makes it easier to handle keys while still making sure that data can be shared safely and privately in healthcare settings with limited resources.

Overall, this thesis makes IoT security better by offering application-specific, scalable, and trust-enhanced solutions for ITS and IHS. The suggested methods work together to safeguard privacy, keep data safe, and allow for secure interaction. This makes them a good fit for the forthcoming generations of smart transportation and healthcare infrastructures.

1.3.1 A Pairing-Free Data Authentication and Aggregation Mechanism for IHS.

This contribution proposed a scheme that integrates the aggregator with an authenticated key agreement mechanism to create a reliable connection between a patient and the medical server, ensuring that both parties are thoroughly checked before sharing critical information. Managing all the health-related information over the internet becomes a challenging chore as numerous people rely on it implicitly or explicitly. The framework is built to stop prevalent threats like man-in-the-

middle, impersonation, and replay, which makes online medical care safer. Pairing-free identity-based cryptography is used to make key management easier and cut down on the amount of computing power required. An aggregator node is used to reduce latency and make it possible to collect data on a large scale. An integrated key agreement phase makes sure that medical data stays private while it is being sent. Provable security has been provided with the Random Oracle model (ROM), which proves the robustness of the presented scheme. The performance analysis through MIRACL shows that the scheme presented outperforms the similar prior mechanisms in terms of computational-communicational cost and energy overheads.

1.3.2 An Improved Certificateless Mutual Authentication and Key Agreement for Cloud-Assisted WBAN

This contribution incorporates WBAN infrastructure with cloud computing facilities. The emergence of certificateless cryptography has been a promising solution to address the issues of certificate management and revocation in public key cryptography. A session key can be established between two participating nodes using the AKA protocol, which gives both nodes confidence in the legitimacy of their counterparts. Our study demonstrates that existing certificateless-based AKA schemes are vulnerable to attacks such as man-in-the-middle and simple impersonation attacks by the key generation center (KGC) itself. we identified critical flaws and limitations in Cheng et al.'s [8] scheme. To address these, we propose an enhanced certificateless security protocol that eliminates key escrow and certificate management issues. Formal security analysis in the Random Oracle Model (ROM) confirms secure mutual authentication between patients and medical servers via session keys. The scheme resists man-in-the-middle, impersonation, forward secrecy violations, replay attacks, and others. The suggested scheme's security depends on the intractability of the Elliptic-Curve-Discrete-Logarithm problem (ECDH) in the ECC group and the Computational Diffie-Hellman (CDH) problem.

1.3.3 Quantum-Safe UAV-Assisted Blockchain Authentication for Secure Vehicle Communications in ITS

This contribution integrates the blockchain technology with the ITS communication network, where each vehicle wants to communicate securely with others, can access the real-time information about road traffic, etc., with the help of RSU and UAV. Blockchain is utilized as a distributed ledger to facilitate a secure and distributed transportation system. The proposed architecture can eliminate the risks associ-

ated with a single point of failure and provide secure distributed database access and storage among multiple vehicles. Moreover, this approach enables seamless authentication of vehicle, UAV when they move from one location to another, ensuring that their shared road traffic information is accessible and protected throughout the network. To ensure authentication and confidentiality in the proposed ITS architecture, a novel quantum-safe, UAV-assisted, blockchain-enabled, authenticated key agreement protocol is proposed for Intelligent Transportation System (ITS) environments. It employs lattice-based cryptography to ensure resistance against both classical and quantum adversaries and uses blockchain for robust, privacy-preserving identity management. UAVs are integrated to reduce latency and deployment costs compared to purely RSU-based infrastructures. The protocol supports conditional privacy, allowing vehicles and UAVs to be authenticated without disclosing real identities while still enabling misbehavior traceability. It ensures message integrity, mutual authentication, and resistance to replay, impersonation, and data tampering attacks. A formal security analysis in the Quantum Random Oracle Model demonstrates strong security against quantum-capable adversaries, and experimental results confirm its scalability and higher efficiency than existing ECC- and lattice-based ITS schemes. We found positive, more efficient results, as it reduces the latency 16.8% and 37.8% in comparison to ECC- and LBC-based schemes, respectively. It is also efficient in terms of communication and computation costs, along with storage requirements. Experimental results demonstrate a 2.1% failure rate within PBFT $f < 1/3$ tolerance, outperforming ECC/LBC baselines by 28% in mobility scenarios. This scalable solution enables secure, real-time vehicle communications essential for safety-critical ITS applications.

1.3.4 Quantum-Resistant Anonymous Authentication for Blockchain Enabled Smart Healthcare

This contribution integrates an authentic blockchain with the smart healthcare system. A quantum-resistant lightweight mutual authentication protocol is designed to secure communication between patients and the medical server in a smart healthcare environment. The protocol incorporates a blind signature mechanism so that collector points can be validated on the blockchain, enabling other nodes to verify that the received medical data is authentic while preserving confidentiality. The Quantum Random Oracle Model is used to analyze the authentication and key agreement phases, ensuring quantum-safe mutual authentication and secure session key establishment. Furthermore, the medical server is implemented over a blockchain-based architecture to enhance security, privacy, and transparency, while removing

single-point-of-failure issues and achieving low computational and communication overhead.

1.4 Thesis Organization

The remaining part of the thesis is organized as follows: Chapter two provides a detailed review and comparative analysis of existing cryptographic schemes for IoT applications, such as intelligent transportation systems, and Intelligent Healthcare systems. Chapter three describes some basic preliminary knowledge regarding the cryptographic security schemes proposed in the subsequent chapters. Chapter four introduces the first contribution in which a pairing-free authenticated key agreement scheme has been developed, along with the aggregation for the intelligent healthcare system using identity-based cryptography. Moreover, Chapter Five first performs the cryptanalysis on Cheng's scheme before introducing an improved certificate-less mutual authenticated scheme for the cloud-assisted WBAN model. In chapter six, blockchain is integrated into the multiple RA environment and introduces a quantum-safe authenticated key agreement approach for the intelligent transportation system for secure V2V communication. Chapter seven presented a quantum-safe authenticated key agreement scheme in its first phase. Then, to verify the authenticity of nodes belonging to the blockchain, we proposed a blind signature-based mechanism for a smart healthcare system. Facilitating authenticated and confidential data transfer between smart medical sensor devices and the medical servers. Lastly, chapter eight serves as a conclusion to the thesis and suggests some future research directions. Figure 1.8 illustrates the complete block diagram of the thesis organization.

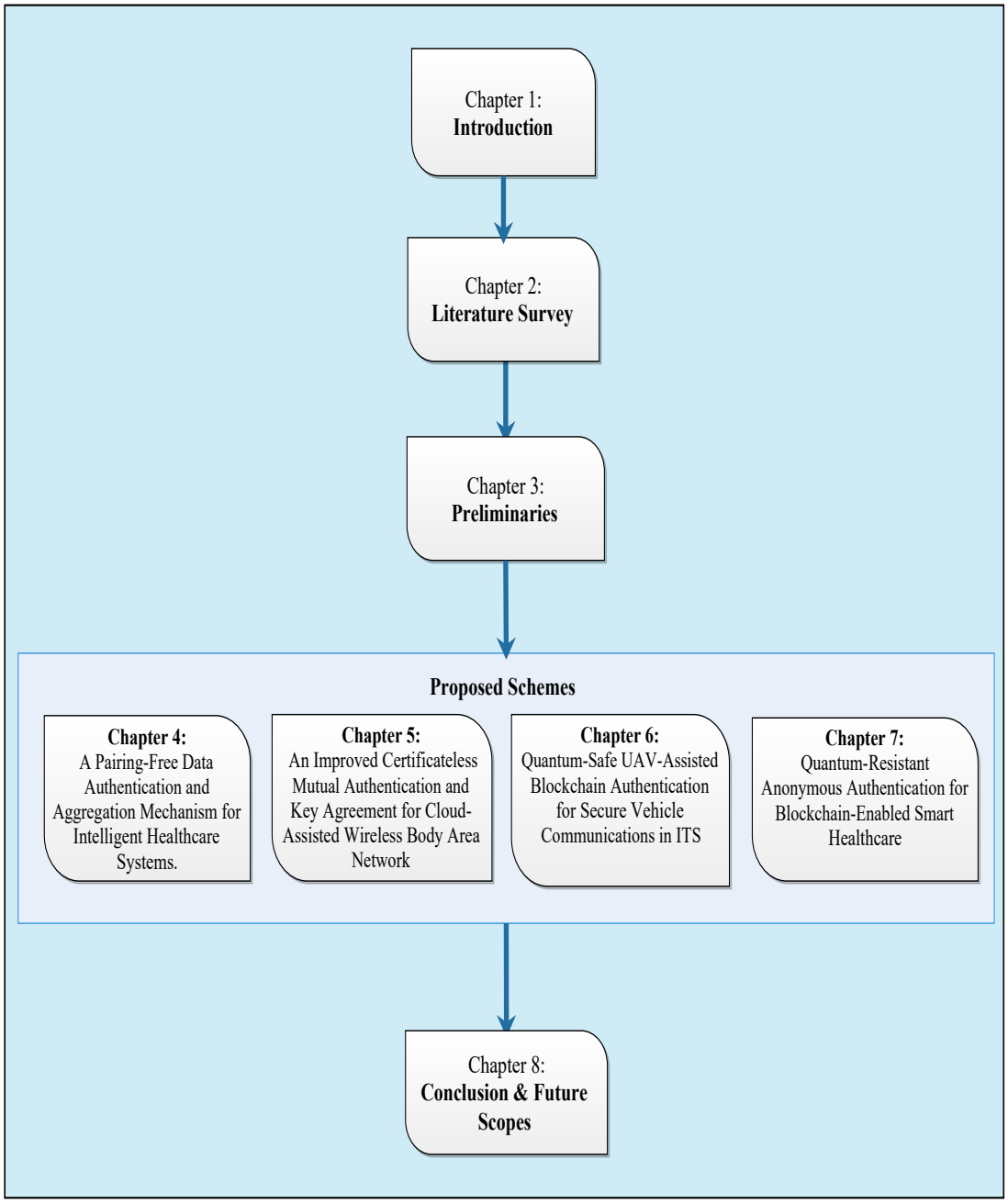


Figure 1.8: Organization of the Thesis.

CHAPTER 2

Literature Survey

2.1 Introduction

Over the past few years, there has been a significant amount of effort put into addressing the security vulnerabilities in IoT environments. IoT environments have several crucial security requirements, like any other information system, including authentication, anonymity, integrity, and confidentiality. Researchers have proposed various signature schemes, mutual authentication, and key agreement schemes to meet these requirements. However, existing schemes face several challenges, such as high computational and communication costs, key escrow problems, and vulnerability to security attacks. In the last few years, scholars have begun to investigate the potential of blockchain technology in addressing security vulnerabilities within IoT environments. The integration of blockchain into the existing infrastructure of IoT environments, such as edge/cloud nodes, makes it possible to address issues related to centralization and a single point of failure. This chapter provides an overview of the existing cryptographic techniques and discusses the various security schemes currently in use in IoT applications, such as intelligent healthcare systems, intelligent transportation systems, and smart Healthcare system.

In the literature, researchers have employed various cryptographic techniques to design cryptographic security mechanisms. In 2008, Raya and Haubax [9] presented a Public Key Infrastructure PKI-based authentication system for vehicular networks, which utilized anonymous certificates to enable both authenticity and anonymity. Devices are provided with public and private key pairs to facilitate anonymous communication in PKI-based schemes. Public key certificates are uti-

lized as a secure and trustworthy mechanism to authenticate a device that contains the device's public key and the certification authority's (CA) digital signature for authentication purposes. The devices generate a signature by using a secret key and short-term pseudonyms. The sender attaches the certificate and the signature to the message, and the receiver can verify the certificate-based signature without disclosing the sender's true identity. CAs are responsible for certificate management and issuance. However, the certificates are huge in size. Therefore, PKI-based schemes are inefficient due to the overhead of certificate management.

Identity-based cryptography (IBC) can be a useful alternative for the design of security schemes, as it has the potential to avoid certificate management and lower communication costs. In 2008, Heo et al. [10] utilized IBC for device authentication and key distribution in IoT. With IBC, the need for public key certificates is eliminated, which simplifies the deployment and management of security credentials. IBC makes use of an individual's identity or other easily remembered information to generate private keys. Unlike traditional public key cryptography, where users must create and distribute their own public keys, IBC utilizes a "private key generator" (PKG), a trusted third party that creates private keys for each user based on their identity or an identifier. This simplifies the key management process but also means that the security of the entire system is reliant on the PKG's trustworthiness. In comparison to traditional PKI, IBC removes the need for certificates when verifying public keys. This eliminates the necessity of distributing public keys along with their corresponding certificates. Additionally, IBC avoids the burden of managing CRLs, which can be quite resource-intensive in PKI-based schemes. However, in IBC, the key escrow problem is a major concern because all private keys are generated by a PKG, meaning that the PKG possesses knowledge of each vehicle's private key in VANETs.

Al-Riyami and Paterson [11] presented Certificateless Cryptography (CLC) intending to eliminate problems associated with PKI-based and IBC-based schemes. It has many advantages over PKI and ID-based schemes. Unlike PKI, CLS-based schemes do not need any certificate to ensure the validity of the public key. Therefore, the absence of certificates reduces communication, computation, and storage overhead. In CLS-based schemes, the Key Generation Center (KGC) acts as a semi-trusted authority. It computes a partial private key using the user's identity and gives it to the user. The user then combines its secret with the partial private key to obtain the complete private key. The Key Generation Center (KGC) in CLS does not possess any access to the user's private key, eliminating concerns about the key escrow problem. As a result, numerous certificateless cryptography-based schemes

have emerged in recent years.

The mathematical hard problems in the area of lattice were firstly initiated by M. Ajtai [12] in 1996 and attracted because of a connection between the worst-case and average-case of these problems. These hard problems play a very important role in the construction of various cryptographic primitives. In a combined work, Ajtai and Dwork [13,14] proposed a probabilistic PKE scheme (known as AD cryptosystem) based on unique shortest vector problem (u-SVP) on lattice problems. They showed that their cryptosystem is secure unless the worst-case of u-SVP can be solved in polynomial time. Goldreich, Goldwasser and Halevi [15] also published a lattice-based cryptosystem known to be GGH cryptosystem. In their work, they familiarized a new trapdoor one-way function which is based on the difficulty of finding the closest vector close to the given vector in a lattice space. Using this trapdoor function, they presented a PKE and a signature scheme. Their proposal did not come with a security proof, but it was very efficient in contrast to the [13,14].

This thesis employs identity-based, certificateless, and also lattice-based cryptography for designing signature and key agreement protocols for blockchain-enabled IoT-based applications.

2.2 Pairing Free Data Authentication Aggregation Mechanism for Intelligent Healthcare System

This thesis presents an authentication and aggregation mechanism for intelligent healthcare systems using identity-based cryptography without pairing. The literature of approach has been delineated.

Many authenticated key agreement schemes have surfaced in the last few years to deal with the privacy and security problems in the IHS. As we know, the online tracking and monitoring of patients' health conditions has become possible and successful with the support of IoT facilities, which has grabbed the full attention of stakeholders. Utilizing the medical sensors in many countries is becoming increasingly expanded to their limits as the number of patients continues growing [8]. Hence, emerging technology in computers and networking technology is evolving towards smart sensors with the emergence of the use of smart devices of the IHS [16]. To mitigate the key escrow & certificate storage problem, Shamir [17] propounded its novel idea of ID-based cryptosystem (IBC) where a user's identity can be used as a public key. But later on, it is found that the protocol based on the computational

cost of bilinear pairing is so high [18]. Thus, to address these issues pairing-free-based protocol [19] [20] [21] [22] was announced. Some of them still do not provide mutual authentication and authorization properly, and some of them also fail to transfer data efficiently over open channels. As lots of security issues like confidentiality, integrity, and authentication regarding patient data. To tackle these issues, various schemes are being reviewed as follows.

Das et al. [23] suggested an authentication scheme for remote users with the health server. The computational cost is so high as to not be focused on the categorization of the collected health data. Authors Othman et al. [24] and Gupta et al., [25], [20] found that each smart device faced the challenges regarding energy consumption, computation, and short storage capacity. Thus, the presented homomorphic encryption method enables aggregators to prevent attacks in regard to confidentiality and integrity while saving energy. Therefore, data aggregation is being used to reduce energy consumption by smart devices while the information travels over networks [24]. But they also mentioned that an aggregator is not safe to maintain the confidentiality and integrity of sensitive information of patients.

In 2017, Islam et al [21] presented an approach without the use of pairing operations for secure and efficient communication purposes. They demonstrated the security of AKA protocol attacks using formal security analysis and BAN logic. In 2018, Kaur et al. [26] introduced an authentication scheme for IoT-based healthcare systems using factors: password, biometric, and smart card. Generate a session key between the medical profession (MP) and the cloud server (CS), which preserves mutual authentication. Over the AVISPA tool, they performed the formal security analysis against several attacks. Dang et al. [27] present a pairing-free key agreement scheme with one round of communication. However, formal analysis schemes claim to prevent attacks like IA, replay, MA, and a few more.

Table 2.1 represents the countermeasures against security attacks of several above-discussed schemes, focusing on providing secure communication and generating authenticated key agreement between patients and servers through aggregators based on the priority for the Intelligent Healthcare Environments. Also, the table shows the advantages of these existing schemes, including their limitations.

In 2019, the author Nassoro et al. [31] present a scheme for single-server and multi-server environments. It works on 3-factor authentication, involving passwords, smart cards, and biometric data. It uses symmetric and asymmetric encryption for single-server and multi-server architectures, respectively, to reduce the computational cost. Comprehensive security analysis shows that it is reliable through mutual authentication. Time cost analysis also shows less time required to complete the au-

Table 2.1: Represents Comparison Among Existing Pairing-Free AKA Schemes with Their Limitations.

Cites	Year	Formal Model	IAR	ESR	MA	PGR	Advantages	Limitations
[23]	2013	DYT	No	No	Yes	No	-Design a Authenticated agreement protocol .	-Failed to prevent the impersonation attacks. -Privacy and authentication security goals compromised.
[28]	2015	NA	No	No	Yes	No	- Design Anonymous robust authentication scheme.	-Possibility of the Impersonation and Password Guess attack.
[29]	2015	BGN	Yes	Yes	No	No	-Record the patient's health data into two categories such as: Spatial and Temporal. - Designed two: multifunctional aggregation MHDA+, MHDA$\oplus$ function.	- Failed to provide the Mutual Authentication between the nodes. -An Intruder can guess the password to interrupt the communication.
[21]	2017	BAN	Yes	Yes	Yes	Yes	- Design a robust secure authentication protocol. - Maintain the data security and privacy.	- Very high computational cost - High Energy overhead.
[27]	2018	eCK	No	Yes	No	No	Designed an agreement protocol. -ID-based cryptography used.	- Failed to maintain the authenticity. -Intruder can impersonate the participating nodes.
[30]	2018	BAN	Yes	Yes	Yes	No	- Robust secure authenticated scheme designed.	- Password guessing attacks possibility.
[31]	2019	mBR	No	Yes	No	Yes	- Design a user authentication scheme for multi server.	- Not mutually authenticated nodes participate.
[32]	2020	ROM	Yes	Yes	Yes	Yes	- Lightweight authenticated agreement scheme designed. -Provide robust, secure communication.	-Energy overhead at the agreement phase.
[19]	2021	NA	No	Yes	Yes	Yes	-An Efficient agreement protocol is designed. -Maintain the data confidentiality and Privacy.	- An Intruder can impersonate the participating nodes.
[8]	2021	NA	No	Yes	No	Yes	-Key escrow issue resolved. - A certificatless approach.	- Can impersonate the allied parties. - Possibility of Men-in-Middle attacks.

IAR* Impersonation attack resist, MA* Mutual attack, ESR* Ephemeral secret resist, PGR* Password Guessing resist, eCK* extended Cantti-Krawcryk, BAN* Burrow-Abadi-Needham, mBR* modified Bellare Rogaway, DYT* Dolevo-Yao threat model, BGN* Boneh-Goh-Nissim model, ROM* Random Oracle Model, NA* Not Available.

thentication process. Deebak et al. [7] focused on providing secure communication and preserving the privacy of users during communication, along with focusing on reducing the time for both parties- the doctor and patients.

Subsequently, in 2020, Masud et al. [33] proposed an AKP for handling the data confidentiality and authentication between Doctors and COVID-19 patients. Physical Unclonable Function (PUF) is used to verify the doctor' and patient's authenticity before creating a session between them. This protocol is prone to a few attacks, like man-in-the-middle, impersonation, etc. Maria et al. [34]introduced a scheme based on edge computing along with dual signature techniques that aimed to preserve the privacy of patients' records in an IHS environment. The Computational cost is also very low as they applied the elliptic curve digital signature authentication techniques.

In the year 2021, Mamdiwar et al. [35], reviewed Wearable sensors in IoT-assisted E-healthcare and focused on recent improvements. They rigorously analyzed and presented several IoT-based designs and questions for data processing. Also mentioned lots of security flaws, designing flaws faced by schemes, which helps researchers to propose a scheme that provides security aspects with less computational and communicational cost. Further, Sowjanya et al. [36] observed that there is a requirement for an enhanced and lightweight agreement protocol to provide secure communication between patients and servers. Thus, they propound an ECC-based authenticated scheme. Jiliang et al. [37] mainly aimed at an optimal solution for computation, communication cost, and space. Thus, they presented a novel work that is a lightweight mutually authenticated agreement protocol that preserves the secrecy of transmitted records. Moreover, Hailong et al. [38] introduced a novel work based on a Symmetric-key-based authenticated agreement scheme. As they found flaws like imperfect forward secrecy, etc., in a few existing symmetric-based schemes, their scheme provides perfect secrecy of keys. Aljumaie et al. [16] presented a comprehensive study over several existing schemes based on the IHS environment. In which situation, what kind of attacks are possible is studied very well by them. They also presented a very helpful study of lots of recent protocols that provide authentication, confidentiality, and privacy as well. Zhang et al. [39] have suggested a lightweight privacy-preserving secure protocol for the IHS environment through the use of an aggregator. They focused on the few main security goals like preserving data integrity, confidentiality. This scheme successfully prevents communication by a few attacks, such as MA, IA, and key escrow. This is a certificateless approach. Authors in the paper [22] mainly focused on the reduction of the latency that occurs between the communicating nodes. The protocol was implemented by using the

ASCII hexadecimal ECC methods. They also proved in regards of how suggested protocol is better than a few existing schemes in terms of computational cost and latency.

In the present year 2022, author Othman et al. [40] introduced a privacy-preserving scheme based on the observation of high computational and energy consumption during the aggregation process. To reduce such issues and preserve the few security goals, like as: maintain the privacy and integrity of patients' data (by utilizing the homomorphic encryption concept), verification of participating (patients or medical staff),f) etc. They utilised green computing for IoT-based healthcare. Cheng et al. [41] aimed to maintain the privacy of biomedical sensors used in smart healthcare systems. By utilizing the homomorphic signature concept, they designed a certificateless secure protocol, which prevents the key escrow issue as well as achieves other security goals as confidentiality and authentication.

Author Hurtado et al. [42] introduced a protocol that is focused on the remote health monitoring system using sensors. By applying the convolution neural network method author analyzed the patients' movement and categorized the analyzed data according to semi-supervised techniques. Chakraborty et al. [43] observed that handling of IoT-based generated health records becomes a challenging task, especially in terms of latency. Thus, they introduced a novel technique with the aim of reducing the latency by applying fog computing and machine learning for IoT-based health records. Chaudhary et al. [44] suggest a technique to identify the doppelganger attack, i.e., a type of impersonation attack for IHS where all the sensors come under a high-risk zone.

2.3 Improved Certificateless Authentication and Key Agreement Protocols for Cloud-Assisted WBAN

Certificateless authenticated key-agreement protocols are widely used in Wireless Body Area Networks (WBANs) because they strike a balance between strong security and the limited power resources of wearable or implantable sensors. Since WBAN devices operate very close to the human body and constantly exchange medical data, each authentication step needs to be lightweight and must guarantee that both ends genuinely know who they are talking to. Most of these schemes rely on identity-based features so that sensors don't have to store bulky certificates, and on ECC (Elliptic Curve Cryptography) because it provides good security with very

small key sizes—essential in energy-constrained environments.

Across the literature, researchers typically aim to solve one or more recurring challenges: keeping authentication fast, ensuring mutual trust between devices, resisting impersonation and replay attacks, avoiding heavy computations like pairings, and preserving the privacy of the user's identity. Although all approaches try to keep the communication overhead low, each design takes a slightly different angle—some try to maximize energy efficiency, some focus on anonymity, others enhance forward secrecy or reduce key-management complexity. Shamir [45] introduced the novel concept of an ID-based cryptosystem (IBC), from which a participant's identity could be utilized as a public key, to alleviate the issue with the key escrow and certificate management. However, it was later discovered that such an approach depended on the huge computational cost of bilinear pairing [18]. Therefore, pairing-free-based techniques have been introduced to deal with such issues. Islam et al. [46] introduced an authentication protocol for wireless networks, which is mainly based on ID-based cryptography. They concentrated on pairing-free operation due to the high computational cost of bilinear pairing. He et al. [47] recommended a protocol for two-party communication, in which an authenticated key is generated. They provide the formal security analysis using the mBR model for the robustness security of the proposed protocol. Sun et al. [48] found that in [47], designing a flaw, an attacker would determine the session key between the two parties, as it could successfully calculate the secret key.

Kim et al. [49] introduced an authenticated agreement scheme. The author generates the session key by applying the concept of a certificateless cryptosystem, and they also proved their scheme using the eCK model. However, Tu et al. [50] analyzed the above scheme [49] and found that it failed against the impersonation attacks, further introducing an improved version of it. Xiong et al. [51] proposed a scheme based on the anonymous behavior of participants inside the network. Thus, an attacker could not find any information about the sender or receiver. Wang et al. [52], based on the investigation of Truong et al. [53] works, authors searched that there is the possibility of impersonation, known as session key attacks. Thus, they proposed an improved protocol using IBC to resist such attacks. Author Gupta et al. [54], introduced work with the aim to eliminate the computational cost during the key exchange process by using the KGC. They achieved their goals by applying the elliptic curve cryptography and justifying it using the provable random oracle model (ROM). Verma et. al [55] suggested a pairing-free operation base authentication key agreement protocol using IDB cryptography for the intelligent Healthcare environment. An aggregator is being used to reduce the computational cost as well

Table 2.2: Comparison of Certificateless-Based Authenticated Key Agreement Schemes for WBAN

Scheme	MA	Cryptogr--aphy Used	Formal Analysis	Main Goal	Advantages	Limitations
ECC-based CL-AKA	Yes	ECC	BAN Logic, AVISPA	Lightweight mutual an authentication	Low overhead, energy-friendly	PKG compromise risk
Pairing-based CL-AKA	Yes	Pairings + ECC	BAN Logic	High security strength	Strong features	Too heavy for sensors
Hybrid Hash + ECC CL-AKA	Yes	Hash + ECC	ROR, ProVerif	Reduce computation cost	Very efficient	Needs careful curve choice
Non-pairing ECC CL-AKA	Yes	ECC	AVISPA, BAN	Remove pairing operations	Sensor-friendly	Fewer advanced features
Symmetric-assisted CL-AKA	Partial	Hash + CL keys	Informal + BAN	Extremely low computation	Fast, minimal footprint	Lacks full mutual auth
Energy-aware ECC CL-AKA	Yes	ECC + MAC	BAN + Simulation	Save energy in WBAN nodes	Highly energy-efficient	May weaken security
ID-based Lightweight ECC CL-AKA	Yes	ID-based CL + ECC	BAN Logic	Faster identity mapping	Simple and light	PKG leakage risk
ECC CL-AKA with Hash Acceleration	Yes	ECC + ID-hash	ProVerif	Reduce ECC multiplications	Efficient	Limited privacy
ID-ECC Mutual Authentication	Yes	ID-ECC	ROR Model	Secure patient-device binding	Strong MITM resistance	Scalability issues
Ultra-light ECC CL-AKA for Implants	Yes	ECC + hashing	Informal + BAN	Reduce sensor computation	Ultra-lightweight	PFS may be weak
Timestamp-based ECC CL-AKA	Yes	ECC + timestamps	AVISPA	Prevent replay attacks	Strong replay protection	Time sync required
Dual-Identity ECC CL-AKA	Yes	ECC + dual ID	BAN Logic	Strengthen identity binding	Prevents impersonation	Higher setup complexity
ID-assisted ECC Medical Monitoring CL-AKA	Yes	ID-ECC + hash chains	ROR + AVISPA	Real-time secure data sharing	Replay/MITM resistant	Partial identity exposure
ID-Obfuscated ECC CL-AKA	Yes	ECC + pseudo-ID	AVISPA	Maintain identity privacy	Anonymous authentication	Pseudo-ID update overhead
Energy-Minimal ID-ECC CL-AKA	Yes	ECC + ID-hash	BAN Logic	Reduce computation energy	Very low cost	Weak PFS unless enhanced
ID-derived Ephemeral ECC CL-AKA	Yes	ECC + ID-derived keys	ROR Model	Improve forward secrecy	Fresh keys each session	Extra ephemeral key cost

MA: Mutual Authentication, AKA: Authenticated Key Agreement, ECC: Elliptic Curve Cryptography, ID-based: identity-based.

as storage overhead.

The following table 2.2 represents all of these ideas together in a clean, consolidated view. Overall, It is observed that challenges related to certificate management are not tackled efficiently. Thus, a certificate-less-based mechanism had been introduced in which a participant can generate its own final key pair instead of only depending on the key generated by the key generator center (KGC) [56]. Shen et al. [57], introduced a certificateless-based scheme using cloud computing to reduce computational and storage issues. In this scheme no one can track the identity of users; therefore, the privacy of participant's privacy should be preserved in WBAN. Amin et al. [30], observed the overload issue for single stations. Thus, multi-layer communication is utilized. The author introduced a novel scheme for the user's authentication scheme for multilayer-based communication. Using the BAN logic, they demonstrated the security and privacy of their proposed protocol. Abiramy et al. [58] designed a multi-level authentication scheme based on elliptic curve cryptography in WBAN. In the first level, create an authenticated session between the sensors and remote device aimed to hide the trackability, and second, generate an authenticated session between the remote device and service provider by utilizing the certificateless concept. Kapito et al. [59] introduced an authentication protocol using machine-based authentication mechanisms for IoT networks. In this scheme, they select the user's identity dynamically and use a machine for biometric authentication purposes, therefore they successfully hide the identity of the user and maintain data privacy.

Numerous research works based on mutual authentication key agreement have been studied. We observed the requirements of a provable security analysis model to deploy such authenticated key agreement schemes in practical use. Early, the Bellare and Rogaway (BR) model [60] and its extension (mBR) [61] failed to inspect the scheme against a few attacks like perfect forward secrecy (PFS), key compromise impersonation (KCI), and ephemeral key leakage (EKL). After that, the CK model [62] was put forth, allowing the attacker to request state data for sessions beyond the test stage. However, the CK model is unable to deal with KCI and EKL robustness. To assure the feature of optimal exposure resistance, Lamacchia et al. [63] presented the extended CK model (eCK), which enables an opponent to request the non-trivial pair of fixed and ephemeral secret keys, irrespective of the testing session. It was verified by Yoneyama et al. [64] that it was difficult to accomplish provable security in the strengthened eCK model. Furthermore, the random oracle model (ROM) [65] is introduced to ensure the robustness and security of the protocol. Using ROM, the formal security analysis of schemes based on authentication key agreement (AKA)

against attackers like impersonation, man-in-the-middle attacks, etc., is performed using a few oracle query sessions.

2.4 A Quantum-Safe UAV-Assisted Blockchain Authentication for Secure Vehicle Communications in Intelligent Transportation Systems.

This thesis focuses on addressing the security concerns related to V2V (vehicle-to-vehicle) communications within vehicular networks. As a solution, a quantum-safe authentication system based on blockchain has been introduced to ensure secure V2V communications. Here, the literature review related to the proposed scheme is described.

Table 2.3 represents the comparison of the existing AKA-based scheme for ITS environments along with their advantages and limitations. To provide security for the transmitted information in the communicating network, numerous cryptography-based algorithms are used, like ECC [82] [54] [83] [84] [85] , IDC with pairing and pairing-free [3] [86], and LBC [87] [88] [89] [90] [91]. Initially, researchers' schemes suffered from a key escrow issue due to the management of certificates. To alleviate these issues associated with key escrow and certificate storage, Shamir [17] introduced the innovative concept of identity-based cryptography (IBC), wherein a user's identity serves as a public key. Although it offers advantages, IBC often requires bilinear pairings, which are costly to compute and expensive for real-time ITS scenarios. To address this problem, pairing-free IBC has been introduced. It is also somehow unable to offer secure key agreement or strong mutual authentication. The Certificateless Public Key Cryptosystem (CL-PKC) [92] was created to help with this problem. In CL-PKC, a semi-trusted Key Generation Center (KGC) gives the user a partial private key. The user then makes the rest of their private key and their public key on their own.

Authors Du et al. [93] introduced an identity-based AKA mechanism for Vehicle Ad-hoc Network (VANET). To prevent traffic jams, accidents, and procurement, they provide the V2V communication without the involvement of the RSU. Thus, the computational and communicational overhead will be reduced as there is minimal delay in response. Authors Lee et. al [94] introduced an AKA scheme for a vehicular cloud computing system (VCC). The suggested protocol improves the integrity, secrecy, and credibility of messages in VCC systems by employing identity-based key agreement. It is a viable approach that doesn't require additional overhead. Lu Wei

Table 2.3: Comparison of Prior Authentication Key Agreement Schemes for Intelligent Transportation Environments.

Scheme	**Aims**	**Cryptogr--aphy Used**	**Form--al SA**	**AKA**	**MA**	**BCT Used**	**Advantages with Limitations**
[66]	Secure V2V authentication	ECC	✓	✓	✓	×	Low delay, scalable; lacks quantum resistance.
[67]	Anonymous authentication	Lattice-based	✓	✓	✓	×	Quantum-resistant, but high computational overhead.
[68]	Privacy-preserving communication	ZKP + Hash	✓	×	✓	×	Strong privacy, moderate latency; complex implementation.
[69]	Blockchain-based trust management	Hash + Blockchain	×	×	✓	✓	Decentralized, tamper-resistant; blockchain overhead impacts latency.
[70]	Lightweight mutual authentication	Hash + XOR	✓	×	✓	×	Efficient, suitable for low-power devices; limited security features.
[71]	Post-quantum secure communications	Lattice-based (Kyber)	✓	✓	✓	×	Strong security; signature size and computation cost high.
[72]	Multi-factor authentication	ECC + Biometric	✓	✓	✓	×	Enhanced security; increased computational and communication cost.
[73]	Edge-assisted authentication	ECC + Blockchain	✓	✓	✓	✓	Efficient and decentralized; needs edge infrastructure support.
[74]	Quantum communication protocol	Quantum cryptography	⊗	✓	✓	×	Unconditional security; requires quantum channel and hardware.
[75]	Batch authentication	Group signatures	✓	×	✓	×	Scalable; signature verification overhead can be high.
[76]	Adaptive privacy control	ZKP + Blockchain	✓	✓	✓	✓	Strong privacy, decentralized; blockchain-induced latency.
[77]	Efficient V2I authentication	ECC	✓	✓	✓	×	Good tradeoff performance-security; assumes trusted CA.
[78]	Resource-constrained device compatibility	Hash + Symmetric keys	×	×	✓	×	Very lightweight; limited forward secrecy.
[79]	Sybil attack resistance	Hash + Blockchain	✓	×	✓	✓	Effective attack mitigation; blockchain delay overhead.
[80]	Decentralized trust	Blockchain only	×	×	⊕	✓	Full decentralization; throughput limited by consensus protocol.
[81]	Secure V2V, V2I communication	Tree Based	✓	✓	✓	×	Mutually authenticate; Not scalable, failed to prevent the past forward secrecy of key and real identity.

MA: Mutual Authentication, BCT: Blockchain Technology, AKA: Authenticated Key Agreement, SA: Security Analysis, ECC: Elliptic Curve Cryptography, ZKP: Zero-Knowledge Proof, CA: Certificate Authority, LBC: Lattice-based Cryptography, V2V: Vehicle to Vehicle communication, V2I: Vehicle to Infrastructure Communication, ⊗: Theoretical, ⊕: Partial, ✓: Yes, ×: No

et al. [81] proposed a mechanism to facilitate the V2I and V2V communication. Tree tree-based mechanism is used to verify the authenticity of a vehicle during joining and leaving the system. The whole scheme is divided into two parts first provides the authentication among vehicles, RSU, and TA. The second part introduced a tree-based mechanism for an authorized vehicle to join or leave the channel. They mainly focused on using the XOR operation to make it lightweight and efficient.

Tan et al. [71] introduced a Certificateless-based group signature scheme where UAVs and vehicles can check the authenticity of each other. In this author creates a mobile base by using UAVs, which reduces the cost of active edge computing of IoV infrastructure. Authors Zhang et al. [69] proposed a solution for secure vehicle auxiliary communication with UAV. They also archived the identity anonymity, thus the other party cannot gain information about the requester and responder identity. Gupta et al. [95] in 2016 present a cryptanalysis of a lattice-based scheme. Subsequently, they also proposed a novel work [96] using LBC to provide data security and privacy. Gupta et al. [32] also introduced an LBC-based access control scheme for IoT environments. Gupta et al. [91] present a post-quantum-based AKA scheme for IoV, where OBU and RSU check the authenticity of each other before starting the communication and construct the key exchange. Gupta et al. [97] also introduced a quantum-defended authentication scheme using blockchain technology. In this way, they maintain more robustness and transparency over transactions in the network as compared to their previous works. This scheme only focused on authentication, not on key agreement to encrypt the information between OBU and RSU. Verma et al. [87] introduced an AKA based scheme for P2P communication that aims to maintain data privacy and secrecy. They used the smart contract concept for registration purposes.

Mishra et al. [89] introduced a quantum-safe mechanism for the IoV network to provide secure communication between the RSU and the Vehicle. The major drawbacks of this scheme are that it consumes more storage space and is expensive in terms of communication as well as computational costs. Also failed to maintain the transparency and decentralized communication network. Author Mundhe et al. [88] introduced a LBC-based scheme for VANET using a ring signature. They provide the authentication between the RSU and the vehicle. This scheme is expensive due to the more and more RSU station installation requirements.

Over the past few years, researchers have explored the potential of leveraging the distributive nature of blockchain technology to develop authentication schemes for ITS with purpose to provide the secure communication between the vehicles. In 2019, Lu et al., [98] introduced a novel blockchain-based authentication scheme for

VANET. They extended the standard blockchain structure by using the Merkle Patricia tree to provide distributed authentication with no CRL in VANET. However, due to the usage of multiple certificates per vehicle, their scheme incurs the requirement of higher computational and storage power. Later, Zheng et al. [99] provided an authentication scheme with anonymous access based on blockchain, though it is unsafe against a compromised certificate authority [100]. Feng et al. in [101], proposed an ABE (Attribute-Based Encryption) based authentication scheme with blockchain assistance. They used a consortium blockchain, mainly managed by TA and blockchain managers, where TA is provided with all access rights, and blockchain managers have only READ permission. Nevertheless, their scheme does not support batch message authentication. Lin et al., [100] integrated a key derivation algorithm with public blockchain to achieve efficient certificate management in public key infrastructure; on this basis, they proposed a PKI-based authentication scheme with privacy preservation. Xu et al. [102] utilized multiple TAs and blockchain to create a multi-TA environment and proposed an authentication and key agreement scheme for the created environment. All TAs manage the common ledger and store vehicle-related information to achieve efficient and cross-TA authentication. Wang et al. [103] introduced an authentication scheme for Vehicle-to-Infrastructure (V2I) communication, utilizing bilinear pairing and incorporating scalable trust computation through blockchain technology. Ren et al., [104] presented a certificateless signature scheme using blockchain technology for VANET. They employed two Merkle tree structures for the management of vehicle pseudo-identities. However, they have not mentioned the type of blockchain, nor have they analyzed the performance of blockchain. Additionally, their scheme is bilinear pairing-based, which causes a higher computational cost. Bagga et al. [105] recently proposed a blockchain-assisted authentication scheme that supports both V2V authentication and batch authentication. However, in their scheme, blockchain usage does not impact the authentication process; instead, it is used only to analyze and store data generated by vehicles. Moreover, their scheme is also bilinear pairing-based, which causes more computational cost.

Most of the blockchain-based authentication schemes proposed in the literature till now involve blockchain interaction during the signature verification phase, which causes huge computation cost and makes them unfit for a real-world scenario. Apart from that, in most of the existing blockchain-based schemes, access rights are not equally distributed among all the participating entities. The single trusted authority has all the rights, such as UPDATE and REVOKE, whereas other entities can only query the blockchain. Hence, the existing works are not able to fully utilize the

properties of the blockchain. Table 2.4 illustrates the survey done over numerous schemes based on the integration of blockchain technology, along with cryptography approaches used to design the secure protocols for ITS environments.

Table 2.4: Authenticated Key Agreements Schemes for secure V2V communications in Blockchain-Enbaled ITS.

Schemes	Year	Main Contribution	Major Limitations
Raya et al. [9]	2007	PKI-based authentication scheme	Huge storage and computation burden
Lu et al. [106]	2008	Short-term anonymous certificate-based scheme	Certificate management problem
Zhang et al. [107]	2008	Identity-based signature scheme	Key escrow problem
Zhang et al. [108]	2011	Identity-based signature scheme	Vulnerable towards replay attack
Lee et al. [109]	2013	Identity-based authentication scheme	Does not support non-repudiation and suffers from impersonation attack
He et al. [110]	2015	Identity-based authentication scheme	Key escrow problem
Horng et al. [111]	2015	Certificateless aggregate signature scheme	Malicious-but-passive KGC attack
Azees et al. [112]	2017	PKI-based authentication scheme	Identity revocation list management
Cui et al. [113]	2019	Certficateless authentication scheme	No coordination among semi-trusted authorities in mutliple semi-TA environment
Lu et al. [98]	2019	Blockchain-assisted PKI-based scheme	High computation and communication cost
Zheng et al. [99]	2019	Blockchain-based authentication scheme	Vulnerable to compromised certificate authority
Feng et al. [101]	2019	ABE (Attribute-Based Encryption)-based authentication scheme	No batch message authentication
Wang et al. [114]	2020	Identity-based authentication scheme	Key escrow problem
Wang et al. [115]	2020	Hybrid authentication scheme	Certificate management problem
Xu et al. [102]	2021	Blockchain-based authentication scheme	Multiple trusted authorities
Ren et al. [104]	2021	Blockchain-based certificateless signature scheme	High computation cost due to bilinear pairing operations
Bagga et al. [105]	2021	Blockchain-based certficatless signature scheme	Bilinear pairing-based and limited usage of blockchain

2.5 Quantum-Resistant Anonymous Authentication for Blockchain-Enabled Smart Healthcare

Security and privacy have long been central concerns in smart healthcare research due to the sensitive nature of medical data and the increasing use of networked medical devices. Early solutions primarily relied on conventional public-key cryptographic techniques, including RSA, elliptic-curve, and identity-based schemes [86]. While these approaches provide acceptable security against classical adversaries, their reliance on quantum-vulnerable assumptions limits their applicability in long-term healthcare deployments. Lattice-based cryptography, especially systems based on Learning with Errors (LWE), Ring-LWE, and Short Integer Solutions (SIS/ISIS), has become a top choice for standardization after quantum computing. This field of study covers digital signatures (like Dilithium and BLISS), key encapsulation methods (like Kyber and FrodoKEM), and general frameworks for making authentication protocols based on lattices [116]. A lot of research has been done on how safe these primitives are under the Quantum Random Oracle Model (QROM). Some of this research shows how oracle reprogramming and quantum query techniques can provide an adversary an edge.

There have been more recent efforts to use post-quantum cryptography in healthcare and the Internet of Things (IoT). These studies primarily address lightweight key exchange, authenticated encryption, and lattice computations with minimal overhead [95]. Nonetheless, there exists a scarcity of research that integrates lattice-based authentication, blind-signature privacy, and blockchain decentralization into a unified architecture specifically designed for smart healthcare. Moreover, there is a shortage of research that provides comprehensive QROM-based security proofs that meticulously analyze mutual authentication, session-key secrecy, unforgeability, and unlinkability.

This study aims to fill these gaps by proposing a cohesive framework that combines lattice-based verified key agreement, lattice-supported blind signatures, and blockchain-enabled decentralized trust.

Chaum [117] introduced blind signatures, which remain a fundamental tool for enabling unlinkable authentication in contexts where privacy is essential. Numerous suggestions introduced by authors [118], [119], [130], [120], [121], [131], [124], and [125] have been made for enhancements and additions to e-voting, credential systems, and secure transactions. However, most traditional blind-signature architectures rely on RSA or discrete-logarithm assumptions, making them vulnerable to quantum attacks. Recently, there has been a growing interest in post-quantum

Table 2.5: Comparison of Existing Anonymously Mutually Authenticated Key Agreement Schemes for SHS environment.

Cite	Crypto––graphy	BCT	Security Aims	Mechanism Used & Formal Model	Limitations
[117]	RSA-based blind signatures	No	Unlinkable anonymous signatures	Message blinding + signing Informal definitions	Not quantum secure; classical assumptions
[118]	Symmetric + PKC	Yes	Secure and efficient healthcare data sharing	Blockchain + off-chain storage Informal, prototype evaluation	Not quantum-safe; privacy leakage possible
[119]	Various (survey)	Yes (survey)	Comprehensive review of blockchain in healthcare	Taxonomy + literature survey Not applicable	Survey; no protocol/proof
[120]	PKI + encryption	Yes	Privacy-preserving medical data sharing	Blockchain + encryption + access policies Informal analysis	Performance overhead; partial privacy only
[121]	Post-quantum cryptography (survey)	No	Lightweight PQC for IoT systems	Comparison of PQC families Not applicable	Survey only; no practical deployment
[122]	Lattice-based signatures (SIS)	No	Trapdoor-free lattice signatures	Fiat-Shamir with aborts ROM/QROM-related analyses	Parameter tuning; higher signature size than classical
[123]	Lattice-based signatures	No	Practical Gaussian-based signatures	Gaussian sampling + lattice constructions ROM-based security proofs	Sampling complexity; implementation challenges
[124]	LBC	No	Quantum random-oracle foundations	Compressed oracle model Formal QROM proofs	Theoretical; not a system-level solution
[125]	Module-LWE KEM	No	Post-quantum key encapsulation	Module-LWE + CCA transform ROM/CCA security analysis	Parameter trade-offs; implementation overhead

blind signatures based on lattice structures. But most of the work done so far is on independent signature constructions, not on how they can be used in healthcare-driven blockchain architectures. Table 2.5 demonstrates the comparison of numerous security schemes based on the SHS, based on parameters like: used cryptography primitives, blockchain technology, their aims, mechanisms, and used formal methods, and also highlights their schemes' limitations.

Blockchain-based approaches have been proposed to improve data integrity, auditability, and trust management in healthcare systems [87]. These works demonstrate the advantages of decentralized record management; however, many of them pay limited attention to privacy issues arising from transparent ledger structures. Without additional safeguards, blockchain transactions may reveal information about

user identities or operational behavior.

Over the past few years, few efforts have been made on blockchain-based signcryption. Eltayieb et al. [133] introduced a novel signcryption scheme based on blockchain technology, aiming to facilitate secure data sharing within cloud computing environments. To provide confidentiality and unforgeability, they combined attribute-based signature and encryption with blockchain technology. Elkhalil et al. [134] proposed a blockchain-based online/offline signcryption scheme for a heterogeneous vehicular environment to provide secure data sharing among vehicles and servers. In their scheme, the blockchain and smart contract prevent data tampering and wrong data transmission to IoV nodes. However, their scheme uses pairing operations that incur high computational cost. Xu et al. [135], utilized blockchain as a public key directory and proposed a blockchain-assisted pairing-based encryption scheme. As per their scheme, to address the issue of key escrow, users opt to publish their public keys on the blockchain, thereby enhancing the security of their private keys. Recently, Yang et al. [136] proposed an aggregate encryption scheme using blockchain technology for IoT-enabled Maritime Transportation System (IMTS). They utilized blockchain for the verification of IoT devices and the preliminary verification of signcryption. Both Xu et al. [135] and Yang et al. [136]'s schemes are pairing-based schemes, which incur more computational cost than ECC.

Blind signatures have been widely studied as a tool for anonymous authentication in applications such as electronic payments and voting. Most existing blind-signature schemes, however, are constructed using number-theoretic assumptions that do not offer post-quantum security [117]. Recent advances in lattice-based cryptography have introduced quantum-resistant signatures and key exchange mechanisms with formal security guarantees. Nevertheless, relatively few studies combine lattice-based authentication, blind signatures, and blockchain within a single framework tailored to smart healthcare systems and supported by rigorous QROM-based analysis. Table 2.6 illustrates the comparison of several blind signature-based scheme based on lattice and other public cryptosystems along with their scheme's limitations.

2.6 Summary

In the next chapter, the technical backgrounds on the existing cryptographic techniques, definitions, mathematical models, and blockchain technology, etc., which are used to develop the contributory works in the thesis, are explained.

Table 2.6: Comparison of Blind-Signature and Anonymous-Authentication-Based Schemes for IoT Application environments

Schemes	Application Domain	Cryptographic Primitives	Security Aims & Formal model	Limitations
Chaum Blind Signature [117]	General privacy systems	RSA-based blind signature	Anonymity, unlinkability Informal cryptographic definitions	Classical assumptions; not post-quantum
He et al. [126]	Wireless medical sensor networks	ECC-based anonymous / blind-style authentication	Authentication, privacy preservation Informal adversary analysis	High computation; quantum-vulnerable
Alzahrani et al. [127]	Internet of Medical Things (IoMT)	ECC-based anonymous authentication	User anonymity, authentication ROM-based reasoning	Not resistant to quantum attacks
Islam and Biswas [128]	IoT-based healthcare	ECC-based privacy-preserving authentication	Identity protection, authentication BAN logic analysis	Weak formal guarantees; classical crypto
Li et al. [129]	Smart healthcare data sharing	PKI + blockchain-based access control	Secure sharing, access control Informal / system-level analysis	Metadata leakage; no anonymity
Xu et al. [130]	Blockchain-enabled e-health	Blind-style authentication + blockchain	Privacy preservation, auditability Informal security analysis	Ledger transparency issues
Yang et al. [121]	IoT systems (survey)	Post-quantum cryptography (lattice, hash)	Quantum-resistant security Not applicable (survey)	No integrated healthcare scheme
Lyubash et al. [131]	General authentication systems	Lattice-based signatures (SIS)	Unforgeability, PQ security ROM-based security reduction	Not designed for IoT or blockchain
Ducas et al. (BLISS) [132]	General-purpose signatures	Lattice-based signature scheme	Unforgeability, efficiency ROM-based proof	Sampling complexity
Our Scheme	Smart healthcare with blockchain	Lattice-based AKA + lattice blind signature	MA, SK security, anonymity, unlinkability QROM (game-based proof)	Higher cost than ECC; parameter tuning

CHAPTER 3

Preliminaries

This chapter outlines cryptographic fundamentals, hard challenges, and blockchain technology utilized throughout the thesis, encompassing elliptic curve cryptography, identity-based cryptography, lattice-based cryptography, and Hyperledger Fabric.

3.1 Public-Key Cryptosystem

In this section, public-key encryption (PKE) and digital signature schemes are explained with their definitions and examples.

3.1.1 Public-Key Encryption

The definition of the PKE scheme is presented as follows:

Definition 1. A public-key encryption (PKE) scheme ε comprises a triad of polynomial-time algorithms (K, E, D) such that for every pair of keys (s, p) of K (1^k) and any k-bit message P, we have

$$Pr[D(s, E(p, P)) = 1] = 1$$

i.e. E(p,P) is a valid encryption of an n-bit message with respect to the private key s.

In this definition, K is a key generation algorithm whereas E and D are, respectively encryption and decryption algorithms. s is a secret key and p is the corresponding public-key.

A PKE scheme is said to be secure if the probability of decryption of an encrypted message by any adversary is negligible even if he knows all public information about the encryption algorithm and the algorithm itself.

Definition 2. A public-key encryption (PKE) scheme ε comprising algorithms (K, E, D) is said to be secure if for every PPT randomizes adversary $\hat{A}$, the probability that after intercepting E(p,P) of any message P of his choosing, adversary $\hat{A}$ can derive ŝ in such a manner that D(ŝ,E(p,P))=1, is negligibly small. K, E and D are randomized polynomial-time algorithms and the probability is taken over the randomness of K, E, D and $\hat{A}$.

A number of PKE schemes are proposed in the literature. RSA [137] and ElGamal [138] systems are well-known asymmetric cryptographic constructions. Here, we describe ElGamal's PKE as required for this proposal.

ElGamal's Encryption Scheme [138]. The ElGamal cryptosystem is one of the most fundamental public-key cryptosystems based on the discrete logarithm problem (DLP). This cryptosystem is described by three algorithms: *keyGen*, *encrypt* and *decrypt* as formulated here.

1. *keyGen*(1^k): An authentic user, say Alice selects a large prime number q and a primitive root g in a group $G_1 =< Z_q^*, \times >$ where group G_1 is a cyclic group of prime order q. Now, Alice chooses a random integer $X_A \in Z_q$ and calculates $Y_A = g^{X_A} \pmod q$ and publishes his public key as $< g, Y_A >$ and kept his private key X_A.

2. *encrypt* (g, Y_A, q, m): Anyone can send a message to Alice using his public key. Given a message $P \in Z_q^*$, any sender, say Bob chooses a random integer $r \in Z_q$ and calculates $h = g^r \pmod q$ and sends cipher texts *(h, l=* $P \times {Y_A}^r \pmod q$*)* to Alice.

3. *decrypt* (X_A, q, h, l): On receiving ciphertext *(h, l)*, Alice computes $(h^{X_A})^{-1} \times l \pmod q = (g^{rX_A})^{-1} \times P \times Y_A^r \pmod q = (g^{rX_A})^{-1} \times P \times g^{rX_A} \pmod q = P$ and recover the original message.

An adversary might try to retrieve the secret key X_A, but he is unable to do so because of the hardness of the discrete logarithm problem. The adversary might also try to discover g^{rX_A} from public parameters g , g^r and g^{rX_A}. The assumption that it is hard for any adversary to derive g^{xy} from g, g^x, and g^y is known to be the computational Diffie-Hellman assumption. Under this assumption, ElGamal's cryptosystem is secure.

3.1.2 Digital Signature

In this sub-section, another primitive of public-key cryptosystem, named *digital signature* is discussed. In addition, the security issues regarding the signature schemes are also described.

Definition 3. A public-key digital signature scheme S comprises a triad of polynomial-time algorithms (K, σ, V) such that for every pair of keys (s, p) of K (1^k) and any k-bit message P, we have

$$Pr[V(p, \sigma(s, P)) = 1] = 1$$

i.e. σ(s, P) is a valid signature of n-bit message P with respect to the public key s.

In the definition defined above, K is a key generation algorithm, whereas σ and V are, respectively, signing and verifying algorithms. s is a secret key and p is the corresponding public key. A digital signature scheme is given by the following three algorithms:

1. *keyGen* (1^k): It is a probabilistic polynomial-time (PPT) algorithm K which takes security parameter k as input and outputs a public-private key pair *(s, p)*.

2. *sign (s, P)*: A PPT algorithm σ which takes a message P, a secret key s as input and outputs a signature σ.

3. *Verf (P, σ, p)*: A deterministic polynomial-time algorithm which takes a message P, a signature σ and a public key p as input and outputs **T**rue if σ is a valid signature on message P, else it returns **F**alse.

A public-key digital signature scheme S is said to be strongly unforgeable if the probability that any adversary, after intercepting any number of signatures for adaptively selected messages of his choice, can produce a new signature is negligible.

Definition 4. A public-key digital signature scheme S comprising algorithms (K, σ, V) is said to be strongly unforgeable if the probability of success of the following experiment is negligible for every polynomial-time randomizes adversary $\hat{A}$: choose a pair of keys (s, p) of K (1^k), give the public key p to the adversary Â to generate an interrogation message P $\leftarrow$ Â(p), generate a digital signature σ for the message P i.e. $\sigma(s, P)$, send the generated signature σ to the adversary to obtain a forgery $(P', \sigma') \leftarrow$ Â(p, σ) , and check that $V(p, P', \sigma')$=1 and $(P', \sigma') \neq$ (P, σ).

Many signature schemes are proposed in the literature etc. The traditional ElGamal's signature is described here:

ElGamal's Signature Scheme [138]. The DLP-based ElGamal's signature scheme is described by the following three algorithms:

1. *keyGen*(1^k): An authentic user, say Alice selects a large prime number q and a primitive root g in a group $G_1 =< Z_q^*, \times >$ where group G_1 is a cyclic group of prime order q. Now, Alice chooses a random integer $X_A \in Z_q$ and calculates $Y_A = g^{X_A} \pmod q$ and publishes his public key as $< g, Y_A >$ and kept his private key X_A.

2. *sign* (X_A, P): To sign a message $P \in Z_q^*$, Alice randomly chooses an integer $r(0 < r < q)$, and calculates $h = g^r \pmod q$ and sends *(P, h,* $l = r^{-1}(P - X_A h \pmod q)$ to Bob.

3. *Verf (P, h, l,* Y_A*)*: On receiving P, h, l, Bob checks that $g^P \equiv Y_A^h.h^l \pmod q$.

3.2 Key Exchange Protocol

Symmetric key cryptography uses a common secret key to secure the network communication. This secret key is shared between the sender and receiver of messages and is used by sender and receiver to encrypt and decrypt the messages. To share this secret key between sender and receiver, there must be some protocols. For this reason, the key exchange protocols are used and identified as an important tool in the field of cryptography. The key exchange protocols use the public-key cryptography to negotiate a session-key among parties. In modern cryptography, key exchange protocols are used to build many cryptosystems which are used to secure the network communication. To authenticate a key exchange protocol, an authenticator must be applied to it. The formal definition of an authenticated key exchange protocol is laid down now.

Definition 5 (AKE protocol)**.** *: An authenticated key Exchange (AKE) protocol* π *comprises two triads of PPT randomized algorithms* (P, x, K) *and* (S, σ, V) *such that for every pair (s, p) of* $S(1^k)$ *and k-bit message P, we have*

$$Pr[K(x, P) = 1] = 1 \quad and \quad Pr[V(p, \sigma(s, P)) = 1] = 1$$

i.e. σ(s, P) is a valid authenticator of k-bit message P with respect to the public key s.

3.2.1 Diffie-Hellman (DH) Key Exchange Protocol

The Diffie-Hellman key exchange protocol [139] is first public-key cryptosystem that allows two end users to exchange a common key among them. This protocol is based on the exchange of two public values over an open communication network. In the DH protocol, two participants, say, Alice and Bob, agree upon two public numbers q and g, where q is a large prime with g as a generator of order q in a multiplicative group defined by $< Z_q^*, \times >$. The DH protocol comprises three steps as provided below:

- *Step-1*:Alice choses a large random number $x_a < q$, calculates public message $u = g^{x_a} \pmod q$ and sends to the Bob.
- *Step-2*: Bob choses another large random number $x_b < q$, calculates $v = g^{x_b} \pmod q$ and sends the same to Alice.
- *Step-3*: Alice calculates the secret key $K = v^{x_a} \pmod q = g^{x_a x_b} \pmod q$. Similarly, Bob calculates the same secret key as $K = u^{x_b} \pmod q = g^{x_a x_b} \pmod q$.

It is clear that both Alice and Bob secretly negotiate the same key $K = g^{x_a x_b} \pmod q$. However, it may be pointed out that the DH key exchange protocol suffers from a number of attacks including MITM attack.

3.3 Elliptic Curve Cryptography

Here, an explanation of the elliptic curve cryptography technique is given. ECC is a mathematical concept with some fascinating qualities deployed in cryptography [140]. The curve's points demonstrate an abelian group characteristic with the addition operation. The following equation defines a standard elliptic curve used in cryptography: $\mathbf{y^2 = x^3 + ax + b}$; $\mathbf{4a^3 + 27b^2 \neq 0}$.

Where $\mathbf{a, b, x, y}$ all are real numbers. Elliptic curves comprise two operations: addition and scalar multiplication [54] over their points. Here we present the outlines of how to formulate these operations.

- Let X=(x_1, y_1) and Y=(x_2, y_2)be the abscissa and coordinate of X and Y, respectively. Over an elliptic curve, the addition of these two points $(X + Y)$ introduces a new point Z [140]. To determine the Z, connect X, and Y with a straight line that intersects the curve at the point $(-Z)$. The point Z, i.e. Z=$X + Y$, is the mirror reflection of this point $-Z$ w.r.t. the x-axis.

- Suppose $X = (x_1, y_1)$ and $Y = (x_1, y_1)$ are the identical points on the elliptic curve ($X = Y$), where the addition of these two points $(X + Y)$ introduces a new point Z. To obtain the Z, draw a tangent to the point X, or Y that intersects the elliptic curve at the point $(-Z)$. The point Z, i.e. $Z = X + Y$, is the reflection of this point $(-Z)$ about the x-axis [140].

- Let $X = (x_1, y_1)$ and $Y = (x_1, -y_1)$ which demonstrates that X and Y are the mirror images of one another $(X = -Y)$. This elliptic curve yields a point O after the summation of points $(X + Y)$. To determine the O, connect X and Y like a straight line that would not intersect the respective elliptic curve but is presumed to intersect at infinity O [140]. This point is known as the elliptic curve group's additive identity, i.e. $X + (-X) = O$.

3.4 Identity-Based Cryptography

In cryptography, the IBC is a crucial primitive. Shamir [17] was the first to propose the IBC in 1984. In an IBC, the public key is the identity, which is known to everyone. Therefore, the role of a third party to provide an extra certificate for authenticity purposes is eliminated. As a result, the IBC eradicates the need for a CA-based PKI. On the other hand, Shamir does not provide an encryption strategy. Boneh and Franklin [65] were the first to design the whole roles of an encoding method entitled IBE, after a significant number of research attempts towards IBC (IBE).

For the setup of IBC, a private key generator (PKG) is required. It has four phases: Setup, Key Extraction, Encrypt, and Decrypt [18]. which are depicted as follows:

1. ***Setup***: This phase considers security parameters as input and provides the system's parameters as well as a master key.

2. ***Key Extraction***: This phase generates the long-term private key of the user by using the user's identity and PKG's master key.

3. ***Encrypt***: The sender node can only apply this algorithm to encode the message m as ciphertext c by using the identity of the receiver node.

4. ***Decrypt***: The original message m can be obtained as output by the receiver, only after decoding the ciphertext c by using its private key.

3.5 Homomorphic Encryption

Homomorphic encryption [25] encrypts data sufficiently that together we can use it to operate on the encrypted message to get the same operation encrypted on the original message. It consists of three phases: key generation, encoding and decoding, followed by the evaluation process. Which are defined as follows:

1. ***Key Generation***: Accepts the security parameter to generate the public *(Pb)* and private *(Pr)* key pair.

2. ***Encrypt (Pb, m)***: This command accepts a public key with a message and returns an encrypted message c.

3. ***Decode (Pr, c)***: This command accepts ciphertext c and uses a private key Pr as input to produce an outputs plaintext m.

4. ***Evaluate (Pb, ci, f)***: This algorithm takings a public key, a set of ciphertexts, and a feature, where f is used to be performed on ciphertext c and produces a ciphertext that is the encryption of the same function computed on the original plain texts.

Homomorphic property: Let there are two plain text $P1$ and $P2$ and we got $C1$ and $C2$ after the applying the above-mentioned encryption phase over respective plain texts [20]. Let authorized node requesting for storing the addition of encrypted data over server, thus aggregator need to perform addition over $C1$ and $C2$ i.e C_e=C_1+C_2 by calling the evaluate phase and forward this value to only authorized server. Next, decrypt phase being applied to retrieve the original plaintext in form of $(P_1 + P_2)$ from C_e. As per encrypt phase: $C_1 = P_1 .P \in G$ *and* $C_2 = P_2 .P \in G$. Then after the decryption we evaluate and found that C_e is like C_e= C_1 + C_2= $P_1 .P + P_2 .P$=$(P_1 + P_2).P \in G$.

3.6 Lattice-Based Cryptography

Lattice-Based cryptography (LBC) has emerged as a promising area in post-quantum cryptography due to its efficiency, provable security, and resistance to attacks from quantum computers. Built upon the computational complexity of problems defined over high-dimensional lattices, this approach provides the foundation for constructing secure encryption schemes, digital signatures, identity-based encryption, and fully homomorphic encryption systems.

A lattice in $\mathbb{R}^n$ is a discrete set of points generated by linear combinations of linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ with integer coefficients. Formally, a lattice $\mathcal{L}$ generated by a basis $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is defined as:

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}_i \,\middle|\, a_i \in \mathbb{Z} \right\}.$$

Lattices exhibit geometric properties that make certain problems defined on them computationally hard, especially in high-dimensional spaces. In the appearance of quantum mechanics, the cryptographic concept could be hardened by the rigid assumptions on lattices. A lattice would be any conventional architecture. By employing the characteristics of lattice, lattice-based cryptographic concepts demonstrate a sophisticated security mechanisms and effective network establishment [116]. The profound challenges are robust sufficiently to conquer the quantum attacks were carried out in a quantum computing system. Using the mentioned specifications, one can define a lattice mathematically:

- **Integer Lattices:** Let $\mathfrak{R}^m$ be a collection of feature vectors $l = \{l_1, l_2, ...l_n\}$ that are linearly independent [141]. Where $\mathfrak{R}^m$ is Euclidean space of dimension m, from which a lattice $\mathcal{L}$ is achieved by l could be define as:

$$\mathcal{L}(\ l_1,\ l_2, ...l_n)\ = \{\sum_{k=1}^{n} \{z_k l_k : z_k \in Z\}\} \tag{3.1}$$

  where linear vectors $l_1, l_2, ...l_n$ are known as basis with rank n and the dimension of lattice $\mathcal{L}$ is m. Moreover, every lattice has a common basis. A basis is being represented by the basis matrix $\mathbf{L} = [l_1,\ l_2, ...l_n] \in Z^{m \times n}$, in which the matrix $\mathbf{L}'s$ column refers to a set of basis vectors. Thus, $\mathcal{L}$ derived from $\mathfrak{R}^m$ Euclidean space of dimension m is interpreted as: $\mathcal{L}(\mathbf{L}) = [\mathbf{L}z : z \in Z]$ where general matrix-vector multiplication performed represented by expression $\mathbf{L}z$.
  Definition 1: The minimum distance of $\mathbf{L}$, which corresponds to the shortest non-zero vector, could be interpreted as:

$$\mathcal{D}_{min}(\mathcal{L}) = min_{l \in \frac{\mathcal{L}}{0}} || \ l|| \tag{3.2}$$

- **q-ary Lattice:** The Lattice $\mathcal{L}$ which fulfilled the condition: $Z_q^n \subseteq \ \mathcal{L} \subseteq Z^n$ over the q modulus value is termed as q-ary lattice [32].
  Definition 2: Assume an integer matrix: $\mathbf{M} \in {Z_q}^{m \times n}$ with a modulus value q,; therefore,q-ary lattices are represented as: $\bigwedge_q^{\perp} = \{\mathbf{x} \in Z^n : \mathbf{M}\mathbf{x} = 0 \ mod\ q\}$ and $\bigwedge_q = \{\mathbf{x} \in Z^n : \mathbf{x} = \mathbf{M}^T\mathbf{w} \ mod\ q, \forall\ \mathbf{w} \in Z^n\}$.

Table 3.1: Comprehensive Comparison of Cryptographic Primitives

Parameter	Elliptic Curve Cryptography (ECC)	Identity-Based Cryptography (IBC)	Certificateless Cryptography (CLC)	Lattice-Based Cryptography (LBC)
Main Concept	Elliptic curve algebra for small-key high security	User identity acts as public key; removes certificates	Eliminates certificates while avoiding key escrow	Based on hard lattice problems such as LWE, RLWE, SIS
Quantum Resistance	No (Shor's algorithm breaks ECC)	No	No	Yes (Post-quantum secure)
Key Management	Requires PKI and certificates	No certificates; KGC generates full private key	No certificates; partial key from KGC + user secret	PKI optional; scheme-dependent
CC Cost	Low	Low to Moderate	Moderate	Higher (large keys)
Storage Requirement	Very Low	Low	Moderate	High (large public/private keys)
Main Advantage	High efficiency with small key sizes	Simple PKI-free architecture	Solves certificate + key escrow problems	Quantum-resistant, strong hardness assumptions
Main Limitation	Vulnerable to quantum computing	Full trust on KGC (escrow)	Higher complexity	Large key sizes; heavier computation
Weakness	Not secure against quantum attacks	Key escrow issue	Complex setup and key generation	Large keys and computations

3.7 Hard Assumptions of Cryptography Primitives

The security of any of the cryptography primitive-based schemes is rooted in well-defined computational problems that are conjectured to be hard, even for quantum algorithms. Table 3.2 represents the mathematics behind all hard assumptions. Few hard problems faced difficulties to solve [142], which are defined as below:

1. ***Computational Diffie-Hellman Problem:*** To derive the value $(a \cdot b \cdot P)$ from the known variable pair such as $(a \cdot p, b \cdot P,\ P)$ is comes under the computational hard challenge. Where a, b belong to the multiplicative prime integer set Z_q^* and P belongs to the group G [143].

2. ***Elliptic Curve Discrete Logarithm Problem:*** Determine the value of r from the already defined and known variable, which $Q = r \cdot P$ comes under the hard challenge where $Q, P \in G$ and $r \in Z_q^*$ [143].

3. **Shortest Vector Problem (SVP)** The SVP asks for the shortest non-zero vector in a given lattice. It is computationally hard, especially in high dimensions, and serves as the basis for many cryptographic hardness reductions. Approximating SVP within polynomial factors is known to be NP-hard. The SVP involves finding the shortest non-zero vector in a lattice, based on the Euclidean norm. Formally:

$$\text{Find } \mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\} \text{ such that } \|\mathbf{v}\| = \lambda_1(\mathcal{L}),$$

Where $\lambda_1(\mathcal{L})$ denotes the length of the shortest non-zero vector in the lattice. SVP is NP-hard under randomized reductions, and its hardness increases exponentially with lattice dimension.

Table 3.2: Summary of SIS, ISIS, CVP, SVP, and Hash Hard Problems

Problem	**Input / Given**	**Output / Goal**	**Mathematical Expression**	**Cryptographic Role**	**Hardness Basis**
SIS (Short Integer Solution)	Matrix $A \in \mathbb{Z}_q^{n \times m}$	Find nonzero short vector $\mathbf{z} \in \mathbb{Z}^m$	$A\mathbf{z} \equiv \mathbf{0} \pmod q$, $\|\mathbf{z}\| \leq \beta$	Basis for signatures, hash functions	Reduces to worst-case SVP/SIVP
ISIS (Inhomogeneous SIS)	Matrix $A \in \mathbb{Z}_q^{n \times m}$, $\mathbf{u} \in \mathbb{Z}_q^n$	Find short vector $\mathbf{z} \in \mathbb{Z}^m$	$A\mathbf{z} \equiv \mathbf{u} \pmod q$, $\|\mathbf{z}\| \leq \beta$	Identification protocols, trapdoor functions	Reduces to worst-case lattice decoding
CVP (Closest Vector Problem)	Lattice basis $\mathbf{B} \in \mathbb{R}^{n \times n}$, target $\mathbf{t} \in \mathbb{R}^n$	Find lattice vector $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{t}$	$\min_{\mathbf{v} \in \mathcal{L}} \|\mathbf{t} - \mathbf{v}\|_2$	Security reductions, decoding attacks	NP-hard to approximate
SVP (Shortest Vector Problem)	Lattice basis $\mathbf{B} \in \mathbb{R}^{n \times n}$	Find nonzero shortest vector $\mathbf{v} \in \mathcal{L}$	$\min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|_2$	Foundational problem for lattice cryptosystem	NP-hard to approximate
Collision Resistance	Hash function $H : \{0,1\}^* \to \{0,1\}^n$	Find $x \neq x'$ such that $H(x) = H(x')$	$\exists x \neq x' : H(x) = H(x')$	Fundamental for hash integrity and digital signatures	Computational hardness (ideally $2^{n/2}$)
Preimage Resistance	Hash output $y \in \{0,1\}^n$	Given y, find x such that $H(x) = y$	Find x where $H(x) = y$	Protects against inversion attacks	Computational hardness (ideally 2^n)
Second Preimage Resistance	Input x, hash $H(x)$	Find $x' \neq x$ such that $H(x') = H(x)$	Given x, find $x' \neq x$ with $H(x') = H(x)$	Important for signature forgery resistance	Computational hardness (ideally 2^n)

4. **Closest Vector Problem (CVP)** It is a fundamentally computationally difficult problem in lattice-based cryptography. It is delineated as follows: Consider a lattice $\mathcal{L}$ formed by the basis vectors $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ in$\mathbb{R}^n$. and a target vector $\mathbf{t} \in \mathbb{R}^n$, which may not reside in $\mathcal{L}$. The CVP inquires: Identify

the lattice vector $\mathbf{v} \in \mathcal{L}$ that minimizes the distance to $\mathbf{t}$, often concerning the Euclidean norm. CVP: $\min_{\mathbf{v}\in\mathcal{L}}\|\mathbf{t}-\mathbf{v}\|_2$

5. **ISIS Hard Problem** Find a short vector mapping to a given syndrome under the matrix modulo q. To determine a vector $\mathbf{x} \in \frac{Z^n}{0}$ with the satisfying condition: $\mathbf{M}^T\mathbf{w}$ *mod* q with $||x|| \leq \chi$ become a hard challenge, where matrix $\mathbf{M} \in {Z_q}^{m\times n}$ and $\chi \in Z^+$ are given.
($\text{ISIS}_{n,m,q,\beta}$)] Given: A matrix $A \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$ sampled uniformly at random, and A target vector $u \in \mathbb{Z}_q^m$, The Inhomogeneous Short Integer Solution (ISIS) problem asks to find a vector $z \in \mathbb{Z}^n$ such that $Az \equiv u \pmod q$ and $\|z\|_2 \leq \beta$

6. **SIS Hard Problem** Find a short, nonzero vector mapping to zero under the matrix modulo q. To determine a vector $\mathbf{x} \in \frac{Z^n}{0}$ with the satisfying condition: $\mathbf{Mx} = 0$ *mod* q with $||\mathbf{x}|| \leq \chi$ become a hard challenge, where the matrix $\mathbf{M} \in {Z_q}^{m\times n}$ and $\chi \in Z^+$ are given.
Let $n, m \in \mathbb{N}$ be positive integers, $q \geq 2$ be an integer modulus, and $\beta > 0$ a norm bound. Let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. We define SIS as: ($\text{SIS}_{n,m,q,\beta}$): Given a matrix

$$A \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$$

chosen uniformly at random, the Short Integer Solution (SIS) problem asks to find a non-zero vector $z \in \mathbb{Z}^n \setminus \{0\}$
such that $Az \equiv 0 \pmod q$ and $\|z\|_2 \leq \beta$.

3.8 Hash Function

A hash function is a mathematical operation that processes a parameter (or "message") and produces a fixed-size byte number, usually a digest that seems random. Let us assume that: M: Set of possible input messages (of arbitrary length), H: Set of possible hash values (of fixed length n), $h : M \to H$: A hash function mapping an input message to a hash value:
$h : M \to \{0,1\}^n$ Given a message $m \in M$, the hash value is: $h(m) = d, \quad d \in \{0,1\}^n$

Blockchain, digital signatures, cryptography, data integrity, and conserving passwords all make extensive use of hash functions.

3.8.1 Property of Hash Function

The "HARD Assumption" for hash functions denotes the mathematical impossibility of inverting the hash (pre-image resistance), identifying second pre-images, or

discovering collisions within a feasible (polynomial) timeframe for security systems. These attributes are crucial for the application of hash functions in security contexts, including digital signatures, password hashing, and message authentication code.

- **Pre-images:** Given a hash output $y = \mathcal{H}(x)$, it is computationally impractical for any competitor to identify the input $x^{'}$ like that $\mathcal{H}(x^{'}) = y$. This ensures that the genuine input can't be retrieved from its hashing algorithm.
- **Second Pre-images:** For every single fixed input x with its hash $\mathcal{H}(x)$, it is computationally impractical to identify an alternative input $x^{'} = x$ such that: $\mathcal{H}(x^{'}) = \mathcal{H}(x)$. This inhibits an assailant from replacing one legitimate message with another undetected.
- **Collision Resistance:** It is computationally challenging for any competitor to discover two distinguished variables $x \neq x'$ such that $\mathcal{H}(x) = \mathcal{H}(x^{'})$. That greater resilience property ensures of the fact the hash function does not permit conjunction collisions.

3.9 Blockchain Technology

Blockchain technology, initially established as the foundation for digital currencies, has evolved into a transformational force with applications in multiple industries. Blockchain functions as a decentralized peer-to-peer (P2P) ledger system on the Internet, integrating cryptographic methods, including encryption, with a communal database. Blockchain effectively addresses issues pertaining to trust. Data traceability, unpredictability, and modification are guaranteed by secure communication with a hash chain-based encryption method and a timestamp technique for certificate values. The linkage of the nodes and the block data is maintained by the implementation of consistency mechanisms [144]. The synergy between the autonomous script program and the Turing virtual machine guarantees the programmable smart contract. Consequently, the blockchain comprises a formidable security framework that depends on cryptography and communication technology, in addition to a consensus method.

1. **Blocks** A block in the context of blockchain is a group of data that symbolizes a collection of information or transactions. Each block incorporates a cryptographic hash of the preceding block, establishing an unbroken sequence of blocks—thus the term "blockchain". Blocks play a crucial part in the essential characteristics of blockchain technology [145]. Primarily, blocks function

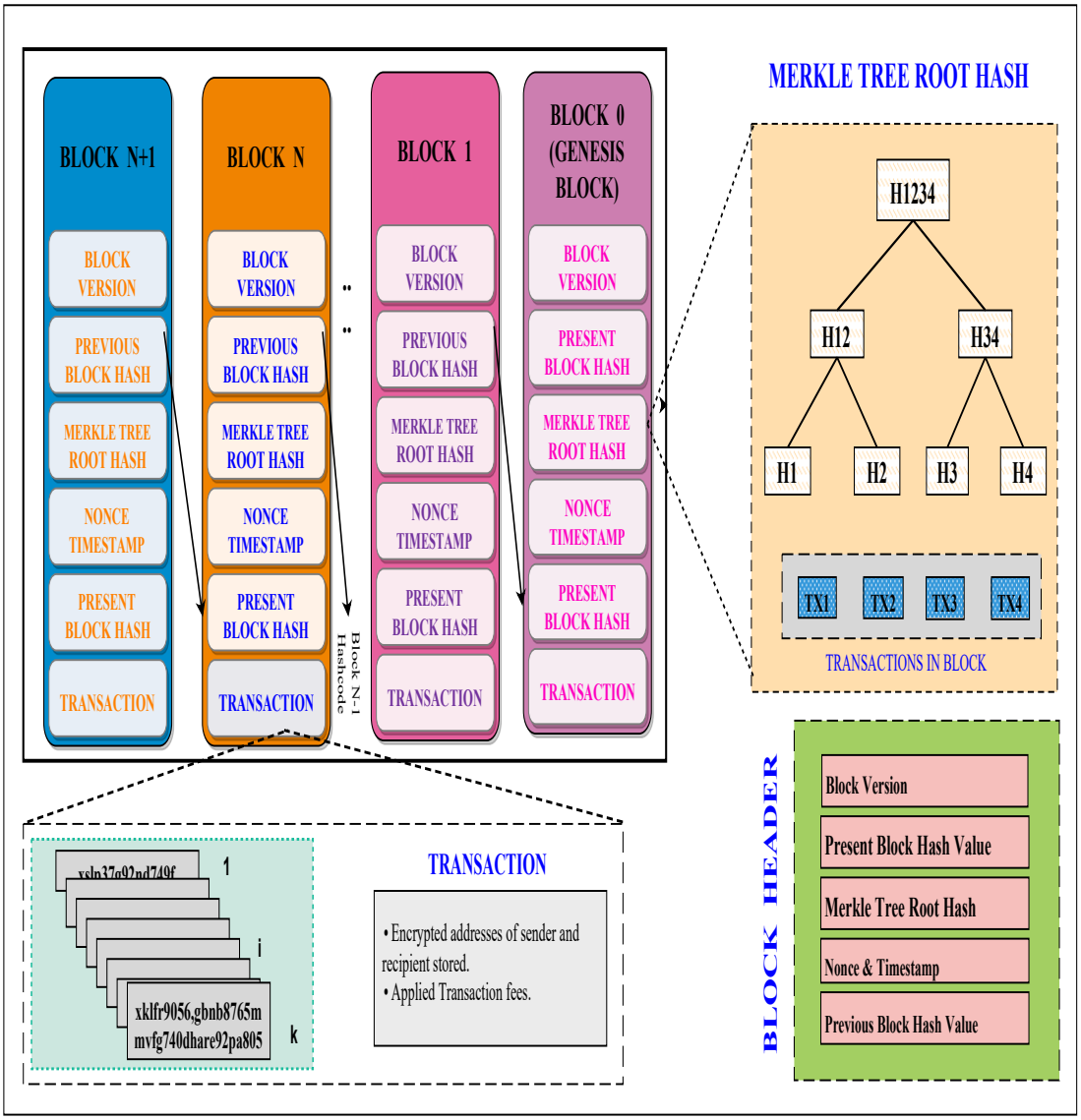


Figure 3.1: Block Architecture in Blockchain Technology.

as repositories for data, encapsulating records of transactions, contracts, or pertinent information within the blockchain network. The interlinked structure of blocks via cryptographic hashes guarantees the integrity and security of the entire chain. modifying the data in a block would necessitate recalculating the hash and modifying the block's contents, which would then impact all blocks that follow. This interdependence establishes a counterfeit-proof system, as any illegal alteration would be promptly evident to all members of the network. Moreover, blocks are essential for sustaining a transparent and consensus-oriented atmosphere.

Figure 3.1 represents the block architecture used in blockchain technology, where we can see how a block holds the hash of the previous block and the present block along with the Markle tree hash, which contains the transactions. In this way, we can say that blockchain holds the immutable property.

2. **Smart Contract** A significant utilization of blockchain technology is smart contracts, which enable a consumer or company to create a legally enforceable contract using the blockchain's decentralized network. Smart contracts provide legal assistance and support services by functioning as autonomous agents, embedded within blockchains, equipped with encryption abilities that convert transactions into contracts or other legal documents. Each script under these smart contracts possessed a distinct network address [87]. Consequently, smart contracts might be readily identified and verified using their encrypted identifier. Within decentralized infrastructure, smart contracts provide a mechanism that minimizes participant engagement while maintaining a high level of transaction efficiency. To efficiently transfer assets and execute transactions seamlessly, a consumer must be able to draft their contracts without the aid of a legal advisor or notary. This addresses the substantial concerns regarding cost or time [146].

3.10 Consensus Mechanism

In simple terms, blockchain is a collection of machines that analyze and record information to preserve the Integrity and security of data exchanges [5]. Now, how can we ensure that all such transactions appear to be secure and authenticated? Owing to the dispersed and decentralized features of blockchains, there is no monolithic authority that can execute system-wide administration. Blockchain employs a variety of techniques to accomplish what would be referred to as consensus across

trust-free parties to validate protocol guidelines have been observed and prohibit any malfeasance [147]. The agreement of a set of entities regarding a common value through local contact is referred to as consensus. Consensus seems to be the process through which members of a Blockchain network agree on an awareness regarding the present value of the information in the system. These consensus mechanisms are granting the legacy to blockchain platforms in terms of consistency and credibility [148].

Why Consensus? A blockchain is an electronic record that is dispersed, decentralized, and generally public, which is recorded as transactions [148]. Such transactions are represented as a block of contents that must first undergo impartial validation by the P2P computer network in favor of being included in the chains. These techniques help to eliminate the issue of "double-spending" by protecting the blockchain from nefarious transactions. All cryptocurrency applications are developed using consensus principles, which also give them intrinsic security. We must initially understand how much consensus signifies within the context of blockchain systems, before proceeding with the talk about the various consensus protocols [149]. Consensus strategies are used by blockchain systems such as Ethereum and Bitcoin to verify that all users (members) agree on a consistent view of events. The goal of such procedures is the system's fault tolerance.

How does it work? In the case of a Centralized system, there is a central administrator who holds the authority to operate and run the database [150]. Further modifications, such as the insertion, deletion, or alteration of records, are carried out by the central authority, which continues to be the only one in a position to maintain adequate data. Blockchain systems run on a worldwide platform without needing a single centralized body and are distributed, autonomous systems. Blockchain systems run on a worldwide platform without needing a single centralized body and are distributed, autonomous systems [151]. It comprises the contributions of the many participants, who concentrate on block mining along with the tasks of verifying and authenticating transactions that take place on the ledger. Table 3.3 illustrates the differences between all types off consensus mechanisms used in blockchain technology.

3.10.1 Transaction Processing Steps

The transaction processing steps are explained below figure 3.2.

1. Client API sends a proposal to invoke a chaincode function with the input

Table 3.3: Performance Analysis of different types of consensus algorithms.

Protocol	**P/ V**	**Block Chain Type**	**DN**	**Latency**	**Comp Cost**	**Network Cost**	**Extens- -ibility**	**DR**	**AA**	**TPS**
PBFT	V	PR	PO	Large	Less	Expensive	Small	Small	< 33% FR	< 50000
DBFT	V	PR	PO	Large	Less	Expensive	Large	Large	< 33% CP	< 1000
Raft	V	PR	PO	Large	Less	Moderate	Large	Large	< 50% CF	< 800
PoW	P	PB	PL	Large	Expensive	Less	Large	Large	< 25% CP	< 20
PoC	P	PB	PL	Large	Less	Less	Large	Small	NA	< 300
PoH	P	PB	PL	Medium	Less	Less	Large	Large	< 51% H	< 500
PoI	P	PB	PL	Medium	Less	Less	Large	Large	< 51% I	< 500
PoS	P	PB	PL	Medium	Moderate	Expensive	Small	Small	< 51% SP	< 20
PoB	P	PB	PL	Large	Moderate	Moderate	Large	Small	< 25% CP	NA
DPoS	P	PB	PL	Medium	Moderate	Moderate	Large	Large	< 51% VA	< 500
PoET	P	PB	PL	Small	Less	Less	Large	Large	NA	< 100
Steller	V	PR	PL	Medium	Less	Moderate	Large	Large	< VR	< 10000
Ripple	V	PR	PL	Medium	Less	Moderate	Large	Large	< 20% FU	< 10000
Tendermint	V	PR	PO	Small	Less	Less	Large	Large	< 33% VA	< 10000

P* Proof, V* Voting, PR* Private Blockchain, PB* Public Blockchain, PO* Permission Oriented, PL* Permission Less, DN* Distributed Nature, AR* Adversarial Alliance, DR* Delivery Ratio, CC* Computational Cost

parameter to perform operations, such as insert, update, delete, query, etc.

2. The endorsing peers verify the proposal by checking the signature and freshness. Further, they verify if the requesting client has the access rights to invoke the corresponding chaincode function. Next, the proposal response is prepared, and endorsement is added. For that, the endorsing peer invokes the chaincode by taking proposal inputs as arguments to the corresponding function. Next, the chaincode is run against the existing state database, generating transaction results that consist of a response value, write set, and read set. Now, the proposal response is sent along with the endorsing peer signature to the target peer. It should be noted that no ledger updates are made at this point.

3. Target peer verifies the response and checks if it has received enough endorsements from valid peers.

4. Now, the target peer forwards the transaction proposal and response to the ordering service.

5. Ordering service receives transactions and creates a block by ordering them

according to the occurrence.

6. The blocks containing transactions are distributed to all peers on the channel. Within each block, the transactions are validated to ensure that the endorsement policy is satisfied and that the ledger state for the variables in the read set has not changed since the read set was generated during transaction execution. Additionally, each transaction in the block is labeled as either valid or invalid.

7. All the peers update the ledger, and the client is notified.

3.11 Hyperledger Fabric

In 2008, Satoshi Nakamoto introduced blockchain technology [152]. Since its inception, blockchain has attracted researchers worldwide due to its features such as immutability, distributivity, and fault tolerance. Blockchain is a chain of blocks that contains a list of transactions in each block. These blocks connect through a hash in such a way that each block includes the previous one's hash, which means a single bit change in a block will reflect on the entire ledger. Blockchain employs a decentralized architecture, where multiple participating peers maintain the ledger. A transaction is added to the block by following a consensus algorithm that runs among participating peers. Blockchain has three types based on the needs of the application, i.e., public, private, and consortium. A public blockchain is open to all. The nodes on the blockchain network have equal rights to perform any transaction. Any new node can join the public blockchain without permission and perform transactions. Private blockchain follows a centralized model in which only one node manages the blockchain and has all the rights to perform any operation. The rest of the nodes have only read permission. In a consortium blockchain, access privileges such as writing, reading, and auditing are granted to the participating peers, which is maintained by a defined set of peers. This thesis utilizes the open-source consortium blockchain platform Hyperledger Fabric. Hyperledger Fabric follows a highly configurable and modular architecture with a permissioned environment where participating nodes know each other but do not fully trust one another. Hyperledger Fabric has the following key components:

1. **Peer:** Peers are a basic building block to form a fabric network as they are part of the consensus process and also keep the smart contract and ledger. A peer can host multiple ledgers and also be part of multiple channels.

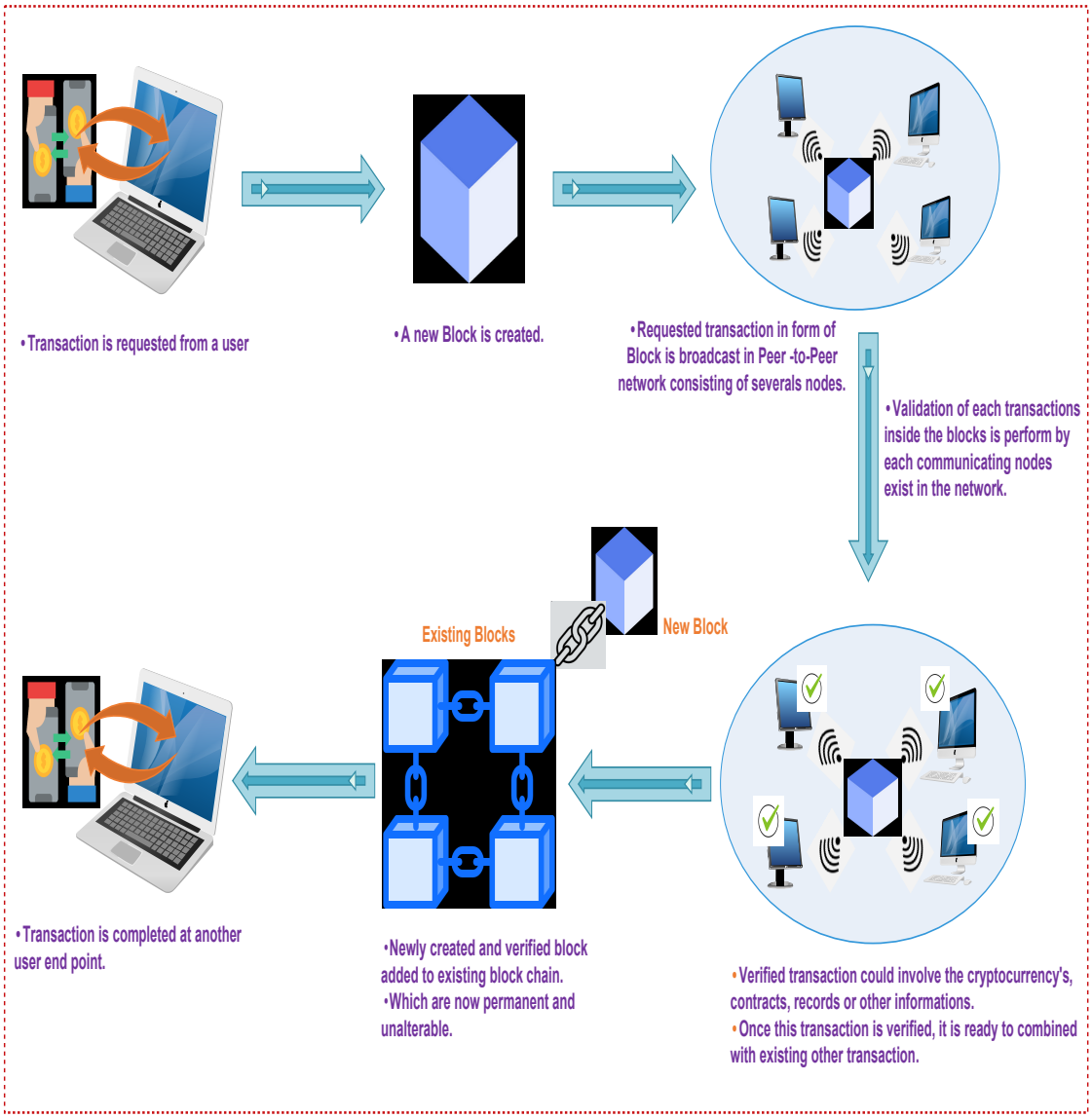


Figure 3.2: Node Creation, Validation, and Addition in Blockchain.

2. **Orderer:** The orderer node in the fabric network performs the task of transaction ordering, creating blocks, and providing them to other peers on the network.

3. **Ledger:** The ledger is shared among all the peers. It has two parts: blockchain and world state, where blockchain is an immutable chain of data blocks containing a transaction list, and world state contains the current values of attributes of an object. Transactions are collected inside blocks that are added to the blockchain. This enables to check updation history on the world state. The world state enables avoiding traversing the whole blockchain to calculate the current value of the object. It can be extracted directly from the world state.

4. **Smart contract:** A smart contract is an executable code that defines the set of rules for transactions between parties. The transaction logic defined inside a smart contract controls the life-cycle of objects in the world state. It is packaged inside a chaincode, which is later deployed on peers. A chaincode may contain multiple smart contracts. When the chaincode is deployed, the smart contracts inside it are available to applications.

5. **Channel:** A channel is a way for consortium members to communicate with one another. Numerous channels can exist in a fabric network, and a peer can join multiple channels. Channel provides infrastructure sharing efficiently with communication and data privacy. When a chaincode is deployed or committed on a channel, all the applications on that channel can access smart contracts inside the chaincode.

3.12 Summary

This chapter presents some important cryptographic tools, some public-key cryptosystems like elliptic curve cryptography, ID-based cryptography, homomorphic encryption, and lattice-based cryptography, with their formal definition are discussed. Additionally, a hard assumption of all cryptosystems is also described and followed by the definition of a hash function and blockchain technology, which are useful in the subsequent chapters. With these preliminaries, we would move forward to present the research works.

CHAPTER 4

A Pairing-free Data Authentication and Aggregation Mechanism for Intelligent Healthcare System

4.1 Introduction

According to the World Health Organization (WHO) [153], every minute around 44 people get sick which is more than 43 million people a year. Out of which 4700 lose their life. These deaths are caused alone due to consuming contaminated food. [153] As a part of the diagnosis of these diseases, we require a strong healthcare department. The Internet of Things (IoT) is revolutionizing the healthcare system by announcing the Intelligent Healthcare System (IHS), which provides the services of anyone, anytime, and from anywhere with all connecting medical sensors [154]. Medical sensors are rapidly being adopted in many countries as the number of patients continues to rise [155]. These medical sensors are implanted over or in the body as per their roles, which collect health monitoring records like glucose level, blood pressure, ECG reading, etc. Patients can share their health history and current records with healthcare professionals in real time without visiting the hospital or having a physical meeting for better medical services and advice. So, the healthcare team can provide the proper treatment as per the received patient's sensitive health records [154], [156]. Therefore, it also has a positive impact on health services, as

professionals can monitor and diagnose a patient's health status intermittently and provide better treatment in time. The adoption of the Intelligent Healthcare System significantly plays a very impactful role over the traditional healthcare system by reducing the cost, providing faster and more timely better treatment, improving the quality, etc. [32]. Today, the number of connected devices in IoT expands exponentially [157]. Therefore, in a single day, billions of data points are exchanged, whereas all the communication is completed via open channels [56]. As we analyzed all the sensitive and confidential information about patients and recommended treatment, like as name, their medical history, address, doctors' personal data, etc, traveled over an open channel. This makes it very easy for attackers to perform several malicious activities like modification, deletion, falsification, or misuse of the confidential records of patients or doctors [156]. For example, a Man-in-Middle attacker, who modifies the recommended treatment, will cause serious health issues, maybe loss of life. Hence, the security concern about Privacy, Authentication, Integrity, etc., will become a major concern in the IHS environment to prevent several attacks like Man in the middle, impersonation, data falsification, etc., which are correlated to the entity authentication issues and key exchange issues [154], [158], [56], [82].

As per the Health Insurance Portability and Accountability Act (HIPAA) [157], IHS must prevent attacks. All internet-connected devices are restricted by the consumption of energy and computational cost. An improved, authentic key agreement mechanism will be obligatory for the IHS environment, which delivers the proper authentication and key exchange between the connecting nodes to prevent such malicious activities. An aggregator [24] is used to remove the redundancy of collected data from medical sensors, hence it reduces the energy consumption and computational cost of nodes by transferring data. Despite this, it also faces several security issues. To resolve the above-mentioned issues, we are enthused to design a protocol for an e-Healthcare system that prevents malicious activities of the attacker by preserving the confidentiality, integrity, and availability security goals. Therefore, we introduce an authentication key management scheme along with protected data accumulated using homomorphic encryption for the IoT-based healthcare environment based on ECC. Figure 4.1 illustrates the working architecture of the Intelligent Health System, in which numerous medical sensors are implanted over or in the body of patients to observe the patient's health condition. These health records are associated with various parameters like Blood Pressure, Glucose level, ECG reading, etc. Such collected records forwarded towards the Medical server through open channel internet connections. Medical Server received that health records and stores it, forwarded towards the associated health professionals teams like doctors or nurses, etc,

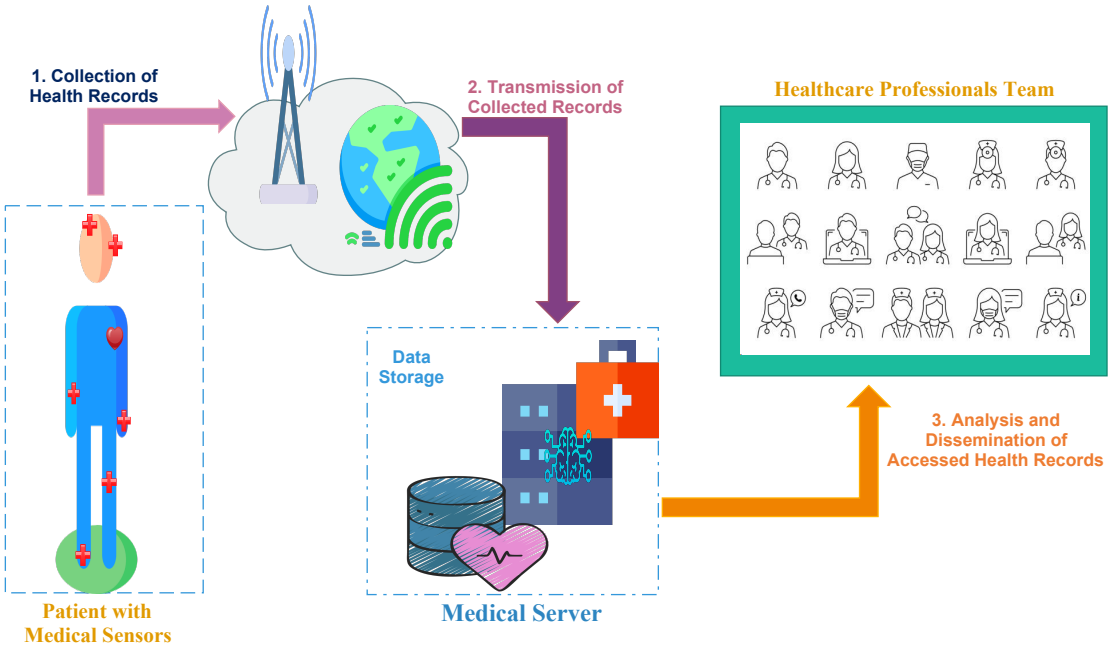


Figure 4.1: Basic Architecture of Intelligent Healthcare System.

who can recommend the proper treatment according to readings like precaution, prescribe the medicine, recommends admitting in emergencies, etc.

4.1.1 Problem Statement

We found that the patient's health records are categorized into three major categories: Emergency, Vital and Regular data, where we set the highest priority for emergency data, so that, it should be accelerated on an urgent basis from the medical sensor to the medical server aimed for immediate action. Some patients' health conditions will require observation on daily basis as per the doctor's request such data comes under the vital category. Patient data that do not come under emergency and vital will come under the regular category. These should be updated periodically on the medical server. After observing the numerous schemes, we analyzed that the detection of attackers was not performed on the medical sensors as well on the aggregation phase. That's why these schemes also suffered from high computational and communicational cost issues as already abortive to maintain data confidentiality and integrity. Thus, we were stimulated to design a protocol in which at all phases we prevent the attacks and are also attentive to diminishing the computational and communicational cost. As the public key infrastructure (PKI) has a central authority to generate and maintain the key pair for each and every node, it faced a key escrow problem. Using a private key generator (PKG), these key overhead issues

are tackled. An Identity-based private key is generated for each user. Further, we observed that the ID-based bilinear pairing scheme has a high computational cost, which inspired us to go through pairing-free computations so that we meet our other goals and reduce the computational and communicational costs.

4.1.2 Main Contributions

Motivated by all of these security and privacy challenges of the IHS, the main objective of this proposal is to farm protected data accumulated along with an authentication key management scheme using homomorphic encryption for the IoT-based e-healthcare environment based on ECC, which is responsible for secure and authentic verification. This newly proposed protocol contributes in the following directions:

1. Protect data against data imitation and data amendment. Moreover, it ensures data sharing, credibility, and security through its verification phase.

2. A formal security model with informal security analysis is given and further adopted to show the semantic security of the protocol, considering the different adversaries and their attacks in the IHS.

3. The proposed model satisfies mutual authentication. Subsequently, it also prevents man-in-the-middle, replay, node impersonation, etc attacks.

4. The model under a suitable environment is shown to outperform other related performances in sustained computation, communication, storage, and energy overheads.

4.2 Background

This section has outlined the network model, security model, and security goals defined for the proposed scheme.

4.2.1 Network Model

Figure 4.2 depicts our network approach, which could be used either by healthcare organizations or by the patient who is situated at a remote location. Our designed schemes consist of three functional modules: Patients, Aggregator, as well as Medical Server. The roles of these modules are described below

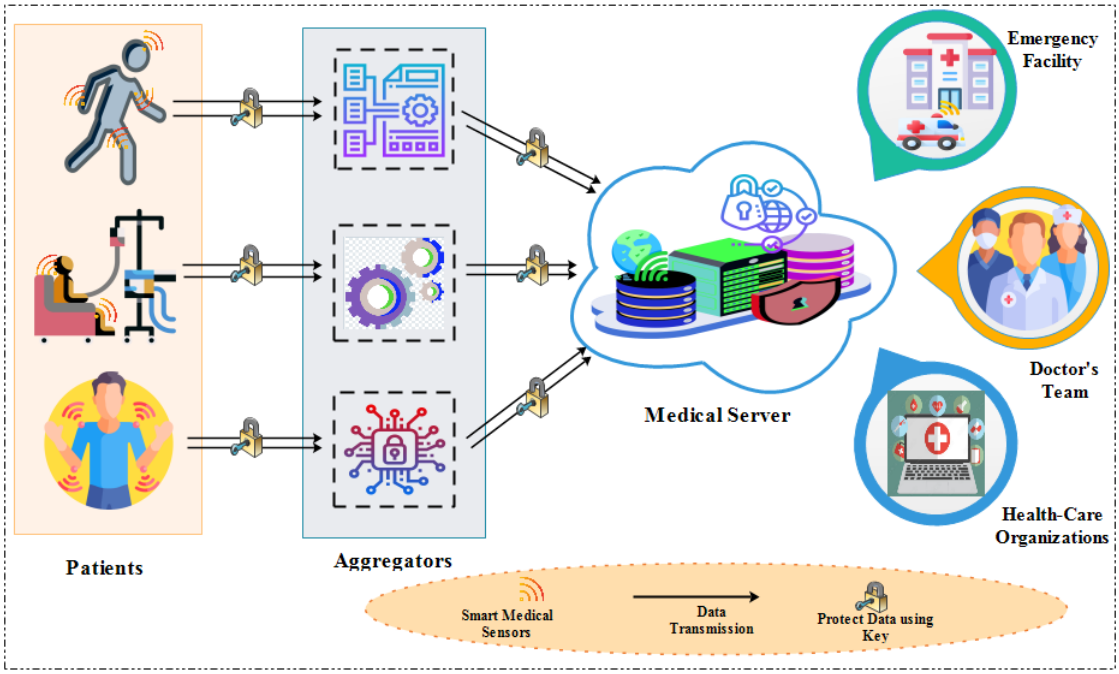


Figure 4.2: Proposed Intelligent Healthcare System Network Model

1. **Patients:** Patients are being outfitted with a portable smart medical sensor, which seems to be wearable equipment. Such sensors are mounted on or in the person's body and used to monitor physiological functioning as well as the ambient. Where thermometers, electromyograms, echocardiograms, cardiovascular, and oximeters are all integrated into smart sensors. The task of these smart sensors is to report to the Aggregator regarding the health information that has been observed.

2. **Aggregator**: It is just a patient smartphone, which behaves like a gateway between the patients and the medical server. it is a unique smart point, which offers better computation and communication range capabilities. Aggregation, as even the title indicates, has to use aggregate procedures to consolidate the records. To prohibit malicious units from accessing the network, the Aggregator accumulates the patient's health records and verifies the authenticity of the patient's sensor, seeking to interact with the server using it, then computes the aggregation on them.

3. **Medical Server:** Health care professionals are the users of the Medical Server who could be doctors, medical staff, researchers, etc. It has virtually unlimited storage and powerful computational potential. Therefore, disease diagnosis and prognosis could be performed on the information stored at the medical server end. However, we assume a setup in which only authorized entities

and the pertinent doctor/emergency health professional are allowed to access the server. Moreover, now the doctors are able to get real-time information regarding health status only by acquiring the patient's health data.

4.2.2 Security Model

In this part, we designed and defined a formal attack model for the proposed scheme, which consists of participating nodes $\wp$ that can be PS_i, MS_i, AG_i. Any two of these participating nodes can communicate; one of the participants should initiate the conversation, while the other would react. Furthermore, let's glance at the relevant parameters and abbreviations.

1. $\mathfrak{S}^n_{\wp_i}$: it states the information regarding the nth iteration performed by Participating Node $\wp$. Where it $\wp_i$ generates a list that contains a set of variables. That group of variables keeps track of the protocol's present state inside a list, which is automatically updated during the execution of scheme.

2. $\mathfrak{A}^n_{\wp_i}$: This is a value that is utilized to appraise the session's uniqueness. All participants, along with the oracle, must be aware of the respective session's identity.

3. $\mathfrak{L}^n_{\wp_i}$: It refers to the actual relevant data about specified participants' credentials (private key as well as additional specific identities). Those are required to obtain the protocol's secret session key.

The following are a few definitions relating to the proposed scheme, along with its security.

- $\wp_i$ Participant's $\mathfrak{S}^n_{\wp_i}$ state would be considered as the accepted state only after it generates the genuine and non-nil session key with its counter participant.

- For every instance $\mathfrak{S}^n_{\wp_i}$ of participant's $\wp_i$, identifier of the session key, it should be identified with respect to the session n and maintained in the list $\mathfrak{A}^n_{\wp_i}$, which is a public list.

- Participant's Identity $\mathfrak{I}^n_{\wp_i}$ should be shared at instance $\mathfrak{S}^n_{\wp_i}$, when a participant $\wp_i$ wants to share a secure session key with the other. $\mathfrak{I}^n_{\wp_i}$ is also a public value.

- To establish the partnership between two instances $\mathfrak{S}^n_{\wp_i}$ and $\mathfrak{S}^m_{\wp_j}$ the requirements are as follows:

  – Both instances $S^n_{\wp_i}$ and $\mathfrak{S}^m_{\wp_j}$ exist in acceptable state.

– Session and participant identity are same for both instances i.e. $\mathfrak{A}^n_{\wp_i} = \mathfrak{A}^m_{\wp_j}$, $\mathfrak{I}^n_{\wp_i} = \mathfrak{I}^m_{\wp_j}$.

Assume $\mathbb{A}$ is the PPT attacker, looking to violate the semantic protection. However, $\mathbb{A}$ as well as the other involved nodes like sensor, server aggregator should communicate with each other via only the random oracle queries, which stated as follows. $\mathbb{A}$ could address the accompanying questioning to see whether the protocol's security will be compromised.

1. Reveal Private key ($\wp_i$): When $\mathfrak{S}^n_{\wp_i}$ execute queries for $\mathbb{A}$, the output received by adversary $\mathbb{A}$ will be participating node's $\wp_i$ private keys.

2. Send $(\mathfrak{S}^n_{\wp_i}, m) : \mathfrak{S}^n_{\wp_i}$ accomplish the queries when $\mathbb{A}$ sends a message m. Therefore, it generates results for $\mathbb{A}$ as described within the introduced scheme. Moreover, the response will be $NULL$ in the presence of an inappropriate message.

3. Reveal Random secret($\mathfrak{S}^n_{\wp_i}$): When $\mathfrak{S}^n_{\wp_i}$ execute queries for $\mathbb{A}$, the output received by adversary $\mathbb{A}$ will be ephemeral keys of participating node $\wp_i$.

4. Reveal Public Key ($\mathfrak{S}^n_{\wp_i}$): When $\mathfrak{S}^n_{\wp_i}$ execute queries for $\mathbb{A}$, the output received by adversary $\mathbb{A}$ will be participating node's $\wp_i$ public keys.

5. Reveal State ($\mathfrak{S}^n_{\wp_i}$): When $\mathfrak{S}^n_{\wp_i}$ execute queries for $\mathbb{A}$, it generates all state details $\mathfrak{A}^n_{\wp_i}$ for nth instances of nodes and send to adversary $\mathbb{A}$.

6. Reveal Session key ($\mathfrak{S}^n_{\wp_i}$): When $\mathfrak{S}^n_{\wp_i}$ run this query for $\mathbb{A}$. Therefore, for nth session it produces $\mathfrak{S}^n_{\wp_i}$ as session key, which being generated between participating nodes and directed to $\mathbb{A}$.

7. Test ($\mathfrak{S}^n_{\wp_i}$): When ${\wp_i}^n$ is a new participating node,$\mathbb{A}$ will only perform such oracle queries once throughout every session. This inquiry produces bit t selected randomly once it detects the sessions keys $SK^n_{\wp_i}$ of $\mathfrak{S}^n_{\wp_i}$; else, it delivers a random number.

An algorithm is supposed to guarantee the protection of data aggregation by ensuring confidentiality, integrity, and authenticity, all of which are basic needs that adversaries potentially try to target.

- **Threats affects Confidentiality:** Adversaries choose either of the below attacks to acquire the accessibility of keys: known-plaintext, chosen-ciphertext, or chosen-plaintext attack. The aggregated records could be deciphered whenever this adversary acquires access to the key.

- **Threats against Integrity:** Adversaries could breach one or much more aggregators or sensors, which results in losing a few important medical data or aggregated records being modified with the purpose of disseminating an erroneous aggregation towards the Medical Server (like replays hit).
- **Threats against Authenticity:** There are two different kinds of threats that have been existed, which would bring authenticity to the risk range.
  - An attacker pretends to be an authentic patient's sensor or aggregating point thus, it injects fake information entering its network.
  - Adversaries pretend to be like true medical servers and inject false records across the network.

4.2.3 Security Goals

The suggested scheme for an IHS structure would satisfy the standards mentioned below;

1. ***Mutual authentication:*** All entity legitimacy must be authenticated before generating the session keys and transmitting data.
2. ***Confidentiality***: Patient health records must be restricted to access by only authorized nodes (eg, Medical Server).
3. ***Integrity***: All data must be provided with appropriate authenticity and integrity, which ensures that the data has not been tampered with during transmission.
4. ***Protecting Node Anonymity:*** Make sure that adversaries never acquire anything about the patient's or health professional's identities.
5. ***Intractability:*** Make sure that an invader is incapable to trace any target nodes.
6. ***Energy Consumption:*** The computational-communicational cost and energy overheads of schemes based on IoT applications should be sustainable, as IoT consists of resource-constrained devices.

4.3 Proposed Scheme

Under this section, we primarily outline all the steps that should be taken to provide an authentic and secure data aggregation scheme for an e-healthcare based IHS

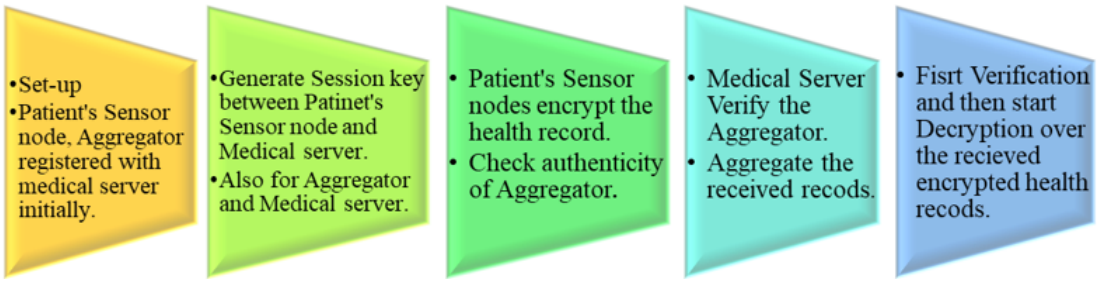


Figure 4.3: Steps involved in the proposed scheme

Table 4.1: Notations Used in Our Proposed Scheme.

Notations	Description
x	Master key of PKG
P	Generator of E/F_q
P_{Pub}	Public key of PKG
$H_i,\ i = 1, 2$	Secure Hash definition
q	Prime modulus
a, b	Random number belongs to
$s,\ r$	Ephemeral key belongs to
ID_{PS}	Patient sensor's Identity
ID_{MS}	Medical Server's Identity
ID_{AG}	Aggregator's Identity
$\lambda_{PS}, \Upsilon_{PS}$	Signature pair of Patient Sensor node.
$\lambda_{MS}, \Upsilon_{MS}$	Signature pair of Medical server Node
$\lambda_{AG}, \Upsilon_{AG}$	Signature pair of Aggregator Node.
K_{PS}	Key computed by patient's sensor node.
K_{MS}	Key computed by Medical server Node.
K_{AG}	Key computed by Aggregator.
SK_{PSMS}	Session Key generated for Patient's and medical server
SK_{AGMS}	Session Key generated for Aggregator and medical server

environment. Figure 4.3 depicted all involved steps in our proposed scheme, which has been consisting of five steps as follows: (i) Setup and Registration. (ii) Key Generation. (iii) Encryption and Authentication. (iv) Verification and Aggregation. (v) Verification and Decryption.

All notations used by our scheme are signified in Table 5.5.

Figure 4.4 represents the flow chart of the proposed scheme, which depicts that first the generation of a session key for a pair of patients with the server and for the aggregator with the server is performed. Then the Encryption using that key being done over the collected category-wise medical history of patients. Further authentication of the aggregator is being performed to check its authenticity. After this Aggregator generates a message S, which is forwarded to the server for validation with it. Then, after the validation of the aggregator, the server sends a permission message to begin aggregating the encrypted data and forwards it to the server. The server will accept this data only after its verification, else discarded.

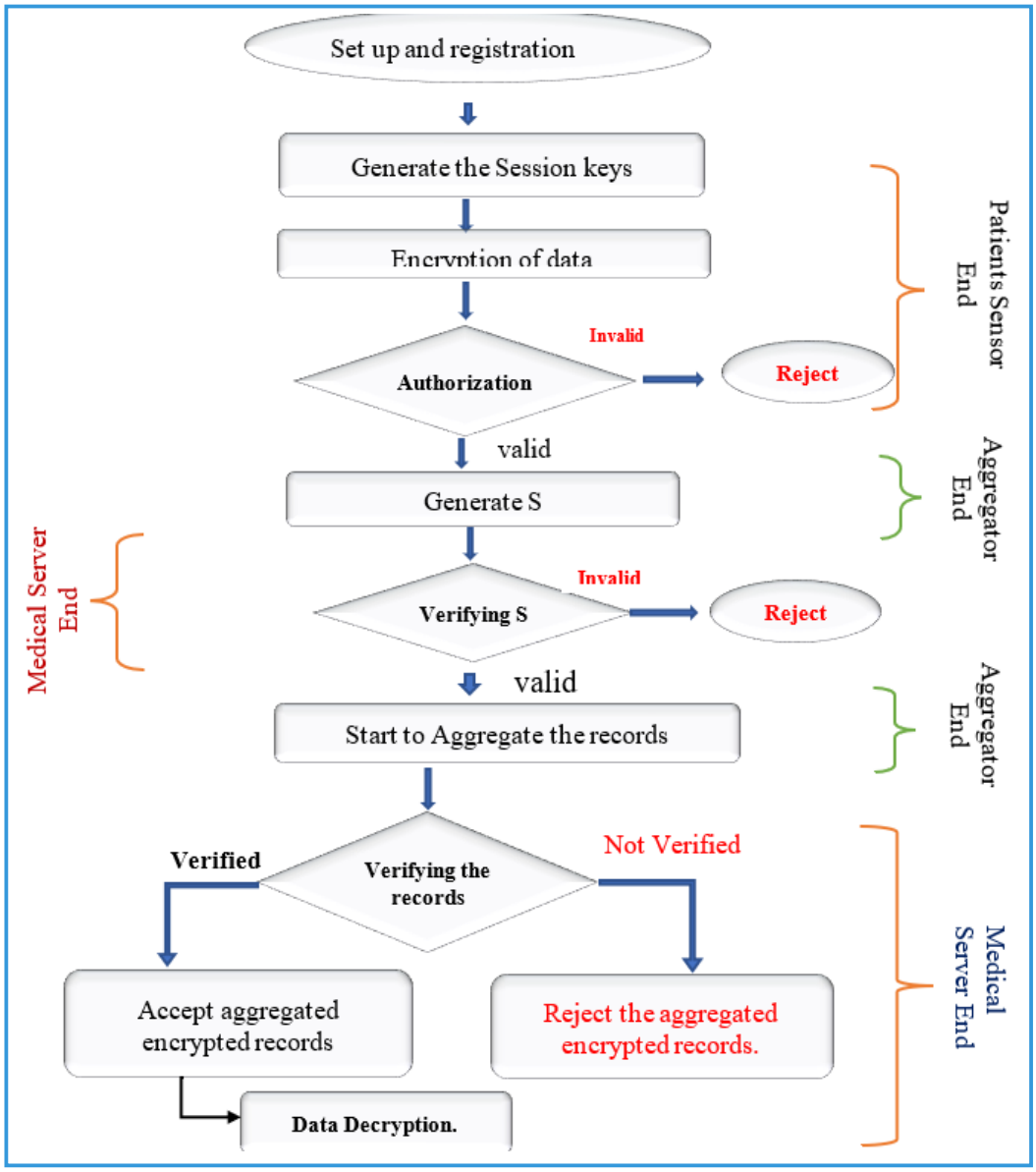


Figure 4.4: Work flow of all involved components in Proposed model.

4.3.1 Setup and Registration Phase

It has been founded that most health organizations instead of admitting patients, adopt the sensor-based management solution based on the doctor's endorsement. Therefore, Healthcare-Staff implant different types of sensors on the human body depending on the specific patient's medical data prerequisite by health monitoring team. Moreover, before going to implant the sensors, all the patients along with the aggregator should always be registered with the Medical Server. The Medical Server monitors the requested key by each sensor whenever the hardware setup is accomplished. The patient's sensor starts responding to the request once acquiring it from the Aggregator. By accomplishment the following steps we finish the setup process.

- Additive group G with prime q order define curve E/F_q with generator P of G.
- Select $x \in Z_q^*$ as master private key to determine the master public key P_{Pub}: $P_{Pub} = x.P$.
- Secure hash functions can be defined like:
  - $H_1 : \{0,1\}^* \times G \rightarrow Z_q^*$.
  - $H_2 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \times G \times G \rightarrow Z_q^*$.
- Now publish system parameter param as: $\{G, q, P, P_{Pub}, H_1, H_2\}$ as well as maintaining the secrecy of master key x.

Key Extraction: With the Patient's sensor (PS) identity: $ID_{PS} \in 0, 1^*$, PKG generate long term private keys.

- PKG select a random number $a \in Z_q^*$
  then computes $U_{PS}, h_{PS}, and V_{PS}$ as follows:
  - $U_{PS} = a \cdot P$
  - $h_{PS} = H_1[ID_{PS} || U_{PS}]$
  - $V_{PS} = a + h_{PS} \cdot x$
- Sends these tuples $[V_{PS}, U_{PS}]$ to Patient's Sensor *(PS)* via a secure channel.
- PKG validate its private key if $(P_{PS} = U_{PS} + H_1[ID_{PS} || U_{PS}]P_{Pub} = V_{PS} \cdot P)$. Similarly, for Aggregator AG and Medical Sensor also received values such as:

Patient's Sensor (PS)	Medical Server (MS)
Step 1: i. Selects an ephemeral number: $s \in Z_q^*$ ii. Computes: $T_{PS} = s \cdot P$, $\lambda_{PS} = [s + h_S]^{-1} \cdot V_{PS}$. $\Upsilon_{PS} = V_{PS} \cdot P$. iii. Send the tuple set $[ID_{PS}, T_{PS}, \Upsilon_{PS}, \lambda_{PS}, U_{PS}]$ towards the Medical Server MS	
	Step 2: i. First MS Verify the Patient's Identity: ii. Check if $(\lambda_{PS} \cdot [T_{PS} + H_1[ID_{PS}\|\|U_{PS}]P] == \Upsilon_{PS})$ In Case of false, it rejects the Patient's Request. Otherwise, Authentication was done successfully. iii. Now, MS selects an ephemeral number: $r \in Z_q^*$ iv. Computes: $T_{MS} = r \cdot P$, $\lambda_{MS} = [r + h_S]^{-1} \cdot V_{MS}$. and $\Upsilon_{MS} = V_{MS} \cdot P$. v. Send the tuple set $[ID_{PS}, T_{PS}, \Upsilon_{PS}, \lambda_{PS}, U_{PS}]$ towards PS vi. Also, Computes Key K_{MS} as: $K_{MS} = (r+V_{MS})\ (T_{PS}+U_{PS}+H_1[ID_{PS}\|\|U_{PS}] \cdot P_{Pub})$ vii. Generates it's the Session Key SK_{MS} as: $SK_{MS} = H_2[ID_{PS}\|\|ID_{MS}\|\|T_{PS}\|\|T_{MS}\|\|\lambda_{PS}\|\|\lambda_{MS}\|\|K_{MS}]$
Step 3: i. First PS Verify the Medical Server's Identity: ii. Check if $(\lambda_{MS} \cdot [T_{MS} + H_1[ID_{MS}\|\|U_{MS}]P] == \Upsilon_{MS})$ iii. In Case of false, it rejects the Medical Server's Request. Otherwise, it starts to compute the Session key after the verification is done successfully. iv. First, Compute Key K_{PS} as: $K_{PS} = (s+V_{PS})\ (T_{MS}+U_{MS}+H_1[ID_{MS}\|\|U_{MS}] \cdot P_{Pub})$ v. Generate the Session Key SK_{PS} as: $SK_{PS} = H_2[ID_{PS}\|\|ID_{MS}\|\|T_{PS}\|\|T_{MS}\|\|\lambda_{PS}\|\|\lambda_{MS}\|\|K_{PS}]$	

Figure 4.5: Steps in Mutual Authentication and Key Agreement Phase in Proposed Scheme.

- $U_{AG} = b \cdot P$, and $U_{MS} = c \cdot P$
- $h_{AG} = H_1[ID_{AG}||U_{AG}]$, and
  $h_{MS} = H_1[ID_{MS}||U_{MS}]$
- $V_{AG} = b + h_{AG} \cdot x$, and $V_{MS} = b + h_{MS} \cdot x$

4.3.2 Session Key Generation

We first compute an agreement key between the patient and the medical server, which are mutually authenticated to each other. Moreover, this generated agreement key is being used to encrypt all collected patients' health records by the sensor at the patient's endpoint. After that, similarly, we also compute an authenticated agreement key between the aggregator and the medical server for the secure transmission of aggregated data.

4.3.2.1 For Patient and Medical server

- At Patient PS end- it selects an ephemeral number $s \in Z_q^*$ and compute: T_{PS} = s ·P, $\lambda_{PS} = [s + h_S]^{-1} \cdot V_{PS}$ and $\Upsilon_{PS} = V_{PS} \cdot P$. Further, sends the tuple set $[ID_{PS},\ T_{PS},\ \Upsilon_{PS}, \lambda_{PS}, U_{PS}]$ towards the Medical Server MS to generate the session key.

- *At Medical Server MS* end- after receiving the tuples, Medical server MS needs to verified the Patient's sensor authenticity. For this Medical server computed: $\lambda_{PS} \cdot [T_{PS} + H_1[\ ID_{PS}||U_{PS}]P]$ and check is it equal to the received value Υ_{PS}? if it holds then verification of Patient's sensor is done successfully, otherwise reject its request for generating agreement key. After authentication of patient's sensor PS is done successfully, Medical server MS again choose an ephemeral key $r \in Z_q^*$ to compute $T_{MS} = r \cdot P$ and generate $\lambda_{MS} = [r + h_{MS}]^{-1} \cdot V_{MS}$ and $\Upsilon_{MS} = V_{MS} \cdot P$. At this state, Medical server MS sends tuples: $[ID_{MS}, T_{MS}, \Upsilon_{MS}, \lambda_{MS}, U_{MS}]$ towards the Patient's Sensor and simultaneously generates the session key as:

$$SK_{MS} = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K_{MS}]$$

  , only after the computation of key i.e. $K_{MS} = (r + V_{MS})\ (T_{PS} + U_{PS} + H_1[\ ID_{PS}||U_{PS}] \cdot P_{Pub})$ where $H_1[ID_{PS}||U_{PS}] = h_{PS}$.

- *Now at the Patient PS side*, after verifying the medical server's identity by computing the value of expression i.e. $\lambda_{MS} \cdot [T_{MS} + H_1[\ ID_{MS}||U_{MS}]P]$. If this calculated value is equivalent to Υ_{MS} then authentication is done successfully, otherwise patient reject the request to generate the session key as authentication failed. After medical server's authentication completed, Patient's sensors computes the key K_{PS} =$(s + V_{PS})(T_{MS} +\ U_{MS} + H_1[\ ID_{MS}||U_{MS}] \cdot P_{Pub})$ and session key by using it as:

$$SK_{PS} = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K_{PS}]$$

Where $H_1[ID_{MS}||U_{MS}]$ =h_{MS}. Figure 6.4 demonstrates the working of the proposed key agreement scheme in the IHS environment, where we can see how the patient and medical server start to compute the session key only if authentication verification is completed successfully at both ends, else they reject the request at their end accordingly.

Table 4.2: Encryption of collected health records using session key SK_{PSMS}.

Input: $M_i, N_i,\ SK_{PSMS}$ **Output:** $C_i,\ h_i$
1. Map all M_i into the points of elliptic curve P_i. *i. For the Emergency condition:* a. Compute: $C_i^{EC} = E_{SK_{PSMS}}(P_i \| \| N_i)$. b. Compute: $h_i^{EC} = h(C_{i,} \| \| M_i)$. *ii. For the Vital condition:* a. Compute: $C_i^{VC} = E_{SK_{PSMS}}(P_i \| \| N_i)$. b. Compute: $h_i^{VC} = h(C_{i,} \| \| M_i)$. *ii. For the Regular condition:* a. Compute: $C_i^{RC} = E_{SK_{PSMS}}(P_i \| \| N_i)$. b. Compute: $h_i^{RC} = h(C_{i,} \| \| M_i)$. 2. Send $(C_i^{EC} \| \| \ h_i^{EC}\ ), (C_i^{VC}\ \| \ h_i^{VC})$ and $(C_i^{RC} \| \ h_i^{RC}\ )$ to aggregator.

4.3.2.2 For Aggregator and Medical Server we also generate the key between the aggregator and medical server similarly, as the session key being generated between the patient and medical server:

$SK_{AGMS} = SK_{AG} = H_2[ID_{AG}||ID_{MS}||T_{AG}||T_{MS}||\lambda_{AG}||\lambda_{MS}||K'] = SK_{MS}$. And $K' = K_{AG} = K_{MS} = (r \cdot t \cdot P + \ r \cdot V_{AG} \cdot P + t \cdot V_{MS} \cdot P + V_{MS} \cdot V_{AG} \cdot P)$. $K_{MS} = (r + V_{MS})\ (T_{PS} + U_{PS} + H_1[\ ID_{PS}||U_{PS}] \cdot P_{Pub})$

$SK_{MS} = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K_{MS}]$ $K_{PS} = (s + V_{PS})\ (T_{MS} + U_{MS} + H_1[\ ID_{MS}||U_{MS}] \cdot P_{Pub})$

$SK_{PS} = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K_{PS}]$

4.3.3 Encryption and Authentication

Confidentiality and anonymity are essential requirements for the effective data aggregation, which assures that information flowing through the internet could not be examined by unauthorized entities. Thus, to preserve the point-to-point data secrecy the homomorphic encryption (Homo Encryp) techniques being used in this scheme. The main benefit of using Homo Encyp would be that it enables complicated mathematics's calculations over the ciphertexts despite of revealing the details of initial plain data. Thus, confidentiality and anonymity being preserved as all computations has been performed over the encrypted plaintext. Such that, we can say that all the communication done between the patient and medical server are safe and secure with respect to any nasty amendment or unauthorized access.

Table 4.2 described about the encryption process performed using the session key SK_{PSMS} generated for the patient's sensor and medical server based on the

Table 4.3: Steps involved in Validation and Aggregation Process.

Validation Process
At Aggregator End: i. Generate S as: $S = E_{SK_{AGMS}}(ID_{AG})$ ii. Send this computed S towards the Medical Server. **At Medical Server End:** i. Generate $S^{'}$ as: $S^{'} = E_{SK_{AGMS}}(ID_{AG})$ ii. Cross check this computed $S^{'}$ with received S: if $(S == S^{'})$ Accepted as it is Authorized Aggregator. else Rejected as Aggregator Validation Failed.
Aggregation Process
Input:$(C_i^{EC} \| h_i^{EC})$, $(C_i^{VC} \| h_i^{VC})$ and $(C_i^{RC} \| h_i^{RC})$ For Nonce N, Ciphertext C_i and MAC h_i, Aggregator start computation as follows: *i. For the Emergency Condition:* a. Compute: $C_{AG}^{EC} = \sum_{i=1}^{N} C_i^{EC}$ b. Compute: $h_{AG}^{EC} = \oplus\, h_i^{EC}$ *ii. For the Vital Condition:* a. Compute: $C_{AG}^{VC} = \sum_{i=1}^{N} C_i^{VC}$ b. Compute: $h_{AG}^{VC} = \oplus\, h_i^{VC}$ *iii. For the Regular Condition:* a. Compute: $C_{AG}^{RC} = \sum_{i=1}^{N} C_i^{RC}$ b. Compute: $h_{AG}^{RC} = \oplus\, h_i^{RC}$

category-wise data. Correspondingly, use the nonce to retrieve the information about the freshness of data. And also compute MAC h as $hi(Ci||Mi)$, which is used to validate the integrity of all transferred data category-wise, respectively. Before transmitting the encrypted pair of $(C_i||h_i)$ towards the aggregator, we need to check its authenticity. Thus, we use here password-based authentication or biometric-based mutual authorization techniques. After checking the authenticity of the aggregator, we direct all encrypted data $(C_i||h_i)$ towards the aggregator.

4.3.4 Validation and Aggregation

Aggregator forwarded a message S towards the medical server only after the mutual authentication has been done between patients' sensor and aggregator. This message S is generally generated using the agreement key SK_{AGMS} to validate the aggregator

with respect to server, which specify as:

$$S = E_{SK_{AGMS}}(ID_{AG}) \tag{4.1}$$

Next, the S' is computed at the end of the medical server using the identity of the aggregator ID_{AG} and session key SK_{AGMS}. Check it with the received S value; if it doesn't match, then the medical server prevents the Aggregator from accessing the networks, whether it is hostile and unauthenticated. Next, the Medical Server directs the authorized message to the Aggregator when the received and computed value are matched. The Aggregator begins its Information Aggregating process once received a permission notification from the same Medical Server. The data aggregation process of the proposed scheme is based on information prioritization. In fact, the specific ciphertexts having similar information priorities must be concatenated instead of concatenating all different priority-based data together. At the end category wise Aggregated computed output $(C_{AG}^{EC}, h_{AG}^{EC})$,$(C_{AG}^{VC}, h_{AG}^{VC})$ and $C_{AG}^{RC}, h_{AG}^{RC})$ should transferred to the medical server via secure channel.
Table 4.3 depicts how, only after the aggregator validation process, the aggregator will start to aggregate all the received health data by patients based on priority. Thus, prior to any significant health information being transferred, the sensors, aggregator, and server all mutually authenticate throughout such a process.

4.3.5 Verification and Decryption

The medical server initiates the decryption and verification process after getting the aggregated information. First, the medical server performs decryption of received encrypted aggregated data, and Next, verifies the point-to-point integrity. All the involved computation is signified by the following step, working as follows:

- Perform the decryption over received aggregated data $(C_{AG}^{EC}, h_{AG}^{EC})$, $(C_{AG}^{VC}, h_{AG}^{VC})$ and $(C_{AG}^{RC}, h_{AG}^{RC})$ using the session key SK_{PSMS}, which is generated for patient and server and Compute $M\prime$ such as: $M\prime = D_{SK_{PSMS}}(C_{AG})$.

- Next, compute h' as $h'= h(M_i||C_i)$. and verify it with received h_{AG}. Check whether $h' = (h_{AG})$ holds or not. The aggregated data would be approved only if the verification is done successfully; otherwise, it will be discarded by the medical server. Afterward, these approved records should be accessible to numerous healthcare organizations, such as hospitals, doctors, etc.

Figure 4.6 illustrates the stepwise working procedure regarding the verification and decryption procedure as per above in steps i and ii.

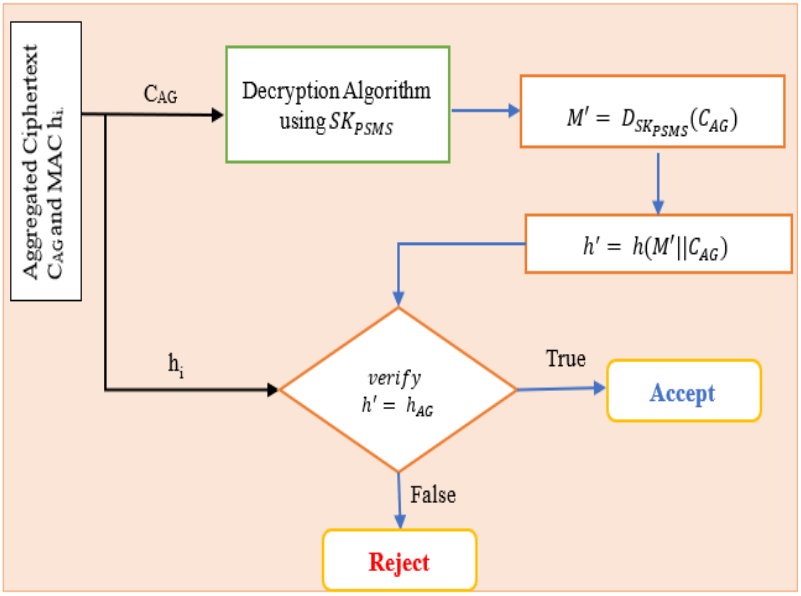


Figure 4.6: Block diagram of Verification and Decryption process.

4.4 Security Analysis and Proof

In this section, we will discuss the correctness of our proposed scheme and its formal analysis.

4.4.1 Correctness of Scheme

With the help of a few theorems, we will prove the correctness of our work.

Theorem 4.1. *When both patient PS and medical server MS follow the proposed scheme, then both can compute the same key* $K_{MS} = K_{PS}$.

Proof. Patients and medical servers can transmit their information over networks using this session key, $SK_{(PS,MS)}$ as both ends generate the same session key. This will be possible only when the used key K_{PS} and K_{MS} value must be equal. Below, we also present the proof that these values are equal. At the medical server side: Computes the key as follows:

K_{MS}
$= (r + V_M S)(T_P S + U_P S + H_1[ID_{PS}||U_{PS}] \cdot P_P ub$
$= (r + V_{MS})(s \cdot P + a \cdot p + h_{PS} \cdot x \cdot P)$
$= (r + V_{MS})(s + V_{PS})P$
$= (s + V_{PS})(r \cdot P + b \cdot p + h_{MS} x \cdot P)$
$= (s + V_{PS})(T_{MS} + U_{MS} + h_{MS} \cdot P_P ub)$

$= K_{PS}$ (at Patient side).
Thus, we can write:
$K_{PS} = K_{MS} = K = (r \cdot s \cdot P + r \cdot V_{PS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P)$. □

Theorem 4.2. *Both patient PS and medical server MS must be computing the same session key before starting the communication between them.*

Proof. As per the above-mentioned theorem, we found that both node patient and the medical server compute the same key K at their respective ends as: $K_{MS} = (r + V_{MS})(s + V_{PS})P, K_{PS} = (s + V_{PS})(r + V_{MS})P$. Therefore both patients and medical server can start the computation of the session key at their respective eends,which are correct and equal too, such as; $SK_{PS} = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K]$ $= SK_{MS}$.

Thus, we can write: $SK_{PSMS} = SK_{PS} = SK_{MS}$, hence the session key being generated by both endsthe to communicate with each other is the same. □

4.4.2 Provable Security Analysis

Formal analysis of the proposed scheme has been covered under this subsection of the security analysis part.

Theorem 4.3. *Let for a PPT attacker $\mathbb{A}'s$ CDH and ECDL issues are numerically challenging tasks, then ROM accomplishes concerning AKA as well as MA security for the proposed scheme.*

Proof. Imagine a PPT adversarial node $\mathbb{A}$ participates in a Defender-responses game-scenario with a defender $\mathfrak{D}$ in order to compromising AKA and MA security that are semantic protections of proposed scheme. $\mathbb{A}$ will plan to win this competition by resolving the ECDL and CDH difficulties. Therefore, $\mathfrak{D}$ first provides the global parameters: $= (G, q, P, P_Pub, H_1, H_2)$ to the adversary $\mathbb{A}$. In order to answer for $\mathbb{A}'s$ query, $\mathfrak{D}$ preserves the three major preliminary empty lists.

1. H_1_list: It is necessary to answer about the H_1 query. Thus, it holds few sequences of tuples, such as (ID_i, h_i, r, s, t_i) where $ID_i \in \{ID_{PS}, ID_{MS}\}$, similarly consider for h_i.

2. Pb_list: In order to respond against the public key inquiry, defender need this collection. Therefore, tuples' sequence such as; $(ID_i, \mathfrak{S}^n_{\wp_i}, T_i)$ are maintained in Pb_list. Where $\wp_i \in \{PS, MS\}$.

3. *SK_list*: Defender preserves this list compulsorily, which hold tuples as $\mathfrak{S}^n_{\wp_i}$, $SK^n_{\wp_i}$, to answer against the session key inquiry by adversary $\mathbb{A}$. Where $SK^n_{\wp_i}$ $\in$ { SK^n_{PS}, Sk^n_{MS}}

□

Defender $\mathfrak{D}$ will be responded against all oracle query raised by attacker $\mathbb{A}$. The following are the answers of all queries.

1. H_1Query: Initial status of the H_1*_list* is empty and such type of query will be answered by help of H_1*_list* only. Whenever $\mathbb{A}$ rise such query with ID_i at most d times then defender $\mathfrak{D}$ will be answered as subsequent fashion;

   (a) *Defender check whether this* ID_i *already belongs to* H_1*_list or not?* If it exists then defender $\mathfrak{D}$ do the following computation and send the values to attacker.

      i. Check $ID_i = ID_{PS}$ if it is true then $\mathfrak{D}$ randomly choose $t_1 \in Z^*_q$, to calculate h_{PS} =$t_1.P$. Now $\mathfrak{D}$ forward this h_{PS} to $\mathbb{A}$ and also add a row i.e. $(ID_{PS}, h_{PS}, r, t_1)$ in H_1*_list*.

      ii. Check $ID_i = ID_{MS}$ if it is true then $\mathfrak{D}$ randomly choose $t_2 \in Z^*_q$, to calculate h_{MS}=$t_2.P$. Now $\mathfrak{D}$ forward this h_{MS} to $\mathbb{A}$ and also add a row i.e. $(ID_{MS}, h_{MS}, r, t_2)$ in H_1*_list*.

      iii. When $ID_i \neq ID_{PS}$ or ID_{MS} then $\mathfrak{D}$ replied to $\mathbb{A}$ with value T_i.

   (b) Whenever $\mathfrak{D}$ found that $ID_i \neq ID_{PS}$ or ID_{MS}. Then $\mathfrak{D}$ randomly pick a number $t \in Z^*_q$ and forwards the calculated value h_i=$t.P$ to $\mathbb{A}$. Defender also add the row (ID_i, h_i, t) in H_1*_list*.

2. *Pb Query*: Attacker $\mathbb{A}$ executes the *Pb* queries to discover the information regarding the participating node's public key of the proposed scheme. The above query's pass $(ID_i, \mathfrak{S}^n_{\wp_i})$ as an argument, after receiving this inquiries defender $\mathfrak{D}$ will respond as follows.

   (a) Defender check whether $(ID_i, \mathfrak{S}^n_{\wp_i}, h_i)$ exists in *Pb*_list or not? Defender $\mathfrak{D}$ answered with h_i if it successfully found in list.

   (b) Apart from above case, Defender $\mathfrak{D}$ randomly select a value $e \in Z^*_q$ to fetch the h_i= $e.P$ and send it to Attacker $\mathbb{A}$. Next $\mathbb{A}$ add the row $(ID_i, \mathfrak{S}^n_{\wp_i}, h_i)$ in the *Pb_list* and simultaneously also add row (ID_i, h_i, ψ) into H_1*_list*.

3. *Send* $(\mathfrak{S}^n_{\wp_i}, m)$ *Query*: Defender $\mathfrak{D}$ will answered about this oracle inquiry in following manner:

   (a) Check whether (m=ψ) or not? If it not true then defender will be replied as per the well-defined proposed scheme.

   (b) While in the case of (m=ψ), defender $\mathfrak{D}$ looks for the tuple (ID_i, t_i, h_i) on the particular($\mathfrak{S}^n_{\wp_i}$ into the H_1_list and also examines for the node$\wp_i$, which is presumably paired to $\mathfrak{D}$ looks for the tuple (ID_i, t_i, h_i) on the particular ($\mathfrak{S}^n_{\wp_i}$ inside the H_1_list, is available or not? $\mathfrak{D}$ will initiate H_1 oracle query using identity ID_i only when it found that tuple (ID_i, t_i, h_i) don't exist in list. Finally, $\mathfrak{D}$ responds as such;

      i. If the supplied instance $\mathfrak{S}^n_{\wp_i}$ differs from the instance that is coupled with $\mathfrak{S}^n_{PS}$ or $\mathfrak{S}^n_{MS}$, then $\mathfrak{D}$ quantifies $T_i = t_i \cdot P$, $\lambda_i = [t_i + h_i]^{-1} \cdot V_i$ and $\Upsilon_{\mathrm{i}} = V_i \cdot P$, where $t_i \in Z_q^*$ with $i = 1, 2$.

      ii. If the supplied instance $\mathfrak{S}^n_{\wp_i}$ same with the instance that is coupled with $\mathfrak{S}^n_{\wp_{PS}}$ or $\mathfrak{S}^n_{\wp_{MS}}$, then $\mathfrak{D}$ computes $T_{PS} = s \cdot P$, $\lambda_{PS} = [s + h_S]^{-1} \cdot V_{PS}$ and $\Upsilon_{\mathrm{PS}} = V_{PS} \cdot P$ and $T_{MS} = r \cdot P$, $\lambda_{MS} = [r + h_{MS}^{-1} \cdot V_{MS}$ and $\Upsilon_{\mathrm{MS}} = V_{MS} \cdot P$, where $r, s \in Z_q^*$ for $i = 1, 2$ respectively. And at the end $\mathfrak{D}$ will send $\lambda_{PS}, \Upsilon_{\mathrm{PS}}$ and $\lambda_{MS}, \Upsilon_{\mathrm{MS}}$ to attacker $\mathbb{A}$.

4. *Reveal Session key* $\mathfrak{S}^n_{\wp_i}$ *Query*: The list SK_list is originally null. Once $\mathfrak{D}$ receives such inquiry, it replies in the prescribed sequence such as;

   (a) $\mathfrak{D}$ examines SK_list and delivers the key $SK^n_{\wp_i}$ to $\mathbb{A}$ once it found that tuples $(\mathfrak{S}^n_{\wp_i}, SK^n_{\wp_i})$ in such list.

   (b) $\mathfrak{D}$ examines SK_list and provides the key $SK^n_{\wp_j}$ to $\mathbb{A}$ once the tuple $(\mathfrak{S}^n_{\wp_i}, SK^n_{\wp_j})$ is detected in list, where $\mathfrak{S}^n_{\wp_i}$ is coupled with the instance $\mathfrak{S}^n_{\wp_i}$.

   (c) Furthermore, even though prior expectations were not fulfilled, then $\mathfrak{D}$ selects the key exchange $SK^n_{\wp_j} \in G$ of $SK^n_{\wp_j}$ with randomness as well as transmits this to $\mathbb{A}$. The resulting set $(S^n_{\wp_j}, SK^n_{\wp_j})$ is subsequently inserted into SK_list.

5. *Reveal Random secret* (ID_i) *Query*: Defender $\mathfrak{D}$ examines the H_1_list, looks for the tuple $(ID_i,\ h_i,\ r,\ s,\ t_i)$, so that it can answer such an oracle request, where $ID_i \in ID_{PS},\ ID_{MS}$. Next $\mathfrak{D}$ will deliver the relevant (r *or* s) with respect to identity of node it founded in tuple. Apart from prior case, $\mathfrak{D}$ randomly chooses $t_i \in Z_q^*$ to calculate $h_i = t_i \cdot P$ and deliver it to $\mathbb{A}$. Thereafter, $\mathfrak{D}$ will added to the pair $(ID_i,\ h_i,\ r,\ s,\ t_i)$ into the H_1_list.

6. *Request Public Key* $(ID_i, \mathfrak{S}^n_{\wp_i})$ *Query*: In order to answer such query defender $\mathfrak{D}$ check whether row $(ID_i, \mathfrak{S}^n_{\wp_i},\ h_i)$ exists in *Pb_list or not?* Defender $\mathfrak{D}$ answered with h_i to attacker if row successfully found in list. Apart from above case, Defender $\mathfrak{D}$ randomly select a value $e \in Z_q^*$ to fetch the $h_i = e \cdot P$ and send it to attacker $\mathbb{A}$. Next $\mathfrak{D}$ add the row $(ID_i,\ \mathfrak{S}^n_{\wp_i},\ h_i)$ in the Pb_list and simultaneously also add row (ID_i, h_i, ψ) into H_1_list.

7. *Reveal State* $(\mathfrak{S}^n_{\wp_i})$ *Query*: In regard to this inquiry, whenever an appropriate entity $\mathfrak{S}^n_{\wp_i}$ belongs to the accepted state, then state $\mathfrak{L}^n_{\wp_i}$ will be delivered by defender. Otherwise, $\mathfrak{D}$ would return as an empty set to adversary $\mathbb{A}$.

8. *Test* $(\mathfrak{S}^n_{\wp_i})$ *Query*: This represents the game's concluding phase. Whenever $\mathfrak{D}$ accomplishes either of these below stated criteria, then the present league between defender and attacker will be terminated.

   (a) $\mathfrak{S}^n_{\wp_i} = \mathfrak{S}^n_{\wp_1}$ or $\mathfrak{S}^n_{\wp_i} = \mathfrak{S}^n_{\wp_2}$,

   (b) $\mathfrak{S}^n_{\wp_1}$or $\mathfrak{S}^n_{\wp_2}$ are never paired with $\mathfrak{S}^n_{\wp_i}$ in any way.

   (c) When $\wp_x \in \mathfrak{A}^n_{\wp_P S}$n or $\wp_x \in \mathfrak{A}^n_{\wp_P S}$ existed that answers for its reveal private key demand, indicating that $\wp_x$ is compromised.

   (d) When an instance $\mathfrak{S}^n_{\wp_i}$ nor $\mathfrak{S}^n_{\wp_i}$ or either of its couple required to inquired about either for reveal session-key or reveal state queries.

Apart from above case, if $t = 1$ is being achieved, where defender $\mathfrak{D}$ picks randomly a bit t then $\mathfrak{D}$ will returned session key $\mathfrak{S}^n_{\wp_i}$ of the instance $\mathfrak{S}^n_{\wp_i}$; otherwise, it returns any null strings. Next, performs the recommended scheme for $i = 1$ as well as $j = 2$. The attacker $\mathbb{A}$ sends the predicted to defender $\mathfrak{D}$ after the completion of the scheme. Further, the defender will estimates $T_{PS} = s \cdot P,\ \lambda_{PS} = [s + h_{PS}]^{-1} \cdot V_{PS}$ and $\Upsilon_{PS} = V_{PS} \cdot P,\ T_{MS} = s \cdot P,\ \lambda_{MS} = [s + h_{MS}]^{-1} \cdot V_{MS}$ and $\Upsilon_{MS} = V_{MS} \cdot P$ and $K = (r + V_{MS})\,(s + V_{PS})P = (r \cdot s \cdot P + r \cdot V_{PS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P)$. As a result, $\mathfrak{D}$ returns $(V_{PS} \cdot V_{MS} \cdot P)$. Moreover, by knowing pair $(P_{PS}, P_{MS}) = (V_{PS} \cdot P,\ V_{MS} \cdot P)$ the computation of $(V_{PS} \cdot V_{MS} \cdot P)$ is comes under the hard assumptions of difficulty to solve CDHP issues. Thus, introduced scheme that provides highly robust AKA security dealing with this CDH challenge. Next, defender $\mathfrak{D}$ compute and return to attacker $\mathbb{A}$ i.e. $K = r \cdot s \cdot P + r \cdot V_{PS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P$ and then he may know about the $\lambda_{PS} = [s + h_{PS}]^{-1} \cdot V_{PS}$ and $\lambda_{MS} = [r + h_{MS}]_{-1} \cdot V_{MS}$. Once V_{PS} and V_{MS} are published, $V_{MS} \cdot V_{PS} \cdot P$ can be calculated by $\mathbb{A}$, but not $r \cdot s \cdot P,\ r \cdot V_{PS} \cdot P$ and $s \cdot V_{MS} \cdot P$ as s and r are unknown to $\mathbb{A}$. Therefore, adversary was unable to obtain the s and r values from T_{PS} and T_{MS} due to strict assumption that it would be difficult to solve the ECDL problem in polynomial time PPT. Thus, introduced

scheme is a provable secure protocol under ROM.
The following are the definitions of coefficients; $\mathfrak{D}_1$, $\mathfrak{D}_2$, and $\mathfrak{D}_3$:

1. $\mathfrak{D}_1$: It's really activated whenever the game was terminated via $\mathfrak{D}$.
2. $\mathfrak{D}_2$: When $\mathbb{A}$ makes an H_1 inquiry, it activates.
3. $\mathfrak{D}_3$: When Defender $\mathfrak{D}$ properly answered every H_1Query utilizing the H_1_list.

Lemma 4: $Pb[\mathfrak{D}_1] \geq [1/Q_n]$, where the count of send queries is expressed using Q_n.

Proof. few distinct events are represented as;

1. *E1*: This indicates that $\mathbb{A}$ is not performed an *Reveal Session key* ($\mathfrak{S}^n_{\wp_i}$) or *Reveal State* ($\mathfrak{S}^n_{\wp_{PS}}$) *query* for $\mathfrak{S}^n_{\wp_{PS}} = \mathfrak{S}^n_{\wp_{MS}}$ or $\mathfrak{S}^n_{\wp_{PS}}$ will coupled with $\mathfrak{S}^n_{\wp_{MS}}$.
2. *E2:* It denotes the absence of a corrupted node $\wp_i \in \mathfrak{A}^n_{\wp_{PS}}$.
3. *E3:* It denotes how $\mathbb{A}$ usually chooses $\mathfrak{S}^n_{\wp_{MS}}$ or its partners for the next challenge request.

Thus, it states as $\mathfrak{D}_1 = E1 \cap E2 \cap E3$. As here $\mathbb{A}$ usually chooses $\mathfrak{S}^n_{\wp_{MS}}$ or its partners for the next challenge request hence it got information about absence of a corrupted node $\wp_i \in \mathfrak{A}^n_{\wp_{PS}}$ along with that $\mathbb{A}$ is not performed an *Reveal Session key* ($\mathfrak{S}^n_{\wp_i}$) or *Reveal State*($\mathfrak{S}^n_{\wp_{PS}}$) *query* for $\mathfrak{S}^n_{\wp_{PS}} = \mathfrak{S}^n_{\wp_{MS}}$ or $\mathfrak{S}^n_{\wp_{PS}}$ will coupled with $\mathfrak{S}^n_{\wp_{MS}}$. Thus, $E3 = E1 \cap E2$, and $\mathfrak{D}_1 = E1 \cap E2 \cap E3$. As a result, $Pb[\mathfrak{D}_1] \geq [1/Q_n]$ is obtained for participating nodes. □

Lemma 5: $Pb[\mathfrak{D}_2] \geq 2 \in$

Proof. Let $\mathbb{A}$ don't ever makes H_1 inquiry indicates that $Pb[t = t' | \mathfrak{D}'_2] \geq \frac{1}{2}$. However, it should be observed that $|Pb(t = t') - \frac{1}{2}| \geq \in$. By considering the prior mentioned two inequities it would be deduced that;
$Pb[t = t'] = Pb[t = t'|D2']\ Pb[D2'] + Pb[t = t'|D2]\ Pb[D2]$
$\leq Pb[t = t' | \mathfrak{D}'_2]\ Pb[D'_2] + Pb[D_2]$.
$= \frac{1}{2}\ Pb[\mathfrak{D}'_2] + Pb[\mathfrak{D}_2]$.
$= \frac{1}{2} + \frac{1}{2}\ Pb[\mathfrak{D}_2]$ and $Pb[t = t'] \geq Pb[t = t'|D2']\ Pb[D2'] = \frac{1}{2}$ - $\frac{1}{2}\ Pb[\mathfrak{D}_2]$
Thus, it implies that $\frac{1}{2}\ Pb[\mathfrak{D}_2] \geq Pb[(t = t') - \frac{1}{2}] \geq \in$, hence $Pb[\mathfrak{D}_2] \geq 2 \in$.

□

Lemma 6: $Pb[\mathfrak{D}_3] \geq \frac{1}{HQ_n}$.

Proof. HQ_n is used to hold the count of responded H_1 Query. Hence the probability of each H_1 Query responded by defender $\mathfrak{D}$ properly is $Pb[\mathfrak{D}_3] \geq \frac{1}{HQ_n}$. □

Lemma 7: $Pb[\mathbb{A}(P, V_{PS} \cdot P, V_{MS} \cdot P)] = V_{MS} \cdot V_{PS} \cdot P : P \in G; V_{PS}, V_{MS} \in Z_q^* \geq \frac{\in}{Q_n H Q_n}$.

Proof. $\mathbb{A}$ needs to break the CDH problem to compute the value of $V_{MS} \cdot V_{PS} \cdot P$ from pair information $(V_{PS} \cdot P,\ V_{MS} \cdot P)$. Thus, the probability to compute the CDH issue will be;
$Pb[\mathbb{A}(P, V_{PS} \cdot P, V_{MS} \cdot P)] = V_{MS} \cdot V_{PS} \cdot P$: $P \in G : V_{PS}, V_{MS} \in Z_q^*$.
$= Pb[E1 \cap E2 \cap E3] \geq \frac{\in}{Q_n H Q_n}$. □

4.4.3 Informal Security Analysis

1. ***MITM Attack:*** In our proposed scheme, Patient send set of tuples to the medical server: $[ID_{PS},\ T_{PS},\ \Upsilon_{PS}, \lambda_{PS}, U_{PS}]$ (MS). Let there be the presence of an adversary $\mathbb{A}$ between them. Where the adversary computes $T'_{PS} = u \cdot P$ and sends modified tuples:
   $[ID_{PS},\ T'_{PS},\ \Upsilon_{PS}, \lambda_{PS}, U_{PS}]$ to MS. While receiving the tuples from any nodes, it MS needs to verify the identity of the sending node such that it computes: $\lambda_S \cdot [T'_S + H_1[\ ID_{PS}||U_{PS}] \cdot P] = \lambda_{PS} \cdot [u \cdot P + H_1[ID_{PS}||U_{PS}] \cdot P]$
   $= [s + h_{PS}]^{-1} \cdot V_{PS} \cdot [u \cdot P + h_{PS} \cdot P \neq \Upsilon_{PS}$.
   Thus, Server fails to identify the adversary node $\mathbb{A}$. Therefore, the server rejects the request to compute the session key by using received tuples forwarded by the adversary node and sends an authentication failed message to the adversary node.
   Similarly, if adversary send tuples to patient like $[ID_{MS},\ T'_{MS},\ \Upsilon_{MS}, \lambda_{MS}, U_{MS}]$ is also rejected by patients as it also unable to identifying the authentication of sender node $\mathbb{A}$. Similar for the aggregator and Server case. Hence, there is no possibility of man-in-the-middle attacks.

2. ***Known Provisionally Information Attack:*** In our proposed scheme the session key SK is computed as: $SK = H_2[ID_{PS}||ID_{MS}||T_{PS}||T_{MS}||\lambda_{PS}||\lambda_{MS}||K]$, where the secrecy of $K = r \cdot s \cdot P + r \cdot V_{PS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P$ has main role to maintain the secret of the session key SK. However, if the adversary achieves the ephemeral keys: r and s of the current session, still not able to determine the session key SK. Adversary $\mathbb{A}$ achieve session key SK only after determine the value $(V_{PS} \cdot V_{MS} \cdot P)$. Moreover, by knowing pair $(P_{PS}, P_{MS}) = (V_{PS} \cdot P,\ V_{MS} \cdot P)$ the computation of $(V_{PS} \cdot V_{MS} \cdot P)$ is comes under the hard assumptions of difficulty to solve CDHP issues. Similar for the aggregator and server case. Thus, our protocol can prevent the attacks while the provisional evidence is disclosed to the adversary.

3. ***Known Key Attack:*** Our proposed scheme's session key computation depends on two ephemeral values r and s. Such that no session key SK that is in the risk zone affect the new session key SK computation due to the computation of $r$and s from T_{PS}= $r \cdot P$ and T_{MS}= $s \cdot P$ comes under the computationally hard assumption of ECDL issues. Similar happen in the case of AG and MS. Hence, $\mathbb{A}$ is not able to acquire information about the other session key if there is any leakage of the existing session key.

4. ***Perfect Forward Security:*** An $\mathbb{A}$ failed to obtain the previous session key, although the private keys of nodes are compromised. The adversary $\mathbb{A}$ may determine the SK, accordingly it first calculate the $K = r \cdot s \cdot P + r \cdot V_{PS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P$ and then he must be know about the λ_{PS} =$[s + h_{PS}]^{-1} \cdot V_P S$ and λ_{MS} =$[r + h_{MS}]^{-1} \cdot V_{MS}$. Once V_{PS} and V_{MS} are published, $V_{MS} \cdot V_{PS} \cdot P$ can be calculated by $\mathbb{A}$ but not $r \cdot s \cdot P$, $r \cdot V_{PS} \cdot P$, and $s \cdot V_{MS} \cdot P$ as s and r are unknown to $\mathbb{A}$. Therefore, adversary was unable to obtain the s and r values from T_{PS} and T_{MS} due to strict assumption that it would be difficult to solve the ECDL problem in polynomial time. Alternatively, $\mathbb{A}$ tries to derive $r \cdot s \cdot P$, $r \cdot V_{PS} \cdot P$, and $s \cdot V_{MS} \cdot P$ from the pairs (T_{PS}, P_{MS}) =$(s \cdot P,\ V_{MS} \cdot P)$ and (T_{MS}, P_{PS}) = $(r \cdot P,\ V_{PS} \cdot P)$ and calculate $K = r \cdot s \cdot P + r \cdot V_{MS} \cdot P + s \cdot V_{MS} \cdot P + V_{MS} \cdot V_{PS} \cdot P$, λ_{PS} =$[s + h_{PS}]^{-1} \cdot V_{PS}$ and $\lambda_{MS} = [r + h_{MS}]^{-1}$. However, this is also not possible because of hard assumption of difficulty to unravelling CDHP. Similar for aggregator AG and serverMS. Thus, our protocol maintains perfect forward security successfully.

5. ***PKG Forward Security:***As we already proved in above point that $\mathbb{A}$ always failed to obtain the session key information with the use of previous session key due to hard assumption to solve the CDH and ECDL problem. However, all participants' secret key is going to be distressed by revealing the PKG's master key we can state that the current and prior key information will always remain safe. Thus, the PKG forward securities are conserved in proposed scheme.

6. ***Non Key Dominance***:In proposed scheme the session key is equally dependent on the input from the patients and server or aggregator and server. Therefore, none of them are able to dominant over the computation of session key. Thus, both either patient and server or aggregator and server not able dominant on each other's.

7. ***Confidentiality resistant:*** Sensor transmit encrypted form of all recorded health data by use of session key SK_{PSMS}. As it is already proved the robust

nature of key SK that can't be computed or accessed by the $\mathbb{A}$ due to hard assumption of CDH and ECDLproblem. Thus data maintained it confidentiality during travelling over channel. Only the authorized nodes like Medical server MS that have that same key will decrypt data and gain about the original message. Let there are eavesdropping being performed over the transmitted information at the aggregator AG end but it would be useless as aggregatorAG being used to aggregate the data without performing the decryption Thus, our scheme maintains the confidentiality successfully.

8. ***Integrity Resistant:***In proposed scheme we also focused on the integrity of nodes. Let $\mathbb{A}$ perform modifications over the encrypted data instead of accessing the data with the miscellaneous purposed. To prevent this server MS check whether ($h'_{AG} = h_{AG}$) *are matched or not* ? If it is not matched then discard the data, otherwise accept it as it conserved end to end data integrity successfully.

4.5 Performance Analysis

This section outlines the effectiveness evaluation of the proposed scheme's based on computational and communicational costs. Furthermore, the Multi-precision, Integer, Rational, and Arithmetic 'C' libraries (MIRACL) have been utilized to conduct each of the relevant computations to implement the proposed protocol [159]. However, MIRACL Crypto SDK is a 'C'-based library that has been labelled as "golden standard open-source SDK" for numerous complex cryptosystems by programmers. Thus, this library is widely used in the field of security research community to calculate the time it takes for various cryptographic procedures.

4.5.1 Simulation Setup

Simulation work had been done on a desktop having Intel core i7 9700 CPU@3.0 GHz and 8GB RAM processor setup with 64bit Ubuntu 20.04.4 LTS desktop i386 OS. For the Medical Sensor's activities, we used IoT appliance RaspberryPi with ARM Cortex-A53 CPU@1.4Ghz with RAM 1 GB, OS Kali Linux. A number of cryptographic functions were done hundred times on average, and the run-time was determined by computing the mean of such iterations. The suggested protocol is accomplished using a Type-A with a unique elliptic curve: $y^2 mod\ q = (x^3 + x) mod\ q$. This curve comprises a 512-b group, which provides analogous to a 1024-b Rivest–Shamir–Adleman (RSA) algorithm in level of protection.

Table 4.4: Execution time of various Cryptographic Operation.

Operation	**Description**	**Execution Time (*ms*)**
T_{PM}	Execution time for scalar Point multiplication operation in ECC like $a \cdot P$, where P is point,$a \in Z_q^*$	$2.0015ms$
T_{MP}	Execution Time for mapping of hash function like $\{0,1\}^* \; into \; Z_q^*$.	$2.0059ms$
T_{BM}	Execution time for Bilinear multiplicative Pairing operation	$4.0027ms$
T_M	Execution time for scalar multiplication like $a \cdot b, where \; a, b \; \in Z_q^*$	$00.78ms$
T_E	Execution time for exponential operation like a^n where $a \in Z_q^*$.	$8.0238ms$

Table 4.4 and Figure 4.7a, provide an overview of execution times of various cryptographic operations, which we used to compute the computational costs of existing work, including our proposed scheme. In general, the execution cost of hash:(T_H) and point addition: (T_A) operation will be neglected as these needs very tiny amount with respect to other operators like as: $T_H \approx 0.0001ms$ and $T_A \approx 0.0003ms$.

4.5.2 Computation-Communication Cost Analysis

Computational analysis of our scheme has been performed as follows:

- *Set Up:* In this we compute master Public key as: $P_{Pub} = x.P$, here it required $1T_{PM}$ to compute this key, which takes $2.0015ms$ and further after publishing the global variable PKG select a random number $a \in Z_q^*$ then computes $U_{PS}, h_{PS}, and V_{PS}$ as follows: $U_{PS} = a \cdot P$ need $1T_{PM}, h_{PS} = H_1[ID_{PS}|| \; U_{PS}]$ need $1T_{MP}$, $V_{PS} = a + \; h_{PS} \cdot x$ need $1T_M$ and one addition operation. Thus, PKG takes total time: $1T_{PM} + 1T_{MP} + 1T_M = 4.7874ms$ for each individual participating node.

- Next both nodes sensor and server need to mutually authenticate and generate the same session key as follows; first the computation of values: $T_{PS} = s \cdot P, \lambda_{PS} = [s + h_S]^{-1} \cdot V_{PS}$ and $\Upsilon_{PS} = V_{PS} \cdot P$ being performed at end of patients which required of $2T_{PM}$, and verify its identity at another end $\lambda_{PS} \cdot [T_{PS} + H_1[ID_{PS}||U_{PS}]P]$ and after verification Server node also compute and send it towards the patient $[ID_{MS}, \; T_{MS}, \Upsilon_{MS}, \lambda_{MS}]$. Further Key $K_{PS} = (s+V_{PS})(T_{MS} + U_{MS} + H_1[ID_{MS}||U_{MS}] \cdot P_{Pub})$ and session key SK as:$SK_{PS} =$

Table 4.5: Comparison of related scheme with proposed work based on Execution Cost of different phases.

Scheme	**Set-Up**	**PKG Phase**	**Session Key Agreement**
[39]	2.0015 ms	6.0089 ms	12.006 ms
[21]	2.0015 ms	4.7874 ms	32.0728 ms
[27]	2.0015 ms	4.7874 ms	12.7978 ms
[160]	2.0015 ms	6.7889 ms	12.0090ms
[18]	2.0015 ms	2.0015 ms	14.0093 ms
[8]	2.0015 ms	4.7874 ms	24.3741 ms
Proposed	2.0015 ms	4.7874 ms	8.0060 ms

Table 4.6: Comparison of Existing Scheme with Proposed Scheme

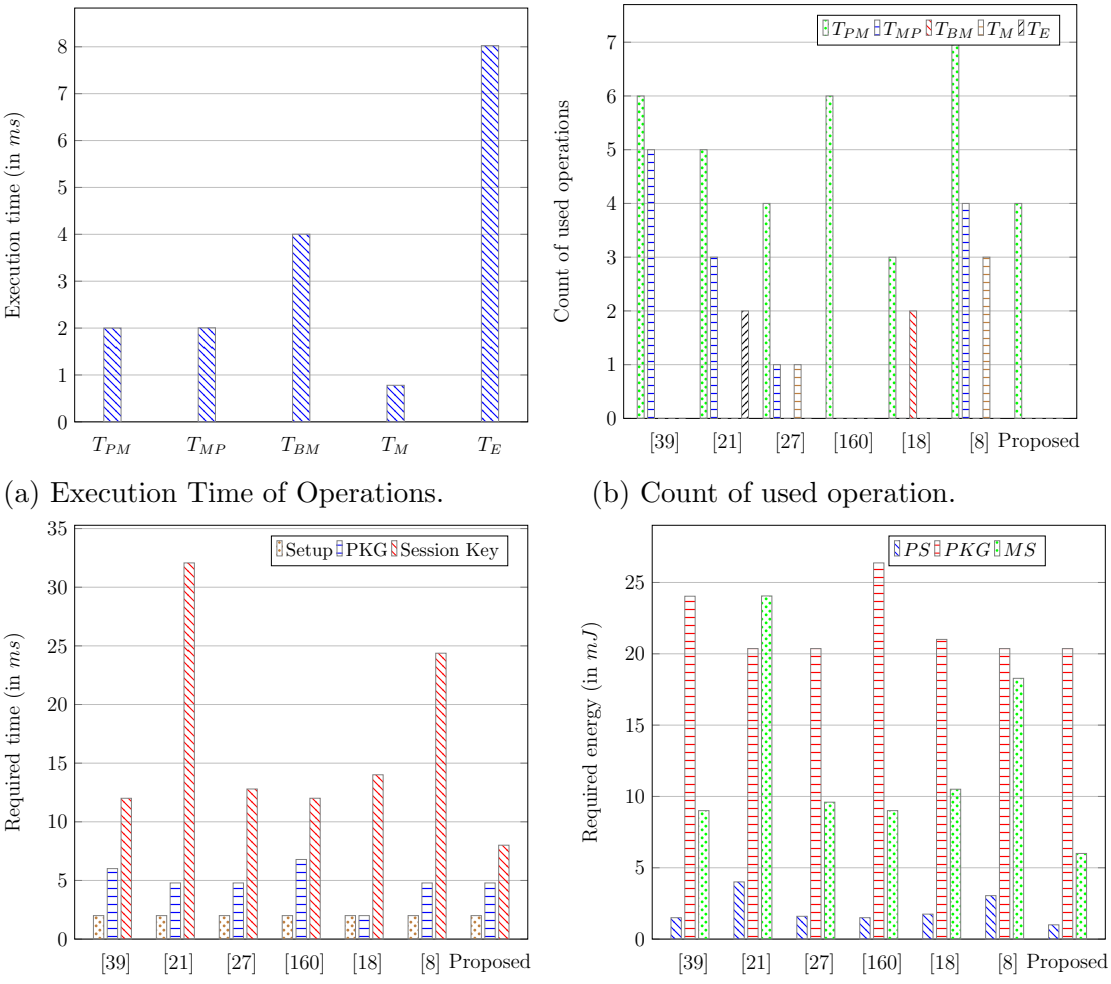
Scheme	**Communicational Cost**	**Computational Cost**	**Hardness Considered**	**Provable Model**	**MA**
Zhang et al. [39]	$2\|G_q\|$	$6T_{PM} + 4T_{MP}$	*CDH, OGDH*	CLASC	*Yes*
Islam et al. [21]	$2\|G_q\|$	$5T_{PM}$+$3T_{MP}$+ $2T_E$	*CDH, ECDL*	BAN	*Yes*
Dang et al. [27]	$2\|G_q\|$	$4T_{PM}$ + $1T_{MP}$ + $1T_M$	*CDH*	eCK	*No*
Wang et al. [160]	$2\|G_q\|$	$6T_{PM}$	*CDH*	ROM	*No*
Gupta et al. [18]	$2\|G_q\|$	$3T_{PM} + 2T_{BM}$	*CDH, BDH*	ROM	*Yes*
Cheng et al. [8]	$2\|G_q\|$	$7T_{PM}$ + $4T_{MP}$ + $3T_M$	*CDH, ECDH*	eCK	*Yes*
Proposed	$2\|G_q\|$	$4T_{PM}$	*CDH, ECDH*	ROM	*Yes*

Table 4.7: Energy overhead of Proposed Scheme compared to existing schemes.

Entity	[39]	[21]	[27]	[160]	[18]	[8]	Proposed
PS	1.5008	4.0091	1.5997	1.5011	1.7511	3.0468	1.0007
MS	9.0050	24.055	9.5984	9.0068	10.5069	18.2805	6.0045
PKG	24.0312	20.3667	20.3667	26.3712	21.009	20.3667	20.3667

$H_2[ID_{PS}||ID_{MS}||T_{PS}$
$||T_{MS}||\lambda_{PS}||\lambda_{MS}||K_{PS}]$. This computation needs to perform $2T_{PM}$ only. Thus, session key being compute finally with the needs of $4T_{PM}$ i.e. $8.006ms$ only.

- During the generation of session key, the participating nodes Patient PS sends $(ID_{PS},\ T_{PS},\ \Upsilon_{PS}, \lambda_{PS})$ towards server and Medical server MS also share $(ID_{MS},\ T_{MS},\ \Upsilon_{MS},\ \lambda_{MS})$ with Patient only after verification. Hence, communicational cost will be $2|G|$, size of G will be denoted as $|G|$.



(a) Execution Time of Operations.

(b) Count of used operation.

(c) Computational Cost.

(d) Storage and Energy overheads.

Figure 4.7: Performance evaluation of related protocols with the Proposed scheme for IHS Environment.

Correspondingly for other existing scheme the computational and communicational costs being calculated as Zhang et al. [39] scheme need $20.0326ms$, as Islam et al. [21] use exponential operation thus, their scheme's coast is high i.e. $32.0728ms$. Dang et al. [27] scheme used $4T_{PM},\ 1T_{MP},\ 1T_M$ i.e. $12.7978ms$. Wang et al. [160]

used $6T_{PM}$ only for session key and for PKG need $2T_{PM}$, $1T_{MP}$, $1T_M$ thus, it take $12.009ms$ and $6.7889ms$ respectively. Gupta et al. [18] scheme depends on only $3T_{PM}$ and $2T_{BM}$ i.e $14.0093ms$. Next, we observed Cheng et al. [8] work that required $7T_{PM}$, $4T_{MP}$, $3T_M$ i.e. $24.3741ms$. Table 4.6 and Figure 4.7c will represent the computational cost of each phase like setup, PKG and Session key agreement of few existing protocol including our suggested scheme.

Thus, a comparative analysis being performed and represented by Table 4.6 and Figure 4.7c, in which proposed work is compared with a few existing protocols by Zhang et al. [39], Islam et al. [21], Dang et al. [27], Wang et al. [160], Gupta et al. [18], Cheng et al. [8], which clearly represents how we reduced the computational cost as compared to the discussed prior existing work.

Table 4.6 and Figure 4.7b demonstrate the operation used by existing and proposed protocols, like our work, which depended on only the elliptic point multiplication operation in count it is 4, in spite of this, other existing schemes also depend on other cryptographic operations. Figure 4.7b clearly shows the quantity of individual operations used by all schemes separately.

4.5.3 Energy Overheads

As we know, energy overhead is also a serious concern, especially when we talk about smart sensors. Thus, we also have to focus on reducing the storage and energy overhead of our scheme as much as possible. The energy estimation is being performed by using the equation $Energy = Power * Time$. Where time is measured in $\times 10^{-2}$ milliseconds(ms), power in watt(W) thus, the energy is estimated in milliJoule (mJ) units [161]. In our proposed scheme, patient sensors PS require $1.00mJ$private key generator (PKG) needs $20.36mJ$, and the medical server MS needs $6.0045mJ$ based on their participation in different phases such as setup, private key generation, and the mutual authentication phase. Similarly, for other existing schemes, energy consumption records are calculated for their sensor, server, and authorized third party, and the comparisons of energy overhead are represented by Table 4.7 and Figure 4.7d, where we can see how we minimized the energy overhead at both ends, the patient's sensor and the medical server, to enhance the performance of the Intelligent Healthcare System.

Thus, as per the performance evaluation and results analysis shown in figure 4.7a, 4.7b, 4.7c, and 4.7d, we can clearly state that we efficiently reduced the computational-communicational cost as well as minimized the storage-energy overheads.

4.6 Summary

This chapter utilizes PKG to eliminate the key overhead issues and utilizes an aggregator to minimize the latency during data transmission in IHS. A privacy-preserving mutual authentication and key agreement scheme is proposed that supports common key establishment between the two parties. The formal and informal security analysis shows that the proposed scheme provides session key security and achieves all the security goals. The performance analysis shows proof of the computational and communication, as well as energy-storage overhead efficiency of the proposed scheme. The key escrow problem always comes with the ID-based algorithms, which is a limitation of our work.

CHAPTER 5

An Improved Certificateless Mutual Authentication and Key Agreement Protocol for Cloud-Assisted WBAN

5.1 Introduction

Nowadays, IoT technology has become a very fast-growing research area. Managing all the information over the internet becomes a challenging task as people directly or indirectly depend on the internet [162]. IoT networking applications are used over wide sectors like Agriculture, military, smart home appliances, smart city, health care sector, etc. and all generated data will be heterogeneous. Traditional health-care organizations are facing lots of issues with constantly increasing the rate of patients with poor handling systems which results in a heavy crowd of patients in the hospital. To improvise, the Internet of Medical Things [IoMT] technology is used as one of the most appreciated applications of the IoT [46]. Real-time transmission has been completed using several networking layers such as Body Area Networks, Neighbour Area Networks, and Wide Area Networks in the IoMT [54]. Smart Medical Devices are used to keep remote monitoring of Blood Pressure, Diabetics cases, cardiovascular disease, etc. The wireless Body area network is established with a connection of several independent sensor nodes, which are positioned over the body or plant inside the skin [55]. Based on the star or multi-hop topology, these sensors

are connected through a wireless medium. The decision-making process for the selection of a sensor is also a complicated task and must be considered a few points like power consumption, size, and the lifetime of that sensor. To assemble physiological data, a large range of sensors is being used in the WBAN system. Hence, the Patient's medical history of medical devices communicates over the wireless medium among the healthcare teams or organizations. Therefore, Smart devices become a vulnerable point to attack as their information travels over wireless channels. The main challenges accompanying WBAN are to sustain the privacy and security of patients' data as sensors collect and transmit sensitive information. There are several risks associated with WBAN such as data privacy leakage or misuse, the medical examination is going to be complicated if a patient's data is altered, loss of life because of untrained medical staff, and so on. In [52] issues with WBAN are as follows- CIA Triad (Confidentiality, Integrity, Authentication), Data Falsification, Cyber Attacks, Zombie attacks, etc mentioned.

Authenticity, as well as confidentiality, are really the fundamental principles of reliable transmission. The actual healthcare information needs to remain safeguarded so that both patients and health experts have access to that as well, in addition, there might be a reliable information authentication protocol to address the increasing patient population. The aforesaid issues with secrecy and authenticity are addressed by public key cryptosystems. (PKC). Attempting to address the certificate maintenance challenges which occurred in certificate-based setups, Shamir [45] suggested the identity-based cryptosystem (IBC). In the IBC, public keys are generated by using their one identity information, and private keys are derived via a private key generator (PKG). Due to this, key escrow is a serious concern in IBC, yet the private key determines by PKG. Since the PKG is aware of the entire device's secrets, including users' private keys, hence there would be neither user anonymity nor authenticity preserved. To address the key escrow challenge, the certificates-based cryptosystem (CLBC) [163] was suggested, in which the user generates their final private key itself. In CL-PKC, the user autonomously generates the public/secret key pair while the partially private key is obtained through a semi-trusted key generation center (KGC). Thus, except for the user, no one has any knowledge about its secret key. To resolve these issues, several mutual authentications and key agreement protocols have been introduced, a few of them described in section related work. Cheng et al. [164] found that the identity-based AKA scheme for WBAN designed by Kumar et al. [165] did not support forwarded secrecy. Therefore, to address the issues related to data privacy and integrity, Cheng et al. [164] proposed an improvement to using certificateless cryptography, which handles the

perfect forward secrecy problem.

5.1.1 Problem Statement

- The sensors in the WBAN periodically collect data and then send the collected health information to the medical server.
- During the cryptanalysis process, we found many challenges and issues with the scheme introduced by Cheng et al., which we mentioned in the comments.
- To send regular collecting data, mutual authentication and key agreement schemes may not be as efficient as the signature schemes.
- The certificateless signature with a designated verifier is a secure and efficient approach for collecting users' health data and ensuring user privacy.
- Existing signature schemes face many issues, like certificate management problems, key escrow problems, higher computational cost, etc.

5.1.2 Main Contributions

The following contributions are made to the fog-based smart grid system to address the challenges discussed above.

1. The Certificateless cryptosystem framework provides significant strengths over the PKI's CA and IBC's PKG in terms of certificate management and key escrow concerns.
2. We performed the cryptanalysis over the scheme proposed by Cheng et al. [164]. We found numerous flaws, such as definition contradictions, the presence of security attacks like MITM, impersonation, etc.
3. We designed an improved version of a certificateless signature scheme that ensures the authentication, integrity, and privacy in the proposed scheme, along with work on all above found flaws.
4. We also defined the security model for our improved version and provided a formal security analysis using the ROM model against various security attacks, such as man-in-the-middle.
5. Additionally, informal security analysis included malicious activities such as key impersonation, MITM, key dominance, and perfect forward secrecy, among others.

6. As a result, we found our proposed improved version of the authenticated key agreement protocol for the WBAN network is safe and secure.

5.2 Background

This section presents the review of Cheng et al.'s network model, comments on their scheme, security model, and security goals defined for the proposed scheme.

5.2.1 Review of CL-AKA by Cheng et al. [164]

In this section, we describe the certificateless authentication and key agreement scheme for secure cloud-assisted WBAN designed by Cheng et al. [164].

5.2.1.1 Cheng et al.'s Network Model Cheng et al. [164] proposed a network model consisting of five entities—Network Manager, Cloud Server, Leaf node, Root Node, and Target Node—whose roles are demonstrated by figure 5.1. The working of this model is described in three different stages. First, the network manager sets the private key and the public key and publishes the system parameter. Second—The network model registers other entities' leaf, root, target node, and cloud server. At the last stage, mutual authentication along with a session key is generated between the leaf node and the cloud server. Cheng et al. [164] only considered the mutual authentication between the leaf node and cloud server by assuming that there is auxiliary communication from the LN to the Root and Target nodes; thus, it is out of the scope of discussion.

5.2.1.2 Cheng's [164] Proposed Scheme The author Cheng et al. Proposed model work has been described by three phases: Phase I- Setup, Phase II- Registration Phase, and Phase III- Mutual Authentication and Session Key Generation. Table 5.3 depicts all the steps taken in the registration phase by CS, LN, and NM. The notation used in Cheng et al.'s [164] scheme is signified in Table 5.2.

1. **Setup Phase:** Let two cyclic groups as G_1 is an additive and group $G_2$is multiplicative with a random k it of order q formed by NM. G_1 has P as a generator and where it x is chosen as the master private, which must belong to Z_q^*. By which the master public key is computed as $P_m = xP$. Five secure hash functions were selected as: $H_1 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$, $H_3 : G_2 \rightarrow \{0,1\}^* \times G_1 \times Z_q^*$, $H_4 : \{0,1\}^* \times G_2^2 \rightarrow Z_q^*$,

Table 5.1: Illustrates the Role of entities present in Cheng's Network Model.

Entity	Role in Model
Network Manager [NM]	i. Behave like a trusted node. ii. Publish system parameters for other entities. iii. Registered LN, CS, RN, and TN.
Cloud Server [CS]	i. Acquire system parameters to interact with LN after Registration. ii. It has ample computing and storage resources.
Leaf Node [LN]	i. It has restricted resources. ii. Sensors of wearable devices. iii. Patients' health records are directly collected and accelerated to RN via TN.
Root Node [RN]	i. An intermediate node for LN and TN. ii. Without accessing anything about transmitted data from LN, it just forwarded to TN.
Target Node [TN]	i. Hold vast and valuable resources. ii. Behave as a service provider for the patients.

Table 5.2: Notation used in Cheng's scheme.

Notations	Description
S	System Parameter
ID_{CS}	Identity of CS
ID_{LN}	Identity of LN
CS'_{ID}	Identity of Adversary CS'
P	Generator of group G_1
q	Large prime number
x	The master private key of NM
P_m	The master public key of NM
$H_i(i = 1, 2, 3, 4, 5)$	Secure hash functions
t_1	Timestamp of LN
t_2	Timestamp of CS
x_{CS}, x_{LN}, a, b, m	Random numbers belong to Z_q^*

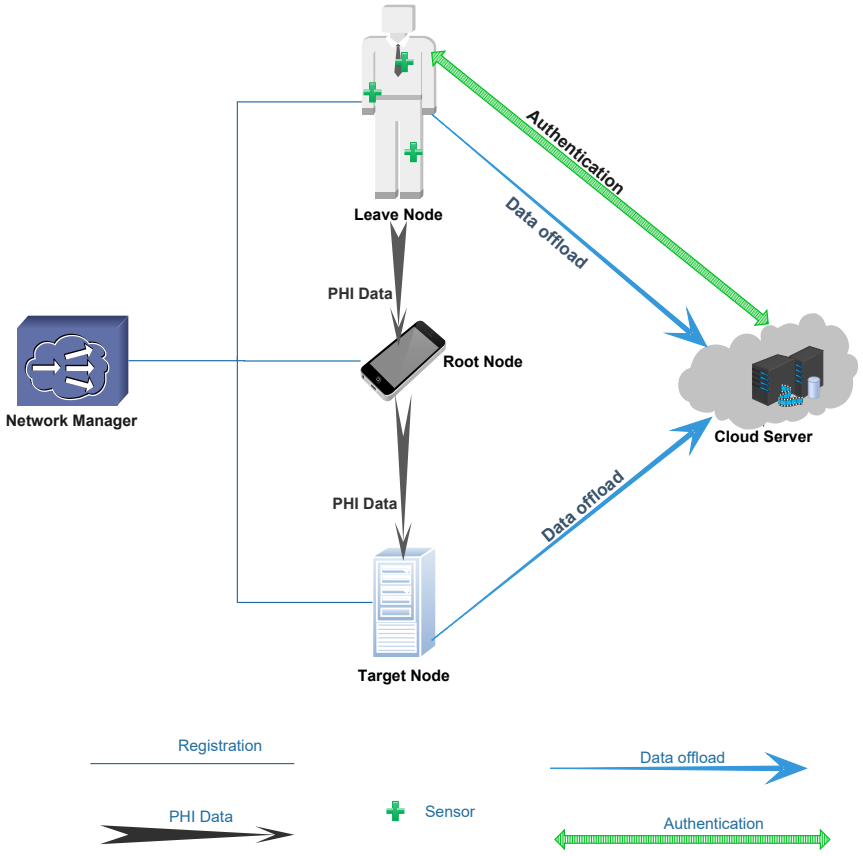


Figure 5.1: Cheng's [164] Network Framework for Wireless Body Area Network

$H_5 : \{0,1\}^* \times G_2^3 \rightarrow Z_q^*$. Finally, now NM can publish the system parameter $(q, k, P, P_m, H_1, H_2, H_3, H_4, H_5, G_1, G_2)$ by keeping x as the master secret key.

2. **Registration Phase:** In this phase, all messages travel over secure channels, where CS and LN are registered with NM.

   (a) *Registrations of CS*: CS selects its identity $ID_{CS} \in \{0,1\}^*$ forward a request to NM with this identity. Now NM chosen a random number $x_{CS} \in Z_q^*$ and then computes $X_{CS} = x_{CS}.P$ and $S_{CS} = x_{CS} + xH_1(ID_{CS} \| X_{CS})$. And directs these computed values (X_{CS}, S_{CS}) to the Cloud server as its private key.

   (b) *Registration of LN*: As like CS, the LN also first selects its identity $ID_{LN} \in \{0,1\}^*$ and forwarded a request along with this to NM. Here NM again select a random value $x_{LN} \in Z_q^*$ and computes $X_{LN} = x_{LN}.P$ and $S_{LN} = x_{LN} + xH_1(ID_{LN} \| X_{LN})$. And directs these computed values (X_{LN}, S_{LN}) to the Leaf Node as its private key.

Table 5.3: Steps involved in the Mutual Authentication Process of Cheng's scheme.

Leaf Node LN	**Cloud Server CS**
Select a random number: $a \in Z_q^*$ Computes: $A = a\ p$ and $e = H_2[ID_{CS} \parallel ID_{LN} \parallel A]$, and $A_{LN} = (a + eS_{LN})P$. Set: $g_{LN} = (a + eS_{LN})(X_{CS} + H_1[ID_{CS} \parallel X_{CS}]P_m)$ $h = S_{LN}(X_{CS} + H_1[ID_{CS} \parallel X_{LN}]P_m)$ $W = H_3(g_{LN}) \oplus (ID_{LN} \parallel X_{LN} \parallel t_1 \parallel h)$ Sends (W, A_{LN}) with timestamp t_1 towards the CS.	
	Computes: $g_{CS} = S_{CS}A_{LN}$ $H_3[g_{CS}] \oplus W = (ID_{LN} \parallel X_{LN} \parallel t_1 \parallel h)$ Verify h with: $h = S_{CS}(X_{LN} + H_1[ID_{LN} \parallel X_{LN}]P_m)$ Check the timestamp t_1 Select a random number: $b \in Z_q^*$ Computes: $B = b\ P$, $d = H_4[ID_{CS} \parallel ID_{LN} \parallel B]$, and $B_{CS} = (b + dS_{CS})P$ Forward the (B_{CS}, t_2) towards the LN. Computes the key: $CS_{key} = (b + dS_{CS})A_{LN}$ Determine the Session key as: $CS_{SK} = H_5[CS_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$.
Check the timestamp t_2 Computes the key: $LN_{key} = (a + e\ S_{CS})B_{CS}$ Determine the Session key as: $LN_{SK} = H_5[LN_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$	

3. **Authentication and Session Key Generation Phase:** In this phase, according to the author Cheng et al. [8] computation of the session keys is performed between the LN and CS only after the mutual authentication process is finished. Table 5.3 depicts the steps involved in the mutual authentication process to determine the session key between LN and CS, which is further used to transfer patient information.

   (a) $a \in Z_q^*$ randomly selected by LN and compute $A = aP, e = H_2[ID_{CS} \parallel ID_{LN} \parallel A]$, and $A_{LN} = (a + eS_{LN})P$. Then, LN sets $g_{LN} = (a + eS_{LN})(X_{CS} + H_1[ID_{CS} \parallel X_{CS}]P_m)$ and $h = S_{LN}(X_{CS} + H_1[ID_{CS} \parallel X_{LN}]P_m)$ and finally LN computes $W = H_3[g_{LN}] \oplus (X_{ID} \parallel X_{LN} \parallel t_1 \parallel h)$. Now LN sends (W, A_{LN}) sends to CS with timestamp t_1.

(b) CS compute $g_{CS} = S_{CS}A_{LN}$ after received (W, A_{LN}) and then performed $H_3[g_{LN}] \oplus W$ to computes $(ID_{LN} \parallel X_{LN} \parallel t_1 \parallel h)$. Check the computed h with its own h value such as $h = S_{CS}(X_{LN} + H_1[ID_{LN} \parallel X_{LN}]P_m)$ if it returns a false value then reject the request as an unauthorized entity found. In the case of true, CS select a timestamp t_2. Next CS, select a random number $b \in Z_q^*$ to computes $B = bP$, $d = H_4[CS_{ID} \parallel LN_{ID} \parallel B]$, $B_{CS} = (b + dS_{CS})P$, and also $CS_{key} = (b + dS_{CS})A_{LN}$. Now finally, CS computes the session key as, $CS_{SK} = H_5[CS_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$. At last CS directs (t_2, B_{CS}) to LN.

(c) LN compute the key $LN_{key} = (a + eS_{LN})B_{CS}$ after cross check the value of timestamp t_2. Next, LN generate its session key $LN_{SK} = H_5[LN_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$.

5.2.2 Comments on Cheng et al. [164] Scheme

Inside Cheng's scheme, network manager NM is used to generate the partial key pair (S_{LN}, X_{LN}) for Leaf nodes, similarly for CS also. Thus, we can say that the NM role is like a key generation center (KGC), which can generate the certificate for the requester entity using their identity. Thus, at the end of the registration process, KGC generates certificates and knows the partial private-public keys of all entities.

Comment 1: Not a Certificateless approach:

Proof. As per author Cheng [164], the proposed protocol is based on a certificateless concept. Where partial key (S_{LN}, X_{LN}) and (S_{CS}, X_{CS}) extracted from the registration process by the Network manager, which plays an important role during the mutual authentication process. Moreover, we can see that only these partial private-public key pairs are used for mutual authentication and session key generation. There is no footprint about the computation of final static private and public keys at the end of the entity by itself in the Cheng scheme, by which the certificate-based keys dependency concept should be exempted. As the key pairs (S_{LN}, X_{LN}) and (S_{CS}, X_{CS}) are generated during the registration process, the respective certificates should be maintained against the requesting identity KGC here NM.
Hence, there is a contradiction with the certificateless concept as here a third party (NM) behaves like a key generation center (KGC). Such that, there is the possibility of the key escrow and trusted third-party problem befalling; if the adversary can attack or NM itself performs some suspicious activities, then the whole system's security will be compromised. □

Comment 2: Imperfect Mutual authentication between LN and CS:

Proof. As per the definition of mutual authentication, both communicating nodes, LN and CS, must be authenticated by each other at their respective side (LN authenticates the CS and CS authenticates the LN).

1. ***CS Authenticate the LN:*** In Cheng's scheme [164], when LN sends a request towards the CS with value (W, A_{LN}), CS starts to verify the requester node LN as follows:

   - First, CS computes $g_{CS} = S_{CS}A_{LN}$ to determine the identity of LN where,

$$\begin{aligned} S_{CS}A_{LN} & \\ &= [x_{CS} + xH_1[ID_{CS} \parallel X_{CS}][(a + eS_{LN})P] \\ &= [x_{CS}P + xH_1(ID_{CS} \parallel X_{CS})P](a + eS_{LN}) \\ &= [(a + eS_{LN})][X_{CS} + H1[ID_{CS} \parallel X_{CS}]P_m] \end{aligned}$$

     thus, here it found that $g_{CS} == g_{LN}$, further to verify the authenticity of LN, CS determines the identity of LN and h by applying the $xor(\oplus)$ operation as $H_3(g_{CS}) \oplus W = H_3(g_{LN}) \oplus W$, by which CS derived the $(ID_{LN} \parallel X_{LN} \parallel t_1 \parallel h)$.

   - Next CS, verify the above-computed value of h with its own h, which is determined as: h

$$\begin{aligned} &= S_{CS}(X_{LN} + H_1[ID_{LN} \parallel X_{LN}]P_m) \\ &= (x_{CS} \ + xH_1[ID_{CS} \parallel X_{CS}])(x_{LN}.P + H_1[ID_{LN} \parallel X_{LN}]xP) \\ &= (x_{CS}.P + xP \ H_1[ID_{CS} \parallel X_{CS}])(x_{LN} + H_1[ID_{LN} \parallel X_{LN}]x) \\ &= S_{LN}(X_{CS} + H_1[ID_{CS} \parallel X_{CS}]P_m) \end{aligned}$$

     In this way, CS successfully authenticates the LN, and further CS directs (t_2, B_{CS}) to LN.

2. ***LN Authenticate the CS node:*** after receiving the value (t_2, B_{CS}) from CS, LN only needs to verify the CS's timestamp value. LN did not perform any verification process about CS's authentication (CS's identity). Thus, there is no authentication verification done at the LN side for the sender CS.
   Thus, as a result, we found that there is imperfect mutual authentication between the nodes LN and CS in Cheng's proposed scheme.

□

Comment 3: Vulnerable to Impersonation attack:

Proof. With the purpose to impersonate any existing entity (either LN or CS) inside the network let it for LN, KGC (NM itself) can replace the chosen ephemeral number $a \in Z_q^*$ by $a' \in Z_q^*$ and computes all dependent variable on value a' such as: $A', e', A'_{LN}, g'_{LN}, h'$ and W'. NM forwards this generated value (W', A'_{LN}) by using identity of LN: ID_{LN}. After receiving the value (W', A'_{LN}), CS starts to check the authenticity and starts the further process to determine the session key. With the help of the following steps, we can see how NM impersonates the LN (or CS) and how CS and NM generate the same session key.

- First, CS compute: $g_{CS} = S_{CS}A'_{LN} = [x_{CS} + xH_1[ID_{CS} \| X_{CS}][a' + e'S_{LN}]P] = [x_{CS}P + xH_1[ID_{CS} \| X_{CS}]P][a' + e'S_{LN}]$. and we found that $g_{CS} == g'_{LN}$, therefore, $H_3(g_{CS}) \oplus W' = H_3(g_{LN}) \oplus W'$, by which CS derived $(ID_{LN} \| X_{LN} \| t_1 \| h')$.

- Second CS need to verify the h' with its own computed value $h = S_{CS}(X_{LN} + H_1(ID_{LN} \| X_{LN})P_m)$, which is depends on S_{CS}, X_{LN} (knows by NM) and P_m is the master public key of NM. Thus, $h == h'$. Hence, here CS again identifies the requester entity as an authorized entity as LN also.

- Third, it is about the keys CS_{key} and LN'_{key}, as it plays an important role to determine the session key. Start with $CS_{key} = (b + d\ S_{CS})A'_{LN} = (b + dS_{CS})(a' + e'S_{LN})P = (a' + e'S_{LN})B_{CS}$. Thus, both entities compute the same keys such as $CS_{key} = LN'_{key}$.

- Last, same session key determined at both ends: $CS_{SK} = H_5[CS_{key} \| ID_{LN} \| ID_{CS} \| B_{CS} \| A'_{LN}] = H_5[LN'_{key} \| ID_{LN} \| ID_{CS} \| B_{CS} \| A'_{LN}] = LN'_{SK}$.

Thus, KGC could impersonate either LN or CS with another existing network and generate the same session key. Figure 5.2 illustrates the impersonation action performed by the network manager, who is trying to impersonate entity LN to CS and successfully calculate the same session key. Thus, all the information transported in such a network fails to maintain its privacy, and integrity as an attacker can perform many actions like data falsification, misused, etc. □

Comment 4: Vulnerable to Men-in-the-middle attack:

Proof. As per the description in comments 2 and 3, we found that there is the possibility of an impersonation attack from the LN to the CS side and vice versa also as well as there is lack of mutual authentication between the LN and CS.

Network Manager: NM	Cloud Server: CS
// Network Manager Impersonating the Leaf Node LN, already knows: (S_{LN}, X_{LN}) 1. Select a random number: $a' \epsilon Z_q^*$ 2. Computes: $A' = a\ p$ $e' = H_2[ID_{CS}\|\|ID_{LN}\|\|A']$ $A'_{LN} = (a' + e'S_{LN})P$ 3. Set: $g'_{LN} = (a' + e'S_{LN})(X_{CS} + H_1(ID_{CS}\|\|X_{CS})\ P_m$ $h' = S_{LN}(X_{CS} + H_1(ID_{CS}\|\|X_{LN})P_m)$ $W' = H_3(g'_{LN}) \oplus (X_{ID}\|\|X_{LN}\|\|t_1\|\|h')$ 4. sends (W', A'_{LN}) with timestamp t_1 towards Cloud server.	5. Computes: $g_{CS} = S_{CS}A'_{LN}$ 6. $H_3(g_{CS}) \oplus W' = (ID_{LN}\|\|X_{LN}\|\|t_1\|\|h')$ 7. Verify the above computed h' with it $h, where$ $h = S_{CS}(X_{LN} + H_1(ID_{LN}\|\|X_{LN})P_m)$ 8. Check the timestamp t_1 9. Select a random number: $b \epsilon Z_q^*$ 10. Computes: $B = b\ p$ $d = H_2[ID_{CS}\|\|ID_{LN}\|\|B]$ $B_{CS} = (b + d\ S_{CS})P$ 11. Forwards (B_{CS}, t_2) towards the Leaf Node. 12. Computes the key: $CS_{key} = (b + d\ S_{CS})A'_{LN}$ 13. Determine the Session key as: $CS_{SK} = H_5(CS_{key}\|\|ID_{LN}\|\|ID_{CS}\|\|B_{CS}\|\|A'_{LN})$.
14. Check the timestamp t_2 15. Computes the key: $LN'_{key} = (a' + e'S_{LN})B_{CS}$ 16. Determine the Session key as: $LN'_{SK} = H_5(LN'_{key}\|\|ID_{LN}\|\|ID_{CS}\|\|B_{CS}\|\|A'_{LN})$	

Figure 5.2: Basic Impersonation Attack by Network Manager (NM)

Therefore, there is a possibility of a men-in-the-middle attack, such that both LN and CS generate the session key with an intermediate attacker. Therefore, all the information traveling in networks lose its privacy and integrity due to the presence of an attacker who can perform malevolent action like misuse or alter or delete important information, etc, which can cause serious health issue for patients or related to doctors' safety as well. □

Comment 5: Design flaws in Cheng et al.'s scheme due to sharing of the private key:

Proof. During the registration phase, the author Cheng et al. [164] mention that NM can generate and forward the private key of LN and CS nodes respectively. So our question is, how can LN access the CS's private key X_{CS} for computing the value of g_{LN} i.e. $g_{LN} = (a + eS_{LN})(X_{CS} + H_1[CS_{ID} \parallel X_{CS}]P_m)$ and h i.e. $h = S_{LN}(X_{CS} + H_1[CS_{ID} \parallel X_{CS}]P_m)$? As we know without making X_{CS} as a public key, LN is unable to use it, and if it is public then there is a contradiction with the registration phase that claims to generate value as a private key. □

Comment 6: Design flaws in Hash function $H_2: \{0,1\}^* \times G_2 \rightarrow Z_q^*$ ***in phase III of Cheng et al.'s scheme:***

Proof. Firstly, we highlight the vagueness of this scheme about the H2. Cheng et. al. [164] use H_2 at the LN side, to compute the value of e in the description part is like $e = H_2[CS_{ID} \parallel LN_{ID} \parallel A]$ which is incompatible with the definition of hash H_2 and in the respective figure they written it as $e = H_2[CS_{ID} \parallel A]$ which followed the definition they mentioned, it seems imprecise. Whenever we focus on the role of H2, we can narrate the role of e and d, which play the same role and computation process at their respective ends. Where CS computes d as $d = H_4(ID_{CS} \parallel ID_{LN} \parallel B)$ using H_4. Therefore, the correct format of e would be like $e = H_4(ID_{CS} \parallel ID_{LN} \parallel A)$, which indicates the flaws in definition H_2. Hence, the correct definition of hash function H2 must be like - $H_2 : \{0,1\}^* \times \{0,1\}^* \times G_2 \rightarrow Z_q^*$. □

Comment 7: Design flaws of Hash function $H_4: \{0,1\}^* \times G_2^2 \rightarrow Z_R^*$ ***in phase III of Cheng et. al.'s scheme.***

Proof. Cheng et al. [164] said that hash H_4 will perform over one string and two points. But at the CS side, Cheng et. al. computes the value of d using hash as follows: $d = H_4(CS_{ID} \parallel LN_{ID} \parallel B)$. Such definition and calling of a hash function are incompatible. As of now, we found that the computation of d is depending on two strings (ID_{CS}, ID_{LN}) and one point (B). Hence, the correct definition of H_4 will be like $H_4: \{0,1\}^* \times \{0,1\}^* \times G_2 \rightarrow Z_q^*$. □

Comment 8: Duplication flaws in the definition of the Hash function H_2 ***and*** H_4***:***

Proof. As we can see, the H_2 can work over two strings and one point at the LN side to compute the value of e. Similarly, on the CS side authors Cheng et.al. [164] compute the d by applying H_4 over two strings and one point.
$e = H_2[CS_{ID} \parallel LN_{ID} \parallel A]$ at the LN side.
$d = H_4[CS_{ID} \parallel LN_{ID} \parallel B]$ at the CS side.
There is no need of calling another hash H_4, as we already mentioned in comments 4 and 5, that both hash functions have the same meaning and role at their respective ends.
$H_2 : \{0,1\}^* \times \{0,1\}^* \times G_2 \rightarrow Z_q^*$
$H_4 : \{0,1\}^* \times \{0,1\}^* \times G_2 \rightarrow Z_q^*$
Thus, there are design duplication flaws during defining the hash function. □

Comment 9: Design flaws with hash function $H_5 : \{0,1\}^* \times G_2^3 \rightarrow Z_q^*$ ***in Phase III of Cheng et. al.'s scheme [164].***

Proof. At both ends during the generation of the session key, the calling of the hash function H_5 is fully incompatible with its definition given by Cheng et. al. [164] scheme's Phase I.
CS computes session key: $CS_{SK} = H_5[CS_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$
LN computes it as: $LN_{SK} = H_5[LN_{key} \parallel ID_{LN} \parallel ID_{CS} \parallel B_{CS} \parallel A_{LN}]$
Here we found that hashing is achieved over two strings (ID_{LN}, ID_{CS}) and three points $(LN_{key},\ B_{CS}, A_{LN})$. However, Cheng et. al. [164]defined hash H_5 using only one string with three points acceptable. Hence, the right definition of Hash Function H5 will be like $H_5 : \{0,1\}^* \times \{0,1\}^* \times\ G_2^3 \rightarrow Z_q^*$. □

5.2.3 Security Goals

By the study [55], [164], and [82], it is observed that an AKA mechanism needs to fulfill the desired security requirements identified under this section as follows:

1. ***Mutual Authentication***: LN and CS should first pursue mutual authentication to ensure the legitimacy and integrity of distributed data.

2. ***Impersonation resist***: Without having access to the stable secret key of an authentic entity, an adversary could not imitate them.

3. ***Known key Secrecy***: Each generated session key should be unique from the other session. Thus, an attacker never determines the session key for the current session, even though it has any previous session key information. After the completion of the execution, the scheme generates a unique session key.

4. ***No key Control***: None of the participating entities could completely or partially influence the process of determining the session key.

5. ***Perfect Forward Secrecy***: Even though each communicating nodes are compromised, adversaries never retrieve the associated session keys when it does not intentionally participate in determining the ephemeral secret keys of those sessions.

6. ***Replay Attack Resist***: If an attacker has each of the session's ephemeral secret information, they must not be capable of determining the session value, only if they can somehow violate at least a single peer implicated.

Table 5.4: Notation used in Proposed Protocol.

Notations	Description
S	System Parameter
G_1	Additive Group
G_2	Multiplicative Group
ID_{CS}	Identity of CS
ID_{LN}	Identity of LN
P	Generator of group G_1
q	Large prime number
x	The master private key of NM
Pb_m	The master public key of NM
$H_i(i = 1, 2, 3)$	Secure hash functions
$(PPr_i,\ PPb_i)$	Partial Private-Public Key pair of $i \in$ (LN,CS)
$(FPr_i,\ (FPb_i)$	Final Private-Public Key pair of $i \in$ (LN, CS)
t_1	Timestamp of LN
t_2	Timestamp of CS
$(\chi_i,\ \omega_i)$	Signature pair for Participating nodes CS and LN.
r, s, a, b	Random numbers belong to Z_q^*
u, v	Ephemeral keys belong to Z_q^*

5.3 Proposed Scheme

This section details the steps of the certificateless aggregate signature scheme that is proposed.

In Cheng's scheme, we have seen that partial keys generated by the Register Authority Network Manager (NM) are being used during the authentication and session key generation process. There is the possibility of an impersonation attack as well as a man-in-the-middle attack. Also, we can see there is a lack of mutual authentication at both endpoints in Cheng's work, as only CS cross-verifies the authorization of LN but on the LN side, there is no cross-verification about the authenticity of CS. Thus, we have studied that maintaining all the generated certificates to publish the key pair for users is also a tough issue. Therefore, key escrow at NM's end will also become a challenging task. Thus, we proposed a protocol that is certificateless protocol to tackle the key escrow challenge. Next, we focused on providing mutual authentication so that we can prevent several attacks, like man-in-the-middle attacks, impersonation attacks, etc. Notations used by the proposed protocol are illustrated in Table 5.4, and we mathematically describe such a concept by the following steps:

5.3.1 Setup (K)

Let two cyclic groups as G_1 is an additive and group G_2 is multiplicative with random k bit of prime order q formed by NM.

1. G_1 has P as a generator and x is chosen as master private which must belong to Z_q^*. Next, master public key is computed as $Pb_m = xP$.

2. Three secure hash functions selected as:
   $H_1 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$,
   $H_2 : \{0,1\}^* \times \{0,1\}^* \times G_1 \rightarrow Z_q^*$,
   $H_3 : \{0,1\}^* \times \{0,1\}^* \times Z_q^* \times Z_q^* \times Z_q^* \times Z_q^* \times Z_q^* \rightarrow Z_q^*$.

3. Finally, now NM can publish the system parameter $(k, q, P, Pb_m, H_1, H_2, H_3, G_1, G_2)$ by keeping x as the master secret key.

5.3.2 User's Partial Key Extraction$(Param, ID_i)$

In this phase, CS and LN can generate their own partial private-public key pair (PPr_i, PPb_i) where i is used to represent the cloud server CS or Leaf Node such that $i \in (CS, LN)$. Let the first cloud server, CS, call the generate function for its partial private-public key as follows:

1. *Generate partial private key* (PPr_{CS}): Cloud server chooses a random number $r \in Z_q^*$ as its partial private key i.e., $PPr_{CS} = r$.

2. *Generate partial public key* (PPb_{CS}): Cloud server computes its partial public key by using its partial private key as $PPb_{CS} = rP$.

Similarly, Leaf node LN also computes its partial private-public key pair as: $(PPr_{LN}, PPb_{LN}) = (s, sP)$. At the end of this phase, all the generated partial private keys PPr_i is known by the node itself; however, it is the node's choice to publish the PPb_i key as its public key or not.

5.3.3 Registration and Key Extraction Process $(Param, ID_i)$

In this phase, both CS and LN are registered with NM before going to start the process of generating session keys and starting the transmission of their information through the internet. When the requester (CS, LN) makes a request using its identity which is a string of an arbitrary length. Next, NM can extract the key pair for the requester corresponding to that identity and share it through a secure channel. Thus, such extracted partial key pairs are known by both NM and the requester (CS, LN).

Table 5.5: Generated Keys detailing during Phases 5.3.2, 5.3.3, and 5.3.4.

Key Pairs	Cloud Server (CS)	Leaf Node (LN)
Partial Key Pair (PPr_i, PPb_i)	$PPr_{CS} = r$ as partial private. $PPb_{CS} = r \cdot P$ as Partial public key.	$PPr_{LN} = s$ as partial private. $PPb_{LN} = s{\cdot}P$ as Partial public key.
NM Key Extracted (λ_i, δ_i)	$\lambda_{CS} = a \cdot P$ and $\delta_{CS} = a + x \cdot H_1[ID_{CS}\|\|\lambda_{CS}\|\|PPb_{CS}]$ extracted key pairs	$\lambda_{LN} = b \cdot P$ and $\delta_{LN} = b + x \cdot H_1[ID_{LN}\|\|\lambda_{LN}\|\|PPb_{LN}]$ extracted key pairs.
Set Final Key Pair (FPr_i, FPb_i)	$FPr_{CS} = PPr_{CS} \times \delta_{CS}$ $FPb_{CS} = (PPb_{CS}, \lambda_{CS})$	$FPr_{LN} = PPr_{LN} \times \delta_{LN}$ $FPb_{LN} = (PPb_{LN}, \lambda_{LN})$.

1. *Registrations of CS:* CS selects its identity $ID_{CS} \in \{0,1\}^*$ and sends a request towards the NM with this chosen *id*. Now NM also chooses a random number $a \in Z_q^*$ and then computes $\lambda_{CS} = a{\cdot}P$ and $\delta_{CS} = a{+}xH_1[ID_{CS}\|\lambda_{CS}\|PPb_{CS}]$. Thus, the partial private-public key extraction at the NM end is being done using requester identity, ID_{CS}, and further, NM transfers these computed key pairs $(\lambda_{CS},\ \delta_{CS})$ towards the cloud server through a secure channel.

2. *Registration of LN:* As with CS, the LN also first selects its identity $ID_{LN} \in \{0,1\}^*$ and forwards a request along with this identity towards the NM. Next, Network Manager selects a random value $b \in Z_q^*$ to compute $\lambda_{LN} = b \cdot P$ and $\delta_{LN} = b + xH_1[ID_{LN} \parallel \lambda_{LN} \parallel PPb_{LN}]$. further NM directs these computed values $(\lambda_{LN},\ \delta_{LN})$ towards LN as its partial private-public key pair extraction is done.

Table 5.5 presents the keys generated by each phase involved in the final key process.

5.3.4 Set Final Key Pair $(Param, ID_i, PPr_i, PPb_i)$

This phase can apply only after the registration and key extraction process has been performed by the network manager. Cloud server or Leaf node can start to set their final private-public key pair by using a generated partial private-public pair by themselves and receive their respective partial private-public key pair by NM. The steps involve during set the final key pair for the cloud server are as follows:

1. *Set Final Private Key* (FPr_{CS}): For its own selected secret key PPr_{CS} and received secret key δ_{CS}, cloud server CS set its full final private key FPr_{CS} as; $FPr_{CS} = PPr_{CS} \times\ \delta_{CS}$ where $PPr_{CS} =\ r$.
   As this final value is dependent on two secret keys, one of which r is only known

by CS itself and δ_{CS} is known by the trusted authority NM also. Therefore, our proposed final private key maintains its secrecy property properly.

2. *Set Final Public Key* (FPb_{CS}): For its own computed public key PPb_{CS} and received key λ_{CS}, Cloud Server CS set its full final public key FPb_{CS} as: $FPb_{CS} = (PPb_{CS}, \delta_{CS})$. Where both public key pairs depend on the secrecy of random values $r, a \in Z_q^*$.

5.3.5 Certificateless Mutual Authentication

In this phase, mutual authentication between LN and CS is performed as follows:

1. LN selects randomly an ephemeral key $u \in Z_q^*$ and computes $T_{LN} = u \cdot P, h_{LN} = H_2[ID_{LN} \parallel ID_{CS} \parallel T_{LN}]$. Next, to verify its authenticity by CS, LN needs to generate its signature like as: $\chi_{LN} = [u+h_{LN}]^{-1} \cdot \delta_{LN}$ and $\omega_{LN} = \delta_{LN} \cdot P$. Further, LN also computes a key using its secret keys such as $S_{LN} = T_{LN} + [h_{LN} \cdot FPr_{LN} \cdot P]$. At last set of tuples $(ID_{LN}, FPb_{LN}, S_{LN}, \chi_{LN}, \omega_{LN}, T_{LN}, t_1)$ sends to the cloud server CS, where t_1 is the current timestamp of LN.

2. After receiving the tuples CS first needs to verify the timestamp and then verify the LN's legitimacy. thus, it checks whether $\omega_{LN} == \chi_{LN} T_{LN} + H2[ID_{LN} \parallel ID_{CS} \| T_{LN}] \cdot P$ or not? With the return value being false, CS finds that LN is an unauthorized node and rejects the request of LN regarding sharing information to generate the session key. If it returns the true value, then it demonstrates that the requesting node verified its authenticity. Thus, next CS, select an ephemeral key $v \in Z_q^*$ and computes $T_{CS} = v \cdot P, h_{CS} = H_2[ID_{LN} \| ID_{CS} \| T_{CS}]$. Next, to verify its authenticity by LN, CS needs to generate its signature as well as LN as follows $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS}$ and $\omega_{CS} = \delta_{CS} \cdot P$.
   Further, CS also computes a key using its secret keys as $S_{CS} = T_{CS} + [h_{CS} \cdot FPr_{CS} \cdot P]$. Next a set of tuples $(ID_{CS}, FPb_{CS}, S_{CS}, \chi_{CS}, \omega_{CS}, T_{CS}, t_2)$ sends to LN, where t_2 is the current timestamp of CS. Simultaneously it also computes a key $K_{CS} = [v + h_{CS} \cdot FPr_{CS}] S_{LN}$.

3. Leaf node LN also needs to verify the timestamp first and then cross-check the authenticity of CS after receiving the tuple sets like the cloud server done at their ends. Such that LN checks whether the equation $\omega_{CS} == \chi_{CS} T_{CS} + H2[ID_{LN} \parallel ID_{CS} \parallel T_{CS}] \cdot P$ is it true or not? LN starts to compute the key K_{LN} only after the verification of CS is done successfully. Otherwise reject the request of CS. Where $K_{LN} = [u + h_{LN} \cdot FPr_{LN}] S_{CS}$.

Table 5.6: Steps in Proposed Scheme's Mutual Authentication Process.

LN	**CS**
• LN select an ephemeral key $u \in Z_q^*$ • Computes $T_{LN} = u \cdot P$ $h_{LN} = H_2[ID_{LN}\|\|ID_{CS}\|\|T_{LN}]$. • Signature as follows: $\chi_{LN} = [u + h_{LN}]^{-1} \cdot \delta_{LN}$ and $\omega_{LN} = \delta_{LN} \cdot P$. • Further, LN also computes a key using its secret keys, such as: $S_{LN} = T_{LN} + [h_{LN} \cdot FPr_{LN} \cdot P]$. Forwards tuple: $\langle \mathbf{ID_{LN}}, \mathbf{FPb_{LN}}, \mathbf{S_{LN}}, \mathbf{Ø_{LN}}, \mathbf{!_{LN}}, \mathbf{T_{LN}}, \mathbf{t_1} \rangle$	
	• CS first needs to verify the timestamp t_1 and then verify the LN's legitimacy. • checks whether $\omega_{LN} == \chi_{LN} T_{LN} + H2[ID_{LN}\|\|ID_{CS}\|\|T_{LN}] \cdot P$ or not? Rejects the request if the return value is false. Otherwise, **LN authenticity is verified successfully by CS**, and then CS follows the next steps as: • Now, CS, selects an ephemeral key $v \in Z_q^*$ • Computes $T_{CS} = v \cdot P, h_{CS} = H_2[ID_{LN}\|\|ID_{CS}\|\|T_{CS}]$. • Signature as: $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS}$ and $\omega_{CS} = \delta_{CS} \cdot P$. • Further, CS also computes a key using its secret keys as $S_{CS} = T_{CS} + [h_{CS} \cdot FPr_{CS} \cdot P]$. • Computes $\mathbf{K_{CS}} = [\mathbf{v} + \mathbf{h_{CS}} \cdot \mathbf{FPr_{CS}}]\mathbf{S_{LN}}$. Next a set of tuples sends to LN as : $\langle \mathbf{ID_{CS}}, \mathbf{FPb_{CS}}, \mathbf{S_{CS}}, \mathbf{Ø_{CS}}, \mathbf{!_{CS}}, \mathbf{T_{CS}}, \mathbf{t_2} \rangle$.
• LN first needs to verify the timestamp t_2 and then verify the CS's legitimacy. • checks whether $\omega_{CS} == \chi_{CS} T_{CS} + H2[ID_{LN}\|\|ID_{CS}\|\|T_{CS}] \cdot P$ or not? Rejects the request if the return value is false. Otherwise, CS is authentic, and then only LN follows the next steps as: **LN also verifies the authenticity of CS**. • Further, LN also computes a key: $\mathbf{K_{LN}} = [\mathbf{u} + \mathbf{h_{LN}} \cdot \mathbf{FPr_{LN}}]\mathbf{S_{CS}}$	

At the end of the phase, we can see how both LN and CS start further computation only when they verified the authenticity of each other. Thus, we introduced a mutual authentication protocol using certificateless concepts.

5.3.6 Session Key Generation

The session key generation process should start only after the successful mutual authentication process has been completed by both the cloud server and the leaf node. During the mutual authentication process, variables pair (T_{CS}, T_{LN}), and (χ_{CS}, χ_{LN}) depend on the secrecy of secret keys, such that the privacy of a variable (T_{CS}, T_{LN}) depends on their respective secret ephemeral keys - (u, v). Similarly, (χ_{CS}, χ_{LN}) a variable's privacy depends on the secrecy of secret keys- (u, v) and $(\delta_{CS}, \delta_{LN})$, which are themselves secure due to random secret keys- (a, b). And we also found that the last computed key pair (K_{CS}, K_{LN}) maintains its privacy due to depending on the full private key pair of both endpoints (FPr_{CS}, FPr_{LN}), which are computed and known by nodes themselves.

Thus, session key computation must depend on such types of variables to preserve the robust secrecy of the session key. Computation of session key is as follows at the cloud server side: $SK_{CS} = H_3[ID_{CS}, ID_{LN}, T_{CS}, T_{LN}, \chi_{CS}, \chi_{LN}, K_{CS}]$ and similarly done at Leaf node endpoints: $SK_{LN} = H_3[ID_{CS}, ID_{LN}, T_{CS}, T_{LN}, \chi_{CS}, \chi_{LN}, K_{LN}]$.

As a result, both invoke the same session key as the computed key K_{CS}, and K_{LN} are the same in nature: $K_{CS} = K_{LN} = K$, as proved in the security analysis section. Finally, CS and LN computed the same session key: $SK_{CS}= SK_{LN}= SK$, which is used for encryption and decryption purposes to maintain the privacy of the transferred message between these endpoints. Therefore, we can write the final shared session key like this: $\mathbf{SK = H_3[ID_{CS} \parallel ID_{LN} \parallel T_{CS} \parallel T_{LN} \parallel \emptyset_{CS} \parallel \emptyset_{LN} \parallel K]}$.

5.4 Proposed Security Model

In this section, we are going to define the security model using the Random Oracle Model (ROM) [65]. Peers P_i involved in communication could be LN and CS. Let's take a brief look at the design variables and symbols as well.

1. $\mathcal{S}_{P_i}^n$**:** This gave relevant data about the nth round carried out through the participant node P of the proposed scheme. Whereas P_i produces a list with a collection of parameters. This variable set maintains a record of the present status of this protocol and automatically syncs throughout the execution of the scenario and maintains a track.

2. $\mathfrak{U}_{P_i}^n$**:** The distinctiveness of sessions has been measured by using this value. Oracle and all participants must always be aware of the specific session's identification.

3. $\mathfrak{L}_{P_i}^n$**:** It corresponds to the precise pertinent information on the credentials of the mentioned individuals (private key as well as additional specific identities). Those are all necessary to acquire the protocol's unique session key.

Listed below are a couple of definitions that pertain to the suggested approach as well as its security.

1. P_i participant's $\mathcal{S}_{P_i}^n$ state won't be acknowledged as approved until it generates a real, non-null session key with a counterparty.

2. The identifiers of n the session must be preserved in a public list $\mathfrak{U}_{P_i}^n$, intended for the respective instances $\mathcal{S}_{P_i}^n$ of the participant's P_i.

3. $\mathfrak{J}_{P_i}^n$ is used to represent the communicating peer's identity whenever it P_i wants to exchange a secure session credential with another, which must be disclosed with each other communicating peers at instance $\mathcal{S}_{P_i}^n$. This $\mathfrak{J}_{P_i}^n$ value is public too.

4. The necessary prerequisites must be achieved to establish the connection between instances $\mathcal{S}_{P_i}^n$ and $\mathcal{S}_{P_j}^n$ are as follows:

   - $\mathcal{S}_{P_i}^n$ and $\mathcal{S}_{P_j}^m$ instances clearly persist in a true condition.
   - All variants share the identical session as well as participant identities at the same iteration as: $\mathfrak{U}_{P_i}^n = \mathfrak{J}_{P_j}^m$ and $\mathfrak{U}_{P_i}^n = \mathfrak{J}_{P_j}^m$.

Assume that $\mathcal{A}$ is the PPT attacker attempting to breach the security definitions. Furthermore, $\mathcal{A}$ and the other associated nodes, such as the cloud server and Leaf Node, might exclusively contact each other through random oracle inquiries, stated below as. $\mathcal{A}$ should be responded to here for preceding inquiries to determine regardless of whether the secrecy of the proposed protocol would be threatened.

1. *Disclose Private Key* (P_i): The result, which an attacker $\mathcal{A}$ would get whenever it asks $\mathcal{S}_{P_i}^n$ to run the query, will consist of the peer's private key information.

2. *Send* $(\mathcal{S}_{P_i}^n, m)$: If $\mathcal{A}$ submits the message m, $\mathcal{S}_{P_i}^n$ wraps up the requests. Consequently, it produces $\mathcal{A}$'s outcomes according to the introduced approach. Also, even when an unsuitable message is present, the reply would be NULL.

3. *Disclose Random Secret* ($\mathcal{S}^n_{P_i}$): While $\mathcal{S}^n_{P_i}$ examining against the $\mathcal{A}$ query, the results that adversary $\mathcal{A}$ receives are the ephemeral credentials of the participating node P_i.

4. *Release Public Key* ($\mathcal{S}^n_{P_i}$): Once $\mathcal{S}^n_{P_i}$ runs queries for$\mathcal{A}$, the information acquired by $\mathcal{A}$ is the public keys of the active nodes in the network.

5. *Release the State* ($\mathcal{S}^n_{P_i}$): $\mathcal{S}^n_{P_i}$ exposes full state information $\mathfrak{U}^n_{P_i}$ whenever it executes requests from $\mathcal{A}$. Forward $\mathfrak{U}^n_{P_i}$ for nth session as answer to $\mathcal{A}$.

6. *Release Session Key* ($\mathcal{S}^n_{P_i}$): When $\mathcal{A}$ request $\mathcal{S}^n_{P_i}$ to execute this request, it answered with generated session key $SK^n_{P_i}$ information for the P_i with its partner at *nth* session. Response to the attacker as an acceptable state for $\mathcal{S}^n_{P_i}$.

7. *Test* ($\mathcal{S}^n_{P_i}$): These oracle queries would be asked by an attacker $\mathcal{A}$ once per session with newer peer P_i. The output of such a query is a randomly selected bit t. if its response is 1 that means session key $SK^n_{P_i}$ for P_i should in accept state $\mathcal{S}^n_{P_i}$; else returned any random bit to adversary $\mathcal{A}$.

5.5 Security Analysis

In this section, we are going to present proof of the correctness of the scheme followed by its provable security analysis using a random oracle model. At the end of the section, we also performed an informal security analysis against several attacks like men in the middle, impersonation, replay attacks, and perfect forward secrecy, and so on. The result signifies the robust security nature of our proposed scheme, which preserves its data privacy, integrity, and authentication against various types of intruders.

5.5.1 Correctness Proof

A couple of theorems would be used to demonstrate the correctness of our suggested scheme.

Theorem 5.1. *Whenever both cloud server CS and leaf node LN adhere to the proposed approach, the same key is derived through both ends:* $K_{CS} = K_{LN}$.

Proof. Since both ends derived the similar session key SK, the cloud server and leaf node should exchange their data via the internet by using the session key. However, it is allowed only if the values of the utilized keys K_{CS} and K_{LN} are equivalent. We

also show the evidence that shows such variables are equivalent here. Cloud server CS side, the key is computed as follows:

$$\begin{aligned} K_{CS} = & \\ & = [v + h_{CS} \cdot FPr_{CS}][S_{LN}] \\ & = [v + h_{CS} \cdot FPr_{CS}][T_{LN} + h_{LN} \cdot FPr_{LN} \cdot P] \\ & = [v + h_{CS} \cdot FPr_{CS}][u \cdot P + h_{LN} \cdot FPr_{LN} \cdot P] \\ & = [v + h_{CS} \cdot FPr_{CS}][[u + h_{LN} \cdot FPr_{LN}] \cdot P] \\ & = [v \cdot P + h_{CS} \cdot FPr_{CS} \cdot P][u + h_{LN} \cdot FPr_{LN}] \\ & = [S_{CS}][u + h_{LN} \cdot FPr_{LN}] \\ & = K_{LN} \end{aligned}$$

Hence, both keys are equivalent generated by the cloud server and the Leaf Node: $K_{CS} = K_{LN} = K = [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$ □

Theorem 5.2. *Prior to initiating their communications, the cloud server CS and leaf node LN needs to have been computing identical session keys.*

Proof. In accordance with Theorem 1, we discovered how both the cloud server and leaf node generate the identical key K at their respective ends as follows:
$K_{CS} == K_{LN} == K == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P.$
Consequently, both the cloud server and leaf node would begin the generation of session key at their respective ends with results that are correct and equivalent to each other's, such that:
$SK_{CS} == SK_{LN} == H_3[ID_{CS} \parallel ID_{LN} \parallel T_{CS} \parallel T_{LN} \parallel \chi_{CS} \parallel \chi_{LN} \parallel K].$
Hence, both start the communication by using the same session key $SK == SK_{CS} == SK_{LN}$. □

5.5.2 Provable Security Analysis

This part of the security analysis procedure was mainly aimed at the formal analysis of the suggested approach.

Theorem 5.3. *If the CDH and ECDL problems for PPT attackers are computationally hard obstacles, then ROM succeeds AKA in addition to MA assurance with the suggested scheme.*

Proof. Consider a malicious PPT node. In an attempt to compromise the AKA and MA security, $\mathcal{A}$ plays in Defender responses role-playing contest against a defender

$\mathcal{D}$, which seems to be a semantic defense of the proposed model. By overcoming the ECDL and CDH challenges, $\mathcal{A}$ intends to succeed in this contest. As a result, $\mathcal{A}$ starts by giving the global variables: $(q, P, Pb_m, H_1, H_2, H_3, G_1, G_2)$. In the ability to respond to $\mathcal{A}$'s questions $\mathcal{A}$ keeps the three principal preliminary empty records.

1. H_i_list: Replies to the H_1, H_2, and H_3 inquiries are essential tasks. Consequently, H_1 only contains a few sets of tuples, like as: $(ID_m,\ a,\ b,\ h_j,\ T_j)$, where ID_m should be either ID_{CS} or ID_{LN}. Similarly, H_2 and H_3 are stored as per their respective definitions.

2. Pb_list: Defender needs public information, intending to respond to such a query. As a result, the set of tuples $(ID_m,\ \mathcal{S}^n_{P_i},\ h_j)$ are retained in such Pb_{list}. Where $ID_m \in ID_{CS},\ ID_{LN}$ and P_i either cloud server CS or Leaf Node LN.

3. SK_list: Defenders must keep such list, that comprises both item-set $\mathcal{S}^n_{P_i}$ and $SK^n_{P_i}$, to respond to such attacker $\mathcal{A}$'s sessions key inquiry. Where $SK^n_{P_i} \in \{SK^n_{CS},\ SK^n_{LN}\}$.

Each oracle query posed by adversary $\mathcal{A}$ would be addressed via defender $\mathcal{A}$. The responses to all questions are listed below.

1. H_1 *Query*: The H_i_list's default status is nil, and queries of that kind are only able to be addressed using the H_i_list. Defender $\mathcal{D}$ can indeed be responded to in the following manner whenever $\mathcal{A}$ raises such a question with ID_m at least n times:

   (a) The defender $\mathcal{D}$ needs to verify the existence of ID_m into the H_i_list or not. In case of true, Defender must follow one of the following scenarios.

      i. Check whether the $ID_m == ID_{CS}$ or not? if true then it computes the $h_{CS} = t_1 \cdot P$ after randomly selection of t_1 from the multiplicative prime integer set Z^*_q. At the end $\mathcal{A}$ added a row with a tuple $(ID_{CS},\ a,\ h_{CS},\ t_1)$ inside H_i_list and response to an attacker with the value h_{CS}.

      ii. Defender again check whether $ID_m == ID_{LN}$ or not. if true then it chooses $t_2 \in Z^*_q$ to determine the value $h_{LN} = t_2 \cdot P$. In the end, $\mathcal{A}$ added a row with a tuple $(ID_{LN},\ b,\ h_{LN},\ t_2)$ inside H_i_list and response to an attacker with the value h_{LN}.

      iii. When $\mathcal{D}$ found requester identity ID_m does not match with either ID_{CS} or ID_{LN}, then $\mathcal{A}$ simple response with any random value $T_j \in Z^*_q$ to $\mathcal{A}$.

(b) When defender found that ID_m not belongs toH_i_list then $\mathcal{A}$ randomly picked an integer from Z_q^* as: $t \in Z_q^*$ and calculate $h_j = t \cdot P$. At the last forward, this computed value h_j to adversary node $\mathcal{A}$ as well add the row with a tuple $(ID_{LN},\ h_j,\ t)$ into the H_i_list.

2. *Pb Query*: With the aim to gain the details about the public key of a participating node, an adversary $\mathcal{A}$ sends a Pb Query to the defender $\mathcal{A}$. Input parameter of such query should be $(ID_m,\ \mathcal{S}_{P_i}^n)$. The defender should respond in the following manners:

   (a) Check whether arguments $(ID_m,\ \mathcal{S}_{P_i}^n)$ including respective h_j existence in Pb_{list} or not? In case of true, the defender responds to that respective value of h_j to an attacker node $\mathcal{A}$.

   (b) Else, $\mathcal{A}$ select a random integer $c \in Z_q^*$ to compute $h_j = c \cdot P$, and this h_j value forwarded towards the $\mathcal{A}$ as a response of such query. Next, Defender $\mathcal{D}$ also add tuple $(ID_m,\ \mathcal{S}_{P_i}^n,\ h_j)$ into the Pb_{list} and update the H_i_list with tuple $(ID_m,\ h_j, \Omega)$.

3. Send $(\mathcal{S}_{P_i}^n,\ m)$: To respond to such a query Defender will follow the following cases:

   (a) If the message, $m \neq \Omega$ then the defender should behave and respond to the adversary as per our introduced protocol.

   (b) When it comes to $(m = \Omega)$, the defender $\mathcal{D}$ checks to see if the tuple $(ID_m,\ \mathcal{S}_{P_i}^n,\ h_j)$ on specific $\mathcal{S}_{P_i}^n$ exist within the H_i_list is present or not. Furthermore, $\mathcal{D}$ checks to see if the peers P_i which is apparently connected to that same specific $\mathcal{S}_{P_i}^n$ exist in the H_i_list, is present or not. Once $\mathcal{D}$ discovers that the tuples $(ID_m,\ T_j,\ h_j)$ are not existed in the H_i_list then $\mathcal{D}$ start an H_1 oracle query utilizing identity ID_m. Next, the defender $\mathcal{D}$ starts to give the response as follows:

      i. Considering the event that the specified case $\mathcal{S}_{P_i}^n$ differs significantly beyond the version, which is associated with $\mathcal{S}_{P_{CS}}^n$ or $\mathcal{S}_{P_{LN}}^n$, defender evaluates $T_j = t_j \cdot P,\ \chi_j = [t_j + h_j]^{-1} \cdot \delta_j$ and $\omega_j = \delta_j \cdot P$, where $t_j \in Z_q^*$.

      ii. When it found that instance $\mathcal{S}_{P_i}^n$ which is provide is similar to associated with instances $\mathcal{S}_{P_{CS}}^n$ or $\mathcal{S}_{P_{LN}}^n$, then defender should calculate the $T_{LN} = u \cdot P$ to determine the values as: $\chi_{LN} = [u + h_{LN}^{-1} \cdot \delta_{LN}$ and $\omega_{LN} = \delta_{LN} \cdot P$, similarly for cloud server computes $T_{CS} = v \cdot P$

to quantifies $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS}$ and $\omega_{CS} = \delta_{CS} \cdot P$. Next, $\mathcal{D}$ forwards these values $\chi_{LN},\ \omega_{LN},\ \chi_{CS}$, and ω_{CS} towards adversary $\mathcal{A}$ as a response of such query.

4. *Disclose Session Key* $(\mathcal{S}_{P_i}^n)$ *Query*: Initially the *SK_list* is empty. After receiving such query defender will start to respond in the following manners:

   (a) $\mathcal{D}$ looks at *SK_list* and returns to $\mathcal{A}$ the value $SK_{P_j}^n$, when it discovers that SK_{list} contains tuples $(\mathcal{S}_{P_i}^n,\ SK_{P_j}^n)$. There must be instance $\mathcal{S}_{P_i}^n$ is interacted with $\mathcal{S}_{P_j}^n$.

   (b) In addition, even if earlier objectives were still not met, $\mathcal{D}$ chooses the key exchange $SK_{P_j}^n$ that belongs to G of $SK_{P_i}^n$ at random and directs it to adversary node $\mathcal{A}$. The resulting set $(\mathcal{S}_{P_i}^n, SK_{P_j}^n)$ is then added to the *SK_list*.

5. *Disclose the Random Secret* (ID_m) *Query:* Defender $\mathcal{D}$ scans over the H_i*_list* for such tuple $(ID_m,\ a,\ b,\ h_j,\ T_j)$ in to respond to an Oracle query wherein ID_m should be ID_{LN} or ID_{CS}. Following that, the defender $\mathcal{D}$ would release the pertinent $(a,\ b)$ according to the identification of the nodes it invented inside the tuple. In contrast to the earlier situation, $\mathcal{A}$ randomly chooses $t_k \in Z_q^*$ and computes $h_k = t_k \cdot P$. Further, $\mathcal{D}$ deliver it to the attacker and appended the information $(ID_m,\ a,\ b,\ h_j,\ T_j)$ inside the H_i*_list*.

6. *Disclose the Public Key* $(ID_m,\ \mathcal{S}_{P_i}^n)$ *Query*: Defender $\mathcal{D}$ determines whether records $(ID_m,\ \mathcal{S}_{P_i}^n,\ h_j)$ belongs in *Pb_list* with aims to respond such query. If an item was satisfactorily identified in the list, Defender greeted the adversary by saying hello. However apart from the scenario mentioned prior, Defender $\mathcal{D}$ will pick a number at randomness $e \in Z_q^*$, to retrieve the $h_j = e \cdot P$ and communicate this to the adversary $\mathcal{D}$. The following step is to add this item $(ID_m,\ \mathcal{S}_{P_i}^n,\ h_j)$ to the *Pb_list* and the tuple $(ID_m,\ \Omega, h_j)$ to the H_i*_list* at the relatively similar time.

7. *Disclose the State* $(\mathcal{S}_{P_i}^n)$ *Query*: With respect to such query, state $\mathfrak{L}_{P_i}^n$ would be given from defender $\mathcal{D}$ once an adequate object $\mathcal{S}_{P_i}^n$ corresponds to accepted state. Instead, $\mathcal{D}$ would've been returned to opponent $\mathcal{A}$ as a null set.

8. *Test* $(\mathcal{S}_{P_i}^n)$: It corresponds to the game's final stage. This ongoing game involving defender $\mathcal{D}$ and adversary $\mathcal{A}$ would stop immediately once $\mathcal{D}$ satisfies any of the given requirements listed as:

   (a) $\mathcal{S}_{P_i}^n = \mathcal{S}_{P_{LN}}^n$ or $\mathcal{S}_{P_i}^n = \mathcal{S}_{P_{CS}}^n$.

(b) $\mathcal{S}_{P_{LN}}^n$ or $\mathcal{S}_{P_{CS}}^n$ will never be communicated with $\mathcal{S}_{P_i}^n$.

(c) If either $P_{LN} \in \mathfrak{U}_{P_i}^n$ or $P_{CS} \in \mathfrak{U}_{P_i}^n$ appeared, then the response should hold its private keys. This gave the confirmation about this peer P_i is already compromised.

Under plenty of all cases, if $t = 1$, the value that the defender $\mathcal{D}$ chooses at randomness, is reached, $\mathcal{D}$ would deliver the session key $SK_{P_i}^n$ of instance $\mathcal{S}_{P_i}^n$; alternatively, this should supply any null messages. Executes the suggested protocol with $j = 1$ and $i = 2$ accordingly. Only after the scheme is executed, the adversary $\mathcal{A}$ transmits the predicted towards the defense node $\mathcal{D}$. The defender $\mathcal{D}$ would also compute $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS},\ \omega_{CS} = \ \delta_{CS} \cdot P$ and $\chi_{LN} = [u + h_{CS}]^{-1} \cdot \delta_{LN}, \omega_{LN} = \ \delta_{LN} \cdot P$ and $K_{CS} == K_{LN} == K == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$ with key pair $(T_{LN},\ T_{CS}) = (u \cdot P, v \cdot P)$. Hence, by expanding, this expression Defender $\mathcal{D}$ got $(u \cdot v \cdot P)$. And determine the value of $(u \cdot v \cdot P)$ from $(T_{LN},\ T_{CS}) = (u \cdot P,\ v \cdot P)$ is comes under CDHP, a challenging task. As a result, a system that offers extremely strong AKA security was devised to address such CDH difficulty.

Defender $\mathcal{D}$ will therefore calculate $K_{CS} == K_{LN} == K == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$ as well as notify it to the adversary $\mathcal{A}$, so that it is aware of the values $\chi_{LN} = [u + h_{LN}]^{-1} \cdot \delta_{LN},\ \omega_{LN} = \ \delta_{LN} \cdot P$ and $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS},\ \omega_{CS} = \ \delta_{CS} \cdot P$. Now adversary would compute the partial private key pair i.e., δ_{LN} and δ_{CS} of the leaf node and cloud node respectively, but still cannot determine the full private key $FPr_{LN} = PPr_{LN} \times \delta_{LN}$ where $PPr_{LN} = s$ and$FPr_{CS} = PPr_{CS} \times \delta_{CS}$ with $PPr_{CS} = \ r$ information. Thus, adversary cannot determine the value of $(u \cdot h_{CS} \cdot FPr_{CS} \cdot P), (\ v \cdot h_{LN} \cdot FPr_{LN} \cdot P)$, and $(\ h_{CS} \cdot FPr_{CS} \cdot\ h_{LN} \cdot FPr_{LN} \cdot P)$ as full private key FPr_{CS} and FPr_{LN} is unknown to adversary. Determining the value of $(r,\ s)$ comes under the ECDH hard challenge problem. Hence, the proposed technique is a ROM-based secure protocol that can be proven. The specifications of coefficient D_1, D_2, and D_3 are as follows:

- $\mathbf{D_1}$: Once the game got ended with $\mathfrak{O}$, it became triggered.
- $\mathbf{D_2}$: It becomes active anytime $\mathcal{A}$ asks an H_1 Query.
- $\mathbf{D_3}$: Once Defender $\mathfrak{O}$ used the H_i_list to accurately reply to each H_1 Query.

□

Lemma 4: $Pb[D_1] \geq [1/T_q^n]$ in which T_q^n is used to denote the number of transmit queries.

Proof. A few selective different movements are shown as follows:

1. $\mathbf{M_1}$: It signifies that the attacker $\mathcal{A}$ is not the target of a Disclose Session key $\mathcal{S}_{P_i}^n$ or Disclose State ($\mathcal{S}_{P_{LN}}^n$) queries where $\mathcal{S}_{P_{LN}}^n = \mathcal{S}_{P_{CS}}^n$ or both $\mathcal{S}_{P_{LN}}^n$, $\mathcal{S}_{P_{CS}}^n$ communicated with each other.

2. $\mathbf{M_2}$: It indicates the disappearance of a compromised component $P_i \in \mathfrak{U}_{P_{LN}}^n$.

3. $\mathbf{M_3}$: It describes how $\mathcal{A}$ typically selects $\mathcal{S}_{P_{CS}}^n$ or its associates for the upcoming contest question.

Consequently, it interprets $D_1 = M_1 \cap M_2 \cap M_3$. Since $\mathcal{D}$ typically selects $\mathcal{S}_{P_{LN}}^n$ or its teammates for the upcoming contest question, it learned that there was no compromised node $P_i \in \mathfrak{U}_{P_{LN}}^n$ and that $\mathcal{A}$ hadn't yet run a query to disclose the session key ($\mathcal{S}_{P_i}^n$) or disclose the state ($\mathcal{S}_{P_{LN}}^n$) query either for $\mathcal{S}_{P_{LN}}^n = \mathcal{S}_{P_{CS}}^n$ or $\mathcal{S}_{P_{LN}}^n$ will be paired with $\mathcal{S}_{P_{CS}}^n$. As a result, $M_3 = M_1 \cap M_2$, and $D_1 = M_1 \cap M_2 \cap M_3$. Consequently, for network participants, $Pb[D_1] \geq [1/T_q^n]$ is achieved. □

Lemma 5: $Pb[D_2] \geq 2 \in$.

Proof. Let $\mathcal{A}$ resist from asking H_1 Query, which demonstrated as $Pb[t = t' \mid D_2'] \geq \frac{1}{2}$. It must be emphasized, although, that $|Pb[t = t'] - \frac{1}{2}| \geq \in$. By taking into account the two inequities described above. It could be concluded as;
$Pb[t = t']$
$= Pb[t = t'|D_2'] \; Pb[D_2'] + Pb[t = t'|D_2] \; Pb[D_2]$
$= Pb[t = t'|D_2'] Pb[D_2'] + Pb[D_2]$
$= \frac{1}{2} \; Pb[D_2'] + Pb[D_2]$
$= \frac{1}{2} + \frac{1}{2} Pb[D_2]$
where $Pb[t = t'| \geq Pb[t = t'|D_2'] Pb[D_2'] = \frac{1}{2} - \frac{1}{2} Pb[D_2]$

Thus, it indicates that $\frac{1}{2} Pb[D_2] \geq Pb[t = t'] - \frac{1}{2} \geq \in$, hence $Pb[D_2] \geq 2 \in$. □

Lemma 6: $Pb[D_3] \geq [1/H_q^n]$

Proof. H_q^n demonstrate the number of answered H_1 Query. Therefore, the likelihood that defender $\mathcal{D}$ will correctly answer from all H_1 Query is $Pb[D_3] \geq [1/H_q^n]$. □

Lemma 7: $Pb[\mathcal{A}(P, \; h_{CS} \cdot FPr_{CS} \cdot P, \; h_{LN} \cdot FPr_{LN} \cdot P)] = (h_{CS} \cdot FPr_{CS} \cdot h_{LN} \cdot FPr_{LN} \cdot P)$ where $P \in G$ and $h_{CS}, \; FPr_{CS}, \; h_{LN}, \; FPr_{LN} \in Z_q^* \geq \frac{\in}{H_q^n \; T_q^n}$.

Proof. To determine the values of $(h_{CS} \cdot FPr_{CS} \cdot h_{LN} \cdot FPr_{LN} \cdot P)$ given pair variables $(P,\ h_{CS} \cdot FPr_{CS} \cdot P,\ \ h_{LN} \cdot FPr_{LN} \cdot P)]$, adversary $\mathcal{A}$ must solve the CDH challenge. The possibility to calculate the CDH concern will thus be; $Pb[\ \mathcal{A}(P,\ h_{CS} \cdot FPr_{CS} \cdot P,\ \ h_{LN} \cdot FPr_{LN} \cdot P)] = (h_{CS} \cdot FPr_{CS} \cdot h_{LN} \cdot FPr_{LN} \cdot P) = Pb\ [M_1 \cap\ M_2 \cap M_3] \geq \frac{\in}{H_q^n\ T_q^n}$. □

5.5.3 Informal Security Analysis

1. ***Men in the Middle resist:*** In our suggested scheme, Leaf node LN sends a tuple set: $(\ ID_{LN},\ FPb_{LN},\ S_{LN},\ \chi_{LN},\ \omega_{LN},\ T_{LN},\ t_1)$ towards the cloud server CS. Assume there is an attacker $\mathcal{A}$ among the communication nodes, that wants to participate and gain the shared information via the internet. For this purpose, let $\mathcal{A}$ compute $T_A\ =\ \sigma \cdot P$ and forward the modified tuple i.e. $(\ ID_{LN},\ FPb_{LN},\ S_{LN},\ \chi_{LN},\ \omega_{LN},\ T_A\ ,\ t_1)$ and send it towards the CS. As per the scheme, first CS needs to verify the authentication of the requester node. For this purpose, it computes $[\chi_{LN}[T_A + H_2[ID_{LN} \parallel ID_{CS} \parallel T_{LN}] \cdot P]] = [[u + h_{LN}]^{-1} \cdot \delta_{LN}[\sigma \cdot P + h_{LN} \cdot P]] = [[u + h_{LN}]^{-1} \cdot \delta_{LN}[\sigma + h_{LN} \cdot P]] \neq\ \omega_{LN}$ (received). Thus, LN's authentication verification failed on the CS side. Similarly, when adversaries perform any modification in forwarded tuples from CS to the LN phase. LN also failed to verify the authenticity of the requester. Thus, there is no possibility of men in the middle attack between the leaf node LN and the cloud server CS.

2. ***Mutual Authentication:*** The proposed scheme provides mutual authentication as both the leaf node and cloud server verify the authenticity of each other at their respective ends. The steps involved in the process of verification of authenticity are as follows:

   (a) In our suggested scheme, Leaf node LN send a tuple set towards the cloud server CS: $(ID_{LN},\ FPb_{LN},\ S_{LN},\ \chi_{LN},\ \omega_{LN},\ T_{LN}, t_1)$. Where first CS needs to verify the Authenticity of the requested node, for this CS crosscheck the computed value $[\chi_{LN}\ [T_{LN} + H_2[ID_{LN} \| ID_{CS} \| T_{LN}] \cdot P]]$ with received value ω_{LN} as follows: Computed value at CS side:
   $== [\chi_{LN}[T_{LN} + H_2[ID_{LN} \parallel ID_{CS} \parallel T_{LN}] \cdot P]]$
   $== [[u + h_{LN}]^{-1} \cdot \delta_{LN}[u \cdot P +\ h_{LN} \cdot P]]$
   $== [[u + h_{LN}]^{-1} \cdot \delta_{LN}[u\ +\ h_{LN}] \cdot P]]$
   $== [\delta_{LN} \cdot P]$
   $==\ \omega_{LN}$ (Received value by CS).

As per the proposed scheme, if it returns a true value, that means it verified the authenticity of the requester node successfully. Hence, authentication is done for LN on the CS side.

(b) Similarly, when CS forwards the tuple set towards the LN, it also computes the value and checks it with what was received ω_{CS} as follows:
Computed value at LN side:
$== [\chi_{CS}\ [T_{CS} + H_2[ID_{LN} \parallel ID_{CS} \parallel T_{CS}] \cdot P]]$
$== [[v + h_{LN}]^{-1} \cdot \delta_{LN}[v \cdot P\ +\ h_{LN} \cdot P]]$
$== [[v + h_{LN}]^{-1} \cdot \delta_{LN}[v\ +\ h_{LN}] \cdot P]$
$== [\delta_{CS} \cdot P]$
$== \omega_{CS}$ (Received value by LN).

Thus, the authentication verification done on the LN side is also about CS' authenticity.

Overall, based on the above proofs, we can state that our proposed scheme provides mutual authentication successfully between communicating points LN and CS.

3. ***Key Compromise Impersonation Resistance:*** The proposed scheme resists the impersonation attack whenever an intruder $\mathcal{A}$ wants to impersonate LN to CS or CS to LN, yet the full private key of LN or CS is known to an intruder. Suppose that there is an adversary $\mathcal{A}$ who impersonates the LN with having the partial $(PPr_{CS},\ \delta_{CS})$ and full private key FPr_{LN}. Still, it $\parallel$ is unable to impersonate the CS node to LN.

   (a) First, $\mathcal{A}$ failed to obtain the signature pair of CS: $\chi_{CS} = [v + h_{CS}]^{-1} \cdot \delta_{CS}$ and $\omega_{CS} =\ \delta_{CS} \cdot P$ from the known private key information of the leaf node. And let us assume that the adversary $\mathcal{A}$ also has the key pair $(T_{LN},\ T_{CS}) = (u \cdot P, v \cdot P)$. The computation of determining the value v comes under the CDH hard assumption. Whereas the private key δ_{CS} itself preserves the secrecy due to its dependency on the variable ($\lambda_{CS} =\ a \cdot P$ and master private key x).

   Thus, by knowing about a few key pieces of information, adversaries are still unable to determine the other's private key information because of the hard assumption of CDH.

   (b) Second, the adversary $\mathcal{A}$ also failed to obtain the key: $K_{CS} == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$ due to its dependency over all secret keys

such as: $(v,\ u)$ and $(FPr_{CS},\ FPr_{LN})$. Thus, acquire the value of $(v,\ u)$ from the known key pair $(T_{LN},\ T_{CS}) = (u \cdot P, v \cdot P)$ comes under the CDH hard problem. And determining the value of FPr_{CS} also depends on two partial private key information which is: $\lambda_{CS} = a \cdot P$ and $PPr_{CS} = r$ which is known by CS itself unless he/she cannot share, and computation of FPr_{CS} it comes under the ECDH problem.

As a result, the adversary $\mathcal{A}$ cannot impersonate the LN (or CS) to CS (or LN).

4. ***Replay Attack Resistance:*** Let's assume that adversary $\mathcal{A}$ gains the information about any previous session key SK information and wants to participate in the present session as an authorized node. As in the proposed scheme, we use the concept of key freshness, concepts like choosing the secret random keys $(a,\ b) \in Z_q^*$ and secret ephemeral key $(v,\ u) \in Z_q^*$ for each new session.
Session key: $SK = H_3[ID_{CS} \parallel ID_{LN} \parallel T_{CS} \parallel T_{LN} \parallel \chi_{CS} \parallel \chi_{LN} \parallel K]$, where $K == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$, $(T_{LN},\ T_{CS})$ and $(\chi_{CS},\ \chi_{LN})$ are also depends on freshness of key $(v,\ u)$ and $(a,\ b) \in Z_q^*$.

   Thus, for each new session, a new session key should be generated so that there are no means to participate in the network using another session's key. Because during the authentication verification process, it should identify as an attack A. Hence, the proposed scheme resists the replay attack.

5. ***Known Key Secrecy Attack:*** Our proposed scheme also prevents the computation of any prior or future session key information based on the known current session or message information by an adversary $\mathcal{A}$. Session key SK computation done as by both LN and CS as: $SK = H_3[ID_{CS} \parallel ID_{LN} \parallel T_{CS} \parallel T_{LN} \parallel \chi_{CS} \parallel \chi_{LN} \parallel K]$
   where $K == [v+h_{CS} \cdot FPr_{CS}][u+h_{LN} \cdot FPr_{LN}] \cdot P$, $(T_{LN},\ T_{CS})$ and $(\chi_{CS},\ \chi_{LN})$ by expand this expression we got $(u \cdot v \cdot P)$. And determine the value of $(u \cdot v \cdot P)$ from $(u \cdot P,\ v \cdot P)$ It comes under CDHP, a challenging task.

   Where also determining the full private key information comes under the ECDH hard challenge problem. Hence, the scheme successfully prevented the K-KS attacks in the proposed scheme.

6. ***No key Dominance:*** the session,In our proposed protocol computation of session, key SK is done as at both LN and CS endpoint like as: $SK =$

$H_3[ID_{CS} \parallel ID_{LN} \parallel T_{CS} \parallel T_{LN} \parallel \chi_{CS} \parallel \chi_{LN} \parallel K]$ where $K == [v + h_{CS} \cdot FPr_{CS}][u + h_{LN} \cdot FPr_{LN}] \cdot P$, $(T_{LN},\ T_{CS})$ and $(\chi_{CS},\ \chi_{LN})$ and the values $(v,\ u) \in Z_q^*$ are chosen by LN and CS respectively. Even other variables like the full private key of both LN and CS $(FPr_{LN},\ FPr_{CS})$ are involved.

Thus, we can clearly see the equally dominant nature of both participating nodes, CS and LN, during the computation of SK. So, whenever an intruder $\mathcal{A}$ or any of the participating node LN over CS (or CS over LN) want to enforce the computation of SK as a fixed value that exists among a short range of Z_q^*, then because of no-key dominance nature, it is not possible for any nodes, whether they are LNs (or CS) or adversary$\mathcal{A}$, to enforce the SK value as a fixed or predefined number. Thus, no one can misuse the SK.

7. ***Perfect forward secrecy:*** Let's suppose that LN and CS share their private key information (FPr_{LN}, FPr_{CS}) with adversary $\mathcal{A}$, still $\mathcal{A}$ could not determine any previous session key information due to freshness nature of other ephemeral keys: $(v,\ u) \in\ Z_q^*$. Where to determine the value of $(v,\ u)$ from the known key pair $(T_{LN},\ T_{CS}) = (u \cdot P,\ v \cdot P)$ it comes under the hard assumption of the CDH problem.
   Thus, our proposed scheme preserves the perfect forward secrecy nature of the session key.

8. ***Ephemeral Key Leakage Resistance:*** Let's suppose that LN and CS share their ephemeral keys: $(v,\ u)$ with adversary $\mathcal{A}$, still, $\mathcal{A}$ could not determine the session key SK due to its dependency over other private keys information like $(FPr_{LN},\ FPr_{CS})$ which is dependent on variable pairs $(a,\ b) \in Z_q^*$ and $(r,\ s) \in\ Z_q^*$. Determining these values also comes under the CDH problem. Due to having any ephemeral keys: $(v,\ u)$ adversary is also unable to gain any information about previous SK due to freshness nature of ephemeral keys and determine the value of $(v,\ u)$ from the known key pair $(T_{LN},\ T_{CS}) = (u{\cdot}P,\ v{\cdot}P)$ it comes under the hard assumption of the CDH problem.
   Thus, in case of leakage of any ephemeral key, our proposed scheme resists such types of leakage attacks successfully.

5.6 Summary

In this chapter, we designed an improved certificate-less authenticated key agreement protocol for the WBAN network assisted by a cloud server. First performs the cryptanalysis on the scheme proposed by Cheng et al. [164]. Where we identify various security challenges, like the possibility of MITM, impersonation attacks, imperfect mutual authentication, design flaws in hash definitions, and most importantly, contradiction with the definition of a certificateless cryptosystem.

As a result, we introduced an improved version of Cheng's scheme. We covered all the identified flaws in the comment section. A thorough formal and informal security analysis has been done to prove the strong security of the proposed scheme.

The formal analysis using the ROM model shows the security against attackers. The correctness proof and the informal analysis show the achievement of various predefined security goals. Thus, all the comments are properly resolved by our designed scheme, which is superior to Cheng et al.'s scheme.

CHAPTER 6

Quantum-Safe UAV-Assisted Blockchain Authentication for Secure Vehicle Communications in Intelligent Transportation System

6.1 Introduction

The Intelligent Transportation System (ITS) refers to the application of advanced information and communication technologies to enhance the efficiency, safety, sustainability, and overall performance of transportation networks. As urban populations grow and traffic congestion intensifies, traditional traffic management methods are becoming inadequate. ITS offers a transformative solution by integrating real-time data collection, analysis, and decision-making into the transportation infrastructure [166]. One of the fundamental aspects of ITS is real-time traffic monitoring. Sensors embedded in roads, along with surveillance cameras and vehicle telematics, provide constant updates on traffic flow, congestion levels, and incidents. This data is processed in centralized traffic management centers, which use algorithms to optimize traffic signals, reroute traffic, and inform road users through dynamic message signs or mobile applications. As a result, travel times are reduced, fuel consumption is minimized, and emissions are decreased—contributing to both economic and environmental benefits [167].

Additionally, ITS is essential for facilitating autonomous driving as well as intelligent vehicles. Vehicles are capable of sharing data regarding position, speeds, and traffic conditions with one another, utilizing Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) networking. As connected and autonomous vehicles (CAVs) become increasingly prevalent, the need for fast, reliable, and secure communication frameworks has grown substantially [167]. Data sharing regarding the traffic networks among the vehicles is possible with the help of On Board Units (OBU) and Road Side Units (RSU), which are mainly used to collect, analyze, and share the data as per requests made by vehicles [97]. To enable V2V and V2I communication, RSUs are placed beside roads; this setup follows the static architecture.

Numerous ITS applications are supported by RSU, such as infotainment, transportation, responding to emergencies, transportation management, and vehicular inspection. The installation of RSUs raised high infrastructure costs, as it involves physical installation, power supply, and network backhaul. This makes it unfeasible for large-scale or remote areas. Therefore, they are less efficient in dynamic traffic situations due to their fixed deployment, expensive setup costs, and also suffer from physical vulnerability and restricted adaptability [168].

Those constraints have prompted research into more adaptable and transportable options, such as Unmanned Aerial Vehicles (UAVs), also known as drones, sensors, or interacting modules [169]. It can process data in real-time. In ITS architecture, by deployment of UAV, several challenges will be resolved efficiently, such as: due to its unmanned nature, it works as a mobile base station, easier for V2V and V2I communication. Easy and quickly deployed whenever an event or when a traffic pattern changes. Flexible in coverage area that doesn't cost too much. UAVs don't need as much ground infrastructure as RSUs do. A few UAVs can cover a larger area, which makes it cheaper to set them up and keep them running. Also, reduce the delay and improve the quality of services (QoS) as UAVs can broadcast and receive real-time information regarding traffic density, hazards, and road conditions to automobiles and traffic control centers. Making the network perform better by balancing loads, UAVs can help busy RSUs and cellular networks by taking traffic off [169].

Recently, researchers have been focusing on the integration of UAVs with vehicles and RSUs in order to optimize the communication and reduce traffic congestion. Researchers are also focused and concerned about the security and privacy of sharing information in ITS [168]. Integration of UAVs in the ITS architecture gives new avenues to attackers, who can perform various malicious activities, which are described below.

- Attackers might be able to intercept or fake UAV control signals, which would let them control UAVs without permission or send them to do horrible things.
- Attacks on wireless communication channels between UAVs and automobiles can use jamming or eavesdropping. These attacks can make the channels unusable or disclose private information.
- Restricted storage: UAVs can't employ complicated encryption methods since their batteries and processor power are restricted.
- Hard to build trust in a dynamic, changing environment, and we need light protocols and work in real time.
- Agreeing on a safe session key. Vehicles and UAVs need to be able to transfer keys fast and safely to stop man-in-the-middle (MITM) and replay attacks.

Consequently, to ensure secure data transmission and storage in ITS environments, cryptographic procedures must be employed to protect vehicles' privacy and prevent network attacks.

Certificate-based batch verification serves as an alternative method for data validation. It employs a public key (pseudonymous) certificate from a reputable certificate authority to verify the vehicle's authenticity. Managing RSUs and similar entities entails significant expenses related to certificate storage and administration [170]. A potential solution to this issue is the incorporation of blockchains to introduce "credibility" and "auto-check" nodes within the ITS architecture.

A blockchain policy is a database that permits only additions and is operated by nodes inside a peer-to-peer (P2P) network. Blockchain-based trust management is significantly more cost-effective than certificate-based trust management [171]. Each node in a blockchain maintains connectivity with its neighbors, transmits and authenticates signed messages, and ensures data blocks remain synchronized. The structure of a blockchain guarantees consumers immutable, decentralized, autonomous, and contractual network advantages. Incorporating auto-credibility into the ITS enhances blockchain as a robust and scalable framework capable of verifying message reliability, assessing vehicle performance, and autonomously monitoring routine communication reports [91].

The emergence of quantum computing challenges numerous traditional cryptographic methods, resulting in improvements in public-key cryptography that are susceptible to contemporary technological advancements [172]. Lattice-based cryptography (LBC) represents a prospective post-quantum solution for traditional and

emerging security challenges, including key exchange, encryption, and digital signatures. LBC algorithms offer enhanced security against quantum cryptanalysis due to their intrinsic properties, while also being straightforward to implement [173]. Thus, we introduced an authenticated key agreement protocol by applying the LBC mechanism to resist against the quantum challenges. And by appending the blockchain technology, we create the most trustworthy and transparent scheme for the ITS architecture associated with UAVs to resolve the challenges related to the RSU. Thus, an attacker can't perform any malicious activity like MITM attacks or replay or impersonate any communicating units (RSU, UAV, and vehicles). Our main goal is to achieve message authenticity, including identity verification and data confidentiality, as well as signature unforgeability.

6.1.1 Problem Statement

There have been various cryptographic techniques used by researchers to realize privacy-preserving authentication in vehicular networks. The PKI-based schemes use certificates to authenticate a vehicle. The Certificate Authority (CA) is responsible for issuing and managing the certificates. However, due to the large size of certificates, PKI-based schemes incur high communication costs. Moreover, certificate revocation is another challenge because CRLs (Certificate Revocation Lists) are huge in size, which leads to the requirement of large storage and high computation cost for storing and CRL checking, respectively. Other than PKI-based schemes, Identity-Based Signature (IBS) is considered another approach for authentication in VANET. In IBS schemes, the node's identity is sent to the Key Generation Center (KGC), which returns the private key. Compared to conventional PKIs, IBS removes the certificate requirement for vehicle public key verification. Thus, certificates are not sent during communication. Additionally, IBS eliminates managing CRLs, which results in low additional overhead. However, since the KGC generates all the private keys in IBS, it can access all of them, which could lead to the key escrow problem [174]. Therefore, a certificateless signature is considered an efficient approach for vehicle-to-vehicle authentication in VANETs. However, these convention mechanism are failed with quantum computer thus, we are motivated to utilize the quantum safe cryptosystem.

Combining blockchain with UAV technologies in Intelligent Transportation Systems could change the way traffic is managed and monitored in real-time and keep roads safe. UAVs are capable of rapidly collecting and disseminating vital traffic data for RSUs and vehicles via their flexible coverage and fast portability. However, the fact that UAV-assisted ITS is highly dynamic and diverse makes it challenging

to maintain privacy and security. The system is vulnerable to challenges, including imitation, replay, and key-compromise counterfeiting assaults, since it often hands off control, transforms domains, and utilizes public wireless channels.

6.1.2 Main Contributions

In the aforementioned fast-changing circumstances, conventional cryptographic techniques typically don't have the speed, adaptability, or post-quantum protection that are needed. As quantum computing becomes increasingly prevalent, many present approaches may become ineffective. This means that we need to create quantum-resistant ideas that use sparse computation as well as communication expenses, while still ensuring mutual authentication, integrity, and anonymity.
This study addresses the problems listed above by creating a robust, lightweight, and quantum-safe authentication and key agreement model that can be utilized with blockchain–UAV-assisted ITS. The main contributions are as follows:

- **Blockchain–UAV-Assisted ITS Framework:** A novel system framework is proposed that integrates UAVs, RSUs, vehicles, and blockchain technology to facilitate robust and viable Intelligent Transportation System (ITS) amenities, while also guaranteeing reliable event recording and system transparency.
- **Post-Quantum Secure Protocol:** We design an AKA protocol founded on lattice-based cryptography. The scheme offers fundamental resilience against quantum attackers and facilitates efficient computation on appliances with limited resources.
- **Integration of Blockchain for Enhanced Auditability:** Blockchain technology is used to ensure that all stored records are authenticated as well as more accurate, which can be verified. Thus, this makes the ITS framework more transparent and reliable.
- **Comprehensive QROM Security Verification:** This article gives a comprehensive security review of the proposed strategy using the Quantum Random Oracle Model (QROM). It shows how strong the protocol is at making sure that both parties can authenticate each other, that session keys are kept secret, and additionally that signatures can't be forged.
- **Evaluation of Performance and Cost Effectiveness:** We do thorough computational analysis and simulations, which demonstrate our solution significantly lowers the costs of computing and communication compared to existing methods, while still keeping security intact.

6.2 Background

This section has outlined the network model, security model, and security goals defined for the proposed scheme.

6.2.1 System Model

Our proposed system architecture mainly consists of six units: Trusted Administrator (TA), Registration Authority (RA), Road Side Unit (RSU), Unmanned Aerial Vehicle (UAV), Vehicles (V), and Cloud server Provider (CSP). These units are grouped into four layers: Vehicle, Edge Device, Consensus, and Cloud layer. Definitions and roles of each involved unit have been described below:

A. **Trusted Administrator**: TA is a fully trusted authority that defines and broadcasts its global system parameters. Mainly used for providing full support to all other units, especially the RSU and Vehicles, with the assessment of CSP.

B. **Registration Authority**: RA refers to trusted units that are incorporated into the blockchain as nodes. The RA produces smart cards and responses per the received registration request messages from units such as RSU, V, or UAV. RA registered them within its local domain, as inside the system, multiple RAs belong. This indicates that only units belonging to the domain range can be registered with their respective RA.

C. **Road Side Unit**: RSU is a fixed device positioned alongside the roadway. It primarily acquires traffic information from the CSP and disseminates this data to the asking vehicles. Likewise, the RSU gathers real-time traffic and vehicular data and transmits it to the CSP.

D. **Vehicles**: In ITS, vehicles function as mobile nodes, collecting and acquiring real-time traffic data from the CSP to facilitate effective route optimization and traffic control. To improve traffic awareness throughout the system, it captures present data (speeds, position, and as well as route circumstances) and sends it to RSUs or directly to CSPs. Vehicles receive notifications regarding accidents, road congestion, or weather conditions, allowing drivers to make informed choices.

E. **Unmanned Aerial Vehicle**: UAV functions as a wireless device that moves according to traffic congestion status. It observes real-time traffic flow, inci-

dents, and route situations from an aerial perspective, therefore offering extensive coverage in areas. UAVs can adaptively reposition themselves to congested locations to collect and transmit essential traffic information to CSPs, RSUs, and Vehicles, hence improving traffic control and emergency response. Moreover, the vehicles inside a UAV's coverage area may regularly alter as the UAV navigates, facilitating adaptable, efficient, and scalable data acquisition in ITS.

F. **Cloud Service Provider**: A Cloud Service Provider (CSP) in ITS is a trustworthy and secure server that offers robust data storage, analysis, and evaluation of real-time traffic data gathered from RSUs, UAVs, and vehicles. It compiles and examines traffic flow, road hazards, and surrounding information to produce actionable conclusions. CSP supplies RSUs with enhanced traffic and route information, enabling them to deliver precise, immediate data for mitigating congestion and selecting optimal routes.

6.2.2 System's Assumption and Architecture

We present a quantum-safe security architecture to safeguard connectivity and data exchange within Intelligent Transportation Systems (ITS). Where each Autonomous Vehicles (AVs) are designed for minimal human input and is integrated with Electronic Control Units (ECUs), cameras, and On-Board Units (OBUs) to capture environmental information. Further, such information is processed almost instantly by edge-based computing resources to support real-time decisions. However, connecting vehicles to the internet makes them more vulnerable to risks like node imitation, MITM, replay, and channel alteration, etc. These kinds of attacks could lead to problems with the connection between vehicles, RSUs, and UAVs, putting both data integrity and reliability of service at risk. To address such risks, we suggest a combination system incorporating blockchain's decentralization, authenticity, and consistency with lattice-based cryptography (LBC) for post-quantum robustness. This will build assurance and make sure that interactions between different nodes, such as vehicles, RSUs, and Unmanned Aerial Vehicles (UAVs), are secure.

The fundamental assumptions of the proposed work are outlined as follows.

i. A permanent distributed ledger, BC, is implemented in the ITS to provide secure administration of certificates while enabling the exchange of inter-domain records and authenticated notifications.

ii. The SC is accessible at all times and is reliable for the creation or preservation

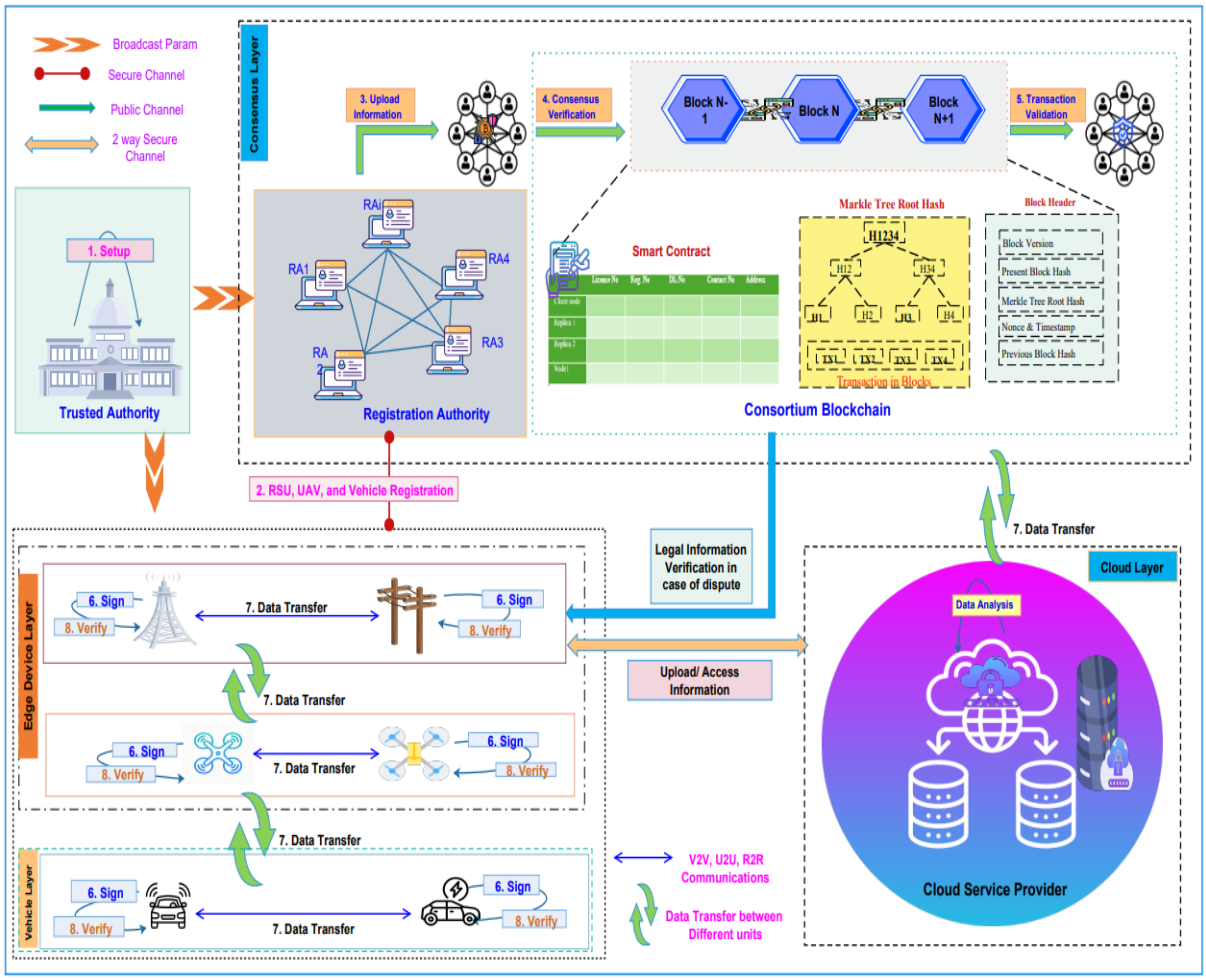


Figure 6.1: Proposed Model for Blockchain-Enabled UAV-Assisted ITS Environment.

of records. An invader can rarely modify entries on the blockchain, which is a continually operating distributed database.

iii. The essential keys of UAV U_i should not be deleted or altered frequently unless there are allegations of malfeasance. The RA will nullify the server's operation, block any communication queries from the Vehicle V_i and the RSU R_i. Then and cut off each connection if the UAV exhibits signs of a compromise.

iv. Exposure to particularly classified sensitive information is restricted to authorized users only.

We designed a secure, authenticated key agreement mechanism that keeps data confidential and protected. The units we stipulated atop are put into layers to do what our proposed scheme needs to do. These layers are: Vehicle layer, Consensus layer, Edge device layer, and Cloud layer, which are defined below:

A. ***Vehicle Layer:*** This layer of connectivity consists of connected vehicles that use V2V and V2I communication methods to share information and work together to find their way. Quantum-safe encryption techniques utilizing LBC have made those conversations secure. It additionally makes guaranteed that vital data, including speeds, locations, and routing recommendations within the Vehicle's ecosystem, is authentic and confidential. These cars have sensors, like OBU, that provide them with real-time information so they can make quick decisions about things like braking assist, changing lanes, and speeding up or slowing down. These are necessary for figuring out the best paths of transportation, easing traffic on roadways, and preventing harmful disasters.

B. ***Edge Device Layer:*** This layer has Roadside Units (RSUs) and UAVs, which render it possible to handle data in an instant right at the edge of the network. This reduces latency and improves decision-making for traffic control and responding to emergencies. UAVs are situated close to vehicles and RSUs on purpose so that they can quickly sort through and evaluate large amounts of information from vehicles. Attackers could use weaknesses in the access points of edge servers and intercept open wireless communication channels. So, to protect against such security dangers, the suggested system infrastructure connects each UAV and RSU to the LBC, which makes sure that the data stays safe and quantum threats don't happen. The consortium mechanism is used to implement to guarantee the consistency and integrity of every transaction and information transmissions inside the network, resulting in strengthening the system's authenticity and consistency.

C. ***Consensus Layer:*** This layer uses the consortium mechanism alongside blockchain technology to preserve a decentralized, that is reliable as well as unalterable of every transaction and information transfer within the ITS environment. At this point, the distributed ledger keeps track of all the information exchanges and transaction flows related to CAVs in blocks that are protected by LBC, which keeps quantum-capable attackers from intercepting or changing the data. So, this agreement step makes sure that data is accurate and can't be changed, which is necessary for reliable and safe vehicle performance in the world of intelligent transportation.

D. ***Cloud Layer:*** This layer is where most of the CAVs ecosystem's data is kept. It stores and processes a lot of information online. It has the processing power and storage space to let CAVs handle all the sophisticated data they need to. The cloud server handles things such as vehicle owner information,

money, and provenance knowledge that aren't necessary and won't put people's safety at risk if they are delayed. The edge server checks vital information to enable it to make quick choices. Subsequently, information is transferred to the cloud for persistent storage, enabling vehicles to easily access it at a later time and to plan and analyze it for an extended period. We store high-definition (HD) maps created using LiDAR and other sensors here. These maps include things like lanes, road markers, and barriers that help with route planning and being aware of the surroundings. This makes the navigator more accurate and reliable.

Figure 6.1 represents the working architecture of each layer included in our proposed scheme. The process is initiated by broadcasting the system parameter by the trusted authority via a secure channel only. Then all the communicating units: vehicles, RSUs, and UAVs, are mandatory to register themselves with the registration authority via a secure channel only. Now, the consensus layer should be updated with each tuple and smart contract done between the units and the system. In this layer, we used the consortium protocol using blockchain to make the system more transparent, immutable, and more robust against various attacks, such as DDoS, etc. Blockchain helps to verify the identity in case any dispute occurs in the communication network. Vehicle information regarding its positions, speed, etc, should be updated by RSU and analyzed before being stored in the cloud. Thus, CSP transfers the requesting traffic information via a secure channel to the RSU whenever a request query is raised by any vehicle. All the communicating nodes append the digital signature to the message; thus, before the response, it should be verified. To resist the quantum attacks, we used lattice-based cryptography to construct our scheme.

6.2.3 Blockchain Justification & Confidentiality Architecture

Traditional Road Side Unit (RSU)-based certificate authorities suffer from three critical limitations: (1) single-point-of-failure vulnerabilities due to physical attacks on static infrastructure, (2) high deployment costs averaging $50K per kilometer for installation and maintenance, and (3) centralized key escrow that creates quantum vulnerabilities. Our consortium blockchain eliminates these weaknesses through distributed Registration Authorities (multiple RA nodes validated through PBFT consensus), fault tolerance ($f < 1/3$ malicious nodes under vehicular churn), and programmable governance via smart contracts.

A. **Confidentiality Model**:

Sensitive vehicle data never appears in plaintext on-chain, as described by table 6.1

Table 6.1: On-Chain vs Off-Chain Storage Model

Data Type	**On-Chain**	**Off-Chain**	**Access Control**
Transaction Hash	Merkle Root (32B)	–	Public (integrity)
Vehicle Pseudonym	$\mathrm{ID}_V^* = \mathrm{H}(\mathrm{ID}_V \parallel \mathrm{T}_i \parallel \mathrm{r})$	–	RA nodes only
Trajectory Data	$\mathrm{C}_i = \mathrm{H}(\mathrm{Enc}_{\mathrm{pk_{CSP}}}(\mathrm{traj}_i))$	IPFS CID	CSP + RA
Session Keys	KDF hash, Ephemeral, Session-only		

B. **RA Smart Contract Access Control (Pseudocode)**

```
{
            function revealIdentity(bytes32 commitment,
bytes32 disputeHash)
public onlyRA
returns (string memory realID) {
 require(disputeHash == keccak256(commitment));
 return decryptCommitment(commitment);
}
```

This hybrid model ensures GDPR-compliant conditional anonymity while maintaining blockchain auditability (detailed in Section 1.1).

6.2.4 Adversary Model I

We established a standard security framework for the proposed approach employing a Quantum-accessible Random Oracle Model (QROM). An adversary $\mathcal{A}$ is granted oracle accessibility to the randomized hash function sets throughout the QROM$\Im$: $(\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5, \mathcal{H}_6)$. It is limited to acquiring the outcome $\Im(\wp)$ by posing an inquiry to an oracle $\Im$ at the historical stage $\wp$. To develop a real system, the random oracle $\Im$ is ultimately converted into the specific hash function values $(\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5, \mathcal{H}_6)$, enabling $\mathcal{A}$ to evaluate such hashed services on quantum states that exist. To address this challenge, we allow the entity $\mathcal{A}$ to communicate in the $\Im$ "in combination phase," which indicates that it can transmit its quantum states $\Sigma\vartheta_\wp|\wp\rangle$ $\Im$ and get the generated reveal $\Sigma\vartheta_\wp|\Im(\wp)\rangle$ (suitably encrypted to achieve a unidirectional transition). We verify the security to counter

the "Quantum Fundamental Unforgeability over History-Free Minimization (QFU-HFM)" associated with the introduced scheme. $\mathcal{A}$ should request $\Im$ a response that resembles a combination of an exponential set of stages. The minimization procedure needs to be verified $\Im$ at every stage of the combination phase. Such inquiries are supposed to be persistently answered by a quantum-obtainable counterfeit random generator(CRG).

QROM Model Assumption: In the QROM, we assume a stronger adversary with quantum computational capabilities:

- The adversary can submit quantum superposition states to the random oracle and receive superposition outputs.
- This models attackers who can evaluate hash functions on multiple inputs simultaneously using quantum algorithms (e.g., Grover's algorithm).
- Despite quantum capabilities, the adversary remains computationally bounded, with a limited number of quantum queries q.
- In our UF-CMA security proof for signatures, the adversary has quantum access to the hash function H but only classical access to the signing oracle.

The QROM provides stronger security guarantees by accounting for post-quantum threats, which is essential for IoT systems that must remain secure against future quantum attacks.

Definition 6. (Counterfeit Random Generator): A quantum obtainable CRG is computed efficiently. $\forall$ quantum algorithms $\mathcal{Q}_{Algo}$:

$$|\mathbf{Pb}_{D1}[\mathcal{Q}_{Algo}^{CRG(k,\cdot)}(1^{\varrho}) = 1] - \mathbf{Pb}_{D2}[\mathcal{Q}_{Algo}^{\Im(\cdot)}(1^{\varrho}) = 1]| < \varepsilon.$$

Where $\varepsilon = \varepsilon(\varrho)$ is not a negative operation and it is nil in ϱ, $\Im$: a random oracle, $\mathbf{Pb}_{D1}$ is a probability-distribution operation on the k length of ϱ, and $\mathbf{Pb}_{D2}$: probability-distribution operation on the random oracles and the sampling value of the outcome from $\mathcal{Q}_{Algo}$.

Definition 7. (Sign QSB-AKA in QROM): Let's $\mathcal{C}$ run key generation for honest parties and give $\mathfrak{a}dv$:

1. Quantum access to $\mathcal{H}$ (all $\mathcal{H}_i$ the scheme uses);

2. Classical signing oracle $\mathsf{Sign}(\cdot)$ for any honest identity except the target identity (in selective forgery, we may permit $\mathfrak{a}dv$ choosing a target identity initially).

$\mathfrak{a}dv$ wins if it outputs $(m^{\star}, \sigma^{\star})$ where $\sigma^{\star}$ is a valid signature and $(m^{\star})$ was not submitted to $\mathsf{Sign}(\cdot)$ previously. We denote the adversary's success probability by $\Pr[\mathsf{Forge}]$.

6.2.4.1 Quantum Computer's Hard Problem The hard challenges in the quantum context are conceptualized as follows:

Definition 8. The Challenge: that is, it $\mathcal{C}_Q = (Game_Q, \chi_Q)$ is founded on the SIS/ISIS lattice hard assumption, which $Game_Q$ delineates a competition that is $\mathcal{A}$ conducted against the conventional challengers $\mathcal{C}$, and this poses significant computing difficulties over quantum computing devices. The subsequent phases outline the activity conducted around $\mathcal{A}$ and $\mathcal{C}$.

- $\mathcal{C}$ calculates a parameter w from the input 1^{ϱ} and transmits the result $\mathcal{A}$ as their input. Subsequently, it $\mathcal{A}$ is run on $\wp$ and is allowed to do standard queries to $\mathcal{C}$.
- The resultant value $\Re$ is subsequently produced by $\mathcal{A}$ and transmitted to $\mathcal{C}$.
- $\mathcal{C}$ then evaluates $\wp$, Re, and the standard questions posed by $\mathcal{A}$, returning either 1 or 0.

χ_Q denotes a number that is real, such as $0 \leq \chi_Q < 1$. Although that is possible to be expressed as an expression of $\wp$, we only needed the constant χ_Q to obtain our security evaluation; more especially, it χ_Q remains 0 or $\frac{1}{2}$.

If $\mathcal{C}$ returns 1, then may conclude the $\mathcal{A}$ won against $Game_Q$. We identify the benefit of $\mathcal{A}$ in challenge $\mathcal{C}_Q$ as $\mathfrak{adv}_{\mathcal{A},\mathcal{C}_Q}$. Which is defined as

$$\mathfrak{adv}_{\mathcal{A},\mathcal{C}_Q} = |\mathbf{Pb}[\mathcal{A} \text{ won the} Game_Q] - \chi_Q|$$

Definition 9. (Computationally intractable problem): An issue. The principle of $\mathcal{C}_Q = (Game_Q, \chi_Q)$ proves computationally challenging under SIS/ISIS premises for quantum systems if the advantage $\mathfrak{a}dv_{\mathcal{A},\mathcal{G}_Q}$ is trivial compared to all polynomial-time quantum attackers $\mathcal{A}$.

6.2.5 Security Model II

All security proofs are conducted in the Quantum Random Oracle Model (QROM). Let us λ_s denote the security parameter.

6.2.5.1 Protocol Instances and Session Identifiers Each party $P \in \{V_i, U_i, R_i\}$ may execute multiple concurrent protocol instances. The j-th instance of party P is denoted by Π_P^j. Each instance maintains the following state variables:

- sid: Session identifier
- pid: Peer identifier
- sk: Session key
- $\mathsf{acc} \in \{0, 1\}$: Acceptance flag

Session Identifier: We define the session identifier as:

$$\mathsf{sid} = (\mathsf{role}, ID_P, ID_Q, N_p, N_Q, \tau_P, \tau_Q, \mathsf{transcript}),$$

where τ_P, τ_Q are timestamps, N_P, N_Q are nonce, and $\mathsf{transcript}$ denotes exchanged authenticated values.

6.2.5.2 Partner Function: Two instances Π_P^j and Π_Q^k are said to be partners if:

i. $\mathsf{sid}_P^j = \mathsf{sid}_Q^k$,

ii. $\mathsf{pid}_P^j = Q$,

iii. $\mathsf{pid}_Q^k = P$,

iv. $\mathsf{acc}_P^j = \mathsf{acc}_Q^k = 1$.

If such an instance exists, we define: $\mathsf{Partner}(\Pi_P^j) = \Pi_Q^k$ otherwise,$\mathsf{Partner}(\Pi_P^j) = \perp$.

6.2.5.3 Freshness Definition: An instance Π_P^j is considered *fresh* if:

i. $\mathcal{O}_{\text{Reveal}}(\Pi_P^j)$ has not been issued;

ii. Neither P nor its partner has been corrupted before session completion;

iii. Long-term keys of both participants are not revealed together with session state;

iv. Π_P^j has accepted.

6.2.5.4 Oracle Queries: The adversary $\mathcal{A}$ is a quantum polynomial-time (QPT) algorithm with access to the following oracles:

1. **Execute Oracle:** $\mathcal{O}_{\text{Execute}}(P, Q)$, which returns the transcript of an honest execution between P and Q.
2. **Send Oracle:**defined as $\mathcal{O}_{\text{Send}}(\Pi_P^j, m)$, which delivers a message m to an instance Π_P^j and returns its response.
3. **Reveal Oracle:** defined as $\mathcal{O}_{\text{Reveal}}(\Pi_P^j)$the function that returns the session key sk_P^j.
4. **Corrupt Oracle:** defined as: $\mathcal{O}_{\text{Corrupt}}(P)$ Returns the long-term private key of party P. The Registration Authority cannot be corrupted.
5. **Quantum Hash Oracles:** Each hash function H_k is modeled as: $\mathcal{O}_{H_k} : |x, y\rangle \to |x, y \oplus H_k(x)\rangle$. The adversary is issuing superposition queries. Let q_H denote the total number of quantum hash queries.
6. **Signing Oracle:** $\mathcal{O}_{\text{Sign}}(m)$ Returns a valid signature on message m.

6.2.5.5 AKE Security Definitions: The pseudocodes for these security experiments are described in Figure **??**.

AKE Advantage is defined as:

$$\mathfrak{Adv}_{\mathcal{A}}^{\text{AKE}} = \left| \Pr[\mathsf{Exp}_{\mathcal{A}}^{\text{AKE}} = 1] - \frac{1}{2} \right|.$$

6.2.5.6 EUF-CMA Security Definitions: Pseudocode is described by the figure 6.2 and

EUF-CMA Advantage is defined as:

$$\mathfrak{Adv}_{\mathcal{A}}^{\text{EUF}} = \Pr[\mathsf{Exp}_{\mathcal{A}}^{\text{EUF}} = 1].$$

Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathrm{AKE}}(\lambda_s)$

1. Generate system parameters and keys.
2. $\mathcal{A}$ may issue adaptive queries: Execute, Send, Reveal, Corrupt, Hash.
3. $\mathcal{A}$ selects a fresh accepted instance Π_P^j.
4. Challenger samples $b \leftarrow \{0,1\}$.
5. If $b = 1$, return real session key; otherwise return random $R \in \{0,1\}^\lambda$.
6. $\mathcal{A}$ continues queries (respecting freshness).
7. $\mathcal{A}$ outputs guess b'.

Return 1 if $b' = b$, else return 0.

Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathrm{EUF}}(\lambda_s)$

1. Generate public/secret key pair.
2. $\mathcal{A}$ may query $\mathcal{O}_{\mathrm{Sign}}(m_i)$.
3. $\mathcal{A}$ outputs (m^*, σ^*).

Return 1 if:

$$\mathrm{Verify}(m^*, \sigma^*) = 1$$

and m^* was not previously queried. Otherwise, return 0.

Figure 6.2: Pseudocode for EUF-CMA Security

6.2.5.7 QROM Reprogramming Bound If at most r oracle inputs are reprogrammed, the compressed-oracle bound yields:

$$\delta_{\text{prog}}(q_H, r) = O\left(\frac{q_H^2}{2^{\lambda_s}}\right).$$

6.2.6 Security and Privacy Requirements

Whenever an attacker breaks into a vehicle, stealing information, and then propagates erroneous facts about its position or traffic circumstances, it can cause significant problems, such as worsening traffic jams and even car accidents. To solve such issues concerning privacy and security, the following measures must be taken:

i. ***Mutual Authentication***: Mutual authentication is a cryptography technique in which both communication parties, such as vehicles, UAVs, and RSUs, authenticate each other's identities before establishing a connection.

ii. ***Traceability:*** Traceability in secure networks denotes the ability to recognize or track the source of correspondence units (vehicle, UAV, or RSU), generally by an authorized entity RA.

iii. ***Unlinkability***: Unlinkability guarantees that an adversary cannot adequately ascertain whether many elements of interest—such as inquiries, activities, or communications—are interconnected. From the adversary's perspective, the likelihood of associating such elements should not rise following the observation of the system.

iv. ***Conditional Anonymity***: Conditional anonymity permits users to maintain anonymity in typical situations. However, their true identities (e.g., the vehicle's actual identification—prominent registration code) should only be disclosed to the authorized registration authority (RA) under certain circumstances. No external observer can infringe upon vehicle confidentiality throughout interactions.

v. ***Resist Signature Unforgeability***: This property ensures the integrity and authenticity of all signed information. Only a recipient can reliably verify that both the message and its signature originate from the genuine signer, and any attempt to create a fake (forged) signature, even with access to other signatures, will fail.

vi. ***Resist MITM Attacks***: A man-in-the-middle (MITM) attack transpires when the perpetrator secretively intercepts, relays, or modifies communication

between two parties without their awareness. Therefore, by introducing secure protocols, including mutual authentication and encryption, we can identify and resist MITM assaults, resulting in ensuring confidentiality and data integrity.

vii. ***Resist Impersonation Attacks:*** Resistance to impersonation ensures that unauthorized units cannot effectively masquerade as genuine users or devices to obtain inappropriate access or privileges. Mechanisms may encompass robust authentication or public-key cryptography with the aim of mitigating illegal access, diminishing the danger of identity theft, and protecting system integrity.

viii. ***Resist Replay Attacks***: An adversary intercepts authentic communications between two entities and retransmits them to attain illegal consequences or achieve accessibility. By utilizing a timestamp and nonce in the protocol, we can resist such attacks due to the freshness and randomness of such countermeasures, respectively.

ix. ***Stolen Smart Card Attacks***: Security standards for stolen smart card breaches usually mean that authentication protocols must have ways to stop an attacker who has a genuine consumer's smart card from pretending to be that user, getting more credentials, or damaging the network.

x ***Resistance to Quantum Attacks***: Resilience to quantum attacks indicates that the protocol maintains its security even in the presence of adversaries utilizing quantum computers. This often entails the implementation of cryptographic algorithms—such as lattice-based or hash-based—that quantum computers are incapable of efficiently compromising.

6.3 Proposed Scheme

Our proposed scheme consists of five phases: Setup, Registration, Integration of Smart contract, Authentication, and Session Key Phase. Table 6.2 represents the meaning of all the symbols used in the proposed scheme.

6.3.1 Set-up Phase

The main purpose of this phase is to broadcast the system parameter once it receives the input 1^k. The broadcasting and production of system parameters are performed by a TA as follows:

Table 6.2: Notations used in Proposed Protocols

Notations Used	Definitions
X	The modular matrix chosen by TA belongs to $Z_q^{m\times n}$
s	The master secret key of TA belongs to Z_q^m
S	Master public key of TA from $Z_q^{1\times n}$
m, n	Positive integer numbers
q	Large Prime number
$\mathcal{H}_0[..] - \mathcal{H}_6[..]$	Defined hash function by TA.
$x_i, y_i \& z_i$	The secret vector belongs to Z_q^n the vehicle, UAV, and RSU, respectively.
$Vid_i, Uid_i \& Rid_i$	Real identity of Vehicle V_i, UAV U_i, and RSU,R_i respectively, from $\{0,1\}^*$
$VPw_i, UPw_i \& RPw_i$	Passwords chosen by Vehicle, UAV, and RSU, respectively, from Z_q^*
k	Random vector selected Z_q^n by RA.
t	Ephemeral Key selected by RA from Z_q^n.
h	Secret vector of RA from Z_q^n.
$Et_v, Et_u \& Et_r$	RA set the Expiry time for requesting units $V_i, U_i \& R_i$ respectively.
$UID_v, UID_u \& UID_r$	Unique identity of $V_i, U_i \& R_i$ generated $w.r.t.$ their corresponding expiry time set by RA.
$EID_v, EID_u \& EID_r$	Encrypted identity of $V_i, U_i \& R_i$ respectively.
$E_{\mathcal{H}(k)}[....]$	Encrypt the information using hashed value k.
$D_{\mathcal{H}(k)}[....]$	Decryption using hashed value k.

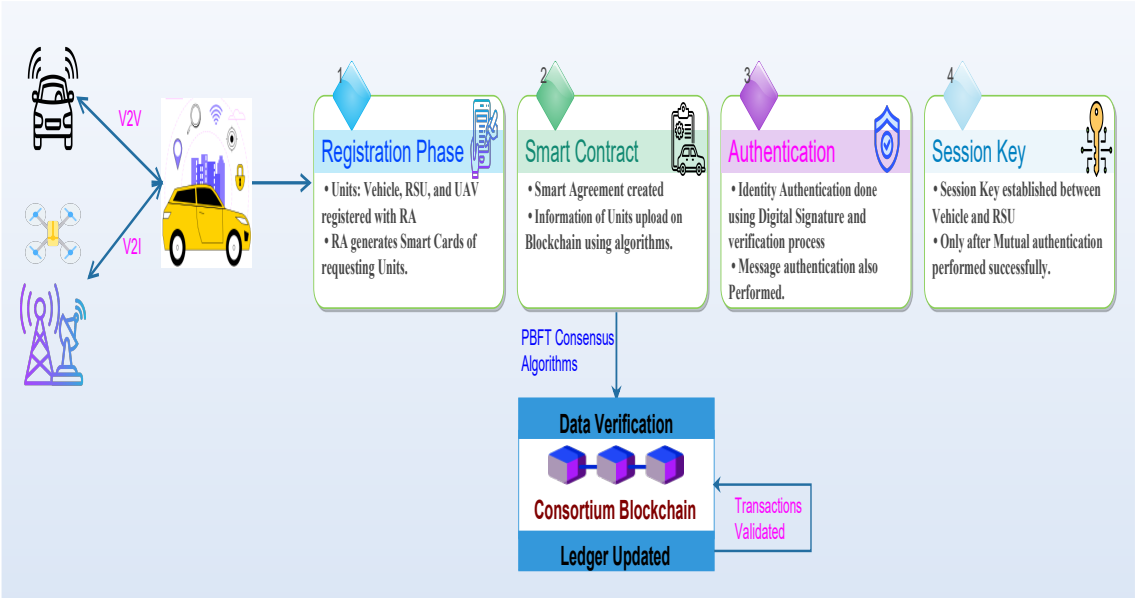


Figure 6.3: Flowchart of Proposed Blockchain-Enabled ITS Model Process.

1. First, TA chooses three prime numbers: m, n, and q
2. Next, select a Matrix $X \in Z_q^{m \times n}$
3. Choose its master secret key $s \in Z_q^m$
4. Generate master public key $S = s^T X$
5. Define a hash function $H[....]$ defined as
   $\mathcal{H}_0[\{0,1\}^l, Z_q^n] \rightarrow Z_q^n$
   $\mathcal{H}_1[Z_q^*, Z_q^n] \rightarrow Z_q^n$
   $\mathcal{H}_2[Z_q^n + Z_q^n + Z_q^n] \rightarrow \{0,1\}^l$
   $\mathcal{H}_3[\{0,1\}^l, Z_q^n, Z_q^*] \rightarrow Z_q^*$
   $\mathcal{H}_4[Z_q^{m \times n}, Z_q^*] \rightarrow Z_q^*$
   $\mathcal{H}_5[\{0,1\}^l, Z_q^{m \times n}, Z_q^{m \times n}] \rightarrow Z_q^*$
   $\mathcal{H}_6[Z_q^{m \times n}, Z_q^{m \times n}] \rightarrow Z_q^*$
   $\mathcal{H}[.....] \rightarrow Z_q^*$

6.3.2 Registration Phase

Figure 6.3 is used to represent and explain the flow of all involved phases in our proposed scheme. We can understand the role of blockchain technology after the registration phase and integration of other phases with blockchain.

All the communication nodes belonging to the ITS infrastructure must register themselves with RA as follows:

A. **RSU Registration**
   RSU side:

   i. First, RSU R_i, selects its identity $Rid_i \in \{0,1\}^l$ and password $RPw_i \in Z_q^*$.

   ii. Next, randomly select a secret vector: $z_i \in Z_q^m$.

   iii. Apply the hash function before requesting to hide the real identity from others using its secret vector z_i, such as $HRid_i = \mathcal{H}_0[Rid_i, z_i]$
   $HRPw_i = \mathcal{H}_1[RPw_i, z_i]$.

   iv. Forward values: $\{Rid_i, HRid_i, HRPw_i\}$ towards the RA via a secure channel to make a registration request (Note: if Rid_i is only sent to support the traceability feature by RA in case of any dispute occurrence by RSU).

RSU R_i	**RA**	**Vehicle** V_i	**UAV** U_i
Computes $HRid_i = \mathcal{H}_0[Rid_i, z_i]$ and $HRPw_i = \mathcal{H}_1[RPw_i, z_i]$		Computes $HVid_i = \mathcal{H}_0[Vid_i, x_i]$ and $HVPw_i = \mathcal{H}_1[VPw_i, x_i]$	Computes $HUid_i = \mathcal{H}_0[Uid_i, y_i]$ and $HUPw_i = \mathcal{H}_1[UPw_i, y_i]$
$\langle HRid_i, HRPw_i, Rid_i \rangle$ To RA		$\langle HVid_i, HVPw_i, Vid_i \rangle$to RA	$\langle HUid_i, HUPw_i, Uid_i \rangle$ to RA
	For RSU Choose a random vector $k \in Z_q^n$ Select a secret vector $h \in Z_q^n$ Computes $R_1 = \mathcal{H}_2[HRid_i + k + h] \oplus Rid_i$ $R_2 = \mathcal{H}_2[HRPw_i + HRid_i + S^T] \oplus R_1$ Set an expiry time Et_r Solicit Algorithm 2 Uploads tuples in database: (R_2, k, Et_r).		
$SC_{R_i} : \langle R1, k \rangle$ to RSU R_i			
Calculates $R_2 = \mathcal{H}_2[HRPw_i + HRid_i + S^T] \oplus R_1$ then Invokes Algorithm 3 with R_2 Key Validation is successfully if Query return: $\langle R_2, k, Et_r \rangle$ Now: RSU R_i stores the secret key: R_1 and computes: $R_3 = z_i \cdot S$ Computes Unique id: $UID_r = \mathcal{H}_3[R_1 \| z_i \| Et_r]$ Call update Algorithm 2 with $\langle R_2, k, Et_r, R_3, UID_r \rangle$ Otherwise, requests to the RA to repeat registration.	**For UAV** Computes $U_1 = \mathcal{H}_2[HUid_i + k + h] \oplus Uid_i$ $U_2 = \mathcal{H}_2[HUPw_i + HUid_i + S^T] \oplus U_1$ Set an expiry time Et_u Solicit Algorithm 2 Uploads tuples in database: (U_2, k, Et_u).		
		$SC_{U_i} : \langle U_1 \rangle$ Forwards from RA to UAV U_i	
	For Vehicle:- Select an ephemeral key $t \in Z_q^n$ Computes $P_0 = \mathcal{H}_0[Vid_i, t]$ $P_1 = \mathcal{H}_2[HVid_i + k + h] \oplus Vid_i$ $P_2 = \mathcal{H}_2[HVPw_i + HVid_i + S^T] \oplus P_1$ $Z_1 = E_{H(k)}[P_0 + t]$ Set an expiry time Et_v Solicit Algorithm 2 Uploads tuples in database: (P_2, k, Et_v) to smart card.		First, Calculates $U_2 = \mathcal{H}_2[HUPw_i + HUid_i + S^T] \oplus U_1$ then Invokes Algorithm 3 using U_2 Key Validated if received values: $\langle U_2, k, Et_u \rangle$ Now, UAV U_i stores the secret key: U_1 and computes: $U_3 = y_i S$ Computes Unique id $UID_u = \mathcal{H}_3[U_1 \| y_i \| Et_u]$ Call update Algorithm 2 with $\langle U_2, k, Et_u, U_3, UID_u \rangle$ Otherwise, requests to the RA to repeat registration.
$\langle Z_1, t, k \rangle$ to RSU R_i		$SC_{V_i} : \langle P_1, t \rangle$ to Vehicle V_i	
RSU stored Z_1, t, k in its database corresponding to vehicle V_i.		Calculates $P_2 = \mathcal{H}_2[HVPw_i + HVid_i + S^T] \oplus P_1$ Call the Algorithm 3 with computed P_2 Validates the received key, if vehicle received values $\langle P_2, k, Et_v \rangle$ if verification done successfully: RSU R_i stores the secret key P_1 and then computes: $P_3 = x_i S$ Computes Unique id $UID_v = \mathcal{H}_3[P_1 \| x_i \| Et_v]$ Call update Algorithm 2 with $\langle P_2, k, Et_v, P_3, UID_v \rangle$ Otherwise, requests to the RA to repeat registration.	

Figure 6.4: Registration Process for RSU, Vehicle, and UAV

RA side:

After receiving the request message from RSU, initially RA verifies the database to check the database. If already registered, then the response as per stored values is the valid expiry time. In case of a new request, it performs the following steps:

i. RA calculates two parameters R_1 and R_2 such as: $R_1 = \mathcal{H}_2[HRid_i + k + h] \oplus Rid_i$
$R_2 = \mathcal{H}_2[HRPw_i + HRid_i + S^T] \oplus R_1$. Where, k is randomly chosen vector, define as: $k \in Z_q^n$ and h chosen as secret vector of RA, define as: $h \in Z_q^n$.

ii. Next RA set an expiry time for requesting RSU R_i as Et_r and then RA generates a smart card: $SC_{R_i} =< R_2, k, Et_r >$ and stored at blockchain (own database as RA acts as node in blockchain) through smart contract then forwards $SC_{R_i} =< R_1 >$ towards the requested R_i via a secure channel.

RSUs side: R_i updates its smart card after verifying the key validation received from RA.

i. First R_i computes: $R_2 == \mathcal{H}_2[HRPw_i + HRid_i + S^T] \oplus R_1$

ii. Now, check whether the computed R_2 value matches the R_2 stored in the blockchain database via RA. For this R_i call the $\mathtt{Q}uery[R_2]$ if value matched, then R_i received tuple $< R_2, k, Et_r >$. Otherwise, RSU sends a registration message again if it fails to validate.

iii. Further, R_i updates the smart card it received by adding two more parameters, such $R_3 = z_i S$ as its public key, where $R_3 \in Z_q^{m \times n}$, and $UID_r = \mathcal{H}_3[R_1||z_i||Et_r]$
Hence, RSU updates smart cards as they are $SC_{R_i} =< R_1, R_2, k, Et_r, R_3, UID_r >$ on the database. As a result, the registration process was done successfully for the RSU R_i.

B. **Vehicle Registration**

Vehicle side:

i. Request vehicle V_i, select its identity $Vid_i \in \{0,1\}^l$ and password $VPw_i \in Z_q^*$.

ii. Next, randomly selects a secret vector: $x_i \in Z_q^m$.

iii. Apply the hash function over the identity and password before requesting as follows:
$HVid_i = \mathcal{H}_0[Vid_i||x_i]$ and $HVPw_i = \mathcal{H}_1[VPw_i||x_i]$.

iv. Now, transmit the calculated hashed value:
$\{Vid_i, HVid_i, HVPw_i\}$ towards the RA via a secure channel. { Vid_i is only shared for traceability feature support by RA in case of dispute}

RA side:

After receiving the request message from V_i, RA initially checks its database to determine whether it is already registered or if it is a new request. If you are already registered, respond if the expiry time is valid. In case of a new request, it performs the following steps:

i. RA computes the following parameters using key h and k:
$P_0 = \mathcal{H}_0[Vid_i, t]$
$P_1 = \mathcal{H}_2[HVid_i + k + h] \oplus Vid_i$
$P_2 = \mathcal{H}_2[HVPw_i + HVid_i + S^T] \oplus P_1$.
$Z_1 = E_{H(k)}[P_0 + t]$
Where Z_1 is computed by an encryption operation over the values P_0, t using the hashed value k. where $H[k] \in Z_q^*$ is used as the encrypted key for encoding.

ii. Next, RA updates its database by using smart contract `update[]` algorithm with arguments as: $< P_2, k, Et_v >$. Where, RA sets Et_v as an expiry time for V_i, and t is chosen as a random ephemeral key: $t \in Z_q^n$.

iii. At last, RA forward $SC_{V_i} =< P_1, t >$ towards the V_i as well as forwards tuple $< Z_1, t, k >$ towards the registered RSU via a secure channel only{RSU use this tuple to verify the requesting vehicles by calling $\mathtt{Q}uery$}.

Vehicle side:
V_i updates its smart card after verifying the key validation received from RA.

i. First V_i computes: $P_2 == \mathcal{H}2[HVPw_i + HVid_i + S^T] \oplus P_1$

ii. Now, to check the validity of received values from RA, V_i call the $\mathtt{Q}uery[P_2]$. In case of failure, RA returns an invalid message. Thus, it V_i makes another registration request. Otherwise, if the key matches, and V_i receives a tuple $< P_2, k, Et_v >$.

iii. Further, vehicle V_i updates its received smart card by adding two more parameters as follows: $V3 = x_i S$ as its public key $\in Z_q^{m \times n}$.
$UID_v = \mathcal{H}3[P_1 || x_i || Et_v]$
Now, vehicle update smart card : $SC_{V_i} =< P_1, P_2, t, k, Et_v, P_3, UID_v >$ on database. As a result, the registration process is done successfully for the requesting vehicle V_i.

Figure 6.4 illustrates all the steps involved in the registration phase for all communicating units (Vehicle, UAV, and RSU) in our proposed scheme.

C. **UAV Registration**
UAV side:

i. First, UAV U_i, selects its identity $Uid_i \in \{0,1\}^l$ and password: $UPw_i \in Z_q^*$.

ii. Next, randomly selects a secret vector: $y_i \in Z_q^m$.

iii. Apply the hash function over the identity and password before requesting as follows:
$HUid_i = \mathcal{H}_0[Uid_i||y_i]$
$HUPw_i = \mathcal{H}_1[UPw_i||y_i]$.

iv. Now, forward the $\{Uid_i, HUid_i, HUPw_i\}$ towards the RA via a secure channel.

RA side:
After receiving the request message from the UAV, RA initially verifies the database to check for its existence. If already registered, then respond as per stored values after verifying the expiry time. In case of a new request, it performs the following steps:

i. RA computes the following parameters, such as: $U_1 = \mathcal{H}2[HUid_i + k + h] \oplus Uid_i$
$U_2 = \mathcal{H}2[HUPw_i + HUid_i + S^T] \oplus U_1$.

ii. RA sets Et_u as an expiry time for U_i and generates a smart card $SC_{U_i} =< U_2, k, Et_u >$ and stores it in RA's database. Then, share the values $SC_{U_i} =< U_1 >$ with the connected UAV via a secure channel.

UAV side: U_i updates its smart card after verifying the key validation received from the RA.

i. First U_i computes: $U_2 == \mathcal{H}2[HUPw_i + HUid_i + S^T] \oplus U_1$

ii. Now, to check whether the computed U_2 value matches with the one U_2 stored in the blockchain database via RA. For this U_i call the $\texttt{Q}uery[U_2]$, if value matched, then U_i the received tuple $< U_2, k, Et_u >$. Otherwise, RA returns an invalid message; therefore, UAV makes a new request to RA. [iii.] Further, it U_i updates its received smart card by adding two more parameters: $U_3 = y_i S \in Z_q^{m \times n}$ as its public key.
$UID_u = \mathcal{H}3[U_1||y_i||Et_u]$
Now, the UAV updates the smart card as: $SC_{U_i} =< U_1, U_2, t, k, Et_u, U_3, UID_u >$ on the database. As a result, the registration process is done successfully for the request of U_i.

Algorithm 1 Initialize EIL

```
contract EIL {
    address owner;
    struct EI {                          ▷ Initialize the structure of components in EIL
        byte16 Para2;                    ▷ Para2 ∈ P_2, R_2, U_2
        uint128[2] k;
        DataTime Et;
        uint128[2] t;
        uint128[2] Z_1; }                ▷ RA define key t and Z_1 only for Vehicle
    EI[] public EIL;
    constructor EIL(){                   ▷ Constructor runs itself once the SC is implemented.
        owner=msg.sender;
        length = 0;
        return 1; }
}
```

Algorithm 2 Update EIL

```
function updateEIL(PrevPara2, Para2, k, Et) {
    if owner ≠ msg.sender then
    return 0;
    else {
        if Exist(EI[i].Para2 == PrevPara2) then {   ▷ If there exists a tuple ¡Para,.¿ in EIL, then update it; else, add a new tuple for it
            EI[i].Para2 = Para2;
            EI[i].k = k;
            EI[i].Et = Et;
            EI[i].Para3 = Para3;
            EI[i].TID = TID;
            return 1; }
        else {
            length++;
            EI[length].Para2 = Para2;
            EI[i].k = k;
            EI[length].Et = Et;
        return 1; }
}
```

Algorithm 3 Query1 EIL

```
function query1(Para2) {
    if Exist(EI[i].Para2 == Para2) then   ▷ Vehicle V_i, UAV U_i and RSU R_i executes retrieve the public key information
        return EI;
    else;
        return 0;
}
```

Algorithm 4 Query2 EIL

function query2($UID_i, Para1$) {
  if Exist(EI[i].$Para1$ == $Para1$) then if Exist(EI[i].UID_i == UID_i) then ▷ V_i,U_i andR_i executes to retrieve the public key information of respective $UID_i \in EID_u, EID_r, EID_u$
    return $Para3$; ▷ $Para3 \in P_3, R_3, U_3$
  else;
    return 0;
}

Algorithm 5 Query3 EIL

function query3($Para1, Z_1$) { ▷ $Para1 \in R_1$
  if Exist(EI[i].$Para1$ == $Para1$) then if Exist(EI[i].$Z1$ == $Z1$) then ▷ RSUR_i executes to retrieve the information of V_i corresponding to Z_1 which is shared by RA
    k;
    t;
    $Z1$;
    return 1;
  else;
    return 0;
}

Algorithm 6 Revoke EIL

function revokeEIL($Para2$){
  if owner ≠ msg.sender then ▷ RA execute to revoke the Vehicle, UAV & RSU
    return 0;
  else {
    if Exist(EI[i].$Para2$ == $Para2$) then { ▷ If there exists a tuple ¡Para,.¿ in EIL, then delete it.
      Release(EI[i]);
      for (; i¡length;i++)
        EI[i] = EI[i+1];
      length−−;
      return 1; }
  else
    return 0; }
}

6.3.3 The Integration of Smart Contract

The Essential Information List (EIL) is administered by the SC. The system framework will quickly monitor and revoke employing algorithms that are encompassed

in the SC's, avoiding asynchronous challenges, while additionally offering a level of conditional secrecy for authentication of information among communication professionals. The pseudo-code for initializing, updating, querying, and revoking EIT is presented in Algorithms 1 to 6

6.3.4 Authentication Phase

For the V2I communication, let V_i seek to obtain traffic information from the RSU, and this request is forwarded via UAV to facilitate efficient and fast communication in our suggested method. Before transmitting a message to the RSU, the UAV must check the integrity and validity of the requested V_i to avert the manipulation of traffic information by any adversary. R_i will respond with useful details only after verifying the legitimacy of the requesting V_i and U_i. R_i additionally verifies the integrity of the received message to ensure it has not been corrupted during transmission across the public channel. By the figure 6.5, we illustrate the steps involved in the authentication process for stage I.

The validity and integrity of the answer from the RSU unit are verified first by UAV and subsequently by the vehicle. In this manner, they mutually validate one another and guarantee that the information transmitted remains untainted by any third party or impostor. The steps involved in the authentication phase for Stage II are represented in Figure 6.6. We employ the digital signature to authenticate the legitimacy of both requesting and responding entities. A one-way hash function and a nonce value are used to ensure the integrity of information. The following steps outline the procedures for signature generation and verification conducted during the first half (communication from the vehicle to the RSU via UAV) and the second half (communication from the RSU to the vehicle via UAV).

Stage I: Request Message From V_i to R_i via U_i

A. **Vehicle Side**

- *Signature Generation:*

  V_i produces the signature and message digest to guarantee the authenticity and integrity of the message. The subsequent steps are as follows:

  i. Initially, V_i chooses at random a Nonce $N_v \in Z_q^{m\times n}$, a Timestamp T_v, and a random secret vector $v \in Z_q^m$.

  ii. Subsequent V_i calculates the ensuing variables: $\delta_v = H_4[P_3||T_v]+\prod_{j=1}^{n}[\theta_v]_j$
  $\theta_v = vS$
  $\omega_v = v + \delta_v x_i$
  $Dig1 = H[m_i||N_v + \theta_v]$

$\lambda_v = N_v + \theta_v$

$EID_v = E_{H(P_0)}[UID_v]$, Call the `Update[]` algorithm to update database with encryptedid EID_v corresponding P_1.

{ Remarks: $Dig1$ used as message digest value used to validate the integrity of message.

V_i can computes P_0 value as query return k and RA already shared t value during registration phase.}

iii. Vehicle V_i transmits tuples towards U_i as: $< m_i, \sigma_v, Dig1, \lambda_v, EID_v, Z_1 >$, with T_v its timestamp where the signature pair is represented as: $\sigma_v =< \theta_v, \omega_v >$.

B. **At UAV Side**

- *Signature Verification:*

i. Initially, the UAV has to confirm the legitimacy of the received timestamp T_v as follows: Verify if $T_v^{'} - T_v \leq \Delta T$. Where $T_v^{'}$ is the timestamp at which the UAV received the message. In the true situation, the UAV calls $\mathtt{Q}uery[EID_v, U_1]$. It returns P_3 in case of match; else it returns false. After that, U_i executes the following step for authenticating the requester unit V_i's signature:

   a. Now, UAV computes two parameters: $a_i = H[P_3||T_v]$ and $b_i = a_i + \prod_{j=1}^{n}[\theta_v]_j$

   b. Check the equation $\theta_v == \omega_v S - b_i P_3$ hold or not? if true the follow the next step; else, reject the request.

ii. UAV starts to verify the integrity by computing nonce: $N_v^* = \lambda_v - \theta_v$. To, check $Dig1 == H[m_i||N_v^*||\theta_i]$ or not ? Where True response ensures that the received message is not tampered with, otherwise reject the request.

- *Signature Generation:*

iii. UAV produces its signature and message digest as follows:

$\delta_u = H[U_3||T_u] + \prod_{j=1}^{n}[\theta_u]_j$

$\theta_u = uS$

$\omega_u = u + \delta_u y_i$

$Dig2 = H[m_i||N_v^* + \theta_u]$

$\lambda_u = N_v^* + \theta_u$

$EID_u = E_{H(k)}[UID_u]$, Call the `Update[]` algorithm to update database with encryptedid EID_u corresponding U_1.

iv. Now, UAV forwards tuples towards R_i: $< m_i, \sigma_u, Dig2, EID_u, EID_v, Z_1, \lambda_u >$ along with timestamp $T_v, T_u,$ where signature pair is represented as: $\sigma_u =< \theta_u, \omega_u >$.

Figure 6.5: Steps involved in Authentication Process- Stage I.

C. **At RSU Side**

• *Signature Verification:*

Initially, RSU verified the genuineness of the incoming timestamps T_v and T_u. In the occurrence of a true state, the RSU initiates verification of UAV U_i and vehicle V_i as follows: Else, reject the request.

i. *For vehicle V_i*: First of all, RSU calls algo $\mathtt{Q}uery[R_1, Z_1]$. This call returns the tuple $< Z_1, t, k >$ only when the received Z_1 value is matched in the database corresponding to R_1, which proves the authenticity of V_i;

Vehicle V_i	**UAV** U_i	**RSU** R_i
		Only After authentication verification done for Vehicle V_i and UAV U_i, RSU Generates it signature: Selects a nonce: $N_r \in Z_q^{m\times n}$, Timestamp T_r. Secret a random vector: $r \in Z_q^m$. Computes the following variables: $\delta_r = H[R_3 \| T_r] + \prod_{j=1}^{n}[\theta_r]_j$ $\theta_r = rS$ $\omega_r = r + \delta_r z_i$ $Dig3 = H[m_j \| N_r + \theta_r]$ $\lambda_r = N_r + \theta_r$ $EID_r = E_{h(P_0)}[UID_r]$ $K_1 = \mathcal{H}_6[UID_v, UID_r]$
		$\langle m_j, Dig3, \sigma_r, T_r, \lambda_r, EID_r, K_1\rangle$ to UAV U_i
	UAV call Algorithm 4 with received EID_r, U_1. Received public key R_3 if true; else return 0. Computes: $c_i^{'} = H[R_3 \| T_r]$ and $d_i^{'} = c_i^{'} + \prod_{j=1}^{n}[\theta_r]_j$ Check equation $\theta_r == \omega_r S - d_i^{'} R_3$ hold or not? Return true if signature verified; otherwise, return false. Next, computes the nonce: $N_r^* = \lambda_r - \theta_r$. Check equation $Dig3 == \mathcal{H}[m_j \| N_r^* + \theta_u]$ is true or not? Return true if the integrity of the response message is preserved; otherwise, reject the request. Now, compute the few more variables: $Dig4 = \mathcal{H}[m_j \| N_r^* + \theta_u]$ $\lambda_u = N_R^* + \theta_u$ $\sigma_u = \langle \theta_u, \omega_u \rangle$	
	$\langle m_j, \sigma_u, Dig4, EID_u, EID_r, K_1, \lambda_u, T_u, T_r\rangle$ to Vehicle V_i	
Initially, verify the legitimacy of the received timestamps as: $T_r^{'} - T_r \leq \Delta T$, $T_u^{'} - T_u \leq \Delta T$. Now, Calculates its P_0 as: $P_0 = \mathcal{H}_0[Vid_i, t]$. Ascertain the value of $UID_r = D_{H(P_0)}[EID_r]$. computes key $K_2 = \mathcal{H}_6[UID_r, UID_r]$ Verifies if $K_2 == K_1$. If affirmative, the vehicle verified the RSU. Otherwise, failed to verify. Now, Vehicle call Algorithm 4 with $[EID_u, P_1]$. Query retrieves the public key of UAV U_3; else, it returns 0. To verify the UAV's signature V_i calculates: $a_i^{'} = H[U_3 \| T_u]$ and $b_i^{'} = a_i^{'} + \prod_{j=1}^{n}[\theta_u]_j$. To Check: $\theta_u == \omega_u S - b_i^{'} U_3$ Verification completes if the equation returns true; else V_i terminates the communication. At last, calculate the nonce as: $N_r^{**} = \theta_u \oplus \lambda_u$. Verifies equation: $Dig4 == \mathcal{H}[m_j \| N_r^{**} + \theta_u]$. If not, then reject as received message is tampered; else the message is validated.		

Figure 6.6: Steps involved in the Authentication Process of Stage II.

otherwise, returns a message indicating unauthorized units. Next, RSU performs the following steps:

a. RSU can compute P_0 by using the returned value t and k by applying decryption over received Z_1 as:
$P_0 = D_{H(k)}[Z_1] - t$
$== D_{H(k)}[E_{H(k)}[P_0 + t]] - t$
$== P_0$.

b. Now, RSU can compute $UID_v = D_{H(P_0)}[EID_v]$ and store it corresponding Z_1 value.

ii. *For* U_i: RSU call $\mathtt{Query}[EID_u, R_1]$ to obtain the information regarding public value U_3. The following steps are used for signature verification of U_i:

a. Start with computation of: $c_i = H[U_3||T_u]$ and $d_i = c_i + \prod_{j=1}^{n}[\theta_u]_j$
b. Then, check equation $\theta_u == \omega_u S - d_i U_3$ hold or not? If not, then reject. Otherwise, accept the request.

iii. To check the integrity of the message, it computes the nonce as: $N_v^{**} = \lambda_u - \theta_u$. Then, verify the equation $Dig2 == H[m_i||N_v^{**} + \theta_u]$ is true or not? If true, then it confirms that the received message has not been tampered with on the way; reject the request.

Stage II. Response From RSU R_i To V_i Via U_i

A. **At RSU Side**

•*Signature Generation:*

Once the authentication verification and validation of the message is performed successfully at the ends of RSU R_i, it sends the response to the vehicle via UAV. For this purpose, RSU performs the following steps:

i. RSU selects its nonce $N_r \in Z_q^{m\times n}$, Timestamp T_r, and secret a random vector $r \in Z_q^m$.

ii. Next, RSU computes the following variables:
$\delta_r = H[R_3||T_r] + \prod_{j=1}^{n}[\theta_r]_j$
$\theta_r = rS$
$\omega_r = r + \delta_r z_i$
$Dig3 = H[m_j||N_r + \theta_r]$
$\lambda_r = N_r + \theta_r$
$EID_r = E_{h(P_0)}[UID_r]$
$K_1 = \mathcal{H}_6[UID_v, UID_r]$

iii. Then, forwards $< m_j, Dig3, \sigma_r, T_r, \lambda_r, EID_r, K_1 >$ towards the UAV where $\sigma_r =< \theta_r, \omega_r >$.

B. **UAV Side**

•Signature Verification

i. First, UAV verifies the legitimacy of the received timestamp T_r. Then call the $\mathtt{Q}uery[EID_r, U_1]$, which delivers the value of R_3 in case of true, else rejected.

ii. Next, it computes two parameters as: $c_i^{'} = H[R_3||T_r]$and $d_i^{'} = c_i^{'} + \prod_{j=1}^{n}[\theta_r]_j$.

iii. Now, start to verify the signature by the equation: $\theta_r == \omega_r S - d_i^{'} R_3$ or not?

iv. After signature verification is successful, to ensure the integrity of the response message, it calculates the nonce as: $N_r^* = \lambda_r - \theta_r$

v. To check is equation $Dig3 == H[m_j||N_r^* + \theta_r]$ hold or not ? If true, then it ensured that the received message it not tampered else reject. So, after successfully verifying that the response is coming from an authorized RSU and the response is also not tampered.

iv. UAV sends the tuple: $< m_j, Dig4, \sigma_u, T_u, T_r, \lambda_u, EID_r, K_1, EID_u >$ towards the requesting vehicle. Where signature is represented as: $\sigma_u =< \theta_u, \omega_u >$. and δ_u, ω_u and θ_u are already computed during Part A at UAV ends. Others define as:
$Dig4 = H[m_j||N_r^* + \theta_u]$
$\lambda_u = N_r^* + \theta_u$

C. **Vehicle Side**

•Signature Verification

i. Initially, verify the legitimacy of the received timestamps T_v, and T_u by ensuring that: $T_r^{'} - T_r \leq \Delta T$, $T_u^{'} - T_u \leq \Delta T$. If false the reject; else, proceed with the subsequent operation.

ii. The vehicle subsequently calculates its P_0 value, having received the key t, such that $P_0 = \mathcal{H}_0[Vid_i, t]$. This is used to ascertain the value $UID_r = D_{h(P_0)}[EID_r]$ by decoding the received EID_r.

iii. Subsequently, V_i computes key K_2 to cross check whether $K_2 == K_1$, as: $K_2 = \mathcal{H}_6[UID_r, UID_r]$. If affirmative, it indicates that the vehicle completes the authentication process for the RSU. Failed to authenticate the responding unit RSU.

iv. To continue authenticating the UAV, the vehicle call $\mathtt{Q}uery[EID_u, P_1]$. This query retrieves the public key of UAV U_3; else, it returns 0.

v. Next, To verify the UAV's signature V_i calculates: $a_i^{'} = H[U_3||T_u]$ and $b_i^{'} = a_i^{'} + \prod_{j=1}^{n}[\theta_u]_j$. TO, check whether $\theta_u == \omega_u S - b_i^{'} U_3$? or not? Verification completes if the equation returns true; else V_i terminates the communication.

vi. At last, calculates the nonce as: $N_r^{**} = \theta_u \oplus \lambda_u$ to verifies integrity of message with the equation: $Dig4 == H[m_j||N_r^{**}||\theta_u]$ is true or not? If not, then reject as the received message is tampered; else, the message is validated.

The vehicle ultimately obtained the needed information securely, ensuring the integrity and authentication processes to prevent any tampering, alteration, impersonation, or man-in-the-middle attacks, and many more.

6.3.5 Session Key

Vehicle V_i and RSU R_i initiate the computation of the session key exclusively after the successful completion of the Authentication phase. This signifies that the requesting and responding units mutually authenticate each other, along with the validation of information exchanges. To confirm the confidentiality of the generated $\mathbf{SK} = \mathcal{H}[UID_v||UID_r||N_v||N_r]$, we elucidate how all parameters UID_v, UID_r, N_v, and N_r effectively maintain their respective secrecy.
Commence with $UID_v = \mathcal{H}_3[P_1||x_i||Et_v]$, which is contingent upon the confidentiality of the secret key of Vehicle x_i, P_1, and Et_v, where P_1 is exclusively computed by the trusted authority RA, Thu, UID_v indirectly depends on k, and h as: $P_1 = \mathcal{H}_2[HVid_i||k||h]$. Here, RA also establishes the expiry time Et_v for V_i during the registration process. Where h is the secret key of RA and k is a randomly chosen matrix defined as $\in Z_q^n$. Likewise, UID_r is contingent upon its own secret key z_i, as well as the keys selected by RA k, h, and Et_r. As a result, we can say that none other than V_i and R_i can compute the value of UID_v and UID_r, respectively.
Next for the nonce values, as Nonce pair (N_v, N_r) are contingent upon the variable pairs: (θ_u, λ_u) and (θ_r, λ_r). These defined pairs rely on the secret vectors (v, r) of the vehicle and the RSU, respectively. Which concludes that the value of nonce also comes under the hard assumption due to secret vectors v and r. Consequently, the produced session key **SK** maintained its confidentiality by relying on the parameters: UID_v, UID_r, N_v, and N_r.

The session key **SK** produced at both ends is symmetrical and is being used by the vehicle V_i and the RSU R_i to securely transmit traffic and vehicle data. The symmetric key **SK** is utilized to encrypt and decode the data exchanged between the vehicle V_i and the RSU R_i.

6.3.6 Updations and Revokation

There are two possibilities whenever any of the communicating units (vehicle, RSU, or UAV) wants to update their key list. Assume here the vehicle V_i requests to update the keys:

1. *Case 1:* RA first verifies the validation of the existence of the requesting V_i units and checks whether the expiry time set Et_i is up or not. In the true

case, RA updates the key information by calling the respective Algorithm 2.

2. *Case 2:* If the private key information is gained by an intruder, then the unit requests to update its key to prevent further malicious activities. Thus, RA generates a new parameter by calling Algorithm 2.

Similar to the update function, the revoke algorithm is also applicable to the following two cases:

1. *Case 1:* If V_i behavior seems to be suspicious to the RA. Then RA calls Algorithm 6 to erase all the transactions and respective entries performed by such units.

2. *Case 2:* Next, RA can call the revocation Algorithm 6 whenever any communicating units want to leave the environment.

Remarks: When a vehicle V_i is identified as suspicious, then, along with performing case 1, RA also maintains a revocation list (RL). RL contains the identity of such a suspicious node. Therefore, when RA receives a request for new registration, it checks the identity before starting any step. Reject the request if the requesting identity belongs to RL; otherwise, start the registration phase.

6.4 Security Analysis and Proofs

6.4.1 Correctness: Session Key

We construct the session key as follows: $\mathbf{SK} = H[UID_v||UID_r||N_v||N_r]$. To communicate securely by using this key, the same session key **SK** value must be obtained by the vehicle V_i and the RSU R_i. For this purpose, both ends must obtain all parameters involved in SK. Thus, the computed **SK** value must be equivalent at both ends.

- ***Verify the Unique id:*** Start with Unique ID UID_v and UID_r: For UID_v: We can see in Stage II, RSU calls the Algorithm 5. This call returns the tuple $< Z_1, t, k >$ only when the received Z_1 value is matched in the database corresponding to R_1. Next, RSU can compute P_0 by applying decryption over received Z_1 as follows:$P_0 = D_{H(k)}[Z_1] - t$. Now, RSU can finally obtain $UID_v = D_{H(P_0)}[EID_v]$ and store its corresponding Z_1 value. Similarly, for UID_r: at the end of the vehicle, it V_i decodes the received EID_r to ascertain the value: $UID_r = D_{h(P_0)}[EID_r]$ after calculating its own P_0 value by using

the already received key t at registration time. Which is also authenticated by equation $K_2 == K_1$, where $K1 = \mathcal{H}_6[UID_v, UID_r] = K_2$.

- ***For Nonce Value:*** Both units V_i and R_i computed same nonce value. When the nonce value of the vehicle is N_v transferred to RSU through UAV:
  Then, RSU computes the $N_v^{**} == \lambda_u - \theta_u == N_v^* == \lambda_v - \theta_v == N_v$ same value of nonce transferred by V_i.
  Similarly, when the vehicle receives a response from R_i via UAV: Then, Vehicle computes nonce values as follows:$N_r^{**} == \lambda_u - \theta_u == N_r^* == \lambda_r - \theta_r == N_r$.

Hence, in our proposed protocol, both units V_i, R_i compute the same session key, as both ends obtain the same value of parameters: UID_v, UID_r, N_v, N_r.

6.4.2 Signature verification

At UAV Side

- For Requesting Vehicle V_i as follows: Check equation: $\theta_i == \omega_i S - b_i P_3$ hold or not?
  $== \omega_i S - b_i P_3$
  $== \omega_i S - [a_i + \prod_{j=1}^{n}[\theta_i]_j]P_3$
  $== [v + \delta_i x_i]S - [H[P_3||T_v] + \prod_{j=1}^{n}[\theta_i]_j]P_3$
  $== [v + \delta_i x_i]S - \delta_i P_3$
  $== vS + \delta_i P_3 - \delta_i P_3$
  $== \theta_i$ matched with received θ_i value
  Thus, the vehicle's signature was verified successfully at the end of the UAV.

- For Responding Unit RSU R_i as follows: Check whether $\theta_r == \omega_r S - d_i^{'} R_3$ or not?
  $== \omega_r S - d_i R_3$
  $== \omega_r S - [c_i{'} + \prod_{j=1}^{n}[\theta_r]_j]R_3$
  $== [r + \delta_r z_i]S - [H[R_3||T_r] + \prod_{j=1}^{n}[\theta_r]_j]R_3$
  $== [r + \delta_r z_i]S - \delta_r R_3$
  $== rS + \delta_r R_3 - \delta_r R_3$
  $== \theta_r$ matched with received θ_i value
  Thus, the signature was verified successfully at the end of the UAV for the responding RSU.

At RSU Side

- Signature verification performed by RSU for requesting UAV as follows: Check if the equation $\theta_u == \omega_u S - d_i U_3$ holds or not?
  $== \omega_u S - d_i U_3$
  $== \omega_u S - [c_i + \prod_{j=1}^{n} [\theta_u]_j] U_3$
  $== [u + \delta_u y_i] S - [c_i + \prod_{j=1}^{n} [\theta_u]_j] U_3$
  $== uS + \delta_u U_3 - \delta_u U_3$
  $== \theta_u$ Signature verified Successfully

At Vehicle V_i Side

- Signature verification performed by V_i for UAV as follows: check equation $\theta_u == \omega_u S - b_i' U_3$ hold or not?
  $== \omega_u S - b_i' U_3$
  $== \omega_u S - [a_i' + \prod_{j=1}^{n} [\theta_u]_j] U_3$
  $== [u + \delta_u y_i] S - [a_i' + \prod_{j=1}^{n} [\theta_u]_j] U_3$
  $== uS + \delta_u U_3 - \delta_u U_3$
  $== \theta_u$ Signature verified Successfully

6.4.3 Formal Security Analysis

Formal security analysis has been performed using the QROM model for our proposed QSB-AKA scheme. In which the adversary $\mathcal{A}$ wants to access the information based on the query asked to challenger $\mathcal{C}$. The security analysis process is described as follows, where the proofs use standard QROM techniques:

1. **Measure-and-reprogram** or **compressed-oracle**-style arguments to program the random oracle at challenger-determined points while controlling adversary disturbance. Concretely, if the adversary makes at most Q_H quantum queries, reprogramming probability loss terms are typically $O(Q_H/\kappa)$ where κ is the ROM space size is (informal; we track Q_H explicitly in bounds).

2. **Classical hybrid games** to progressively replace honest computation with simulated values, bounding changes in success probability using the above QROM bounds.

3. **Reduction algorithms** that simulate oracles and, if the adversary succeeds, extract an SIS/ISIS witness from adversary outputs (often following the algebraic structure of the signature: signatures consist of short vectors satisfying an $\mathbf{A}z \equiv \cdot$ relation).

A. **Signature Unforgeability proof (Sign QSB-AKA)**: We give a full game sequence and reduction to SIS or $ISIS$.

Theorem 6.1 (Sign QSB-AKA in QROM)**.** Let $\mathbf{a}dv$ be a QPT adversary making at most Q_H quantum queries to the random-oracles used by the signature algorithm, and at most q_s classical signing queries to honest signers. Then there exists an algorithm Q_{algo} that uses $\mathbf{a}dv$ to solve an instance of the SIS (or $ISIS$, depending on the exact algebraic form of the scheme) problem with success probability

$$\Pr[SIS\text{-Solver Won}] \geq \frac{\Pr[\mathsf{Forge}] - \epsilon_{QROM} - q_s \cdot \delta_{\text{sim}}}{\alpha}$$

where $\Pr[\mathsf{Forge}]$ is $\mathbf{a}dv$'s forging advantage, $\epsilon_{QROM} = O(Q_H/\mathcal{N})$ captures the QROM reprogramming loss (with $\mathcal{N}$ the size of the oracle domain), δ_{sim} is the simulation failure probability for each signing query (negligible under parameter choices), and α is an explicit small factor (e.g., number of target points or algebraic multiplicity) arising from reduction branching. In particular, for reasonable parameters and bounded Q_H, the scheme is Sign QSB-AKA assuming hardness of $SIS/ISIS$ (as stated in the scheme).

Proof (Game sequence and reduction). We present a game sequence: Game 0 is the real attack; Game 1 transforms to a reduction that extracts an SIS witness.

Game 0. Real Sign QSB-AKA game
$\mathcal{C}$ runs honest key generation (including master public matrix $\mathbf{S}$ derived from the RA/TA), gives public parameters to $\mathbf{a}dv$, answers quantum oracle queries to $\mathcal{H}$ honestly, and answers signing queries by computing valid signatures using the honest secret keys. The adversary outputs a purported forgery $(m^\star, \sigma^\star)$. Using scheme notation where a signature has two parts $\sigma = (\theta, \omega)$ produced from a short vector v (e.g. $\theta = vS$, $\omega = v + \lambda_v x$ or the scheme's stated form), we note that a valid signature implies an algebraic short-vector relation.

Game 1 (Lazy-sampling and compressed-oracle bookkeeping).
The challenger $\mathcal{C}$ implements the QROM via lazy sampling /compressed-oracle bookkeeping. This does not change $\mathbf{a}dv$'s success probability. We record all prior oracle queries (quantum states are modeled according to the standard QROM bookkeeping).

Game 2 (Program one hash value).
The challenger picks at random an oracle input point $u^\star$ (an encoding of $(m^\star)$ combined with other labels as per the signature hash used in the scheme) among the implicit domain points that could appear; using the QROM measure-

and-reprogram argument we will reprogram the $\mathcal{H}$ on this point after the adversary's queries with bounded effect: the probability that the adversary notices reprogramming is at most $\epsilon_{QROM} = O(Q_H/\mathcal{N})$ (where $\mathcal{N}$ denotes the oracle domain size). Thus

$$|\Pr[\mathsf{Forge}]_{Game0} - \Pr[\mathsf{Forge}]_{Game2}| \leq \epsilon_{QROM}.$$

The challenger programs the hash at $u^\star$ to an extractable value that will enable the extraction of a short vector witness if $\mathbf{a}dv$ forges at $(m^\star)$.

1. **Simulation of signing queries.** For signing queries on identities other than $HID^\star$, $\mathcal{C}$ simulates signatures using the trapdoor or using the same simulation technique the scheme supports. Each simulation has a probability of failure δ_{sim}, bounded, negligible given parameters; there are q_s such queries, so total failure mass is $q_s \cdot \delta_{\text{sim}}$.
2. **Extraction from a forgery.** Suppose $\mathbf{a}dv$ outputs a forgery $(m^\star, \sigma^\star)$ that verifies under the public key. Because the challenger reprogrammed the hash at the programmed point and because the signature algebra ties hash outputs to short vectors (as in the scheme: $\theta = vS$ and $\omega = v + \ldots$), the challenger can algebraically manipulate the verification equations to obtain a non-zero short vector z such that $\mathbf{A}z \equiv 0 \pmod{q}$ or $\mathbf{A}z \equiv u \pmod{q}$ (i.e., an SIS or $ISIS$ solution) depending on the exact signature construction. The extraction works as follows:
   - Using the forged signature components and the reprogrammed hash value at the chosen point, form the linear equation(s) that the signature implies (the verification relation in the scheme).
   - Rearrangement yields a vector z bound by the scheme's shortness constraints (since $\sigma^\star$ must be a valid short signature), which is a valid $SIS/ISIS$ witness.

   The extraction is exact if
   (i) The adversary's forgery uses the programmed point.
   (ii) The simulation did not abort during signing queries.

   The probability that the forgery uses the programmed point is at least $1/\alpha$ for a small integer α (e.g., if the reduction randomly selects among the oracle inputs corresponding to attackable message/identity pairs; α can be made explicit by enumerating hash-encoding collisions).

Thus, overall, the reduction succeeds with probability at least

$$\frac{\Pr[\mathsf{Forge}] - \epsilon_{QROM} - q_s \cdot \delta_{\text{sim}}}{\alpha}.$$

This completes the reduction: an SIS/ISIS solver is obtained from the adversary's forgery with only the above losses. □

Remark 1. Concrete bounds: the $O(Q_H/\mathcal{N})$ term is the standard QROM reprogramming cost; for a large random-oracle domain (e.g., 256-bit outputs) and bounded Q_H, this is negligible. The exact α and δ_{sim} depend on the scheme's hash encoding of messages and the signing simulation method (both are explicit in the scheme description). See manuscript sections describing signature construction.

B. ***Mutual Authentication Proof***:

Theorem 6.2 (Mutual authentication)**.** Assume the signature scheme used in the protocol is Sign QSB-AKA in the QROM (Theorem 1) and the hash functions are modeled as random oracles. Then no QPT adversary making at most Q_H queries and q_s signing queries can cause an honest party to complete a session believing it is partnered with an honest peer when no matching honest session exists, except with probability negligible in the security parameter (concrete bound is a function of $\Pr[\mathsf{Forge}]$, QROM losses, and the chance of breaking timestamps or forging nonce).

Proof. **[Sketch (full reduction detail follows standard hybrid argument)]**
We proceed by contradiction: suppose the adversary **a***dv* succeeds in impersonating some identity to an honest party (or causes non-partnered accept). Then either:

1. **a***dv* produced a fresh, valid signature on a message binding the identities and nonce/timestamps used in the session — this contradicts Sign QSB-AKA (Theorem 1); or
2. **a***dv* overcame freshness checks by forging valid timestamp/nonce values that cause acceptance without valid signatures — but freshness checks are based on signed values or on hash-derived nonce; forging these reduces to either breaking the signature scheme or finding hash collisions/preimage, both negligible in QROM under our assumptions.

We formalize this via hybrid games:

i. Replace any incoming authenticated message that the adversary sends (which the honest party verifies) with a check that directly queries the signing oracle or hash value. If the adversary $\mathcal{A}$ causes acceptance without previously issued signatures/queries, it implies a forgery or preimage, hence contradicts Sign QSB-AKA or hash security in QROM.

ii. Each hybrid change is bounded by the same QROM reprogramming bounds as in the EUF proof.

Thus, acceptance without a partner implies either a signature forgery (contradiction by Theorem 1) or breaking hash properties (negligible in QROM). Therefore, mutual authentication holds. □

C. ***Session Key Proof:***

We prove that the session key established by matched honest sessions is indistinguishable from random to any QPT adversary with the permitted oracle access, unless the adversary can solve the underlying lattice problem.

Theorem 6.3 (Session-key secrecy (AKA) in QROM)**.** (Under the same assumptions (SIS/ISIS hardness, QROM RO access limited by Q_H, and honest signing Unforgeability), the advantage of any QPT adversary $\mathbf{a}dv$ in distinguishing a real session key (from a fresh partnered session) from a random key is bounded by

$$\mathbf{a}\mathrm{dv}_{AKE}(\mathbf{a}dv) \leq \Pr[\mathsf{Forge}] + \epsilon_{QROM} + \mathrm{negl}(\lambda),$$

i.e., it is upper-bounded by the EUF forgery advantage plus QROM reprogramming losses and negligible terms. Consequently, session-key secrecy reduces to the same lattice assumptions mentioned previously.)

Proof. **[Proof (sketch with hybrid games)]**

Existing standard AKA proofs proceed by progressively replacing protocol flows and the key derivation function (KDF) with simulated values and showing that any distinguishing advantage implies either

(a) signature forgery or

(b) extraction of an algebraic relation giving an SIS/ISIS solution.

Key steps are as follows:

1. **Partnering and freshness handling.** Restrict attention to a test session that is unexposed: its long-term keys are not revealed, nor its session key revealed. Use timestamps/nonces to block replay (manipulated via adversary queries, but those are classical).
2. **Replace the KDF output for the test session** with a genuine random string when answering the Test query; simulate all subsequent honest computations consistently (using programmed $\mathcal{H}$ values where necessary).
3. **Show that changing the KDF output is indistinguishable** except with probability bounded by forging/hash-breaking events: because key material is derived from authenticated values (signatures, signed nonces, ephemeral values that are validated with signatures), the only way $\mathbf{a}dv$ can notice the substitution is by forging a signature or by finding a hash preimage/collision that contradicts the Sign QSB-AKA bound or hash security in QROM.
4. **Use the Sign QSB-AKA reduction** that if $\mathbf{a}dv$ distinguishes the real key from random with noticeable advantage, then we can build an algorithm that either forges a signature (contradicting Theorem 4) or extracts an SIS witness (by the same algebraic extraction used in the signature proof), with probability loss at most $\epsilon_{QROM} + \mathrm{negl}(\lambda)$.

Quantitatively, the distinguishing advantage is at most the sum of:

$$\Pr[\mathsf{Forge}] + \epsilon_{QROM} + (\text{negligible simulator failures}).$$

Thus, session-key secrecy follows. □

D. **EUF-CMA Proof:** We now prove security via game hopping.

Theorem 6.4. Assume ISIS is hard. Then the signature component is EUF-CMA secure in QROM.

Proof. Define games G_0, G_1, G_2, G_3.

Game G_0: Real EUF-CMA experiment.

Game G_1: Replace hash with compressed QRO: $|\Pr[G_1] - \Pr[G_0]| = 0$.

Game G_2: Define bad event: adversary queries programmed input prematurely $\Pr[\mathrm{Bad}] \leq \frac{rq_H}{2^{\lambda_s}}$.

Game G_3: Reprogram r inputs., By QROM bound: $|\Pr[G_3] - \Pr[G_2]| \leq \frac{8q_H^2}{2^{\lambda_s}}$. If forgery occurs: $Az^* \equiv u \pmod{q}$.

Hence, z^* solves ISIS.

Thus:
$\mathfrak{Adv}_{\mathcal{A}}^{\mathrm{EUF}} \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{ISIS}} + \frac{8q_H^2}{2^{\lambda_s}} + \frac{rq_H}{2^{\lambda_s}}$.

□

E. **AKE Proof:**

Theorem 6.5. Assume SIS/ISIS is hard. Then the protocol achieves AKE security.

Proof. Define games $G_0 \rightarrow G_4$. **Game** G_0**:** Real AKE experiment.

Game G_1**:** Replace hash with compressed QRO.

Game G_2**:** Define bad event for session-key input: $\Pr[\mathrm{Bad}] \leq \frac{rq_H}{2^{\lambda_s}}$.

Game G_3**:** Reprogram session-key hash: $|\Pr[G_3] - \Pr[G_2]| \leq \frac{8q_H^2}{2^{\lambda_s}}$.

Game G_4**:** Replace real key with random. If the adversary distinguishes, then either: case 1: Authentication forgery occurs → reduces to EUF case or case 2: Algebraic relation yields short vector → solves SIS/ISIS will occur:

Thus:
$\mathfrak{Adv}_{\mathcal{A}}^{\mathrm{AKE}} \leq 2\mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS/ISIS}} + \frac{8q_H^2}{2^{\lambda_s}} + \frac{rq_H}{2^{\lambda_s}}$.

□

Theorem 6.6. Tightness: The reduction loss consists of:
$O\left(\frac{q_H^2}{2^{\lambda_s}}\right) \quad + \quad O\left(\frac{q_H}{2^{\lambda_s}}\right)$.

For $\lambda_s = 256$ and $q_H \leq 2^{20}$: $\frac{8q_H^2}{2^{\lambda_s}} \approx 2^{-216}$

Thus, the reduction is tight up to polynomial factors in q_H.

6.4.4 Informal Security analysis

1. **Mutual Authentication**: In our proposed scheme, each unit checks the authenticity of the other. In our proposed scheme, when V_i forwards the tuples towards the UAV: $< m_i, Dig1, \sigma_i, T_v, \lambda_i, Z_1, EID_v >$, where the signature is defined as: $\sigma_i =< \theta_i, \omega_i >$. To verify the authenticity of $V_i$the request for the RSU, the steps are as follows:

   i. First, UAV checks the authenticity of V_i by signature verification equations: $\theta_i == \omega_i S - b_i P_3$ holds or not? Before transmitting this request

to RSU
$== \omega_i S - b_i P_3$
$== \omega_i S - [a_i + \prod_{j=1}^{n}[\theta_i]_j]P_3$
$== vS + \delta_i P_3 - \delta_i P_3$
$== \theta_i$ { matched with received θ_i value}
Hence, the authenticity of V_i is verified successfully at the end of UAV.

ii. Next, RSU to verify the authenticity of requesting V_i and UAV U_i who forwards the requests as tuple: $< m_i, Dig2, \sigma_u, \lambda_u, EID_u, EID_v, Z_1 >$ along with timestamp T_v, T_u, where signature represented as :$\sigma_u =< \theta_u, \omega_u >$. (a). *For* V_i: RSU calls algo $\mathtt{Q}uery[R_1, Z_1]$. This call returns the tuple $< Z_1, t, k >$ only when the received Z_1 value is matched in the database corresponding to R_1. Otherwise, return failed to verify the authenticity of the requesting vehicle. (b). *For UAV*: RSU starts to verify the Signature with equation: $\theta_u == \omega_u S - d_i U_3$ hold or not? First, obtain the U_3 by calling $\mathtt{Q}uery[EID_u, R1]$
$== \omega_u S - d_i U_3$
$== \omega_u S - [c_i + \prod_{j=1}^{n}[\theta_u]_j]U_3$
$== \theta_u$ {Signature verified Successfully}

Thus, at the RSU end, authentication processes are done successfully for V_i and UAV
During Stage II: RSU responded with the tuple: $< m_j, Dig3, \sigma_r, T_r, \lambda_r, EID_r, K_1 >$. Where, $\sigma_r =< \theta_r, \omega_r >$.

iii. UAV checks whether the response comes from an authorized unit or not by equation: $\theta_r == \omega_r S - d_i^{'} R_3$ or not?
$== \omega_r S - d_i R_3$
$== \omega_r S - [c_i{}' + \prod_{j=1}^{n}[\theta_r]_j]R_3$
$== rS + \delta_r R_3 - \delta_r R_3$
$== \theta_r$ matched with received θ_r value
Hence, RSU authenticity is being verified successfully at the end of UAV.

iv. When, vehicle received tuples: $< m_j, Dig4, \sigma_u, T_u, T_r\lambda_u, K_1, EID_r >$, where signature represented as: $\sigma_u =< \theta_u, \omega_u >$. V_i starts to verify the authenticity of UAV and RSU units as follows:

   – The vehicle subsequently calculates its P_0 value, having received the key t, such that $P_0 = \mathcal{H}_0[Vid_i, t]$. This is used to decode the received EID_r to ascertain the value of $UID_r = D_{h(P_0)}[EID_r]$. Subsequently, V_i computes key $K_2 = \mathcal{H}_6[UID_r, UID_r]$ and verifies if $K_2 == K_1$. If

affirmative, it indicates that the vehicle completes the authentication process for the RSU. Failed to authenticate the responding unit RSU.

– Now, V_i verify the authenticity of UAV by equation: $\theta_u == \omega_u S - b_i{}' U_3$ hold or not? First, calling $\mathtt{Query}[EID_u, P_1]$ to obtain the public key of UAV $U3$ Then check :
$== \omega_u S - b_i{}' U_3$
$== \omega_u S - [a_i{}' + \prod_{j=1}^{n} [\theta_u]_j] U_3$
$== \theta_u$ Signature verified Successfully

In this way, V_i completes the verification of responding UAV as well as RSU.

2. **Message Authentication**: Our proposed scheme ensures that during the V2I communication process, all the transferred message hold their validity and integrity. When V_i transmits message m_i towards UAV, UAV first verifies its validation by computation of nonce and digest values as follows:

   - First computes nonce as: $N_v^* = \lambda_i - \theta_i$. To check $Dig1 == H[m_i || N_v^* + \theta_i]$ or not ? value is true as we already proved the computed $\theta_i == \omega_i S - b_i P_3$.

   Similarly, when RSU receives a request message through UAV, RSU also validates the request as follows:

   - Check $Dig2 == H[m_i || N_v^{**} + \theta_u]$ or not ? where $N_v^{**} = \lambda_u - \theta_u$ is calculated by RSU. Hence, the equation returns true as we already proved: computed $\theta_u == \omega_u S - d_i U_3$.

   Further, when the RSU replies V_i through the UAV, then the response message m_j is again validated at both units, the UAV as well as V_i. The steps involved in validating m_j are as follows:

   - Initially, the UAV computes the nonce as: $N_r^* = \lambda_r - \theta_r$ does the equation $Dig3 == H[m_j || N_r^* + \theta_r]$ hold or not ?
   - Next, V_i computes the nonce as: $N_r^{**} = \lambda_u - \theta_u$ to cross-check if the equation $Dig4 == H[m_j || N_r^{**} + \theta_u]$ is true or not.

   The above equations return the true value, as already verified; the signature; and generate the same value of θ_r and θ_u by the UAV and vehicle, V_i respectively.

3. **Conditional privacy preserving**: All the communicating units communicate with each other by hiding their own real identities. In the proposed

scheme, V_i requests registration with the trusted RA by using its real identity, hashed identity, and password, such as the following:$(Vid_i, HVid_i, HVPw_i)$. Even in the future, it V_i uses to communicate its signature $\sigma_i = (\theta_i, \omega_i)$ and EID_v, not its real identity or hashed identities. Similarly, the UAV also registered with RA with credentials $(Uid_i, HUid_i, HUPw_i)$. Even though in the future UAVs will also use them EID_u to communicate, which is an encrypted value by using k. Similarly, RSU requests RA with value:$(Rid_i, HRid_i, HRPw_i)$. Even in a communicating network in the future, never use its identity or passwords after validating the received key R_1 from RA by calling $\mathtt{Q}uery[R_2]$.
Consequently, our proposed scheme ensures that all communication units effectively uphold the conditional privacy-preserving definition by utilizing their UID rather than their actual identity.

4. **Traceability**: The suggested protocol serves the purpose of traceability in the event of any dispute. Only the RA can ascertain the true identity of a vehicle, not any other entity. RA can determine it as the following:$Vid_i = \mathcal{H}_2[HVid_i + k + h] \oplus P_1$. In the network, the vehicle utilized its P_3, which is updated in the database corresponding to the P_1 acquired. Similarly, RA is capable of tracking RSU and UAV in the event of a disturbance.

5. **Unlinkability**: vehicle generates its signature $\sigma_v =< \theta_v, \omega_v >$ by using a randomly chosen secret vector $v \in Z_q^m$. Where $\theta_v = vS$ and $\omega_v = v + \delta_v x_i$. Note, it δ_v depends on keys P_3, T_v, θ_v. Thus, no adversary can link signatures of vehicles, σ_v and $\sigma_v^{'}$ together as keys, they v, T_v uphold the freshness feature and involvement of θ_v in ω_v.
Likewise, no adversary can link signatures of RSU $\sigma_r, \sigma_r^{'}$ and signatures of UAV $\sigma_u, \sigma_u^{'}$ together because of the key freshness of secret vectors, r, u respectively.

6. **Resist Against Impersonation Attack**: In our proposed scheme, no adversary $\mathcal{A}$ can impersonate the communicating units in ITS due to ISIS's hard assumptions. Assume it $\mathcal{A}$ wants to impersonate vehicle V_i, for this, it $\mathcal{A}$ needs to construct the same tuple set: $\langle m_i, \sigma_v, EID_v, T_v, \lambda_v, Z_1 \rangle$, where $\sigma_v = (\theta_v, \omega_v), \theta_v = vS, \omega = v + v\lambda_v$, and $\lambda_v = \mathcal{H}[P3||T_v]$. Thus, it is impossible for $mathcalA$ to impersonate due to the involvement of the vehicle's secret key: x_i, random vector v, and P_0 (calculated by RA). Most importantly, $\mathcal{A}$ the value of the master secret key s is $S = s^T X$ because it comes under the ISIS hard problem. Thus, for the impersonation in our scheme, an adversary needs the $x_i, vP_0, ands$, which is impossible to guess or calculate all these values at the same time, which is impossible. Likewise, for UAV and RSU adversaries,

failure to impersonate is in our proposed scheme.

7. **Resist against Man-in-the-Middle attacks**: As we already know, an adversary failed to impersonate the units VehicleV_i, UAV U_i, and RSU R_i due to the ISIS hard problem. Hence, no adversary can be present between the communicating units to perform the MITM attacks. Let the adversary $\mathcal{A}$ be present between the vehicle V_i and U_i. When the vehicle V_i forwards the request message : $\langle m_i, \sigma_v, \lambda_v, T_v, EID_v, Z_1$ towards the UAV, it $\mathcal{A}$ received the request. It is impossible to forward the received tuple with the same signature along with the encrypted identity EID_v and Z_1.
   To generate the same signature, one $\mathcal{A}$ has to calculate the exact value of the involved secret keys and vectors: x_i, v, s, P_0 where for $P_0 = \mathcal{H}_0[Vid_i, t]$, one $\mathcal{A}$ also needs the real identity of V_i and ephemeral key t, which are only known by RA, and the calculation is impossible due to the collision strength resistance property of the hash function, including preimage and second-preimage properties.
   Therefore, due to being unable to generate the same signature σ and encrypted identity UID of communicating units, our proposed scheme resists the MITM attacks.

8. **Resist against Replay Attacks**: In our proposed scheme, whenever units want to communicate, they send a tuple including a timestamp T_i and signature σ. Assume for the vehicle, send the tuple along with the signature, $\langle m_i, \sigma_v, T_v, EID_v, \lambda_v, Z_1\rangle$ where Signature $\sigma_v = (\theta_v, \omega_v) = (vS, v + \lambda_v x_i)$, and where $\lambda_v = \mathcal{H}[P3||T_v]$ is dependent on the timestamp value. Thus, due to the fresh nature of timestamps for every time a request or response, it prompts the possibility of replay attacks. Similarly, for the UAV and RSU case as well. Hence, our proposed scheme resists the replay attacks.

9. **Perfect Forward Secrecy** To safeguard previously communicated messaging, the system must inhibit attackers from accessing previous session keys, even if $\mathcal{A}$ has access to the communication's private keys. proposed session key SK is depends on the UID_v, UID_r and N_v, N_r by expanding these, we saw that TID depends on the expiry time Et set by RA and parameter $P1 or R2$ which is depends on secret key of RA: K, h. To determine these key is impossible due to the collision resistance property of the hash functions. Next, it SK also depends on the nonce $N_r \& N_v$, which has a freshness property. Thus, attackers cannot determine the previous or future key based on the current session key information.

10. **Resist the known Ephemeral key**: our proposed scheme maintains secrecy of the SK even if we share the ephemeral key of the current session with $\mathcal{A}$. Assume that RA shares t information with $\mathcal{A}$, still failed to obtain the information about the SK as it depends on $k, h, x_i, z_i, N_v \& N_r$. and to determine this key information comes under hard assumptions.

11. **Stolen Verifier table**: In our scheme, all units only know their private key values; they do not share these with any table. They update their smart cards with a few parameters (not the secret key) in the blockchain database, excluding their secret key. Only in case of any dispute, RA can cross-check the identity, and updated parameter values are accessible by only authorized units using Algorithms 3 to 6.

12. **Revocation:** In case of any misbehavior activities done by any units, whether it is a vehicle, UAV, or RSU, that are noticed by RA. Then call the revocation algorithm to erase all the transactions done in that session. In this way, malicious activities are easily identified and erased. So, only verified information should be stored in a database by a verified and authentic entity.

13. **Distributed Data Storage:** A Hyperledger Fabric–based network is established in which multiple RAs function as peer nodes to support distributed ledger storage. Each peer actively participates in the consensus mechanism and independently verifies updates made to the ledger. As a result, the proposed approach enables decentralized access to data along with reliable and distributed storage.

14. **Resist against Distributed Denial of Service**: In our scheme, the computation of the session key is followed by the authentication phase (signature verification). During the signature generation process, units- $V_I, U_i \& R_i$ need random vectors like: v, u, r, privates keys: x_i, y_i, z_i, and Timestamps T_v, T_u, T_r respectively. And on receiving the request from multiple attacker points, Units initially verify the timestamps, then verify the signature to authenticate those points, which depends on a random vector and timestamps. The computation of vectors v, u, r and private keys x_i, y_i, z_i comes under the SIS and ISIS hard problems. As a result, units failed to authenticate when they received the request from multiple sources and rejected the false request immediately. Hence, our proposed scheme successfully resists the DDoS attacks.

15. **De-synchronization Attacks**: In our proposed scheme, all the communicating units shared their hashed identity and passwords with RA, which are

Table 6.3: Comparison of Related Existing Schemes Based on Security Requirements.

Scheme	Various security attacks												
	S0	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**
[84]	√	√	⊡	×	×	√	√	√	√	⊡	×	×	×
[85]	√	√	×	√	√	√	√	√	√	⊡	×	⊡	⊡
[83]	√	√	×	√	√	√	√	√	√	⊡	×	×	×
[89]	√	×	×	√	√	√	√	√	√	×	×	×	×
[88]	√	√	⊡	×	√	√	√	√	√	⊡	√	×	×
[97]	×	√	√	√	√	√	√	√	√	√	√	⊡	⊡
[90]	×	√	√	√	√	√	√	√	√	×	√	⊡	⊡
OUR	√	√	√	√	√	√	√	√	√	√	√	√	√

S0: *Key Agreement*; S1: *Modification*; S2: *Un-linkability*; S3: *Traceability*; S4: *Identity privacy*; S5: *Mutual Authentication*; S6: *Man-in-the-middle*; S7: *Replay*; S8: *Impersonation*; S9: *Stolen verifier table*; S10: *Quantum*; S11: *Desynchronization*; ; S12: *Distributed Denial of Services.*
√: *Safe*; ×: *Unsafe*; ⊡: *Not Mentioned.*

stored in RA's database. And then, RA generates parameters by applying the hash over the received hash ID and passwords. Further, no updated hashed ID or password calculation was done and used for any phases. Thus, attacks to obtain the message of the current session due to desynchronization attacks are impossible and resisted in our scheme.

16. **Resist to Quantum attacks**: The accuracy and formal security evaluation utilizing "history-free reduction" within the framework of QROM can be guaranteed by appropriately designing the scheme with the relevant essential variables. Suppose that in the event the adversary can compromise the scheme by retrieving the private key, it is also able to resolve the SIS/ISIS hard problem on the lattice. The SIS/ISIS complex problems on the lattice are computationally challenging to resolve using correct variables. Consequently, they exhibit resilience against prospective quantum attacks.

6.5 Performance analysis

In this section, we describe the performance analysis of our proposed QS-BAKA scheme for the UAV-assisted ITS environment. Computation and communication cost, along with storage and energy overhead cost, should be evaluated to compare our QS-BAKA scheme against various existing schemes [84], [85], [83], [89], [88], [97], and [90]. We chose the system parameter $m = n = \mathcal{O}(t\ \log q)$, where, $q = \mathcal{O}(t^2)$. Now, to make it more convenient for the performance evaluation, we take it as:

Table 6.4: Comparison of Competing AKA based prior schemes.

Scheme	**CA**	**UAV**	**MA**	**SSK**	**BCT**
[84]	ECC	YES	YES	NO	NO
[85]	ECC	YES	YES	YES	YES
[83]	ECC	YES	YES	YES	NO
[88]	LBC	NO	YES	YES	NO
[97]	LBC	NO	YES	NO	YES
[90]	LBC	NO	YES	NO	YES
Proposed	LBC	YES	YES	YES	YES

CA: *Cryptography Algorithm*; ECC: *Elliptic Curve Cryptography*;MA: *Mutual Authentication*; LBC: *Lattice Based Cryptography*; SSK: *Secured Session Key*; BCT: *BlockChain Technology.*

Table 6.5: Comparison of Storage, Communication & Computation Cost of existing Lattice-based Scheme for the ITS environment.

Scheme	Order of Execution	Type	Primitive	Cost
[88]	$O(mn{\cdot}\|q^2\|)$	Storage	$\langle \mathbb{R}, \mathbb{R}_{\shortparallel}^{\ltimes}, \mathbb{R}_{\shortparallel}^{\urcorner}, \mathbb{R}_{\shortparallel[\mathbb{B}]}^{\ltimes} \rangle \in Z_q^{m\times n}$, $\langle D_\varphi^n, a_i, sk_i \rangle \in Z_q^n$	$(4mn+3m)\cdot\|q\|$ $\approx 32t^2\log^3 t + 12t\log^2 t$
		Communication	$\langle z_1, z_2 \cdots z_3, c', \rangle$	$(3mn+1)\cdot\|q\|$ $\approx 12t^2\log^3 t + 2\log t$
		Computation		$4mn\cdot\|q^2\|{+}3m\cdot\|q\|$ $\approx 64t^2\log^4 t + 12t\log^2 t$
[97]	$O(mn{\cdot}\|q^2\|)$	Storage	$\boldsymbol{d} \in \mathbb{Z}_q^m, \boldsymbol{X} \in \mathbb{Z}_q^{m\times n}$, and $\langle RID, PASS, K, TID \rangle \in Z_q^*$	$(mn+m+4)\cdot\|q\|$ $\approx 8t^2\log^3 t + 4t^2\log^2 t + 8\log t$
		Communication	$\langle m, t, \sigma, TID \rangle$	$(m+n+3)\cdot\|q\|$ $\approx 8t^2\log^2 t + 6\log t$
		Computation		$3mn\cdot\|q^2\|{+}m\cdot\|q\|{+}n\cdot\|q\|$ $\approx 48t^2\log^4 t + 8t\log^2 t$
[90]	$O(mn{\cdot}\|q^2\|)$	Storage	$S_0 \in \mathbb{D}_{\alpha q}^{n\times n}, E_0 \in \mathbb{D}_{\alpha q}^{m\times n}, \boldsymbol{T_0} \in \mathbb{Z}_{\alpha q}^{m\times n}$ where $m = 2n$	$2mn\|q\|{+}n^2\|q\| \approx (4n^2{+}n^2)\|q\|$ $\approx 40t^2\log^3 t$
		Communication	$\langle Time_{reg}, PID_v i, T_{vi}^i \rangle$ $\mathbb{C} = \langle \mu, \upsilon_1, \upsilon_2 \rangle$	$\|q\| + n^2\|q\| + mn\|q\| n^2\|q\| + mn\|q\|$ $\approx (1+2n^2+n^4\|q\|)(2\log t)$ $\approx 32t^4\texttt{log}^6 t + 16t^2\log^3 t + 2\log t$
		Computation		$(mn+3n+3)\|q\|$ $\approx 16t^2\log^3 t + 12t\log^2 t + 6\log t$
Proposed	$O(mn{\cdot}\|q^2\|)$	Storage	$s \in \mathbb{Z}_q^m, \boldsymbol{S} \in \mathbb{Z}_q^{1\times n}$ $\{h, t, k\} \in \mathbb{Z}_q^n$	$(m+4n)\cdot\|q\|$ $\approx 12t\log^2 t$
		Communication	$\langle \sigma_u, Dig4, EID_u, EID_r, K_1, \rangle$ $\langle \lambda_u, T_u, T_r \rangle$.	$(2mn+m+6)\cdot\|q\|$ $\approx 16t^2\log^3 t + 4t\log^2 t + 12\log t$
		Computation		$(2mn+2)\|q\|$ $\approx 16t^2\log^3 t + 4\log t$

Table 6.6: Comparison based on parameters: Storage, Communication, and Computation Cost for ECC-based mechanism using UAV.

Scheme	**Message Count**	**Storage** (in bits)	**Total Communication** (in bits)	**Total Computation** (in milliseconds)
[84]	6	2528 bits	5344 bits	$17T_H + 6T_{BP} + 4T_{MHP} + 6T_{PM} \approx$ 71.6802 ms
[85]	6	2752 bits	4320 bits	$12T_H + 2T_{SE}/T_{SD} + 12T_{PM} + 4T_{PA} \approx$ 65.2106 ms
[83]	4	1760 bits	3872 bits	$10T_{SPM} + 1T_{PA} + 18T_H + 2T_SD \approx$ 58.1961 ms

$m = t \log q$ and $q = t^2$. We assume that these parameters are sufficient to prove security against the ISIS and SIS challenges.

Table 6.3 illustrates the comparison of several schemes based on a few important security parameters denoted by $S0 - S12$. Wherever Table 6.4 is used to highlight the main characteristics of a few existing schemes based on authenticated key agreement concepts using parameters mentioned in table notes, like whether it depends on blockchain or not, whether there is mutual authentication or not, whether LBC is used or not, etc.

Table 6.5 represents the comparison of storage, communication, and computation costs acquired by prior schemes. This table compares only those schemes that are based on LBC. Whereas, Figure 6.7 represents the graphical presentation of comparison based on computational security level, execution time, storage requirement, and computational cost as well as average consumption of the setup, registration, and AKA phase.

Table 6.6 represents how our proposed scheme outperforms comparison with various ECC-based schemes in terms of storage, communication, and computation costs. Whereas, the table inside Figure 6.8 is used to define the execution times taken by numerous operations that are utilized in ECC-based schemes.

Figure 6.8 is used to show the graphical representation of security level, storage, computational, and communication costs of the existing ECC-based scheme against our scheme.

6.5.1 Storage overheads

In our proposed QS-BAKA scheme to store the master secret key s and master public key $S = s^T X$, we need $m|q|$ overhead as $s \in Z_q^m$ and require $n\,|q|$ overhead for the S as $X \in Z_q^{m \times n}$. Further, we need to store RA's secret key $h \in Z_q^n$. Thus, RA needs

$n\,|q|$ overhead. Hence, total storage overhead is computed as:$m\,|q| + n\,|q| + n\,|q| = (m + 2n) \cdot |q|$. which is equivalent to $= (2t \log t + 2 \cdot 2\, t\log t)(2 \log t) \approx 12t \log^2 t$. The same approach is being used for the existing LBC scheme [89], [88], [97], and [90] to calculate the storage primitives.
Which are represented by the help of table 6.5. Graphical representation demonstrated by subfigure 6.7f in Figure 6.7 in a positive, desirable way compared to other existing schemes based on LBC.

We also perform the storage analysis for the ECC-based approaches that utilize the UAV (or drone) system. For the ECC-based scheme in general, we consider, for 'identity, a random generation function' of 16 bits; for '(hash mapping)' it is 320 bits, and for the 'Sign Encrypt and Decrypt' 128 bits are required, 'ECC point additions' should be 320 bits. By this general case, we calculate for ECC-based authenticated schemes that use UAVs. Table 6.6 is used to describe the total storage taken by existing schemes, including their communication cost and computation costs. This is presented by subfigure 6.8f in Figure 6.8.

6.5.2 Communication Overheads

In our proposed scheme, communication is performed between vehicle V_i and RSU R_i via UAV U_i. The RSU first checks the authenticity of the Vehicle as well as the UAV and then responds as per the requested message. RSU forwards the tuple towards the Vehicle, which is forwarded by UAV as: $\langle \sigma_u, Dig4, EID_u, EID_r, K_1, \lambda_u, T_u, T_r \rangle$. Hence, total transmission cost should be

$(2mn + m + 6)|q|$. Which is equivalent to $16t^3 \log^3 t + 4t \log^2 t + 12 \log\, t$. Note that we only considered the computational cost of the time-consuming operation in our proposed scheme, where $|q| = \log q = 2 \log t$.

Table 6.5 is used to represent the comparison of the communication of a few of the existing LBC-based schemes compared to our proposed scheme. This highlights how our scheme is better than others in terms of communication costs. In Figure 6.7, subfigure 6.7e shows the communication overhead for a quantum-safe authentication mechanism based on lattice-based cryptography for vehicular environments.

Regarding estimating goals, the total execution overhead of the different cryptography operations has been determined as the average of fifty executions for $m = n = 128$. The communication costs have been calculated by considering the length of different factors, including hash $|H|$= 64 bytes, number $|Z_q^*|$= 16 bytes,

identity $|ID|$= 16 bytes, timestamps, and as nonce $|T_i|, |N_i|$= 4 bytes.

We also compare our scheme with a few ECC-based schemes that are mainly UAV-assisted for ITS environments. Table 6.6 illustrates how our proposed scheme is better than the existing ECC-based scheme in terms of communication cost as well. Figure 6.8 has subfigure 6.8e that represents the graphical comparison of our work with the ECC-based scheme in terms of communication cost.

Remark 2. *Sometimes, for sure, a few ECC-based AKA schemes can better perform in terms of communication cost, but they fail to stand against quantum attacks and many more, like unlinkability, traceability, DDoS attacks, and many more. Thus, we are motivated to introduce a scheme that is a combination of LBS and blockchain technology, including UAV assistance in ITS environments. Thus, we achieved our aim: to resist quantum attacks; provide decentralization; and resist other attacks, such as DDoS, etc. With the assistance of UAVs, we also reduce the latency between the vehicle and RSU data transmission and also reduce the setup cost of additional RSUs due to their scalable and movable nature.*

6.5.3 Computational Analysis

In our scheme, the vehicle V_i computes the session key: $SK = H[UID_v||UID_r||N_v||N_r]$ only after verifying the authenticity of the RSU and UAV as well. Unique identity $UID_v = \mathcal{H}_3[P_1||x_i||Et_v]$, and nonce $N_v \in Z_q^{m\times n}$ hold computation cost, $|q|, mn|q|$ respectively. Next, to verify the RSU computes P_0 and then K_2 consumes computation cost: $n|q|, \&|q|$ respectively. At last, verifying the signature: $\theta_u == \omega_u S - b' U_3$ takes $mn|q|$ computation cost. Thus, the final computation cost is: $(2mn + n + 2)|q| = 16t^2 \log^3 t + 4t \log^2 t + 4\log t$.

Table 6.5 is used to show the comparison of a few LBC-based schemes with our proposed scheme, and Figure 6.7d represents the computational cost analysis for the quantum-safe scheme for ITS environments. In addition, we also performed a comparison with a few ECC-based schemes, as defined in Table 6.6. And its graphical representation is illustrated by the figure 6.8 with subfigure 6.8d.

6.5.4 Simulation Setup

- ***System Configuration***

  Under this section, we describe the hardware descriptions of all the involved units: V_i, U_i, and R_i in our proposed mechanism. In our proposed scheme, the

Trusted Authority TA broadcasts the system parameters. Next, the RA registered all units with their respective requesting information (identity, hash ID, and passwords). We used a high-performance computing server for the RA, who is responsible for the registration process, which helps to authenticate the units further. HPC with @ AMD EPYC 9654 with 2, 4 GHz base clock, which has 512 GB RAM per node, for secondary storage, 2X NVMe SSD (each node 4 TB). For interconnection, it features a 200G HDR InfiniBand network, thus facilitating low-latency, high-throughput node communication. HPC has Rocky Linux 9 as the OS. All the vehicle's and UAV's functionalities are simulated on the IoT device Raspberry Pi 5 with the features of a BCM2712 Quad-Core 64-bit Arm Cortex-A76 Processor running at 2.4 GHz with cryptography extensions and support SDR104 mode for faster read-write operations. All the RSUs' tasks are performed on desktops with a configuration of Intel Core i7 9700 CPU at 3.0 GHz with 8 GB RAM processor setup, along with 64-bit Ubuntu 20.04.4 LTS desktop i386 OS.

- ***Software Configurations***
  - ***Hyperledger Fabric BC:*** Hyperledger Fabric is a popular freely available permissioned blockchain technology since it can handle smart contracts rapidly and inexpensively and can execute several thousand transactions at once seconds. We utilized Hyperledger Fabric to build our scheme due to its requirement for every participant of the consortium to confirm their real identities beforehand, before becoming part of the network. Thus rendering control over access and security even more robust. We select the RAFT consensus protocol since it is known for being secure, fault-tolerant, and private. RAFT works by choosing the leader point from the scheduling network nodes. This node is in charge of gathering transaction data. The leader generates a new block and broadcasts it across nodes once the number or size of the transaction reaches a certain level. This makes sure that the ledger updates are all in a consistent state. This mechanism is optimal as it is reliable, resilient, and allows for expansion.
  - ***PyCryptodome Library:*** It is a distinct Python module that has a set of basic cryptographic design blocks. It works with PyPy and Python versions 2.7, 3.5, and beyond. PyCryptodome is a revised version of PyCrypto that provides more capabilities and is easier to maintain. It may be used to encrypt, hash, and execute other cryptographic tasks in

Table 6.7: Comparison of Authentication Latencies of ECC- and LBC-based schemes with our proposed scheme.

Scheme	**Average Latency (ms)**
ECC-Based Schemes [84], [85], [83]	24.1
LBC-Based Schemes [89], [88], [97], and [90]	27.8
Proposed	20.3

Python scripts.

- ***Hashlib Library:*** This module facilitates a feasible way to utilize a lot of well-known and safe hashing strategies in Python's code. It has well-known algorithms like MD5 and the SHA family (SHA1, SHA224, SHA256, SHA384, SHA512) that are often used to make distinct digital patterns of information. Hashlib renders simple to build and use cryptographic hash functions whenever it's necessary to validate the authenticity of data or protect login credentials confidentially.

Regarding estimating goals, the total execution overhead of the different cryptography operations has been determined as the average of fifty executions for $m = n = 128$. The communication costs have been calculated by considering the length of different factors, including hash $|H|= 64$ bytes, number $|Z_q^*|= 16$ bytes, identity $|ID|= 16$ bytes, timestamps, and nonce $|T_i|, |N_i|= 4$ bytes.

Figures 6.7 and 6.7 represent the computation and communication as well as the energy consumption details in their first three sub-figures. Energy consumption is estimated with the help of the expression $E = P \times T$, where the variables' units are millijoules (mJ), watts (W), and milliseconds, respectively. Authentication latency is calculated in terms of milliseconds (ms). Table 6.7 represents the comparison of the proposed scheme in terms of accuracy and efficiency. We found that our proposed scheme reduced the latency by 16.8% in comparison with the ECC-based scheme and 37.3% with LBS-based schemes. When we increase the vehicle density, the throughput (in terms of sessions per second) of our proposed scheme is improved as we employ the UAV, whereas in the case of the ECC-based scheme, performance is degraded due to full dependency on the RSU.

6.5.5 PBFT Performance Under Vehicular Mobility

This section analyzes Practical Byzantine Fault Tolerance (PBFT) performance in our Hyperledger Fabric consortium blockchain under vehicular network conditions, addressing latency, throughput, and reliability under high mobility.

A. **Experimental Setup:**

We extended our NS-3 v3.41 + Hyperledger Fabric v2.5 simulations as follows:

* **Network**: 100 nodes (50 vehicles, 30 UAVs, 20 RSUs)
* **Mobility**: RandomWaypoint (20-100 km/h), topology churn $f = 0.3$
* **PBFT**: $f < 1/3$ fault tolerance,
  3-phase (PRE-PREPARE/PREPARE/COMMIT)
* **Metrics**: Consensus latency, TPS, block propagation, fork frequency

B. **Quantitative Results:** Table 6.8 illustrates the quantitative result of our PBFT performance based on static and vehicular mobility as we take 100 units.

Table 6.8: PBFT Performance: Static vs Vehicular Mobility (n=100 nodes)

Metric	Static	Mobility	Degradation
Consensus Latency (ms)	120	290	+142%
Throughput (TPS)	450	150	-67%
Block Propagation (ms)	80	280	+250%
Fork Risk (%)	0.8	10.2	+1175%

C. **Performance Analysis:**

PBFT's $O(n^2)$ message complexity degrades significantly under vehicular mobility:

1. **Intermittent Connectivity**: 30% link churn (Δt=1-5s) delays quorums
2. **View Changes**: Frequent leader elections when faulty links exceed $f = 1/3$
3. **Propagation Delays**: Multi-hop V2I/V2U increases block dissemination time

D. **UAV Mitigation Strategy**

Our UAV-assisted architecture reduces mobility impact by 35%:

* Adaptive positioning maintains 80% link stability
* Aerial relays cut propagation delays to <250ms
* Dynamic quorum formation tolerates 25% churn

Despite degradation, 290ms latency supports real-time ITS applications (safety beacons, traffic events). Section 6.6 compares favorably against PoA alternatives given PBFT's stronger BFT guarantees.

E. **Protocol Phase Performance Analysis:**

We measured latency, end-to-end delay, and failure rates for each protocol phase under realistic vehicular conditions (100 nodes, 20-100 km/h mobility, 20% link churn):

Table 6.9: Protocol Phase Performance Metrics (n=100 nodes, 20% vehicular churn)

Phase	Latency (ms)	Total Delay (ms)	Failure Rate	Retries
1. Registration (RA)	45	45	0.2%	0.01
2. Auth Phase 1	58	-	0.3%	0.02
3. Auth Phase 2	62	180	0.4%	0.03
4. Auth Phase 3	60	-	0.1%	0.01
5. Key Exchange	85	85	0.3%	0.02
6. PBFT Pre-Prepare	35	-	0.5%	0.04
7. PBFT Prepare	120	290	1.2%	0.15
8. PBFT Commit	135	-	0.4%	0.08
End-to-End	-	465	2.1%	0.36

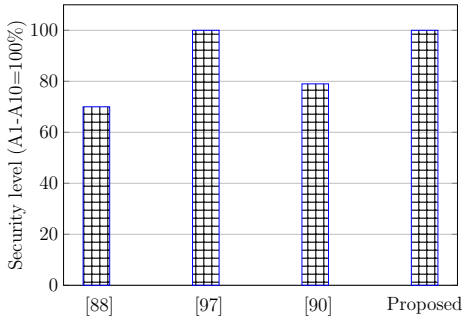
* **Registration**: Fastest phase (single RA round-trip), minimal failures
* **Authentication**: 3-message challenge-response averages 60ms/phase
* **PBFT Prepare phase dominates** (41% of consensus time) due to $O(n^2)$ messaging
* **Total 465ms** supports real-time ITS (safety beacons <500ms requirement)
* **2.1% overall failure** within PBFT $f < 1/3$ tolerance

UAV relays reduce PBFT phase delays by 28% through improved connectivity. All phases meet vehicular network timing constraints.

6.6 Summary

This chapter introduces a blockchain-based, quantum-safe authentication scheme for secure V2V communications in an intelligent transportation system. The proposed

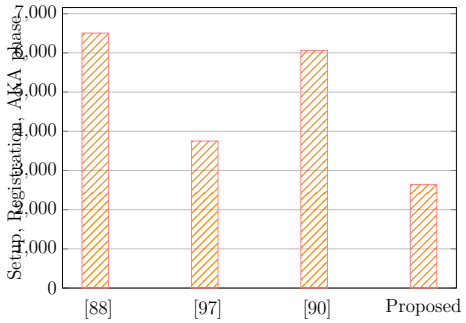
approach integrates the concept of UAV and blockchain technology into the ITS environment by taking multiple semi-RAs as blockchain peers to provide distributed database access and storage. Formal security analysis using the Quantum ROM (QROM) model has demonstrated the proposed scheme's resilience against security attackers, and informal analysis shows the security against various security attacks. Finally, the extensive performance analysis using cryptographic libraries and the Hyperledger Fabric platform demonstrates that the proposed scheme is efficient in computational cost compared to existing schemes.



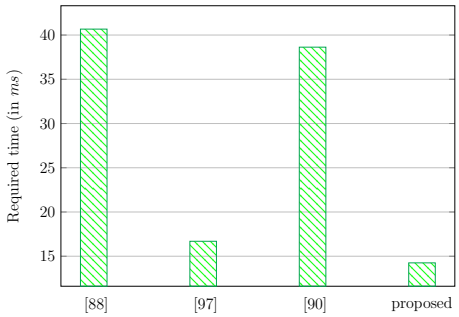
(a) Achieved Security (↑ *Desirable*)

Operation	RA	Vehicl	UAV	RSU
$T_{\mathcal{H}}$	0.1153	.1492	.0954	.2041
T_{Rnd}	0.0251	.1057	.0030	.0049
T_X	.1195	.1416	.1079	.1962
T_s	.0297	.1175	.9931	.9176
$T_{s^T \boldsymbol{X}}$	.1634	.2368	.2017	.2837
T_{sS}	.8172	.9762	.8033	.6174
$T_{\boldsymbol{x}^T}$	0.0112	0.031	0.0117	.0201

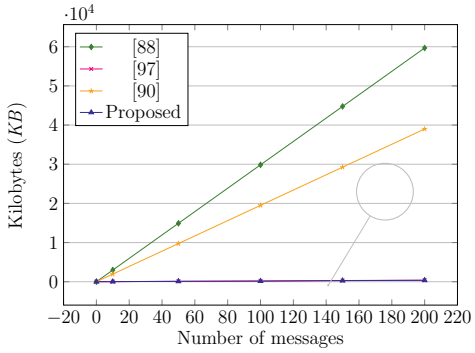
(b) Execution costs (ms) of Units



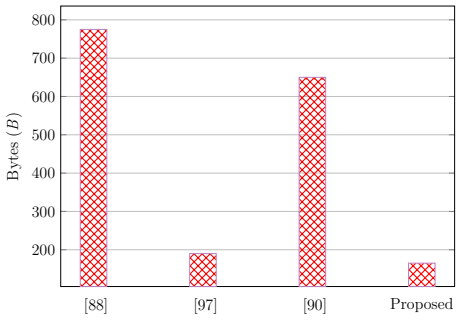
(c) Average Consumption (↓ *Desirable*)



(d) Computation Cost (↓ *Desirable*)



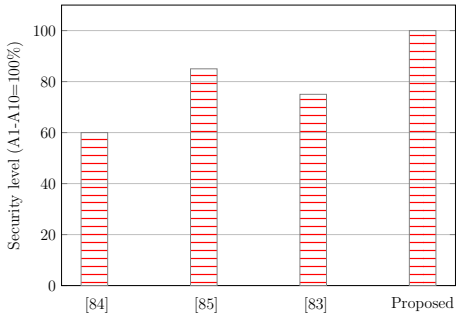
(e) Communication Cost (↓ *Desirable*)



(f) Storage Space Requirements (↓ *is desirable*)

Notations: $T_{\mathcal{H}}$: Hash Digest; T_{Rnd}: random vector; T_X: Matrix from $Z_q^{m \times n}$; T_s: Vector from Z_q^m; $T_{s^T X}$: Product of s^T and X, form a vector; $Z_q^{1 \times n}$ T_{xS}: product of vector and matrix; T_{x^T}: transpose of x.

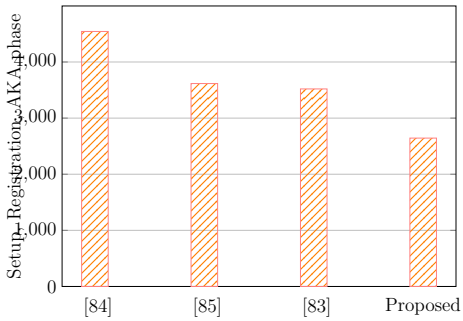
Figure 6.7: Performance Analysis of Existing Quantum-Safe AKE Schemes in ITS Environments



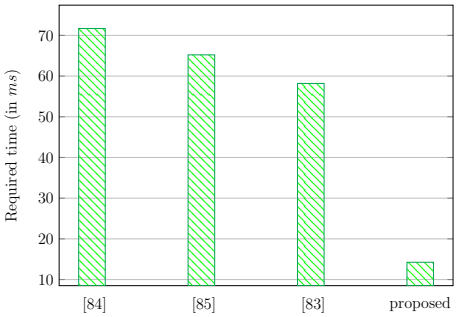
(a) Achieved Security (↑ *Desirable*)

Operation	Definition	UAV
$T_{\mathcal{SPM}}$	ScalarPoint Multiplication	2.0057
T_{SM}	Sclar Multiplications	1.0051
T_{PM}	Point Multiplications	4.8047
T_{MHP}	Hash mapping to points	2.0068
T_{BP}	Bilinear Pairing	8.1769
T_{SE}	Sign Encryption	0.9249
T_{SD}	Sign Decryption	0.9436
T_{PA}	Points Addiions	0.0587
T_H	One Way Hash function	0.4558

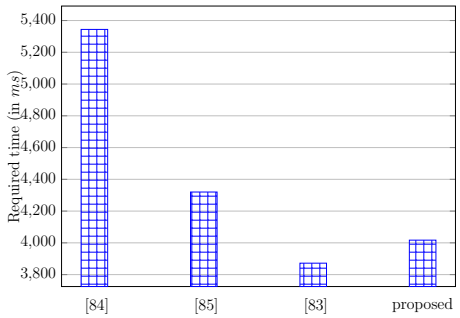
(b) Execution costs (ms) of Units



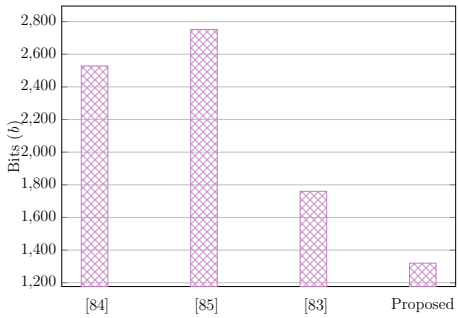
(c) Average Consumption (↓ *Desirable*)



(d) Computation Cost (↓ *Desirable*)

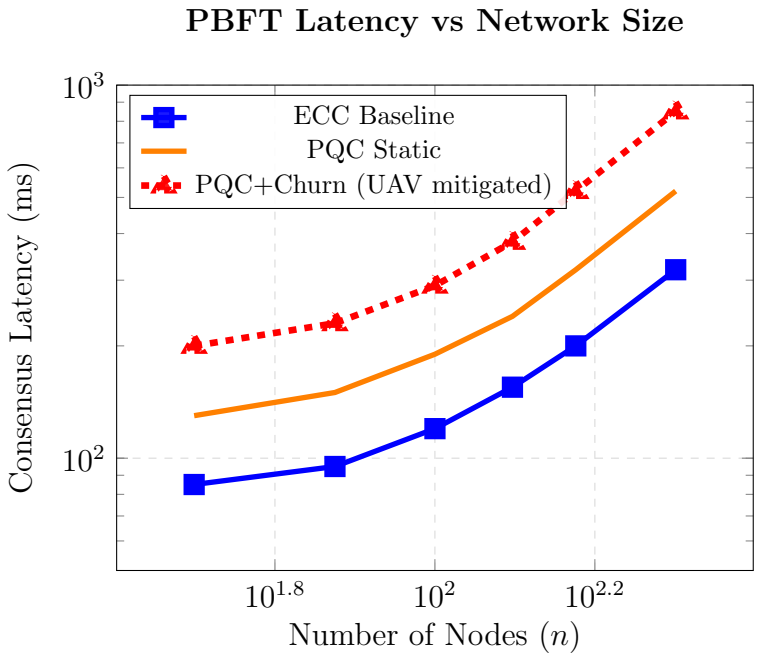


(e) Communication Cost (↓ *Desirable*)



(f) Storage requirements (↓ *Desirable*)

Figure 6.8: Performance Analysis of Existing UAV-Assisted ECC-Based Schemes.



PBFT Consensus Latency vs. Network Size ($n = 50 - 200$nodes). *Where:***ECC Baseline** (blue): 85–320 ms. **PQC Static** (orange): +58% overhead from 4.8 KB Dilithium signatures vs 64 B ECC. **PQC+Churn** (red dashed): +142% total degradation under vehicular mobility ($f = 0.3$, 20–100 km/h). UAV relays mitigate 35% penalty, maintaining real-time ITS viability (<500 ms).

CHAPTER 7

Quantum-Resistant Anonymous Authentication for Blockchain Enabled Smart Healthcare

7.1 Introduction

A lot has been accomplished concerning healthcare in the past several years. Healthcare facilities, hospitals, and private residences nowadays depend on intelligent devices that monitor the condition of patients in real time and communicate information instantaneously among healthcare professionals and various departments [92]. Implantable sensors, gadgets like portable scanners, and platforms running on the cloud all transmit information through a continuous chain that helps medical professionals determine treatment more efficiently and more accurately. These electronic devices have contributed to better healthcare, but they've also created additional hazards. Highly confidential information is flowing via a sophisticated web of connections and storage platforms [3]. The security of patient information—ensuring that it's confidential and correct as well as secure from tampering—is just as crucial as the treatment procedures patients receive.

Not only is the amount of information passing through emerging smart healthcare systems enormous, but it is also extremely confidential. Laboratory outcomes, past illnesses, and medical treatment sequences can tell you whatever you desire to know concerning a specific individual [86]. In the unlikely scenario of such information being disclosed or tampered with, its disclosure would definitely not purely

be an instance of disrepute; it might compromise the person's safety, credibility, and possibly their life itself. Offensive actions directed at healthcare entities are no longer solely limited to unauthorized surveillance. Hackers can now guess patient routines, change device readings, or even send fake commands that could mess with someone's care [96].

A lot of healthcare systems still employ conventional cryptography like RSA, elliptic-curve crypto, or identity-based solutions. For decades, such protocols have maintained information confidential, but quantum computers are starting to solve the computational challenges that render those secure [96]. Medical records need to be kept confidential for decades, and often even throughout the entirety of one's life. Thus, long-term security is not simply a desired feature; it's an essential requirement. If quantum computers take off, they could break these old protections overnight and expose years' worth of private medical records. This is a significant issue—methods that were effective in the past will not be sufficient for the future. To address these growing vulnerabilities, lattice-based cryptography (LBC) has emerged as one of the most promising post-quantum solutions. Lattice problems such as Learning with Errors (LWE), Ring-LWE, and Short Lattice-based cryptography (LBC) have become one of the strongest emerging post-quantum proposals for these increasing weaknesses [32]. Lattice hard challenges like Learning with Errors (LWE), Ring-LWE, and Short Integer Solutions (SIS/ISIS) have excellent security enhancements from the most vulnerable circumstance to the standard scenario. They are also thought to be relatively resistant to quantum attackers at this point. Short-reminiscent quantum attacks don't compromise the security of LBC, unlike ECC or identity-based networks. This makes lattice-based schemes ideal candidates for long-term medical data protection, secure device authentication, and robust key agreement mechanisms in smart healthcare systems. In addition, lattice primitives support lightweight operations suitable for resource-constrained medical IoT devices, enabling efficient implementation even on low-power sensors.

7.1.1 Why we Integrate Blockchain Technology?

It's not enough to only use strong cryptography to fix all security shortcomings. In most healthcare systems, everything depends on central servers or authority. Such centers supervise content, keep patient information, and regulate which devices can connect to the network and when. But putting all your faith in one place can cause enormous challenges: systems don't handle growth well, privacy gets compromised, and if anything happens improperly at the center, the entire process might fall apart. Just a single violation, like an intruder breaking through the database, a trusted

individual going rogue, or a good old DDoS, may trigger healthcare facilities, laboratories, and insurance organizations to become crazy [87]. Furthermore, whenever information needs to be transmitted across all of these parties, relying upon one central authority is not really compatible with the prerequisites for transparency and availability. If one of these central points goes down—whether from hacking, insider mistakes, or ransomware—the whole hospital, maybe even a whole region, can grind to a halt. With lives on the line, you can't afford that kind of fragility. These problems have pushed people to look for better ways to run healthcare systems—ways that spread out control so no single group holds all the keys.

Blockchain technology offers a natural complementary solution by decentralizing trust, enabling tamper-evident logging, and distributing control among multiple nodes. When integrated with healthcare infrastructures, blockchain allows medical events (such as patient admissions, diagnostic updates, medication changes, or device interactions) to be recorded in immutable and verifiable ledgers [118]. Such transparency is crucial for avoiding manipulation of clinical data, ensuring accountability among healthcare providers, and complying with regulatory requirements. Additionally, blockchain mitigates the single-point-of-failure problem by distributing responsibility across multiple nodes, ensuring system resilience even under targeted attacks. Its decentralized ledger makes it tough to secretly change records, and it lets all the different healthcare players check that information hasn't been tampered with. By eliminating the central intermediary, blockchain significantly reduces the vulnerability to large-scale, coordinated attacks [141]. But there's a catch: just putting medical transactions on a blockchain doesn't mean they're private. Since everyone on the network can see the ledger, clever people can still figure out things like when devices are used and how often, or even pick up patterns in patient care—unless you add extra privacy protections.

Nevertheless, applying blockchain directly to healthcare introduces new challenges. Most notably, healthcare systems must preserve patient anonymity and restrict unnecessary exposure of sensitive medical activities. Recording data on-chain without careful privacy controls may reveal behavioral patterns, device usage, or treatment histories. Traditional blockchain mechanisms—such as simple public-key-based authentication or miner-based consensus—could inadvertently expose identities or link user actions across transactions. Thus, although blockchain provides decentralization and auditability, an additional privacy-preserving authentication mechanism is essential.

To address these needs, blind signatures serve as an effective cryptographic tool for privacy-preserving authentication. Blind signatures allow a user to obtain a sig-

nature on a message without revealing its content to the signer [117]. The signer cannot later link the signed message to the blinded request, guaranteeing anonymity. In the context of our smart healthcare architecture, blind signatures enable nodes that contribute new blocks or data entries to authenticate themselves without revealing identifiable information to the blockchain network. This preserves privacy while ensuring that only authorized healthcare devices or servers update the ledger. Furthermore, blind signatures prevent traceability attacks and linkability issues, giving patients and medical entities freedom to interact securely without disclosing their identities during blockchain operations.

As a result, we designed a lightweight authentication protocol for smart healthcare systems using the LB cryptosystem and also introduced an LB-based blind signature for blockchain authentication purposes.

7.1.2 Problem Statement

The open and wireless nature of SHS makes security and privacy the most significant issues to be addressed. The open communication channel can be exploited by the adversary to seize the data flow and impersonate the medical server to extract the private data of the patient. Additionally, other attacks, for example, spoofing attacks, replay attacks, man-in-the-middle attacks, etc., are also faced by SHS networks. Therefore, in SHS, security measures are essential to provide secure communication and data security. A proper security scheme requires that devices and patients be registered with a trusted authority, and data should be transmitted in encrypted form with the sender's signature. Alternatively, a scheme should achieve both data confidentiality and sender authentication [175].

Lattice-based cryptography (LBC) has emerged as one of the most promising post-quantum solutions. Lattice problems such as Learning with Errors (LWE), Ring-LWE, and Short Lattice-based cryptography (LBC) have become one of the strongest emerging post-quantum proposals for these increasing weaknesses [32]. Lattice hard challenges like Learning with Errors (LWE), Ring-LWE, and Short Integer Solutions (SIS/ISIS) have excellent security enhancements from the most vulnerable circumstance to the standard scenario. They are also thought to be relatively resistant to quantum attackers at this point. Short-reminiscent quantum attacks don't compromise the security of LBC, unlike ECC or identity-based networks. This makes lattice-based schemes ideal candidates for long-term medical data protection, secure device authentication, and robust key agreement mechanisms in smart healthcare systems. In addition, lattice primitives support lightweight operations suitable for resource-constrained medical IoT devices, enabling efficient im-

plementation even on low-power sensors. It leads to issues including single point of failure, higher latency, and inefficient and insecure data handling.

Nevertheless, applying blockchain directly to healthcare introduces new challenges. Most notably, healthcare systems must preserve patient anonymity and restrict unnecessary exposure of sensitive medical activities. Recording data on-chain without careful privacy controls may reveal behavioral patterns, device usage, or treatment histories. Traditional blockchain mechanisms—such as simple public-key-based authentication or miner-based consensus—could inadvertently expose identities or link user actions across transactions. Thus, although blockchain provides decentralization and auditability, an additional privacy-preserving authentication mechanism is essential.

To address these needs, blind signatures serve as an effective cryptographic tool for privacy-preserving authentication. Blind signatures allow a user to obtain a signature on a message without revealing its content to the signer [117]. The signer cannot later link the signed message to the blinded request, guaranteeing anonymity. In the context of our smart healthcare architecture, blind signatures enable nodes that contribute new blocks or data entries to authenticate themselves without revealing identifiable information to the blockchain network. This preserves privacy while ensuring that only authorized healthcare devices or servers update the ledger. Furthermore, blind signatures prevent traceability attacks and linkability issues, giving patients and medical entities freedom to interact securely without disclosing their identities during blockchain operations.

As a result, we designed a lightweight authentication protocol for smart healthcare systems using the LB cryptosystem and also introduced an LB-based blind signature for blockchain authentication purposes.

7.1.3 Main Contributions

Our proposed system incorporates a lattice-based authenticated key agreement method that lets medical servers and patient devices share quantum-safe keys. This ensures that confidentiality, integrity, and authenticity security aims are all protected at the communication layer, even when dealing with advanced adversaries who can eavesdrop and manipulate messages. We also utilize a blind-signature method based on a lattice to support privacy-preserving authentication during block creation and data transfer. Our layered strategy not only protects private health information, but it also makes users' anonymity and the system stronger.

By putting these innovative ideas together into a blockchain-powered healthcare system, our architecture gets past the difficulties that centralized healthcare systems

have had for a long time. With decentralization, no one person or organization is in charge of everything. Cryptographic immutability makes sure that medical events can't be changed or made up. Blind signatures make sure that no one can connect them or observe them, which prohibits profiling or inference attacks. Lattice-based cryptography is safer than identity-based or elliptic-curve-based choices and safeguards the system from future quantum threats.

The contributions of our proposed scheme are described below:

1. **A quantum-resistant, secure key agreement protocol for smart healthcare;** We present a lightweight lattice-based AKA protocol that is based on SIS/ISIS hardness assumptions. The system ensures mutual authentication, reliability, and secure session-key establishment between the patient devices and medical servers, even in the presence of quantum-capable adversaries.

2. **Decentralized framework empowered by blockchain technology to eradicate reliance on central authorities;** We use a distributed ledger to resolve the challenges encountered with centralized healthcare systems having a single point of failure. All important events and identity checks are saved in a way that can't be changed, without needing a trusted central authority.

3. **A privacy-preserving blind-signature mechanism for anonymous blockchain authentication:** We propose a lattice-based blind-signature framework enabling devices to authenticate themselves when appending new blocks or submitting clinical data—without exposing their identity or creating linkable patterns.

4. **Comprehensive QROM-based formal security analysis;** We offer robust game-based security assurances for forward secrecy, KCI/UKS resistance, blind-signature unforgeability, and blindness, as well as mutual authentication and session-key secrecy. In the Quantum Random Oracle Model (QROM), all proofs are made. This model has higher assurances than conventional models.

5. **Improved protection against real-world health challenges;** Our combined architecture mitigates impersonation, replay, data manipulation, identity linkage, misuse of authentication servers, and quantum-based cryptanalysis. The design remains efficient enough for resource-constrained medical IoT devices.

6. **A scalable framework suitable for long-term deployment;** Our system is well-positioned for sustainable and scalable smart healthcare deployment be-

cause lattice cryptography is considered one of the most mature post-quantum standards, and blockchain provides long-term auditability.

7.2 Background

This section has outlined the network model, security model, and security goals defined for the proposed scheme.

7.2.1 Network Model's Entity

We introduce a security protocol: a quantum-safe authentication key agreement protocol to maintain the data privacy and secrecy, whereas we use a blind signature mechanism to check the authenticity of the collecting node in a blockchain-enabled smart healthcare system.

There are two communicating ends (patient and medical server as a node in blockchain) and a trusted authority TA. All the communicating entities are defined below:

i. **Trusted Authority** TA is a trustworthy entity. The primary function TA is to retrieve the key pairs of the corresponding communicating points. By employing the off-site method for point authenticity. TA computes the key pair with a signature. The real identity of any point is only known by TA.

ii. **Patient's End:** In SHS, all the required smart sensors are implanted on the patient's body. An accumulator is being used to accumulate all the patient's health information. This point is represented as HIA: Health Information Accumulator. We assume that all the smart sensors are safe and secure to transmit the health information with the patient's collecting device (smartphone or any other smart device here represented as HIA).

iii. **Blockchain Architecture:** In our proposed model, blockchain consists of five nodes: N1: Collecting node, N2: Hospital Equipment, N3: Emergency Treatment, N4: Medical Research, and N5: Government Authority. Where each node has its specific role, which is explained in Table 7.1.

7.2.2 Proposed Model's Working Architecture

In our proposed SHS model, the smart sensors are implanted on/in the body of the patient. Then, HIA accumulates all the transmitted sensors' information and creates

Table 7.1: Definition of Role Play by Nodes in Blockchain: Phase II.

Node	Role
N1	Used for collecting the health information from the accumulator from the patient's end. This node role is also decided by other nodes belonging to the blockchain based on the PoW mechanism.
N2	Holds the information regarding the hospital's equipment. Maintain the information and make a schedule regarding the usage of equipment as per the patient's visit.
N3	Used for emergency treatment required. Maintain the senior doctor's duty schedule as per the emergency call as per the information received.
N4	Used for medical research purposes. Researchers work on the most recent available medical information of patients with the aim of finding better treatment for the respective disease.
N5	Government Authority, who can handle and monitor all the activities of the hospitals to prevent the malpractice of treatments.

a block. Now HIA needs to share this health record with the hospital receiver point via the internet. As there is a possibility of leakage of medical history and other security and privacy issues, we need a secure method to maintain data confidentiality as well as authenticity. Therefore, HIA generates a session key with the collecting point of the blockchain. This process is well defined in phase I.

Now, at the Medical Server (Blockchain) end, whether the collecting points are valid to receive the blocks from HIA, it must be verified by other nodes. Thus, we need a mechanism to check the authenticity of the collector belonging to the blockchain. For this purpose, we use the blind signature mechanism, which is defined and explained in phase II.

At last, when it is decided which node can collect the HIA data and its verification is done, then the final phase of the proposed model proceeds, which is data transmission. Now, we need to forward a consent form, a type of smart contract, which permits access to HIA data by the collecting node of the blockchain. This step is defined in phase 3.

Figure 7.1 illustrates the working process of our proposed protocol for secure communication performed between the patient and the medical server. And we can easily understand how, during phase I, patients' sensor information is collected by HIA, and HIA then transfers this information to the collector node via the internet. For secure communication purposes, after receiving the system parameters from TA, the entity patient (HIA) and the medical server (N1, known as the collector node) must be registered with TA, and then, only after receiving the key information, they start to perform the mutual authentication and determine the session key at their respective ends. The role of the collector node is also decided by the nodes that

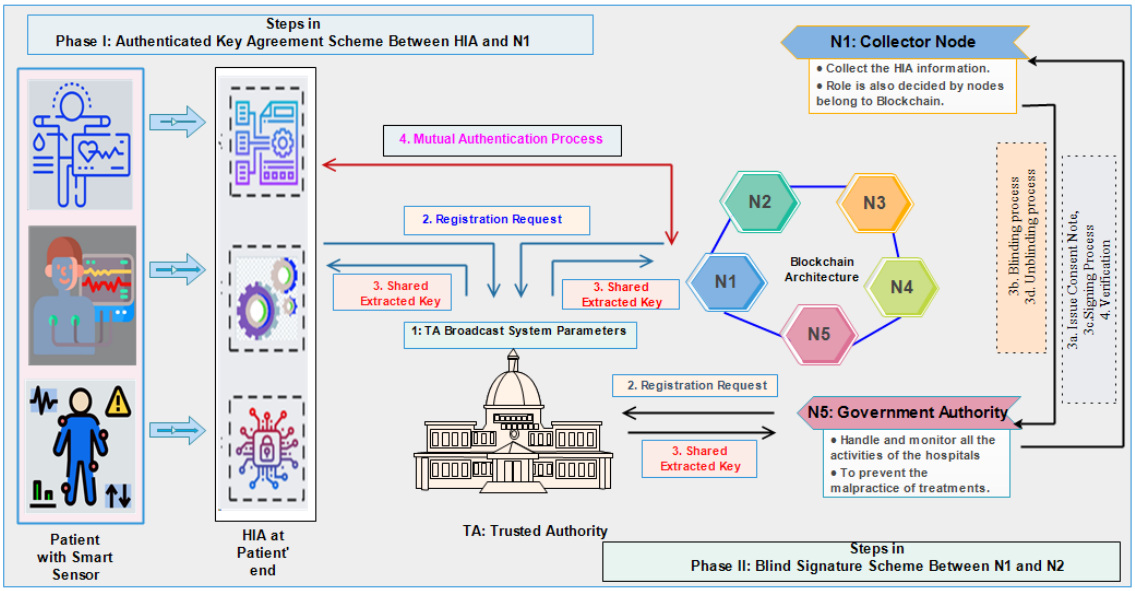


Figure 7.1: Proposed System Model For Blockchain-Assisted Smart Healthcare System.

belong to the blockchain. Thus, we introduce a blind signature-based quantum-safe mechanism to check the genuineness of the collector node. For this, we assume N5 as the signer node, which wants to check whether N1 is eligible to collect HIA information or not. Steps involved in this process are defined in phase II. Finally, after these phases, the data transmission process will take place between the HIA and N1 (in simple words, between the patient and the medical server).

7.2.2.1 Phase I: LBAKA Scheme: To maintain secure communication in SHS, we designed a two-level authenticated secure protocol. First, we designed an AKA protocol using LBC to prevent quantum attacks for IHS and meet other defined security requirements. This phase consists of four steps: setup, registration, key extraction, mutual authentication, and session key generation, described as follows;

i. **Setup:** In this step TA broadcasts its system parameters.

ii. **Registration and Key Extraction:** In this step, all the communicating nodes (patient: HIA and medical server: N1) must be registered with TA. Then TA generates the key pair for the requesting nodes.

iii. **Mutual Authentication:** In this step, the patient and medical server points must mutually authenticate each other before starting to share the information over the internet.

iv. **Session Key:** To maintain the confidentiality of the shared information over the public channel, we must generate the session key SK. This session key is used to encrypt and decrypt the information.

Thus, there is secure communication performed between the patients and the medical server using this authenticated session key in the SHS environment.

Remark: In this protocol, in general, we mentioned communication done between the patient and medical server. During the scheme description, HIA is represented by P and the collecting node, which belongs to the blockchain, which is originally the medical server M.

7.2.2.2 Phase II: Blind Signature Scheme:

We designed a blind signature-based authentication scheme to authenticate the collector node (N1) that wants to add the HIA data block to the existing blockchain. Such verification of the authenticity of the node shows that the collecting node is a genuine one. For this purpose we use the lattice-based blind signature concept, which consists of four steps: Setup, registration, blind signature generation, and blind signature verification. Definitions of these steps are mentioned below:

i. **Setup:** The Trusted Authority TA takes input to generate the system parameter.

ii. **Registration and Key Extract:** In this step, TA takes input from the nodes (who want to verify and to whom to check) to register them. Afterwards, it TA generates the key pair (public/private) for the further steps.

iii. **Blind Signature Generation:** By using the blind factor, a signature is generated over the message by the signer node.

iv. **Blind Signature Verification:** Verification is done after the unblind process is performed. This step is used to validate the signature and obtain the original message.

7.2.3 Security Model

In this section, we formalize the adversarial capabilities against our proposed two-level protocol consisting of the lattice-based authenticated key agreement (LBAKA) in Phase-I and the blind-signature-based collector authentication (BSBCA) in Phase-II. The adversary is assumed to be quantum polynomial-time (QPT) and interacts

Table 7.2: Notations used in the Adversary Model for our Proposed Scheme

Query Type	Oracle	Symbol	Bound
Quantum hash queries	H_1, H_2, H_3	$q_{H_1}, q_{H_2}, q_{H_3}$	$\leq \mathsf{poly}(\vartheta)$
Registration queries	$\mathcal{O}_{\text{Reg}}$	q_{reg}	$\leq \mathsf{poly}(\vartheta)$
Authentication sessions	$\mathcal{O}_{\text{AKA}}$	q_{aka}	$\leq \mathsf{poly}(\vartheta)$
Blind-signature queries	$\mathcal{O}_{\text{BS}}$	q_{bs}	$\leq \mathsf{poly}(\vartheta)$
Corruption queries	$\mathcal{O}_{\text{Cor}}$	q_{cor}	$<$ number of parties

with the protocol through classical and quantum-accessible oracles modeled in the Quantum Random Oracle Model (QROM).

System Entities: The system consists of the following participants:

- **TA:** Trusted Authority, holding the master secret key (non-corruptible).
- **P:** Patient-side node.
- **M:** Medical server/collector node.
- N_5**:** Blind-signature signer node used in Level II.
- $\mathcal{A}$**:** Quantum polynomial-time adversary.

7.2.3.1 Numeric Query Bounds The adversary $\mathcal{A}$ is permitted the following types of queries, each bounded polynomial ϑ is demonstrated in table 7.2. The total number of quantum hash queries is:$[q_H = q_{H_1} + q_{H_2} + q_{H_3}]$.

Definition 10. QPT Adversary A QPT adversary $\mathcal{A}$ is any interactive quantum algorithm that:

1. Runs in time polynomial in the security parameter ϑ,
2. Generates quantum superposition states,
3. Queries each hash function H_i via a quantum random oracle (QRO),
4. Fully controls the communication channels between honest parties.

Formally, $\mathcal{A}$ is represented by a polynomial-size quantum circuit

$$\mathcal{A} : |x\rangle \mapsto U_{\mathcal{A}}|x\rangle.$$

Definition 11. ***Quantum Random Oracle*** Each hash function H_i is modeled as an ideal random function accessible by $\mathcal{A}$ through the unitary operator: $[\mathcal{O}_{H_i} : |x, y\rangle \mapsto |x,\ y \oplus H_i(x)\rangle]$. The adversary may input quantum superpositions: $\sum_x \alpha_x |x, 0\rangle,]$.

and the simulator answers using the compressed-oracle technique to ensure consistency and allow safe reprogramming with bounded disturbance.

QROM Model Assumption: In the QROM, we assume a stronger adversary with quantum computational capabilities:

- The adversary can submit quantum superposition states to the random oracle and receive superposition outputs.
- This models attackers who can evaluate hash functions on multiple inputs simultaneously using quantum algorithms (e.g., Grover's algorithm).
- Despite quantum capabilities, the adversary remains computationally bounded, with a limited number of quantum queries q.
- In our UF-CMA security proof for signatures, the adversary has quantum access to the hash function H but only classical access to the signing oracle.

The QROM provides stronger security guarantees by accounting for post-quantum threats, which is essential for IoT systems that must remain secure against future quantum attacks.

7.2.3.2 Adversarial Oracles

i ***Registration Oracle***: $\mathcal{O}_{\text{Reg}}$ Allows $\mathcal{A}$ registering arbitrary identities: $\mathcal{O}_{\text{Reg}}(ID) \rightarrow (A, \text{Sig})$ with at most q_{reg} queries.

ii ***Authentication Oracle***: $\mathcal{O}_{\text{AKA}}$ Simulates LBAKA protocol sessions

- Initiates sessions,
- Returns accept/reject outcomes,
- Exposes session identifiers.

Limited to q_{aka} sessions.

iii ***Blind-Signature Oracle***: $\mathcal{O}_{\text{BS}}$ Allows blind-signature requests: $\mathcal{O}_{\text{BS}}(m) \rightarrow \text{BlindSig}(m)$. with at most q_{bs} queries.

iv ***Corruption Oracle***: $\mathcal{O}_{\text{Cor}}$ Allows adaptive corruption: $\mathcal{O}_{\text{Cor}}(P_i) \rightarrow$ long-term keys. The TA cannot be corrupted, and neither partner of the test session may be corrupted.

v ***Quantum Hash Oracles***: The adversary may access: H_1, H_2, H_3 (with q_{H_i} queries each).

7.2.3.3 Adversarial Success Conditions

Definition 12. ***Mutual Authentication Break*** The adversary succeeds if it causes an honest party to output "accept" while the corresponding partner session does not exist or is under adversarial control.

Definition 13. ***Session Key Indistinguishability*** Let $\mathrm{SK}_{\mathrm{real}}$ denote the true session key and $\mathrm{SK}_{\mathrm{rand}}$ a random string of equal length. The adversary issues a Test query and receives SK_b. It outputs a bit b'. Its advantage is defined as:
$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{SK}} = \left| \Pr[b' = b] - \frac{1}{2} \right|$.

Definition 14. ***Blind-Signature Unforgeability*** The adversary outputs (m^*, ϱ^*) such that:

1. $\mathrm{Verify}(m^*, \varrho^*) = 1$, and
2. m^* was never queried to the blind-signature oracle.

Its advantage is defined as:
$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{UF}} = \Pr[(m^*, \varrho^*) \text{ is a valid new forgery}]$.

Definition 15. ***Blindness*** Given two messages of equal length, the adversary must not distinguish which one corresponds to a given signature. The advantage is: $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Blind}} = \left| \Pr[b' = b] - \frac{1}{2} \right|$.

7.2.3.4 Adversary Model Summary

Overall, the success probability $\mathcal{A}$ is bounded as:

$$\mathrm{Adv}_{\mathcal{A}} \leq \underbrace{\mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS/ISIS}}}_{\text{hardness assumption}} + \underbrace{\epsilon(q_{\mathrm{aka}}, q_{\mathrm{reg}}, q_{\mathrm{bs}})}_{\text{bad events}} + \underbrace{\delta(q_H, o)}_{\text{QROM reprogramming penalty}}$$

where o denotes the number of programmed oracle points.

7.2.4 Security Goals

The following security objectives are defined for the proposed scheme [176]:

1. **Data Integrity and Confidentiality:** Any malicious attacker should not be able to extract health-related data from the transmitted message, and the data should also be protected from malicious modifications.
2. **Authentication:** Both smart medical devices and patients should be authenticated. All the communicating nodes must be authentic in the blockchain, also.

Table 7.3: Notations used in the proposed scheme's phase I.

Notations	Definition
1^m	Security Input Parameter
$n,\ q$	Integer, Prime number
$\mathcal{M}$	Matrix derived from $Z_q^{n \times n}$
$ID_P,\ HID_P$	The real and hashed identity belongs to the patient P
$ID_M,\ HID_M$	Real and hashed ID of the medical server M
$H_1, H_2, H3$	Secure cryptographic hash function.
x	Master private key of the TA, $x \in Z_q^n$.
$\mathfrak{M}_{Pb}$	Master public key of the TA, $\mathfrak{M}_{Pb} \in Z_q^{m \times 1}$.
a, b	TA select secret key for the P and M chosen by TA, where $a, b \in Z_q^n$
r, s	P and M select their secret key, respectively, from Z_q^n

3. **Quantum Resilience:** Conventional cryptography-based schemes do not resist the quantum attacks; thus, by utilizing the quantum-safe cryptographic algorithm, lattice-based, our scheme should resist quantum attacks.

4. **Identity Anonymity:** The original identity of a device should be hidden during communication. The attacker should not be able to access the original device identity by any means.

5. **Traceability:** If any device is found to be involved in malicious activities, it should be traced out and revoked by the proposed scheme.

6. **Distributed Data Storage:** The data will be managed by servers in multiple hospitals using blockchain architecture.

7. **Data Immutability:** The data cannot be modified in an unauthorized way. All the modifications are carried out by a consensus mechanism running between participating nodes.

8. **Resistance towards the Attacks:** The proposed scheme will be proven secure against modification, impersonation, unforgability, and replay attacks.

7.3 Proposed Scheme

Our proposed scheme is a combination of two schemes: Phase I and Phase II, which are described below:

7.3.1 Phase I: LBAKA Scheme

The proposed model comprises four phases in total: setup, registration and key extraction, mutual authentication, and the last key agreement phase. All the symbols used in our scheme's phase I are defined in Table 7.3.

Patient P	**RA**	**Medical server** M
Computes $HID_P = H_1[ID_P]$		Computes $HID_M = H_1[ID_M]$
$\xrightarrow{\langle HID_P, ID_P\rangle \text{ To TA}}$		$\xleftarrow{\langle HID_M, ID_M\rangle \text{to TA}}$
	For Patient Choose a secret vector $a \in Z_q^n$ Computes $A_P = \mathcal{M} \cdot a$ Computes $Sig_P = x + \sigma_P \cdot a$ $\sigma_P = H_2[HID_P, A_P, \mathfrak{M}_{Pb}]$ Uploads tuples in database: (A_P, Sig_P).	
$\xleftarrow{\langle A_p, Sign_P\rangle \text{ to P}}$		
Verify the received key as: if $(\mathcal{M} \cdot Sig_P == \mathfrak{M}_{Pb} + \sigma_P \cdot A_P)$? True then follow next phase otherwise, again make request	For Medical Server Choose a secret vector $b \in Z_q^n$ Computes $A_M = \mathcal{M} \cdot b$ Computes $Sig_M = x + \sigma_M \cdot b$ $\sigma_M = H_2[HID_M, A_M, \mathfrak{M}_{Pb}]$ Uploads tuples in database: (A_M, Sig_M).	
		$\xrightarrow{\langle A_M, Sign_M\rangle \text{ To TA}}$ Verify the authenticity of the received tuple as: if $(\mathcal{M} \cdot Sig_M == \mathfrak{M}_{Pb} + \sigma_M \cdot A_M)$? repeat if return value false, otherwise follow subsequent phase.

Figure 7.2: LBAKA: Steps in Registration Phase

7.3.1.1 Set up Phase This phase is mainly used to generate the system parameter and broadcast it to other nodes, which is performed by TA. It computes the system parameters after receiving the input 1^m as follows:

i. TA selects an integer n and a prime modulus q, all matrix calculations are computed with modulus q.

ii. TA Select a matrix $\mathcal{M} \in Z_q^{n \times n}$.

iii. TA define hash functions as:
$H_1 : [\{0,1\}^*] \rightarrow \{0,1\}^m$
$H_2 : [Z_q^n || \ \{0,1\}^m || Z_q^n] \rightarrow \ Z_q^*$
$H_3 : [Z_q^{1\times n} || Z_q^{1\times n} || Z_q^{n\times n} || Z_q^{n\times n} || Z_q^{1\times 1}] \rightarrow \{0,1\}^m$.

iv. Now, TA select the master private key: $x \in Z_p^n$.

v. TA computes the master public key:$\mathfrak{M}_{Pb} = \mathcal{M} \cdot x$.

v. Now broadcast its system parameter param as:
$\{\mathcal{M}, q, \mathfrak{M}_{Pb}, H_1, H_2, H_3\}$ and kept the secrecy of its master private key $\mathbf{x}$.

7.3.1.2 Registration and Key Extraction Step

The trusted authority TA will determine the key combination as well as signatures used for authorization. All steps involved in this part are described in Figure 7.2 defined below:

i. First, the communicating node computes the hashed identity HID with the aim of hiding its real identity from other communicating nodes, including intruders. Let's say a patient node P wants to register with TA with a tuple $ID_p, HID_p\rangle$ where:$HID_P = H_1[ID_P]$.

ii. On request from the patient node P, TA call the REG Algorithm 7 with argument HID. The registration algorithm is written in a smart contract. The algorithm returns the key pair and signature to the requesting node.

iii. Similarly, when the medical server MS sends the request to register, it $\langle ID_M, HID_M\rangle$ then RA returns the values as per the REG algorithm called, which are as follows: $A_M = \mathcal{M} \cdot b$, $Sig_M = \mathbf{x} + \sigma_M \cdot b$, where $\sigma_M = H_2[HID_M, A_M, \mathfrak{M}_{pb}]$.

iv. On receiving the key pairs, nodes need to validate the received signature, such as here, where the patient node verifies it as: if $(\mathcal{M} \cdot Sig_P == \mathfrak{M}_{Pb} + \sigma_P \cdot A_P)$ exist or not. Validation of Signature: RHS: $\mathcal{M} \cdot Sig_P$
$== \mathcal{M} \cdot (x + \sigma_P \cdot a)$
$== (\mathcal{M} \cdot x + \sigma_P \mathcal{M} \cdot a)$
$== (\mathfrak{M}_{Pb} + \sigma_P \cdot A_P)$ LHS

Algorithm 7 LBAKA REG()

```
1: function REG(HID_i) {
2:   if HID_i ∈ database then
3:   return 0; { ▷ TA returns the variables from the table corresponding to < ID_i, HID_i > }
4:   else
5:       TA choose ephemeral key a ∈ Z_q^m; { ▷ Here we computes the variables for patient node as per received < ID_P, HID_P > }
6:       A_P = M · a;
7:       Sig_P = x + σ_P · a;
8:       σ_p = H_2[HID_p, A_p, M_Pb];
9:     return ⟨A_p, Sig_P⟩;
10: }
```

7.3.1.3 Mutual Authentication Phase

This step is mainly used to perform the authentication of the requesting node to share the information in the IHS environment. Once a communication is done, the agreement key expires for that session.

Patient P	**Medical server** M
Choose a secret vector $r \in Z_q^n$	
Computes $\tau_P = r^T \cdot \mathcal{M}$	
Computes $V_P = \mathcal{M} \cdot Sig_P \cdot r^T$	
Uploads tuples in database: (τ_P, V_P)	
$\langle HID_P, \tau_P, A_P, V_P, t_1 \rangle$ To M $\longrightarrow$	
	First, validate the time stamp:$t_1 - t_1^{'} \leq \Delta t$
	reject Request if false
	Otherwise Verify the received signature as
	$V_p \cdot \mathcal{M} == [H_2[HID_P, A_P, \mathfrak{M}_{Pb}] \cdot A_P]$ hold or not ?
	Only if above equation return true
	Then M Choose a secret vector $s \in Z_q^n$
	Computes $\tau_M = s^T \cdot \mathcal{M}$
	Computes $V_M = \mathcal{M} \cdot Sig_M \cdot s^T$
	Uploads tuples in database: (τ_M, V_M)
	Calculate its key K_1 as;
	$K_1 = (s^T \cdot Sig_M)[\tau_P \cdot [\mathfrak{M}_{Pb} + H_2[HID_P, A_P, \mathfrak{M}_{Pb}] \cdot A_P]]$
$\langle HID_M, \tau_M, A_M, V_M, t_2 \rangle$ To P $\longleftarrow$	
	else terminate the request with fail to authenticate alert.
First, validate the time stamp:$t_2 - t_2^{'} \leq \Delta t$	
Reject Request if false	
Else Start to Verify the received signature as	
$V_M \cdot \mathcal{M} == [H_2[HID_M, A_M, \mathfrak{M}_{Pb}] \cdot A_M]$ hold or not ?	
When the Authentication is done successfully	
P Calculate its Key $K_2 = (r^T \cdot Sig_P)[\tau_M \cdot [\mathfrak{M}_{Pb} + H_2[HID_M, A_M, \mathfrak{M}_{Pb}] \cdot A_M]]$	

Figure 7.3: LBAKA: Mutual Authentication Phase

That means for each communication session, both communicating nodes need to establish a fresh key. Figure 7.3 represents all the steps involved in the mutual authentication process. By following the steps, we can see how both communicating nodes mutually authenticate each other before generating the session key for that particular session.

i. **At Patient P ends:** P chooses a secret value $r \in Z_q^n$ and computes: $\tau_P = r^T \cdot \mathcal{M}$, and $V_P = \mathcal{M} \cdot Sig_P \cdot r^T$. Now, the patient P sends the tuples: $\langle \tau_P, A_P, HID_P, V_P \rangle$ with timestamp t_1 towards the M to generate the **SK**.

ii. **Authentication of P at M ends :** First, the timestamp needs to be validated: $t_1 - t_1^{'} \leq \Delta t$. Reject the request if it returns the value false. Otherwise, the node M starts to verify the received signature in the smart contract to check the authenticity of patient P. Next, Node M checks the equation: $V_p \cdot \mathcal{M} == [H_2[HID_P, A_P, \mathfrak{M}_{Pb}] \cdot A_P]$ hold or not?. Node M rejects the request if it returns the value false. Once the authentication is done successfully, it M computes variables: τ_m, V_M and $K_1 = (s^T.Sig_m)[\tau_p \cdot [\mathfrak{M}_{Pb} + H_2[HID_P, \mathfrak{M}_{Pb}, A_p] \cdot A_p]]$. at last, M forwards tuples towards the node P: $\langle T_M, U_M, HID_M, V_M, t_2 \rangle$.

iii. **Authentication of Node M at P ends:** P also first verify the validity of the received timestamp t_2, P terminates the subsequent steps when it is invalid. Otherwise, it starts to check the authenticity by verifying the signature received using the equation: $V_M \cdot \mathfrak{M} == [H_2[HID_M, A_M, \mathfrak{M}_{Pb}] \cdot A_M]$? Once it returns a true value, it P computes its key $K_2 = (r^T.Sig_P)[\tau_M \cdot [\mathfrak{M}_{Pb} + H_2[HID_M, \mathfrak{M}_{Pb}, A_M] \cdot A_M]]$.

7.3.1.4 Session Key Generation Phase

Now, nodes can start the computation of the session key based on the variable secrecy level and the computed key at their ends. For this, we select the variable pairs $\langle \tau_P, \tau_M \rangle$ and $\langle V_p, V_M \rangle$, their computations as well as privacy, which depends on the secrecy of secret keys $(r, s) \in Z_q^n$ and signatures, (Sig_P, Sig_M) respectively. (Sig_P, Sig_M) maintain their privacy due to dependency on secret values chosen by TA: $(a, b) \in Z_q^n$. The computation of such information comes under SIS/ISIS's hard challenges. Hence, the session key $\mathcal{SK}$ depends on such variables T_P, T_M, V_P, V_M, which ensures strong preservation of the secrecy of $\mathcal{SK}$. By appending the key K, nodes can increase the privacy and security level of the session key. As a result, both nodes invoke the same session key, such as: $\mathcal{SK} = H_3[\tau_P, \tau_M, V_P, V_M, K]$. where $K == K_1 == K_2$ correctness is proved by theorem 1 defined under section 7.4.1.

7.3.2 Phase II: BSBCA Scheme

Under this section, we designed a blind signature mechanism using lattice-based cryptography to check the authenticity of the node. As a result, it shows that the collecting node N1 is eligible to receive the data forwarded from the HIA. Nodes belonging to the

Table 7.4: Notations used in the proposed scheme's phase II.

Notations	Definition
m	Security Parameter
$n,\ q$	Integer, Prime number
$\mathcal{P}$	Matrix derived from $Z_q^{m\times n}$
$ID_{N5},\ HID_{N5}$	Real and respective hashed ID of the Signer Node M
$H_1, H_2, H3$	Secure cryptographic hash function.
$H[....] \rightarrow Z_q^*$	Hash function to generate the number.
$G1, G_2$	Generating functions take input and return output also in a defined length.
p	Master private key of the TA, $p \in Z_q^n$.
$\mathcal{M}_{Pb}$	Master public key of the TA.
c	TA select secret key for the $N5$ chosen by TA, where $a \in Z_q^*$
t	$N5$ select its secret key, respectively, from Z_q^n

blockchain check the eligibility of the collecting node. Here, we assume N1 is the collecting node, and N5 is the signer node. The steps for the blind signature mechanism are described below. All the symbols used in our scheme's phase II are defined in Table 7.4.

7.3.2.1 Set up Step This phase is mainly used to generate the system parameter and broadcast it to other nodes, which is performed by TA. It computes the system parameters after receiving the input 1^k as follows:

i. TA selects positive integers m, n and a prime modulus q, all matrix calculations are computed with modulus q.

ii. TA Select a matrix $\mathcal{M} \in Z_q^{m\times m}$.

iii. TA define hash functions as:
$H_1 : [\{0,1\}^*] \rightarrow \{0,1\}^m$
$H_2 : [\{0,1\}^m || Z_q^{m\times n} || Z_q^{1\times m}] \rightarrow\ Z_q^*$
$H_3 : [\{0,1\}^m || Z_q^{1\times m}] \rightarrow Z_q^{n\times 1}$.

iv. Define generating functions as:
$G1 : \{0,1\}^{l1} \rightarrow \{0,1\}^{l2}$
$G2 : \{0,1\}^{l2} \rightarrow \{0,1\}^{l1}$, where $l1 + l2 = |q|$
$G : [\{0,1\}^q] \rightarrow Z_q^n$.

iv. Now, TA selects its master private key: $p \in Z_p^n$.

v. TA computes the master public key:$\mathcal{M}_{Pb} = p^T \cdot \mathcal{P}^T$.

v. Now broadcast its system parameter param and kept the secrecy of its master private key p; Param as: $\langle \mathcal{P}, m, n, q, \mathcal{M}_{Pb}, H_1, H_2, H_3, G1, G2, G\rangle$

7.3.2.2 Registration and Key Extraction Step During this step signer node N5 sends a request TA for the registration with $\langle ID_{N1}, HID_{N5}\rangle$, where $HID_{N5} = H_1[ID_{N5}]$. On receiving the request, it TA computes the following keys: $\alpha = c \cdot \mathcal{P}^T$, $\beta = H_2[HID_{N5}||\alpha||\mathcal{M}_{Pb}]$, and $\gamma = (c + p^T \cdot \beta)$. Next, TA shares $\langle \alpha, \gamma\rangle$ as a key pair to node N5.

7.3.2.3 Blind Signature Step The message $msg = \{0,1\}^q$ should be blind with the signing node N5. Thus, by following the steps, collecting and signing nodes generate the blind signature as mentioned below:

a. ***Signer node N5***: first choose a secret vector, $x \in Z_q^n$ then computes $\chi = x^T \cdot \mathcal{P}^T$. Now, N5 just issued a consent note $\langle \alpha, \chi\rangle$ to the collecting node N1.

b. ***Blinding Process:*** Collecting Node N1, select blinding factors $bf_1, bf_2 \in Z_q^*$ to calculate:
$\Delta = G[G1(msg)||G2(G1(msg))]$
$\theta = bf_1 \cdot \chi + bf_2 \cdot \Delta^T \cdot \mathcal{P}^T$
$A = H_2[HID_{N5}||\alpha||\theta]$
$B = bf_1^{-1} \cdot A$
Then, collecting node N1 sends this blinding factor B to signer node N5.

c. ***Signing Process:*** N5 signs the received factor as: $Z1 = [x + \gamma \cdot BF]$ and shares it with Node N1.

d. ***Unblind Process:*** Collecting node N1 generates the blind signature as follows:
$Z2 = [bf_1 \cdot Z1 + bf_2 \cdot \Delta]$
$u = H_3[HID_{N5}||Z2^T \cdot \mathcal{P}^T]$
$v = u - \Delta$
Now collector node N1 generates variable: $\langle msg, \theta, \alpha, v\rangle$, where $\sigma = (\theta, \alpha, v)$is the blind signature on message msg.

7.3.2.4 Signature Verification Step To verify the signature σ on msg the respective identity HID_{N5}, the verifying node performed the below steps:
$\beta' = H_1[HID_{N5}||\alpha||\mathcal{M}_{Pb}]$
$A' = H_2[HID_{N5}||\alpha||\theta]$
$u' = H_3[HID_{N5},\ \theta + A'^T(\alpha + \beta' \cdot \mathcal{M}_{Pb})]$ and
$\Delta' = u' - v$.
At last calculate $msg' = [G_1(\Delta')_{l2}||G_1(G_2(|\Delta'|)_{l1})$ and then cross-check if equation: $G[msg'] \equiv \Delta'$ is hold or not. Returning true means authentication was done successfully, and the original message was retrieved. Otherwise, authentication verification failed.

7.3.3 Data Transmission

When the patient wants to share their medical history, first, it needs to send a consent form to the receiver with consent that it can receive my information. For this blind signature mechanism, we introduced one that can decide which node is eligible to receive the HIA information. And then only a session key must be generated between the sender (P) and the receiver node MS. During the data transmission process, the patient uses this $\mathcal{SK}$ to encrypt the medical information. On receiving the encoded text, the collector node decrypts the text using the session key determined at the MS end. As we proved that both ends determine the same session key, the original medical history was received by the server. The medical profession team now provides accurate treatment within time. In this way, data transmission is safe and secure over the public channel in SHS.

7.4 Security Analysis

Under this section, we are going to explain the formal security analysis of both Phase I and Phase II, including the correctness proof of key K_1 and K_2 along with the session key SK. Subsequently, we also demonstrate the informal analysis of our proposed scheme based on numerous security requirements in the IoT environment.

7.4.1 Correctness Proof

By the definition of a few theorems, we can prove the correctness of our proposed scheme.

Theorem 7.1. Both node patients P and the medical server M participated in our proposed scheme; then they must calculate the same key ($K_1 == K_2$) computes the same session key.

Proof. Communicating nodes can transfer the information over the network by using the session $\mathcal{SK}$. The original message will be encrypted and decrypted by appending $\mathcal{SK}$ by the sender and receiver, respectively. This action is possible only when both ends must calculate the same key on their ends. We show the evidence demonstrating that both of these numbers are equivalent as follows: At the medical server M ends:

K_1
$=(s^T.Sig_M)[\tau_p \cdot [\mathfrak{M}_{Pb} + H_2[HID_P, \mathfrak{M}_{Pb}, A_p] \cdot A_p]].$
$=(s^T.Sig_M)[r^T \cdot \mathcal{M} \cdot [\mathcal{M} \cdot x + \sigma_P \cdot \mathcal{M} \cdot a]].$
$=(s^T.Sig_M)\mathcal{M}\mathcal{M}[r^T \cdot [x + \sigma_P \cdot a]].$
$= (s^T \cdot \mathcal{M} \cdot Sig_M \cdot \mathcal{M})(r^T \cdot Sig_P)$
$= (s^T \cdot \mathcal{M} \cdot (x + \sigma_M \cdot b) \cdot \mathcal{M})(r^T \cdot Sig_P)$
$= [\tau_M \cdot [\mathfrak{M}_{Pb} + H_2[HID_M, \mathfrak{M}_{Pb}, A_M] \cdot A_M]](r^T \cdot Sig_P)$
$= K_2$ (at Patient side).

Thus, we can write:
$K_1 == K_2 == K$ □

Theorem 7.2. The same session key $\mathcal{SK}$ must be generated between the patient node P and the medical server$M$ before sharing the information in a confidential way.

Proof. Our scheme $\mathcal{SK}$ is dependent on variables: $\langle \tau_P, \tau_M, V_P, V_M, K$. With reference to Theorem 1, we can see how both nodes calculate the same agreement key, K_1 and K_2, on their respective ends individually. And other used variables are shared using secure channels, which are demonstrated in section **??**. Hence, the generated $\mathcal{SK}$ by both nodes is correct and equal to: $\mathcal{SK} = H_3[\tau_P, \tau_M, V_P, V_M, K]$, which is used to start the communication in a secure way to maintain the data privacy and confidentiality. □

7.4.2 Formal QROM Game-Based Security Proofs and Analysis

This section presents formal, game-based proofs in the Quantum Random Oracle Model (QROM) for the security properties claimed for our two-level protocol: Level I (LBAKA) and Level II (BSBCA). Each proof is constructed as a sequence of games that are related by quantified indistinguishability or negligible differences. All hash functions H_1, H_2, H_3 are modeled as quantum random oracles, and the adversary is quantum polynomial-time (QPT).

- **Preliminaries and assumptions:** We restate the assumptions and black-box lemmas used across proofs.

  Assumption 1 (Lattice hardness)**.** *SIS (and ISIS as required) is hard for QPT adversaries: no QPT algorithm solves SIS/ISIS with non-negligible probability for the parameters chosen in Setup.*

  Assumption 2 (Signature security)**.** *If a reduction uses signing-unforgeability as a target, we assume the signer's signature scheme (the TA-generated SigP/SigM structure) is secure against existential forgery under chosen-message attack (UF-CMA) in the QROM (or is reducible to SIS/ISIS).*

  Lemma 7.1 (Compressed-oracle simulation)**.** There exists an efficient simulator that perfectly simulates QRO access for an adversary making at most (q_{H_i} quantum queries to each H_i. Inspecting or reprogramming at most o points in the compressed record changes the adversary's success probability by at most $\delta_{\text{prog}}(q_H, o)$, where $q_H = \sum_i q_{H_i}$.

  Lemma 7.2 (SIS/ISIS extraction (informal))**.** Suppose an adversary produces values satisfying the algebraic equalities used in verification (e.g., short relations

involving $\mathfrak{M}_{\mathfrak{M}_{\mathfrak{P}_b}}, \mathcal{M}, A_P, A_M$ and signature terms). In that case, an extractor can transform these into a solution to an SIS/ISIS instance sampled during setup with non-negligible probability.

Let $\mathcal{A}$ be any QPT adversary bounded by:q_{H_i} representing quantum queries to H_i, and $q_{\text{reg}}, q_{\text{aka}}, q_{\text{bs}}, q_{\text{cor}}$, registration, AKA sessions, blind-sign queries, and corruption queries, respectively. Let $q_H = \sum_i q_{H_i}$ and o denote the number of oracle points reprogrammed by a reduction in a given game hop.
We denote probabilities of events in game G_j: $= \Pr[G_j]$. Each hop provides a bound $|\Pr[G_i] - \Pr[G_{i+1}]| \leq \epsilon_i$.

1. **Mutual Authentication (MA):**

   Theorem 7.3 (MA security)**.** Assume SIS/ISIS is hard for QPT adversaries. Then for any QPT adversary $\mathcal{A}$ making at most q_{H_i} QRO queries and at most $q_{\text{reg}}, q_{\text{aka}}$ registration and AKA-session queries, there exists an algorithm $\mathcal{B}$ that solves SIS/ISIS with advantage at least

   $$\text{Adv}_{\mathcal{B}}^{\mathsf{SIS}} \geq \text{Adv}_{\mathcal{A}}^{\mathsf{MA}} - \Pr[\text{bad}] - \phi_{\text{prog}}(q_H, o),$$

   where $\Pr[\text{bad}]$ is the probability that $\mathcal{A}$ queries a programmed oracle point (bounded by $(q_H \cdot o)/2^m$ for an m-bit hash output under simple bounds).

   Proof. We give a game sequence below as:

   Game G_0: Real MA experiment. TAs, honest parties, and QROs as in the protocol. Let's $\Pr[G_0]$ denote the probability an honest party accepts without a matching honest partner (bad MA event).

   Game G_1: Replace each QRO H_i by the compressed-oracle simulator (Lemma 7.1). Since the simulator is perfect before any reprogramming, $|\Pr[G_1] - \Pr[G_0]| = 0$.

   Game G_2: The reduction $\mathcal{B}$ chooses an (honest) target registration tuple or oracle input (depending on the embedding strategy) and will attempt to embed an SIS/ISIS instance into the verification equations. For example, it $\mathcal{B}$ receives a random SIS matrix $M^\star$ or vector relation to be solved, sets the public matrix parameter $\mathcal{M} := \mathcal{M}^\star$ (or uses it $\mathcal{M}$ from the challenge), and simulates TA by answering registration queries using randomness consistent with the challenge (but leaving one key/signature algebraic term dependent on the SIS witness to be solved). The simulator programs at most o oracle points (for instance, specific inputs to be H_2 used in the forgery). Abort (the game outputs $\perp$) if $\mathcal{A}$ queries any of those programmed points before the adversary produces the MA-bad event. Let's $\Pr[\text{bad}]$ denote the abort probability. By union bound $\Pr[\text{bad}] \leq \frac{r \cdot q_H}{2^m}$ under simple random hashing

assumptions, in QROM use the exact ϕ-style bound you prefer. The transition yields

$$|\Pr[G_2] - \Pr[G_1]| \leq \Pr[\text{bad}].$$

Game G_3: Conditioned on not aborting, reprogram the compressed oracle at the chosen inputs to values that allow $\mathcal{B}$ extracting an SIS/ISIS witness from any MA forgery. By Lemma 7.1, programming o points yield at most $\phi_{\text{prog}}(q_H, o)$ a distinguishing advantage:

$$|\Pr[G_3] - \Pr[G_2]| \leq \phi_{\text{prog}}(q_H, o).$$

In G_3, any successful MA attack, yields algebraic relations that $\mathcal{B}$ can transform (via linear algebra and the equations used in verification, e.g., $\mathcal{M} \cdot \text{Sig} = \mathfrak{M}_{Pb} + \sigma \cdot AP$) into a non-trivial short vector satisfying an SIS/ISIS instance. Lemma 7.2 ensures it $\mathcal{B}$ extracts a solution with at least a non-negligible probability at least $\Pr[G_3]$.

Combining the games:

$$\Pr[G_0] \leq \Pr[G_3] + \Pr[\text{bad}] + \phi_{\text{prog}}(q_H, o).$$

Hence $\mathcal{B}$'s SIS/ISIS advantage is at least $\Pr[G_3]$, giving the claimed bound. □

2. **Session Key (SK):**

Theorem 7.4 (SK Indistinguishability in the QROM)**.** Let it $\mathcal{A}$ be any QPT adversary that interacts with honest parties under the protocol, makes at most q_{H_i} quantum queries to each hash oracle H_i (total q_H), at most q_{aka} authentication sessions, and at most q_{reg} registration queries. Suppose it $\mathcal{A}$ issues a single test query on an honest, uncorrupted session. Then there exists a QPT reduction $\mathcal{B}$ such that

$$\text{Adv}_{\mathcal{A}}^{\text{SK}} \leq 2 \cdot \text{Adv}_{\mathcal{B}}^{\text{SIS/ISIS}} + \Pr[\text{Bad}] + \phi_{\text{prog}}(q_H, o),$$

Where, Pr[Bad] bounds the probability that $\mathcal{A}$ queries any programmed input before the test session completes, and o is the number of programmed points (usually $o = 1$).

Proof. We reduce distinguishing the real session key from random to solving SIS/ISIS (or producing a forgery/short relation). As before, we give games G_0–G_5 and construct reduction $\mathcal{B}$.

Game G_0 **(Real experiment).** Run the real LBAKA protocol with honest parties and QROs. The adversary chooses a completed honest session sess^* and issues a Test query; the challenger returns the real SK computed as

$$\text{SK} = H_3[\tau_P, \tau_M, V_P, V_M, K],$$

where K is the algebraic value shared by the two parties. Let $\Pr[G_0]$ denote $\Pr[\mathcal{A}$ guesses $b]$ in this experiment.

Game G_1 (Compressed-oracle). Replace QROs by a compressed-oracle simulator $\mathcal{S}$ (Lemma 7.1). Again, $\Pr[G_1] = \Pr[G_0]$ (or differ by negligible simulator error).

Game G_2 (Choose Test session and pick a programming point). The reduction $\mathcal{B}$ waits until the Test session sess^* completes (or until the session state that defines SK is determined). Let x^* be the hash input that H_3 will be queried on to derive SK in this session; typically

$$x^* = (\tau_P^*, \tau_M^*, V_P^*, V_M^*, K^*),$$

so $o = 1$ (the point to program is the H_3 input for the test session). $\mathcal{B}$ marks x^* as the programmed point; if $\mathcal{A}$ queried x^* previously (Bad), the game aborts. Let $\Pr[\text{Bad}]$ be this abort probability. Then

$$\Pr[G_2] \geq \Pr[G_1] - \Pr[\text{Bad}].$$

Game G_3 (Reprogram H_3 at x^* and answer Test with random). Conditioned on no Bad, $\mathcal{B}$ reprograms the compressed oracle x^* so that it $H_3(x^*)$ is set to a uniformly random value $u^* \in \{0,1\}^m$. When $\mathcal{A}$ issuing the Test query, it $\mathcal{B}$ responds with u^* (the random key) instead of the real $H_3(x^*)$. By Lemma 7.1,

$$|\Pr[G_3] - \Pr[G_2]| \leq \phi_{\text{prog}}(q_H, o).$$

Thus, the adversary's distinguishing advantage between real and random is bounded by the difference between $\Pr[G_2]$ and $\Pr[G_3]$ plus ϕ_{prog}.

Game G_4 (Replace random by simulated key derived from SIS solution candidate). We now connect distinguishing G_3 to either (i) breaking authentication/forgery (which we reduce to SIS/ISIS) or (ii) solving SIS/ISIS directly. Intuitively, if $\mathcal{A}$ can detect that it u^* is random versus real SK, then either:

- $\mathcal{A}$ produced a forgery or a collision enabling extraction of a short relation (reduce to UF case), or
- $\mathcal{A}$'s distinguishing ability implies knowledge of algebraic relations among the values (τ_P^*, τ_M^*), (V_P^*, V_M^*) and (K^*) that permit extraction of an SIS/ISIS solution.

Concretely, it $\mathcal{B}$ proceeds as follows: it uses its SIS/ISIS challenger instance to instantiate some public parameters (as in the UF reduction) and simulates all sessions

except that when a signing-like primitive would be required, it $\mathcal{B}$ either simulates using programmable hashes or answers honestly when possible. If $\mathcal{A}$ distinguishes the real SK from random, then with non-negligible probability $\mathcal{A}$ either:

(a) outputs a transcript that violates authentication checks (i.e., a forgery) in some non-test session (this is handled by the UF reduction in Appendix 3), or

(b) produces additional algebraic information (for example, a second valid transcript for the same session with different ephemeral values) that yields a solvable linear relation in the lattice, which $\mathcal{B}$ converts to an SIS/ISIS witness using algebraic elimination (Lemma 7.2).

Thus, we can upper bound the distinguishing advantage by a combination of the forgery advantage (which itself is bounded by $\mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}_1}$ for an appropriate reduction $\mathcal{B}_1$) and the probability of extracting an SIS/ISIS solution directly (bounded by $\mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}_2}$). Combining them yields a reduction $\mathcal{B}$ with success probability at least half the distinguishing advantage of $\mathcal{A}$, i.e. a bound of the form

$$\Pr[G_3] - \frac{1}{2} \leq 2 \cdot \mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}}.$$

Game G_5 (Finalize and combine). Collecting the inequalities:

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{SK}}_{\mathcal{A}} &= \Pr[G_0] - \frac{1}{2} \\
&= \Pr[G_1] - \frac{1}{2} \\
&\leq \Pr[G_2] - \frac{1}{2} + \Pr[\mathrm{Bad}] \\
&\leq \Pr[G_3] - \frac{1}{2} + \Pr[\mathrm{Bad}] + \phi_{\mathrm{prog}}(q_H, o) \\
&\leq 2 \cdot \mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}} + \Pr[\mathrm{Bad}] + \phi_{\mathrm{prog}}(q_H, r).
\end{aligned}$$

This yields the theorem bound. □

Remarks.

- In many authenticators the SK derivation involves a single hash invocation $H_3(x^*)$ for x^* formed from session-specific values; programming a single point ($r = 1$) suffices and minimizes ϕ_{prog}.
- The concrete conversion of a distinguishing adversary into either (i) a forgery (handled by the UF reduction) or (ii) a direct SIS/ISIS witness depends on the algebraic structure K and the verification equations; the extractor uses two different transcripts (or a forgery transcript together with prior registration

data) to eliminate unknowns and produce a short lattice vector (formalized by Lemma 7.2).

- The abort probability Pr[Bad] is bounded by the probability that $\mathcal{A}$ queries the programmed input before it is programmed; with large hash ranges, this is negligible (the QROM compressed-oracle analysis provides the tight bound).

3. **Blind-Signature Unforgeability (UF)**

Theorem 7.5 (UF in the QROM)**.** Assume SIS/ISIS is hard for QPT adversaries. Let $\mathcal{A}$ be any QPT adversary that makes at most q_{bs} blind-signature oracle queries and at most q_{H_i} quantum queries to hash oracle H_i, with total q_H. Then there exists a QPT algorithm $\mathcal{B}$ such that

$$\text{Adv}_{\mathcal{A}}^{\text{UF}} \leq \text{Adv}_{\mathcal{B}}^{\text{SIS/ISIS}} + q_{\text{bs}} \cdot \phi_{\text{prog}}(q_H, o) + \Pr[\text{Bad}],$$

where, o is count of reduction reprogrammed($r = 1$ or 2) and there Pr[Bad] is the probability that $\mathcal{A}$ queries a programmed point prior to programming.

Proof. We give a game sequence $G_0, \ldots, G_4$ and explicitly construct a reduction $\mathcal{B}$ that uses an $\mathcal{A}$ that wins G_0 with non-negligible probability to solve SIS/ISIS.

Game G_0 (Real experiment). Run the real blind-signature protocol (BSBCA) and give $\mathcal{A}$ oracle access to:

$$\mathcal{O}_{\text{BS}}(\cdot) \quad \text{(blind-sign oracle)}, \qquad H_1, H_2, H_3 \text{ (QROs)}.$$

Let $\Pr[G_0]$ be the probability that it $\mathcal{A}$ outputs a valid forgery $(\mathsf{msg}^*, \vartheta^*)$ such that msg^* was not the unblinded result of any prior $\mathcal{O}_{\text{BS}}$ query.

Game G_1 (Compressed-oracle). Replace each QRO H_i by the compressed-oracle simulator $\mathcal{S}$ (Lemma 7.1). The simulator answers quantum queries faithfully and stores a compressed record $\mathcal{R}$. Since $\mathcal{S}$ is a correct simulation,

$$\Pr[G_1] = \Pr[G_0].$$

(If your chosen simulator is not exact, then record the small distinguishing error here as ϵ_1.)

Game G_2 (Pick programming set and abort on bad). Reduction $\mathcal{B}$ (to be described) chooses a small set $\mathcal{P}$ of oracle input points that it will later reprogram to embed the SIS/ISIS challenge. Concretely, it $\mathcal{B}$ chooses indices or hash inputs that appear in the signing/verification algebra (for example, inputs to H_2 or H_3

that tie α, θ, v to algebraic values). The number of points is $r = |\mathcal{P}|$ (we aim for r small; typically $r = 1$). If $\mathcal{A}$ ever queries any $x \in \mathcal{P}$ before $\mathcal{B}$ programs it, the game aborts (Bad). Let Pr[Bad] = Pr[Bad occurs in G_2] Then,

$$\Pr[G_2] \geq \Pr[G_1] - \Pr[\text{Bad}].$$

Game G_3 (Reprogram $\mathcal{P}$). Conditioned on no Bad, $\mathcal{B}$ reprograms the compressed-oracle outputs on $\mathcal{P}$ to carefully chosen values that allow the reduction to map any eventual forgery to an SIS/ISIS solution. By Lemma 7.1,

$$|\Pr[G_3] - \Pr[G_2]| \leq \phi_{\text{prog}}(q_H, r).$$

Game G_4 (Extraction / reduction). Suppose $\mathcal{A}$ outputs $(\mathsf{msg}^*, \sigma^*)$ in G_3 that verify under the public key of the signer and where msg^* was not the result of an earlier blind-sign query. The reduction $\mathcal{B}$ uses the compressed record $\mathcal{R}$ and the algebraic verification equations to compute an object that solves the SIS/ISIS instance that $\mathcal{B}$ was handed by its challenger. We now describe $\mathcal{B}$ concretely.

Reduction $\mathcal{B}$ (construction). $\mathcal{B}$ is given an SIS/ISIS instance by its challenger (public matrix $\mathcal{P}$ or $\mathcal{M}$ and possibly target vector, according to the SIS/ISIS flavor used in setup). $\mathcal{B}$ will act as the challenger for $\mathcal{A}$:

(a) **Simulate Setup.** Use the SIS/ISIS instance to set the public parameters (e.g., set the TA public matrix $\mathcal{P}$ or $\mathcal{M}$ equal to the challenge matrix, and set related public keys so algebraic verifications will contain the unknown challenge vector).

(b) **Answer registration and blind-sign queries.** For honest registration requests and blind-sign queries, it $\mathcal{B}$ uses its own randomness and the compressed oracle $\mathcal{S}$ to produce protocol transcripts consistent with the public parameters. When a blind-sign query requires the signer to compute some secret-dependent value, $\mathcal{B}$, either:

- uses the challenger-provided structure to respond directly (if possible), or
- uses simulated signing (via reprogramming/ programmable hash values) in such a way that later a forgery will yield a relation to be solved.

(c) **Upon forgery.** When $\mathcal{A}$ outputs $(\mathsf{msg}^*, \varrho^*)$ that verify under the scheme's verification equations, use the compressed record $\mathcal{R}$ to recover the hash preimages and associated randomness involved in the forging transcript. Plug those values into the algebraic verification equation (which by design contains the SIS/ISIS challenge matrix or vector) to compute a short vector that satisfies the SIS/ISIS relation.

By design of the programming step in G_3, and by the verification equation satisfied by the forgery, the computed vector is a valid SIS/ISIS solution with non-negligible probability (formalized using the algebraic structure of the scheme and Lemma 7.2). Thus,

$$\Pr[G_4] \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS/ISIS}}.$$

Combine bounds. From the inequalities above:

$$\begin{aligned}\Pr[G_0] &= \Pr[G_1] \\ &\leq \Pr[G_2] + \Pr[\mathrm{Bad}] \\ &\leq \Pr[G_3] + \Pr[\mathrm{Bad}] + \phi_{\mathrm{prog}}(q_H, o) \\ &\leq \Pr[G_4] + \Pr[\mathrm{Bad}] + \phi_{\mathrm{prog}}(q_H, o) \\ &\leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS/ISIS}} + \Pr[\mathrm{Bad}] + \phi_{\mathrm{prog}}(q_H, o).\end{aligned}$$

Recalling $\Pr[G_0] = \mathrm{Adv}_{\mathcal{A}}^{\mathrm{UF}}$ completes the reduction and proves the stated theorem. □

4. **Key-Compromise Impersonation (KCI) and Unknown-Key-Share (UKS)**

Theorem 7.6 (KCI/UKS resistance)**.** Assuming SIS/ISIS hardness, the protocol resists KCI and UKS attacks: for any QPT adversary $\mathcal{A}$ corrupting at most one long-term key (but not TA), the probability of successful impersonation (KCI) or creating an unknown-key-share condition is bounded by

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{KCI/UKS}} \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS}} + \Pr[\mathrm{bad}] + \phi_{\mathrm{prog}}(q_H, o).$$

Proof. KCI: Suppose the $\mathcal{A}$ learns the long-term key of party P and attempts to impersonate an honest peer M to some honest P' (or vice versa). The reduction simulates corruptions; it uses a compressed-oracle simulation and embeds an SIS/ISIS challenge in the verification/signature equations of the impersonated peer. As with MA, any successful impersonation yields algebraic relations allowing the extraction of the SIS/ISIS witness. Abort events and reprogramming costs are bounded as before.

UKS: A UKS attack creates a situation where two honest parties derive the same SK but disagree about the partner identity. Any such attack implies the adversary either forged a signature or caused a verification relation to hold without proper registration, again reducible to SIS/ISIS or signature forgery. The same game sequence and bounds apply. □

5. **Forward Secrecy (FS)**

Theorem 7.7 (Forward secrecy)**.** Under SIS/ISIS hardness and assuming ephemeral secrets are erased after a session, compromise of long-term keys (after session completion) does not allow a QPT adversary to recover past session keys with probability better than

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{FS}} \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{SIS}} + \Pr[\mathrm{bad}] + \phi_{\mathrm{prog}}(q_H, o).$$

Proof. FS follows from the fact that session keys depend on ephemeral secrets r, s and algebraic combinations that produce SIS/ISIS-hard relations. The reduction proceeds by embedding SIS instances into either parties' ephemeral-related equations and by arguing that recovering the session key implies solving SIS/ISIS. Use compressed-oracle simulation and bound abort/reprogramming costs as before. □

6. **Blindness**

Theorem 7.8 (Blindness)**.** Let it $\mathcal{A}$ be any QPT signer that follows the protocol but attempts to link a returned unblinded signature to the blinded signing session. Then:

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Blind}} \leq \Pr[\mathrm{bad}] + \phi_{\mathrm{prog}}(q_H, o),$$

i.e., the signer's distinguishing advantage is negligible up to QROM reprogramming and abort probabilities.

Proof. Construct a simulator that, given two target messages m_0, m_1, simulates the blind-signing interaction with the signer by generating random oracle answers (via compressed oracle) and random-looking blind-signature transcripts consistent with either message. If the adversary (signer) distinguishes which message was actually signed, we show that either:

- The adversary must have queried a programmed oracle point used to link the transcript (bad event), or
- The signer can distinguish because of different oracle outputs on inputs that we could have reprogrammed, which by Lemma **??** changes advantage by at most $\phi_{\mathrm{prog}}(q_H, o)$.

Hence, the advantage is bounded as claimed. □

7. ***Final combined security bound*** Combining all proofs above and summing abort and reprogramming losses, we may state the following high-level theorem:

Theorem 7.9 (Combined security (informal))**.** Let $\mathcal{A}$ be a QPT adversary with query bounds $(q_H, q_{\mathrm{reg}}, q_{\mathrm{aka}})$ $(q_{\mathrm{bs}}, q_{\mathrm{cor}})$. Then for each security property :
$X \in \{\mathrm{MA}, \mathrm{SK}, \mathrm{Blind}, \mathrm{KCI/UKS}, \mathrm{FS}, \mathrm{UF}\}$

$$\mathrm{Adv}^{X}_{\mathcal{A}} \leq \mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}_X} + \epsilon_X(q_H, q_{\mathrm{reg}}, q_{\mathrm{aka}}, q_{\mathrm{bs}}) + \phi_{\mathrm{prog}}(q_H, o_X)$$

where $\mathrm{Adv}^{\mathrm{SIS/ISIS}}_{\mathcal{B}_X}$: is the advantage of a reduction $\mathcal{B}_X$ against SIS/ISIS,
$\epsilon_X(\cdot)$: collects abort / bad-event probabilities (bounded by simple collision arguments or can be taken from compressed-oracle analyses), and
o_X: is the number of points reprogrammed for property X.

Remarks.

- For concrete parameter estimation, plug in the actual q_H (e.g., an upper bound like $q_H \leq 2^{20}$ or $\mathsf{poly}(\vartheta)$) and compute numerical upper bounds for ϵ_X and ϕ_{prog}.
- All reductions preserve polynomial-time complexity: $\mathcal{B}_X$ runs in expected polynomial time given $\mathcal{A}$'s resources.

This completes the QROM game-based security proofs for all six required properties of the protocol.

7.4.3 Informal Security Analysis

Here, the security goals defined in section 7.2.4 have been discussed. Moreover, the security features comparison of the proposed scheme has been detailed with the existing schemes in Table 7.5.

1. **Man-in-the-Middle Attack:** In our scheme, nodes P share calculated values: $\langle \tau_P, HID_P, A_P, V_P, t_1$ with the server node M via the internet. There is a possibility of $\mathcal{A}$ playing as an MITM attacker. For this, it $\mathcal{A}$ needs to calculate $\tau^*_P = a^T \cdot \mathcal{M}$ and $V^*_P = \mathcal{M} \cdot Sign^*_P.a^T$. Now $\mathcal{A}$ share variables: $\langle \tau^*_P, V^*_P, Sig_p, A_P, t1, HID_{\mathcal{A}} \rangle$ with M. Now, the medical server validates the received signature as: $V^*_P \cdot \mathcal{M} = H_2[HID_{\mathcal{A}}, A_P, \mathfrak{M}_{Pb}]$. This returns false; an authentication failed alert is received. Similarly, when the medical server shares the tuple. The patient also needs to verify the authenticity of the responder unit. Hence, we can state that our scheme is resistant to the MITM attack during the mutual authentication and session key generation process.

2. **Known Provisionally Information Attack:** In the proposed scheme, the session key is **SK** calculated as: $\mathbf{SK} = H_3[\tau_P, \tau_M, V_P, V_M, K]$. Even though an adversary retrieved the information regarding the randomly selected ephemeral key: *s and r* belongs to the current session, it is impossible to achieve knowledge about the session key. The adversary $\mathcal{A}$ must calculate the variables: (τ_P, τ_M) and (V_P, V_M). And due to hard challenges problems SIS and ISIS, the computation of value $(\mathcal{M} \cdot Sig_P \cdot r^T, \mathcal{M} \cdot Sig_M \cdot s^T)$ is a complex task and cannot be performed in

Table 7.5: Comparison of Existing Competing Schemes Based on Security Features Requirement.

Silent Features	[177]	[178]	[179]	[180]	[181]	[182]	[183]	[184]	[185]	[186]	[187]	Our
Mutual Authentication	√	√	√	√	√	√	√	√	√	√	√	√
Man-In-The-Middle Attacks	√	√	√	√	√	√	√	√	√	×	√	√
Anonymity	√	√	×	√	√	√	√	√	√	√	×	√
Secure Session Key	√	√	√	√	√	√	√	√	√	√	√	√
Unlinkability	×	√	√	×	√	×	⊗	⊗	⊗	⊗	⊗	√
Verifiability	×	√	√	×	√	×	√	√	√	⊗	√	√
Consensus Mechanisms	×	×	×	×	×	×	×	×	×	√	×	√
Post-Quantum Scheme	×	×	×	×	×	×	×	×	√	×	×	√
Impersonation Attacks	√	√	√	√	√	√	√	√	√	⊗	√	√
Perfect Forward	√	√	√	√	√	√	√	⊗	⊗	√	√	√
Known Session Key	√	√	√	√	√	√	√	√	√	√	⊗	√
Replay Attacks	√	√	⊗	⊗	√	√	√	√	⊗	⊗	⊗	√
Unforgeability	√	√	×	√	√	√	×	×	×	√	⊗	√
Immutability	×	×	×	×	×	×	×	×	×	√	×	√
Decentralized System	×	×	×	×	×	×	×	×	×	√	×	√

√: Achieved the security silent features.
×: Failed against respective security feature

⊗: Not focused, and no information available with respective silent features.

polynomial time. As a result, our proposed LBAKA scheme is resistant against the known provisional information attacks.

3. **Impersonation Attacks:** As we formally proved using the QROM model, our scheme is a perfectly mutually authenticated protocol. And also, we proved that it is resistant to the MITM attacks. Thus, we can easily depict that it is also resistant to impersonation attacks. For example, whenever an adversary $\mathcal{A}$ wants to impersonate the patient or medical server node. $\mathcal{A}$ forwards the tuple based on its own key information along with the hashed id of the patient or server as: $\langle HID_P, \tau_A, A_A, V_A, t_1^{'} \rangle$ towards the medical server. During the verification process medical server received false value as $V_A \cdot \mathcal{M} \neq [H_2[HID_P, A_A, \mathfrak{M}_{Pb}] \cdot A_A]$. Similarly, when $\mathcal{A}$ wants to impersonate the medical server and sends the details to the patient. The patient also failed to verify the authenticity of the sender. In this way, our proposed scheme fully works against the impersonation attack.

4. **Secure Session Key:** As we know, the computation of the session starts only after the mutual authentication process is completed successfully. The session key $SK$is directly dependent on the variables: $\langle \tau_P, \tau_m, V_P, V_M, K \rangle$, where $\tau_P = r^T \cdot \mathcal{M}, V_P = \mathcal{M} \cdot Sig_P \cdot r^T, \tau_M = s^T \cdot \mathcal{M}, V_M = \mathcal{M} \cdot Sig_M \cdot s^T$ and $K = (s^T \cdot Sig_M)\mathcal{M}\mathcal{M}(r^T \cdot Sig_P)$. Retrieve the value of the secret keys r, s from the expression that comes under the hard assumptions of SIS and ISIS. and the final computation depends on the defined hash function: $H_3[\tau_P||\tau_M||V_P||V_M||K]$, and due to the collision and pre-image property of the hash, $\mathcal{A}$never computes the value of the output of the defined

hashed function. Hence, our proposed scheme maintains the privacy of the session key.

5. **Perfect Forward Secrecy:** As per its definition, $\mathcal{A}$ cannot retrieve any information regarding its previous session key even if units share their present value of randomly selected vectors r and s. Despite having information about r, s $\mathcal{A}$ failing to compute the previous session key due to the presence of a timestamp, and most importantly, for each session, there is a signature Sin_P, Sig_M that depends on the secret vector a, b chosen by the trusted registration authority RA. For RA, we already assume that it is a trusted party, and if it is compromised, then the whole system will be compromised. Computation of r, s from variable $V_P = \mathcal{M} \cdot Sig_P \cdot r^T$, and $V_M = \mathcal{M} \cdot Sig_M \cdot s^T$ is a challenging task, which comes under the SIS and ISIS hard assumption.

6. **Resist Replay Attack:** Assume that somehow, an adversary $\mathcal{A}$ get message about its authentication done successfully for any past session. Now, an adversary $\mathcal{A}$ wants to use this information for the current session to gain knowledge of the present session key. During the mutual authentication phase, we share the Timestamps t_1, t_2, thus, authentication verification failed during the verification of the legitimacy of the received timestamps. Hence, our proposed scheme is fully safe from replay attacks. of the authentication message proposed scheme.

7. **Resist the Quantum Attack:** Due to the advent of quantum computers, lots of conventional cryptosystems have failed to resist quantum attacks. Hence, now researchers are moving towards the post-quantum methodology. Lattice-based cryptography is one post-quantum mechanism. As we proposed our authenticated key agreement protocol using LB-based, it will easily resist all the quantum attacks. We defined the SIS and ISIS hard assumptions and also bound the polynomial time against the quantum attacks.

8. **Conditional Privacy Preserving**: In our proposed scheme, the unit patient and the medical server both request the RA with their original identity and the hashed id HID_i. Nodes, whether they are P or MS, calculate their key $K_1 and K_2$ at their respective end by utilizing HID_M, HID_P only. That means in the whole communication, the patient and medical server always use their hashed identity rather than disclose their real identity. Except for RA, no one has any idea about the real identity of the communicating nodes in the protocol. If any dispute occurs, only RA can identify the real identity from its database. In this way, our proposed protocol successfully maintains the property of conditional privacy preserving.

9. **Anonymity**: An adversary $\mathcal{AS}$ or other communicating nodes do not have any information about the sender or receiver nodes, as they always communicate with

their hashed identity HID_i. RA generates $\sigma_P or \sigma_M$ using their hashed identity. That means if any adversary can gain information about σ, then the adversary $\mathcal{A}$ is still unable to get the real identity of the communicating nodes. This way, the anonymity of information is preserved.

10. **Decentralization**: In our proposed protocol, we utilized the concept of blockchain to upload and access the information from the server. Here, the hospital professional team, the collecting node, the government authority, etc., all work as nodes in the blockchain architecture. In this way, every unit has the same power. Thus, the single-point failure issue will be resolved successfully in our proposed scheme.

11. **Data Provenance**: In our proposed scheme, all the activities are performed and saved in the form of blocks (a set of transactions) in the blockchain. As per the blocks' characteristics, it provides and maintains immutability, accountability, auditability, and robust trustworthiness properties in a more precise and secure way. Thus, third parties, including nodes belonging to the system, don't gain knowledge about who, when, where, or what the information is.

12. **Unlinkability**: An adversary $\mathcal{A}$ could not generate the same signature generated by RA for P or the medical server MS as: $Sig_p = x + \sigma_P \cdot a$ and $Sig_M = x + \sigma_M \cdot b$. Also, the adversary failed to compute the value of τ_i and V_i for any patient or medical server node because these variables are directly dependent on the secret vector $r, s \in Z_q^n$ and indirectly dependent on the secret vector selected by RA for P and MS: $a, b \in Z_q^n$. For each session, these values are randomly selected; thus, the adversary $\mathfrak{A}$ is never able to create any link between any past and present signature values. Hence, our scheme supports the unlinkability feature successfully.

7.5 Performance Analysis

Inside this section, we are going to analyze our scheme against existing authenticated key agreement protocols for smart healthcare systems. We will present the storage, computation, and communication cost overheads. For our protocol, we define $n = O(t \log q)$, where $q = O(t^2)$, which ensures the hard security of SIS/ISIS challenges against the security issues. As $m = n$, we analyzed the performance of our protocol in a simplified way, we defined $n = t \log q = m$ and $q = t^2$. As we analyzed all operations and found that the execution costs of the hash functions are very small, we neglected them during the comparison process.

7.5.1 Storage Overheads

TA needs to store the master key $x \in Z_q^n$ and matrix $\mathcal{M} \in Z_q^{n \times n}$. during the initialization phase of system setup. Thus, the total storage cost of x and $\mathcal{M}$ is $n(1+n)|q| = 2t \log t(1 + 2t \log t)(2 \log t)$. which is equivalent to $4t \log^2 t + 8t^2 \log^3 t$. Similarly, for the scheme [32] we compute storage cost as:$n(1 + m)|q| = (2t \log t)(1 + 2t \log t)(2 \log t) = 4t \log^2 t + 8t^2 \log^3 t$. The same procedure is used for other competing schemes to compute the storage cost. From Table 7.6, we can see how our proposed LBAKA scheme is better than other competing schemes in terms of storage costs for IoT-based applications SHS.

7.5.2 Communications and Computation Cost

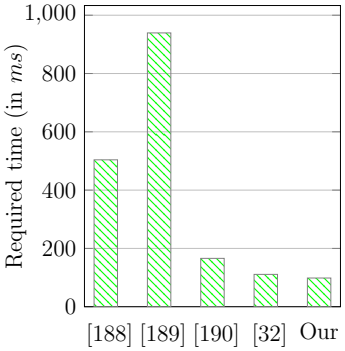
In the proposed scheme, mainly communication is performed between the patient (HIA data) and the medical server (N1). The first patient sends the tuples $\langle \tau_P, A_P, V_P \rangle$ towards the M for the generation of the **SK**. The computation cost is $= (1 + n + n^2)|q| = (1 + 2t \log t + 4t^2 \log^2 t)(2 \log t) == (2 \log t + 4t \log^t k + 8t^2 \log^3 t)$.

We can determine the computation cost of the proposed protocol as parameters: $(Sig, A) \in (Z_q^{n \times 1})$. Thus, the total computation cost of the proposed scheme is $= 3n|q| = 3(2t \log t)(2 \log t) = 12t \log^2 t$. Other schemes' computation cost is also determined by using similar assumptions and values. Like Gupta et al.'s need $m(2n + 1)|q^2|$ and other competing schemes required more cost than our proposed scheme.

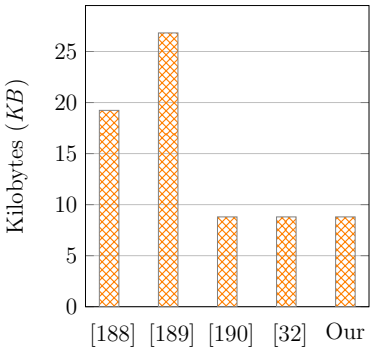
Table 7.6 represents the comparison of our proposed scheme with other competing works based on LB cryptography for IoT-based smart healthcare systems. In this comparison, we mainly focused on parameters: storage, communication, and computation as well. With the length estimation, we can see how our scheme outperformed the schemes mentioned in the table. Figure 7.4 has subfigure 7.4a, which presents the graphical representation of the total computation cost of the competing scheme with our scheme. Subfigures 7.4b, 7.4c, and 7.4d represent the storage, energy, and communication overheads of our scheme compared to others' schemes. These subfigures illustrate how our proposed scheme outperforms the existing scheme based on LB cryptography in IoT environments. We also perform comparisons based on the computation cost taken by the phases: setup, registration, and mutual authentication. By subfigure 7.4e, we can see that approximately all schemes have the same setup cost but differ during the registration process and the mutual authentication process. As a result, we found that our proposed scheme is efficient in terms of comparison parameters: storage, communication, computation, and energy overheads.

7.5.3 Orchestration and Simulation Setup

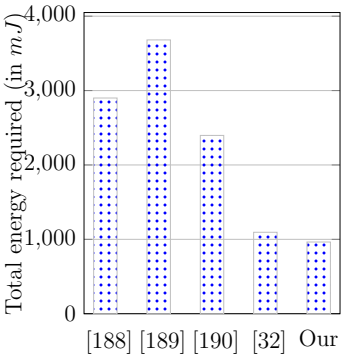
Under this section, we describe the hardware descriptions of the communication units, patient, and medical server used in our proposed mechanism. In our proposed scheme,



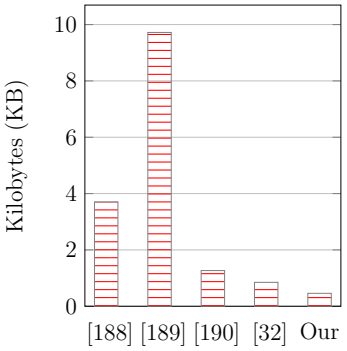
(a) Computation Overheads



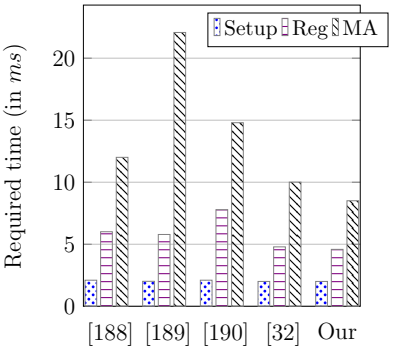
(b) Storage Overheads



(c) Energy overhead



(d) Communication Cost



(e) Computational Costs.

Operat	TA	P	MS
$T_{\mathcal{H}}$	.0953	.1015	.0854
T_{Rnd}	.0151	.9711	.0280
$T_{\mathcal{M}}$	.1081	.1317	.9710
T_x	.0301	.1097	.0001
$T_{\boldsymbol{M}x}$	.1734	.2401	.2515
$T_{x^T\mathcal{M}}$	.9002	.8912	.9106
$T_{\boldsymbol{x}^T}$	.0174	.0915	.0417
$T_{\boldsymbol{M}^T}$	.5902	.9083	.9417

(f) Execution Costs (in ms)

Notations: TA: Trusted Authority; P: At Patient ends HIA; MS: Medical Server; $E_{\mathcal{H}}$: Hash Digest; E_{Rnd}: random vector; $E_{\mathcal{M}}$: Matrix from $Z_q^{n\times n}$; E_x: Vector from Z_q^n; $E_{\mathcal{M}x}$: Product of matrix $\mathcal{M}$ and x, forming a vector $Z_q^{n\times 1}$; $E_{x^T\mathcal{M}}$: Product of transpose vector and matrix; E_x: vector $x \in Z_q^n$; E_{x^T}: Transpose of x.

Figure 7.4: Performance Comparisons of Competing LBC-Based Security Schemes.

Table 7.6: Comparison of Competing LBC-based AKA Schemes based on Storage, Communication, and Computation costs.

Schemes	**Type**	**Fundamentals**	**Length**
[188]	Storage	$\langle R_p, R_q\rangle \in \mathfrak{Z}_q^{n\times n}$, $\langle d, p, e, s, u, v\rangle \in \mathcal{Z}_q^n$	$(6n+2n^2)\|q\| \approx 24t\log^2 t + 16t^2\log^3 t$
	Communication	$\langle \sigma, \mathcal{M}\rangle$, $\langle \sigma_G, M_G\rangle$ $\langle \sigma_{amf}, M_{amf}\rangle$	$(9n+1)\|q\| \approx 36t\log^2 t + 2\log t$
	Computation	$\langle d, p, c, t, z\rangle$	$88t^2\log^4 t$
[189]	Storage	$\langle R, R_a, R_b\rangle \in \mathbb{Z}_q^{n\times n}$, $g \in Z_{q_1}^*$, and $\langle k_x, l_x, m_x, d_x\rangle \in \mathbb{Z}_q^n$	$(4n + 3n^2)\|q\|+\|q_1\| \approx 16t\log^2 t + 24t^2\log^3 t + \log q_1$
	Communication	e_x, $\langle k_x, k_{a_x}\rangle$, m_x, $\langle A_x, B_x, ID\rangle$, $\langle C_y, D_y, ID, v\rangle$	$(n^2 + 3n + 1)\|q\|+4\|q_1\| \approx 8t^2\log^3 t + 12t\log^2 t + 2\log t + 4\log q_1$
	Computation	$\langle e_x, c_x, t_x, m_x, A_x\rangle$	$16t^2\log^4 t + 8K_1^3$
[190]	Storage	$d \in Z_q^n$, $\mathbf{A} \in Z_q^{m\times n}$	$(n+mn)\|q\| \approx 4t\log^2 t = 8t^2\log^3 t$
	Communication	$\langle \mathcal{M}, ANS, \mathcal{R}, \mathcal{S}\rangle$	$(3n+2)\|q\| \approx 12t\log^2 t + 4logt$
	Computation	$\langle PUB,\ sk,\ R,\ S, ANS_1\rangle$	$48t^2\log^4 t + 8t\log^4 t$
[32]	Storage	$d \in \mathbb{Z}_q^n$, $\mathcal{X} \in \mathbb{Z}_q^{m\times n}$	$(n+mn)\|q\| \approx 4t\log^2 t + 8t^2\log^3 t$
	Communication	$\langle \mathfrak{t}_i, \phi_i\rangle$, R_i $\langle ID_i, b_i\rangle$, $\langle \pi_i, ID_i\rangle$, C_{ji}	$(n+7)\|q\| \approx 4t\log^t + 14\log t$
	Computation	$\langle PU, t_i, v_i\rangle$	$32t^2\log^4 t + 8t\log^3 t$
Our	Storage	$x \in Z_q^n, \mathcal{M} \in Z_q^{n\times n}$	$(n+n^2)\|q\| \approx 4t\log^2 t + 8t^2\log^3 t$
	Communication	$\tau_i \in Z_q^{1\times n}$, $V_i \in Z_q^{n\times n}$ and $A_i \in Z_q^n$	$(n+n^2+1)\|q\| \approx 4t\log^2 t + 8t^2\log^3 t + 2\log t$
	Computation	$\langle A_i, Sign_i\rangle$	$12t\log^2 t$

the Trusted Authority TA broadcasts the system parameters. Next, the TA registered P and MS with their respective requesting information (HID_i, ID_i). We used a high-performance computing server for the TA, who is responsible for the registration process, which helps to authenticate the units further. HPC with @ AMD EPYC 9654 with 2, 4 GHz base clock, which has 512 GB RAM per node, for secondary storage, 2X NVMe SSD (each node 4 TB). For interconnection, it features a 200G HDR InfiniBand network, thus facilitating low-latency, high-throughput node communication. HPC has Rocky Linux 9 as the OS. All the patient's end functionalities (HIA collects the sensor information implanted over or in the patient's body) are simulated on the IoT device Raspberry Pi 5 with the features of a BCM2712 Quad-Core 64-bit Arm Cortex-A76 Processor running at 2.4 GHz with cryptography extensions and support for SDR104 mode for faster read-write operations.

Regarding estimating goals, the total execution overhead of the different cryptography operations has been determined as the average of hundreds of executions for $m = n = 128$. The communication costs have been calculated by considering the length of different factors, including hash $|H|$= 64 bytes, number $|Z_q^*|$= 16 bytes, identity $|ID|$= 16 bytes, timestamps, and $|T_i|$= 4 bytes.

Below the table 7.7 explained how our proposed protocol handles scalability based on parameters: computation overheads, quantum complexity, identity, and authentication mechanism.

Table 7.7: Scalability Approaches for Large-Scale IoT Deployments

Scalability Aspect	**Our Approach**
Communication Overhead	We use a **distributed security architecture** with edge/fog computing nodes to localize authentication, reducing traffic to any central server and avoiding bottlenecks.
Consensus Complexity	Our optimized consensus mechanism reduces communication complexity from $\mathcal{O}(N^2)$ (standard PBFT) toward $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$, enabling fast consensus even with 1000+ nodes and sub-second latency.
Computational Load on Devices	We employ **lattice-based cryptography** optimized for resource-constrained IoT devices. This involves simple hash operations and matrix–vector multiplications with negligible decryption overhead, making it suitable for low-power sensors.
Identity and Authentication Management	A **blockchain-based decentralized identity system** eliminates single points of failure while supporting scalable, verifiable authentication for millions of devices without centralized credential stores.
Dynamic Security Adaptation	The system uses **context-aware security policies** that adjust cryptographic parameters and verification depth based on network size, device capabilities, and threat level, balancing security with performance.

We understand that energy consumption is a significant issue, particularly when it comes to smart sensors. Because of this, we need to strive seriously to cut down on the storage and energy overhead of our scheme as much as we can. The equation $Energy = Power * Time$ is being used to figure out how much energy there is. Time is recorded in milliseconds (ms), power is expressed in watts (W), and energy is specified in millijoules (mJ). For our suggested method, the patient sensors need 1.00 mJ, the TA needs 20.36 mJ, and the medical server needs 6.0045 mJ. This is because they are all involved in distinct phases, such as setup, registration, mutual authentication, and session key.

Subfigure 7.4c, belongs to Figure 7.4, which represents the energy overheads of our proposed scheme compared to other schemes used. Whereas subfigure 7.4f represents the execution cost of all matrix-based operations used in others' schemes and our proposed protocol.

PyCryptodome Library: It is a distinct Python module that has a set of basic cryptographic design blocks. It works with PyPy and Python versions 2.7, 3.5, and beyond. PyCryptodome is a revised version of PyCrypto that provides more capabilities and is

easier to maintain. It may be used to encrypt, hash, and execute other cryptographic tasks in Python scripts.

Hashlib Library: This module facilitates a feasible way to utilize a lot of well-known and safe hashing strategies in Python code. It has well-known algorithms like MD5 and the SHA family (SHA1, SHA224, SHA256, SHA384, SHA512) that are often used to make distinct digital patterns of information. Hashlib makes it simple to build and use cryptographic hash functions whenever it's necessary to validate the authenticity of data or protect login credentials confidentially.

7.6 Summary

Security features such as confidentiality and authenticity are major requirements in SHS infrastructure due to the sensitivity of patient health data. Therefore, authors in the literature have proposed various security schemes to achieve confidentiality and authentication using conventional cryptography mechanisms. We examined a different design direction for securing smart healthcare environments. Lattice-based cryptographic techniques were adopted to address concerns related to quantum-enabled attacks and long-term data protection. A decentralized blockchain framework has been designed to lessen reliance on central management and lessen the effects of single points of failure. A lattice-based blind-signature-based approach was added to ensure anonymity and check the authenticity of nodes throughout blockchain transactions. This way, people can participate without having to reveal their identities. Further, the formal and informal security analysis section proves that the proposed scheme strongly resists various attacks and attackers. Furthermore, the result analysis and comparative study prove the computational efficiency of the proposed scheme.

CHAPTER 8

Conclusion and Future Directions

8.1 Summary of Research Works

The evolution of Internet of Things (IoT) technology reshaped network-based applications by enabling device-to-device communication, mini-hardware development, and micro-computing. IoT is a network of devices formed by sensors, actuators, edge devices, and communication technology. IoT devices have computing and communication capabilities that facilitate data processing and exchange with other entities. Hence, the Internet of Things technology has been applied to a large number of applications such as smart grid, smart home, industries, connected vehicles, etc. Despite the advantages of IoT applications, they face many issues related to security and efficiency. The primary objective of this thesis is to address the primary security issues of the IoT applications, such as authentication, integrity, and confidentiality. Moreover, this thesis also covers the advantages of blockchain integration in IoT applications.

This thesis presents a number of approaches to solve the security and efficiency issues of IoT applications. The chapter-wise conclusions of this thesis are summarized below.

Chapter 1 illustrates the basic introduction to IoT and describes various IoT-based applications. This chapter also explains the need for security in IoT applications and the advantages of blockchain integration in IoT. Furthermore, this chapter illustrates the research gap related to security in IoT applications and defines the thesis objectives. Next, the overview of the research contributions related to IoT applications, such as Intelligent Healthcare Systems, Intelligent Transportation Systems, and WBAN have been highlighted.

Chapter 2 discusses the detailed literature related to various cryptographic techniques and schemes in IoT applications. The literature regarding PKI, IBC, CLS, and LB-based

schemes has been discussed and analyzed in this chapter. Moreover, this chapter gives a detailed description and comparative analysis of the existing works regarding cryptographic security schemes for various IoT applications, such as Intelligent Transportation Systems and the Intelligent Healthcare System. Initially, the existing works related to pairing-free ID-based mutual authentication and key agreement schemes, as well as certificateless-based mutually authenticated key agreement schemes, have been discussed in the context of IHS. Next, the literature related to security schemes for secure V2V communications in Intelligent Transportation Systems has been described using LB-based and integrating blockchain. Finally, a detailed review of the existing security schemes, mainly security schemes based on mutual authenticated key agreement, by applying the LB-based cryptography. Additionally, we also introduce a blind signature scheme to check the authenticity of nodes in blocks, which has been discussed and analyzed for the smart healthcare system.

Chapter 3 has highlighted some preliminary knowledge. This chapter discusses the operations of elliptic curve cryptography, identity-based, and lattice-based cryptography. Further, we also highlight their computationally hard problems, such as ECDLP, ECDHP, ECDDHP, and SIS/ISIS. Next, define the one-way hash function and its resistance property. Furthermore, we highlight the importance and working architecture of blockchain, followed by the consensus mechanism used in it. At the end of this chapter, the basic components of Hyperledger Fabric and the transaction processing life cycle have been described in detail.

Following the introductory chapter, literature survey, and background knowledge covered in the initial three chapters, various cryptographic security schemes for Intelligent Healthcare Systems, Wireless Body Area Networks, and Intelligent Transportation Systems are presented in subsequent chapters. In Chapter 4, pairing-free methods are used with identity-based cryptography to resolve the problem associated with the PKI certificate overhead and high computational cost due to bilinear pairing operations for the intelligent healthcare system environment. Furthermore, two party mutual authentication key agreement protocol has been proposed that establishes a common shared key between both nodes. Next, we integrate the concept of aggregation to aggregate the received patient information before sending it to the medical server. For the encryption, we use homomorphic encryption methods to reduce the costs and improve the efficiency. The proposed authentication and key agreement scheme is proven secure using ROM model-based formal proof and informal security analysis.

In Chapter 5, we perform the cryptanalysis over the scheme proposed by the author Cheng et al. and we found numerous design flaws along with possibilities of security attacks such as man-in-the-middle attacks and impersonation. We introduce a certificateless-based secure mutually authenticated key agreement scheme for the wireless body area network, which eliminates the issue of certificate overhead of PKI. Further, after performing the cryptanalysis, we found that our proposed improved version performed and withstood the security attacks successfully. We verify the security analysis of our proposed scheme using

the proposed ROM model and also highlight the informal security analysis.

Chapter 6 introduces an approach to handling the security issues of V2V communications in intelligent transportation systems. This study successfully addresses the critical security and privacy challenges in Internet of Vehicles (IoV) environments by introducing a quantum-safe UAV-assisted blockchain authentication protocol for Intelligent Transportation Systems (ITS). The proposed scheme integrates lattice-based cryptography (LBC) for post-quantum security, Hyperledger Fabric PBFT consensus for decentralized trust, and UAV relays for enhanced network resilience, overcoming the limitations of traditional certificate-based RSU architectures. Experimental evaluation demonstrates end-to-end latency of 465 ms (registration 45 ms, authentication 180 ms, PBFT consensus 290 ms) with 2.1% failure rate within PBFT $f < 1/3$ fault tolerance. The protocol achieves 16.8% and 37.8% latency reductions compared to ECC- and LBC-based baselines, respectively, while UAV relays mitigate 35% of PQC-induced delays and 28% mobility degradation. These metrics confirm real-time viability for safety-critical ITS applications requiring <500 ms response times. Formal QROM analysis proves security against quantum adversaries exploiting SIS/ISIS hardness assumptions, while informal analysis resists MITM, replay, impersonation, and DDoS attacks. Conditional anonymity via pseudonymous IDs and encrypted commitments Balance privacy with traceability.

This scalable, quantum-resistant framework establishes a robust foundation for secure, real-time vehicle-to-infrastructure communications essential for next-generation ITS deployments. Future research may explore group-oriented authentication to support platooning and cooperative driving scenarios, incorporate adaptive learning-based threat detection to counter evolving adversaries, and validate the protocol through real-world ITS testbeds. Such efforts will further advance the practicality and resilience of blockchain-UAV-assisted ITS in the era of quantum computing.

The security of patient health data in smart healthcare systems necessitates robust security measures, particularly in terms of ensuring confidentiality and authenticity. As a result, researchers have presented various schemes using identity- and certificate-less-based solutions to meet the confidentiality and authentication requirements effectively. However, the existing schemes use PKI, PKG, or KGC and follow a centralized architecture for data storage and processing tasks, resulting in high computational cost, a single point of failure, higher latency, and failure with the quantum computing process.

Thus, Chapter 7 of this thesis proposes a blockchain-based, lattice-based authenticated key agreement scheme for the SHS environment. First, the blockchain technology is integrated into a cooperative hospital network by taking medical servers corresponding to each hospital as blockchain peers to create a distributed healthcare environment. Furthermore, to provide authentication of the device and patient and confidential data transfer, a Lattice-based AKA scheme has been presented. Second, we introduce a blind signature-based authentication mechanism to verify the authenticity of nodes in a blockchain. The proposed scheme's security has been proven secure using a formal security proof, QROM,

a game proof-based rule that shows the defense against security attacks like secure session keys, unforgability, etc. Moreover, informal security analysis illustrates the security of the proposed scheme against various types of attacks and also shows the fulfillment of predefined security goals. The proposed scheme's cryptographic operations are implemented and evaluated using cryptographic libraries, respectively. Finally, the comparative analysis with the existing schemes proves the computational and communication efficiency of the proposed scheme.

8.2 Future Directions

The proposed cryptographic schemes offer potential applications in IoT applications, specifically for achieving blockchain-based distributed device/user authentication and secure data transfer. The proposed schemes have demonstrated strong security strength and efficiency through security and performance analysis, making them suitable for IoT applications. However, it is important to acknowledge certain limitations that these schemes currently face. Firstly, there is a requirement for higher latency due to blockchain operations, which may impact real-time responsiveness. Additionally, deploying a blockchain infrastructure introduces the need for additional resources and infrastructure setup. Moreover, the proposed schemes rely on consortium blockchains. Several promising research directions can be pursued in the future to address these challenges and further enhance the field.

1. The proposed schemes introduce additional latency due to blockchain operations, such as insert and update. Although these operations are not performed during signature generation, verification, and key agreement phases, they still contribute to the overall latency. Therefore, it is necessary to design more efficient approaches to address this latency burden imposed by the blockchain. Future research should focus on developing streamlined methods that optimize and minimize the time required for blockchain-related tasks in the proposed schemes. The overall latency can be reduced by addressing this issue, improving the performance and responsiveness of the system.

2. The proposed schemes utilize a consortium blockchain to establish a trusted and secure environment for handling cryptographic security scheme-related data. However, it is essential to acknowledge that modifications are required for these schemes to be compatible with trustless blockchain-based environments that adopt public blockchains for data handling. In trustless environments, where there is no predefined set of trusted entities, additional considerations and adaptations are necessary to ensure the integrity and reliability of the cryptographic operations. Future research should focus on modifying the proposed schemes to suit these trustless

environments and leverage the capabilities offered by public blockchains for data management.

3. The proposed schemes aim to establish authentication of devices and users, thereby creating a trusted environment. However, it is essential to identify and address malicious behavior exhibited by authenticated entities to ensure system security. Cryptographic schemes alone are insufficient in preventing attacks such as the black hole attack. Hence, it becomes imperative to incorporate effective intrusion detection techniques that can detect and subsequently revoke the malicious entities.

References

[1] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: a survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[3] P. Verma and D. S. Gupta, "A pairing-free data authentication and aggregation mechanism for intelligent healthcare system," *Computer Communications*, vol. 198, pp. 282–296, 2023.

[4] M. N. Sadiku, R. A. Jaiyesimi, J. B. Idehen, and S. M. Musa, *Emerging Technologies in Healthcare*. AuthorHouse, 2021.

[5] I. Acharjamayum, R. Patgiri, and D. Devi, "Blockchain: a tale of peer to peer security," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 609–617.

[6] A. Awad, S. J. Trenfield, T. D. Pollard, J. J. Ong, M. Elbadawi, L. E. McCoubrey, A. Goyanes, S. Gaisford, and A. W. Basit, "Connected healthcare: Improving patient care using digital health technologies," *Advanced Drug Delivery Reviews*, vol. 178, p. 113958, 2021.

[7] B. D. Deebak, F. Al-Turjman, M. Aloqaily, and O. Alfandi, "An authentic-based privacy preservation protocol for smart e-healthcare systems in iot," *IEEE Access*, vol. 7, pp. 135 632–135 649, 2019.

[8] Q. Cheng, Y. Li, W. Shi, and X. Li, "A certificateless authentication and key agreement scheme for secure cloud-assisted wireless body area network," *Mobile Networks and Applications*, pp. 1–11, 2021.

[9] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[10] J. Heo, C. S. Hong, M. S. Choi, S. H. Ju, and Y. H. Lim, "Identity-based mutual device authentication schemes for plc system," in *2008 IEEE International Symposium on Power Line Communications and Its Applications*. IEEE, 2008, pp. 47–51.

[11] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*. Springer, 2003, pp. 452–473.

[12] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996, pp. 99–108.

[13] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM, 1997, pp. 284–293.

[14] ——, "The first and fourth public-key cryptosystems with worst-case/average-case equivalence," in *ELECTRONIC COLLOQUIUM ON COMPUTATIONAL COMPLEXITY, REPORT NO. 97 (2007)*. Citeseer, 2007.

[15] O. Goldreich, S. Goldwasser, and S. Halevi, "Collision-free hashing from lattice problems." *IACR Cryptology ePrint Archive*, vol. 1996, p. 9, 1996.

[16] G. S. Aljumaie, G. H. Alzeer, R. K. Alghamdi, H. Alsuwat, and E. Alsuwat, "Modern study on internet of medical things (iomt) security," *International Journal of Computer Science & Network Security*, vol. 21, no. 8, pp. 254–266, 2021.

[17] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*. Springer, 1984, pp. 47–53.

[18] D. S. Gupta, S. H. Islam, M. S. Obaidat, P. Vijayakumar, N. Kumar, and Y. Park, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for iiot environments," *IEEE Systems Journal*, 2020.

[19] I. Ullah, N. U. Amin, M. A. Khan, H. Khattak, and S. Kumari, "An efficient and provable secure certificate-based combined signature, encryption and signcryption scheme for internet of things (iot) in mobile health (m-health) system," *Journal of Medical Systems*, vol. 45, no. 1, pp. 1–14, 2021.

[20] D. S. Gupta and G. Biswas, "A secure cloud storage using ecc-based homomorphic encryption," *International Journal of Information Security and Privacy (IJISP)*, vol. 11, no. 3, pp. 54–62, 2017.

[21] S. H. Islam and G. Biswas, "A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 1, pp. 63–73, 2017.

[22] V. Kumar, S. Ray, M. Dasgupta, and M. K. Khan, "A pairing free identity based two party authenticated key agreement protocol using hexadecimal extended ascii elliptic curve cryptography," *Wireless Personal Communications*, vol. 118, no. 4, pp. 3045–3061, 2021.

[23] A. K. Das and A. Goswami, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *Journal of medical systems*, vol. 37, no. 3, pp. 1–16, 2013.

[24] S. B. Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and integrity for data aggregation in wsn using homomorphic encryption," *Wireless Personal Communications*, vol. 80, no. 2, pp. 867–889, 2015.

[25] D. S. Gupta and G. Biswas, "Secure computation on cloud storage: a homomorphic approach," *Journal of Cases on Information Technology (JCIT)*, vol. 17, no. 3, pp. 22–29, 2015.

[26] P. K. Dhillon and S. Kalra, "Multi-factor user authentication scheme for iot-based healthcare services," *Journal of Reliable Intelligent Environments*, vol. 4, no. 3, pp. 141–160, 2018.

[27] L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang, and X. Fu, "Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, p. 1550147718772545, 2018.

[28] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[29] S. Han, S. Zhao, Q. Li, C.-H. Ju, and W. Zhou, "Ppm-hda: privacy-preserving and multifunctional health data aggregation with fault tolerance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1940–1955, 2015.

[30] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–495, 2018.

[31] N. M. Lwamo, L. Zhu, C. Xu, K. Sharif, X. Liu, and C. Zhang, "Suaa: A secure user authentication scheme with anonymity for the single & multi-server environments," *Information Sciences*, vol. 477, pp. 369–385, 2019.

[32] D. S. Gupta, S. H. Islam, M. S. Obaidat, A. Karati, and B. Sadoun, "Laac: Lightweight lattice-based authentication and access control protocol for e-health systems in iot environments," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3620–3627, 2020.

[33] M. Masud, G. S. Gaba, S. Alqahtani, G. Muhammad, B. Gupta, P. Kumar, and A. Ghoneim, "A lightweight and robust secure key establishment protocol for internet of medical things in covid-19 patients care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 694–15 703, 2020.

[34] M.-D. Cano and A. Cañavate-Sanchez, "Preserving data privacy in the internet of medical things using dual signature ecdsa," *Security and Communication Networks*, vol. 2020, 2020.

[35] S. D. Mamdiwar, Z. Shakruwala, U. Chadha, K. Srinivasan, C.-Y. Chang *et al.*, "Recent advances on iot-assisted wearable sensor systems for healthcare monitoring," *Biosensors*, vol. 11, no. 10, p. 372, 2021.

[36] K. Sowjanya, M. Dasgupta, and S. Ray, "Elliptic curve cryptography based authentication scheme for internet of medical things," *Journal of Information Security and Applications*, vol. 58, p. 102761, 2021.

[37] J. Li, Z. Su, D. Guo, K.-K. R. Choo, and Y. Ji, "Psl-maaka: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 183–13 195, 2021.

[38] H. Yao, Q. Yan, X. Fu, Z. Zhang, and C. Lan, "Ecc-based lightweight authentication and access control scheme for iot e-healthcare," *Soft Computing*, pp. 1–21, 2021.

[39] B. Zhang, "A lightweight data aggregation protocol with privacy-preserving for healthcare wireless sensor networks," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1705–1716, 2020.

[40] S. B. Othman, F. A. Almalki, C. Chakraborty, and H. Sakli, "Privacy-preserving aware data aggregation for iot-based healthcare with green computing technologies," *Computers and Electrical Engineering*, vol. 101, p. 108025, 2022.

[41] J. Chang, Q. Ren, Y. Ji, M. Xu, and R. Xue, "Secure medical data management with privacy-preservation and authentication properties in smart healthcare system," *Computer Networks*, p. 109013, 2022.

[42] A. P. Sandro Hurtado, Jose Garcia Nieto and I. Navas-Delgado, "Human activity recognition from sensorised patient's data in healthcare: A streaming deep learning-based approach," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. InPress, no. 4, pp. 1–15, InPress.

[43] C. C. Amit Kishori and W. Jeberson, "A novel fog computing approach for minimization of latency in healthcare using machine learning," *International Journal of Interactive Multimedia and Artificial Intelligence*, no. 6 (Special Issue on Current Trends in Intelligent Multimedia Processing Systems), pp. 7–17, 2021.

[44] D. Chaudhary and R. Pahuja, "Improvement in quality of service against doppelganger attacks for connected network," *International Journal of Interactive Multimedia and Artificial Intelligence*, no. 7 (Special Issue on Current Trends in Intelligent Multimedia Processing Systems), pp. 51–58, 2022.

[45] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*. Springer, 1985, pp. 47–53.

[46] S. H. Islam and M. K. Khan, "Provably secure and pairing-free identity-based handover authentication protocol for wireless mobile networks," *International Journal of Communication Systems*, vol. 29, no. 17, pp. 2442–2456, 2016.

[47] D. He, J. Chen, and R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings," *International Journal of Communication Systems*, vol. 25, no. 11, pp. 1432–1442, 2012.

[48] H. Sun, Q. Wen, and W. Li, "A strongly secure pairing-free certificateless authenticated key agreement protocol under the cdh assumption," *Science China Information Sciences*, vol. 59, no. 3, pp. 1–16, 2016.

[49] Y.-J. Kim, Y.-M. Kim, Y.-J. Choe *et al.*, "An efficient bilinear pairing-free certificateless two-party authenticated key agreement protocol in the eck model," *arXiv preprint arXiv:1304.0383*, 2013.

[50] H. Tu, N. Kumar, J. Kim, and J. Seo, "A strongly secure pairing-free certificateless authenticated key agreement protocol suitable for smart media and mobile environments," *Multimedia Tools and Applications*, vol. 74, pp. 6365–6377, 2015.

[51] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2327–2339, 2014.

[52] D. Wang, H. Cheng, D. He, and P. Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Systems Journal*, vol. 12, no. 1, pp. 916–925, 2016.

[53] T.-T. Truong, M.-T. Tran, and A.-D. Duong, "Improvement of the more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on ecc," in *2012 26th international conference on advanced information networking and applications workshops*. IEEE, 2012, pp. 698–703.

[54] D. S. Gupta and G. Biswas, "An ecc-based authenticated group key exchange protocol in ibe framework," *International Journal of Communication Systems*, vol. 30, no. 18, p. e3363, 2017.

[55] P. Verma and D. S. Gupta, "A pairing-free data authentication and aggregation mechanism for intelligent healthcare system," *Computer Communications*, 2022.

[56] D. S. Gupta, K. Parai, M. S. Obaidat, and S. H. Islam, "Efficient and secure design of id-3paka protocol using ecc," in *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, 2021, pp. 1–5.

[57] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[58] N. Abiramy and S. Sudha, "A secure and lightweight authentication protocol for multiple layers in wireless body area network," in *Smart Intelligent Computing and Applications*. Springer, 2019, pp. 287–296.

[59] B. Kapito, M. Nyirenda, and H. Kim, "Privacy-preserving machine authenticated key agreement for internet of things," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 13, no. 2, pp. 99–120, 2021.

[60] M. Bellare and P. Rogaway, "Provably secure session key distribution: the three party case," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, 1995, pp. 57–66.

[61] ——, "Entity authentication and key distribution," in *Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings*. Springer, 2001, pp. 232–249.

[62] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings 20*. Springer, 2001, pp. 453–474.

[63] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security: First International Conference, ProvSec 2007, Wollongong, Australia, November 1-2, 2007. Proceedings 1*. Springer, 2007, pp. 1–16.

[64] K. Yoneyama and Y. Zhao, "Taxonomical security consideration of authenticated key exchange resilient to intermediate computation leakage," in *Provable Security: 5th International Conference, ProvSec 2011, Xi'an, China, October 16-18, 2011. Proceedings 5*. Springer, 2011, pp. 348–365.

[65] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*. Springer, 2001, pp. 213–229.

[66] A. Sharma and B. Kim, “Secure v2v authentication using ecc in vanet,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2341–2352, 2021.

[67] C. Li and Q. Liu, “Anonymous authentication for iov networks based on lattice cryptography,” *Ad Hoc Networks*, vol. 112, p. 102345, 2021.

[68] J. Singh and K. Wang, “Privacy-preserving communication in vanet with zero-knowledge protocols,” in *Proceedings of IEEE ICCCN*, 2022, pp. 1–7.

[69] Y. Zhang and S. Patel, “Blockchain-based trust management for vehicular networks,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 4090–4102, 2021.

[70] A. Gupta and M. Chen, “Lightweight mutual authentication for resource-constrained vanets,” *Sensors*, vol. 22, no. 3, p. 1350, 2022.

[71] H. Tan and D. Das, “Post-quantum secure communications for its using lattice-based cryptosystems,” *IEEE Access*, vol. 10, pp. 10 054–10 067, 2022.

[72] R. Park and J. Lee, “Multi-factor authentication in vanet: Combining ecc and biometric authentication,” *IEEE Communications Letters*, vol. 25, no. 7, pp. 2007–2011, 2021.

[73] S. Kumar and Y. Zhao, “Edge-assisted blockchain authentication in vanets,” *Future Generation Computer Systems*, vol. 124, pp. 456–467, 2021.

[74] M. Costa and F. Wang, “Quantum cryptography-based secure communication in vehicular networks,” in *Proceedings of IEEE GLOBECOM*, 2022, pp. 1–6.

[75] I. Niazi and M. Rasheed, “Batch authentication with group signatures in iov,” *Vehicular Communications*, vol. 27, p. 100329, 2021.

[76] E. Hossain and A. Singh, “Adaptive privacy control with blockchain for connected vehicles,” *IEEE Systems Journal*, vol. 15, no. 4, pp. 5320–5331, 2021.

[77] F. Li and Z. Tong, “Efficient v2i authentication with ecc,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2501–2511, 2022.

[78] M. Zhao and L. Xu, “Hash-based authentication for resource-constrained iov devices,” *Internet of Vehicles Journal*, vol. 13, no. 2, pp. 111–121, 2021.

[79] P. Saini and G. Kaur, “Sybil attack resistance using blockchain in vanets,” *IEEE Access*, vol. 9, pp. 143 201–143 215, 2021.

[80] L. Perez and S. K. Gupta, “Decentralized trust infrastructure for its with blockchain,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4004–4016, 2022.

[81] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, “Proven secure tree-based authenticated key agreement for securing v2v and v2i communications in vanets,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3280–3297, 2022.

[82] P. Verma, “A secure gateway discovery protocol using elliptic curve cryptography for internet-integrated manet,” in *Cryptographic Security Solutions for the Internet of Things*. IGI Global, 2019, pp. 181–210.

[83] J. Miao, Z. Wang, X. Ning, A. Shankar, C. Maple, and J. J. Rodrigues, “A uav-assisted authentication protocol for internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10 286–10 297, 2024.

[84] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the internet of drones applications," *Computer Communications*, vol. 155, pp. 143–149, 2020.

[85] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized uav detection and mitigation in internet of drones environment," *Computer Communications*, vol. 166, pp. 91–109, 2021.

[86] P. Verma and D. S. Gupta, "A paring-free id-based authenticated key agreement protocol for iot environment," in *International Conference on Computing and Information Technology*. Springer, 2022, pp. 68–79.

[87] ——, "Sp2p-maka: Smart contract based secure p2p mutual authentication key agreement protocol for intelligent energy system," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2022, pp. 83–92.

[88] P. Mundhe, V. K. Yadav, S. Verma, and S. Venkatesan, "Efficient lattice-based ring signature for message authentication in vanets," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5463–5474, 2020.

[89] D. Mishra, M. Singh, P. Rewal, K. Pursharthi, N. Kumar, A. Barnawi, and R. S. Rathore, "Quantum-safe secure and authorized communication protocol for internet of drones," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 16 499–16 507, 2023.

[90] C. Chen, L. Wang, and Q. Shi, "A lattice-based certificateless secure data transmission scheme for internet of vehicles based-blockchain," *IEEE Transactions on Vehicular Technology*, 2024.

[91] D. S. Gupta, S. Ray, T. Singh, and M. Kumari, "Post-quantum lightweight identity-based two-party authenticated key exchange protocol for internet of vehicles with probable security," *Computer Communications*, vol. 181, pp. 69–79, 2022.

[92] P. Verma and D. S. Gupta, "An improved certificateless mutual authentication and key agreement protocol for cloud-assisted wireless body area networks," *Wireless Personal Communications*, vol. 131, no. 4, pp. 2399–2426, 2023.

[93] Q. Du, J. Zhou, and M. Ma, "Eaia: An efficient and anonymous identity authentication scheme in 5g-v2v," arXiv:2406.04705, 2024, supports RSU-less V2V, Scyther verified.

[94] J. Lee, S. Yu, M. Kim, Y. Park, S. Lee, and B. Chung, "Secure key agreement and authentication protocol for message confirmation in vehicular cloud computing," *Applied Sciences*, vol. 10, no. 18, p. 6268, 2020.

[95] D. S. Gupta and G. Biswas, "Cryptanalysis of wang et al.'s lattice-based key exchange protocol," *Perspectives in Science*, vol. 8, pp. 228–230, 2016.

[96] ——, "A novel and efficient lattice-based authenticated key exchange protocol in c-k model," *International Journal of Communication Systems*, vol. 31, no. 3, p. e3473, 2018.

[97] D. S. Gupta, A. Karati, W. Saad, and D. B. da Costa, "Quantum-defended blockchain-assisted data authentication protocol for internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3255–3266, 2022.

[98] Z. Lu, Q. Wang, G. Qu, H. Zhang, and Z. Liu, "A blockchain-based privacy-preserving authentication scheme for vanets," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2792–2801, 2019.

[99] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in vanets," *IEEE Access*, vol. 7, pp. 117 716–117 726, 2019.

[100] C. Lin, D. He, X. Huang, N. Kumar, and K.-K. R. Choo, "Bcppa: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7408–7420, 2020.

[101] Q. Feng, D. He, S. Zeadally, and K. Liang, "Bpas: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4146–4155, 2019.

[102] Z. Xu, W. Liang, K.-C. Li, J. Xu, and H. Jin, "A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles," *Journal of Parallel and Distributed Computing*, vol. 149, pp. 29–39, 2021.

[103] C. Wang, J. Shen, J.-F. Lai, and J. Liu, "B-tsca: Blockchain assisted trustworthiness scalable computation for v2i authentication in vanets," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1386–1396, 2020.

[104] Y. Ren, X. Li, S.-F. Sun, X. Yuan, and X. Zhang, "Privacy-preserving batch verification signature scheme based on blockchain for vehicular ad-hoc networks," *Journal of Information Security and Applications*, vol. 58, p. 102698, 2021.

[105] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for internet of vehicles," *Journal of Systems Architecture*, vol. 113, p. 101877, 2021.

[106] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 1229–1237.

[107] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 246–250.

[108] C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Networks*, vol. 17, no. 8, pp. 1851–1865, 2011.

[109] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for vanet," *Wireless networks*, vol. 19, no. 6, pp. 1441–1449, 2013.

[110] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[111] S.-J. Horng, S.-F. Tzeng, P.-H. Huang, X. Wang, T. Li, and M. K. Khan, "An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Information Sciences*, vol. 317, pp. 48–66, 2015.

[112] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[113] J. Cui, D. Wu, J. Zhang, Y. Xu, and H. Zhong, "An efficient authentication scheme based on semi-trusted authority in vanets," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2972–2986, 2019.

[114] Y. Wang, H. Zhong, Y. Xu, J. Cui, and G. Wu, "Enhanced security identity-based privacy-preserving authentication scheme supporting revocation for vanets," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5373–5383, 2020.

[115] S. Wang, K. Mao, F. Zhan, and D. Liu, "Hybrid conditional privacy-preserving authentication scheme for vanets," *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1600–1615, 2020.

[116] D. S. Gupta, "A mutual authentication and key agreement protocol for smart grid environment using lattice," in *Proceedings of the International Conference on Computational Intelligence and Sustainable Technologies*. Springer, 2022, pp. 239–248.

[117] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology—CRYPTO*. Springer, 1983, pp. 199–203.

[118] H. Li, Q. Pei, and K. Liao, "A secure and efficient blockchain-based data sharing scheme for smart healthcare," *Future Generation Computer Systems*, vol. 95, pp. 590–599, 2019.

[119] C. C. Agbo, Q. H. Mahmoud, and P. Eklund, "Blockchain technology in healthcare: A comprehensive review and directions for future research," *Applied Sciences*, vol. 9, no. 9, p. 1736, 2019.

[120] R. Xu, M. Song *et al.*, "Blockchain-based privacy-preserving medical data sharing and access control," *Information Sciences*, vol. 526, pp. 1–14, 2020.

[121] W. Yang, L. Chen *et al.*, "Lightweight post-quantum cryptography for iot: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9510–9528, 2020.

[122] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology—EUROCRYPT*. Springer, 2012, pp. 738–755.

[123] L. Ducas, A. Durmus, T. Lepoint, and D. S. Prest, "Bliss: A practical lattice-based signature scheme," *Journal of Cryptology*, vol. 30, no. 3, pp. 399–446, 2017.

[124] M. Zhandry, "How to construct quantum random oracles," in *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2012, pp. 679–687.

[125] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, J. M. Schanck, P. Schwabe, D. Stehlé *et al.*, "Crystals-kyber: A cca-secure module-lattice-based kem," NIST Post-Quantum Cryptography Standardization Project, 2018.

[126] D. He, N. Kumar, M. K. Khan, and J.-H. Lee, "An efficient and privacy-preserving authentication scheme for wireless medical sensor networks," *Journal of Medical Systems*, vol. 42, no. 5, pp. 1–14, 2018.

[127] B. A. Alzahrani, A. Irshad, and S. A. Chaudhry, "Secure and anonymous authentication scheme for internet of medical things," *IEEE Access*, vol. 8, pp. 195 951–195 963, 2020.

[128] S. H. Islam and G. P. Biswas, "A secure and privacy-preserving authentication scheme for iot-based healthcare systems," *Journal of Information Security and Applications*, vol. 58, p. 102706, 2021.

[129] H. Li, Q. Pei, and K. Liao, "A secure and efficient blockchain-based data sharing scheme for smart healthcare," *Future Generation Computer Systems*, vol. 95, pp. 590–599, 2019.

[130] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Blockchain-based privacy-preserving medical data sharing and access control," *Information Sciences*, vol. 526, pp. 1–14, 2020.

[131] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology – EUROCRYPT*. Springer, 2012, pp. 738–755.

[132] L. Ducas, A. Durmus, T. Lepoint, and D. S. Prest, "Bliss: A practical lattice-based signature scheme," *Journal of Cryptology*, vol. 30, no. 3, pp. 399–446, 2017.

[133] N. Eltayieb, R. Elhabob, A. Hassan, and F. Li, "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud," *Journal of Systems Architecture*, vol. 102, p. 101653, 2020.

[134] A. Elkhalil, J. Zhang, and R. Elhabob, "An efficient heterogeneous blockchain-based online/offline signcryption systems for internet of vehicles," *Cluster Computing*, pp. 1–18, 2021.

[135] G. Xu, J. Dong, C. Ma, J. Liu, and U. G. O. Cliff, "A certificateless signcryption mechanism based on blockchain for edge computing," *IEEE Internet of Things Journal*, 2022.

[136] Y. Yang, D. He, P. Vijayakumar, B. B. Gupta, and Q. Xie, "An efficient identity-based aggregate signcryption scheme with blockchain for iot-enabled maritime transportation system," *IEEE Transactions on Green Communications and Networking*, 2022.

[137] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[138] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[139] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[140] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the theory and application of cryptographic techniques*. Springer, 1985, pp. 417–426.

[141] D. S. Gupta, S. Ray, T. Singh, and M. Kumari, "Post-quantum lightweight identity-based two-party authenticated key exchange protocol for internet of vehicles with probable security," *Computer Communications*, vol. 181, pp. 69–79, 2022.

[142] D. S. Gupta and G. Biswas, "On securing bi-and tri-partite session key agreement protocol using ibe framework," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4505–4524, 2017.

[143] D. S. Gupta, S. Hafizul Islam, and M. S. Obaidat, "A secure identity-based three-party authenticated key agreement protocol using bilinear pairings," in *Innovative Data Communication Technologies and Application: ICIDCA 2019*. Springer, 2020, pp. 1–11.

[144] P. Marc, "Blockchain technology: Principles and applications," 2016.

[145] H. Gamage, H. Weerasinghe, and N. Dias, "A survey on blockchain technology concepts, applications, and issues," *SN Computer Science*, vol. 1, pp. 1–15, 2020.

[146] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security framework [future directions]," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, 2018.

[147] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43 620–43 652, 2021.

[148] S. Naz and S. U.-J. Lee, "Why the new consensus mechanism is needed in blockchain technology?" in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 2020, pp. 92–99.

[149] C. Komalavalli, D. Saxena, and C. Laroiya, "Overview of blockchain technology concepts," in *Handbook of research on blockchain technology*. Elsevier, 2020, pp. 349–371.

[150] C. Zhang, C. Wu, and X. Wang, "Overview of blockchain consensus mechanism," in *Proceedings of the 2020 2nd International Conference on Big Data Engineering*, 2020, pp. 7–12.

[151] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.

[152] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[153] W. H. Organization, "Food safety- google chrome," https://www.who.int/health-topics/food-safety, (Accessed on 20/4/2022).

[154] F. Firouzi, B. Farahani, M. Ibrahim, and K. Chakrabarty, "Keynote paper: from eda to iot ehealth: promises, challenges, and solutions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 2965–2978, 2018.

[155] A. K. Sahu, S. Sharma, and D. Puthal, "Lightweight multi-party authentication and key agreement protocol in iot-based e-healthcare service," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 2s, pp. 1–20, 2021.

[156] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities," *Ieee Access*, vol. 5, pp. 26 521–26 544, 2017.

[157] I. G. Cohen and M. M. Mello, "Hipaa and protecting health information in the 21st century," *Jama*, vol. 320, no. 3, pp. 231–232, 2018.

[158] N. McCullagh and P. S. Barreto, "A new two-party identity-based authenticated key agreement," in *Cryptographers' Track at the RSA Conference*, 2005, pp. 262–274.

[159] Miracl, "miracl/miracl," Aug 2019. [Online]. Available: https://github.com/miracl/MIRACL

[160] G. Mwitende, I. Ali, N. Eltayieb, B. Wang, and F. Li, "Authenticated key agreement for blockchain-based wban," *Telecommunication Systems*, vol. 74, no. 3, pp. 347–365, 2020.

[161] M. N. Aman, M. H. Basheer, S. Dash, J. W. Wong, J. Xu, H. W. Lim, and B. Sikdar, "Hatt: Hybrid remote attestation for the internet of things with high availability," *IEEE Internet of Things Journal*, 2020.

[162] S. O. Ogundoyin and I. A. Kamil, "Paash: A privacy-preserving authentication and fine-grained access control of outsourced data for secure smart health in smart cities," *Journal of Parallel and Distributed Computing*, vol. 155, pp. 101–119, 2021.

[163] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*. Springer, 2003, pp. 452–473.

[164] Q. Cheng, Y. Li, W. Shi, and X. Li, "A certificateless authentication and key agreement scheme for secure cloud-assisted wireless body area network," *Mobile Networks and Applications*, pp. 1–11, 2022.

[165] M. Kumar and S. Chand, "A lightweight cloud-assisted identity-based anonymous authentication and key agreement protocol for secure wireless body area network," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2779–2786, 2020.

[166] C. Creß, Z. Bing, and A. C. Knoll, "Intelligent transportation systems using roadside infrastructure: A literature survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6309–6327, 2024.

[167] A. K. Tyagi and N. Sreenath, "Autonomous vehicles and intelligent transportation systems—a framework of intelligent vehicles," in *Intelligent Transportation Systems: Theory and Practice*. Springer, 2022, pp. 75–98.

[168] M. A. H. Al Junaid, A. Syed, M. N. M. Warip, K. N. F. K. Azir, and N. H. Romli, "Classification of security attacks in vanet: A review of requirements and perspectives," in *MATEC web of conferences*, vol. 150. EDP Sciences, 2018, p. 06038.

[169] K. Huang, H. Hu, and C. Lin, "Bakas-uav: A secure blockchain-assisted authentication and key agreement scheme for unmanned aerial vehicles networks," *IEEE Internet of Things Journal*, 2024.

[170] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[171] Q. Feng, D. He, S. Zeadally, and K. Liang, "Bpas: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4146–4155, 2019.

[172] D. J. Bernstein, "Post-quantum cryptography," in *Encyclopedia of Cryptography, Security and Privacy*. Springer, 2025, pp. 1846–1847.

[173] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2775–2785, 2018.

[174] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2018.

[175] J. Li, Z. Su, D. Guo, K.-K. R. Choo, and Y. Ji, "Psl-maaka: Provably-secure and lightweight mutual authentication and key agreement protocol for fully public channels in internet of medical things," *IEEE Internet of Things Journal*, 2021.

[176] B. Zhang, "A lightweight data aggregation protocol with privacy-preserving for healthcare wireless sensor networks," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1705–1716, 2020.

[177] A. A. Omala, K. P. Kibiwott, and F. Li, "An efficient remote authentication scheme for wireless body area network," *Journal of medical systems*, vol. 41, no. 2, p. 25, 2017.

[178] K. Mtonga, E. J. Yoon, and H. S. Kim, "Authenticated privacy preserving pairing-based scheme for remote health monitoring systems," *Journal of Information Security*, vol. 8, no. 1, pp. 75–90, 2016.

[179] A. Hassan, A. A. Omala, M. Ali, C. Jin, and F. Li, "Identity-based user authenticated key agreement protocol for multi-server environment with anonymity," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 890–902, 2019.

[180] Q. Jiang, X. Lian, C. Yang, J. Ma, Y. Tian, and Y. Yang, "A bilinear pairing based anonymous authentication scheme in wireless body area networks for mhealth," *Journal of medical systems*, vol. 40, no. 11, p. 231, 2016.

[181] J. Liu, L. Zhang, and R. Sun, "1-raap: An efficient 1-round anonymous authentication protocol for wireless body area networks," *Sensors*, vol. 16, no. 5, p. 728, 2016.

[182] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1299–1309, 2017.

[183] Z. Wang, J. Zhao, P. Sun, J. Yang, R. Wang, and X. Zhang, "A lightweight three-party mutual authentication protocol for internet of health things systems," *Journal of Healthcare Engineering*, vol. 2023, no. 1, p. 1044282, 2023.

[184] S. Das, S. Namasudra, S. Deb, P. M. Ger, and R. G. Crespo, "Securing iot-based smart healthcare systems by using advanced lightweight privacy-preserving authentication scheme," *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 18 486–18 494, 2023.

[185] A. A. Al-Saggaf, T. Sheltami, H. Alkhzaimi, and G. Ahmed, "Lightweight two-factor-based user authentication protocol for iot-enabled healthcare ecosystem in quantum computing," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2347–2357, 2023.

[186] H. Ghaemi, D. Abbasinezhad-Mood, A. Ostad-Sharif, and Z. Alizadehsani, "Novel blockchain-assisted fault-tolerant roaming authentication protocol for mobility networks without home agent entanglement," *Journal of Network and Computer Applications*, vol. 224, p. 103843, 2024.

[187] T.-Y. Wu, L. Wang, and C.-M. Chen, "Enhancing the security: A lightweight authentication and key agreement protocol for smart medical services in the ioht," *Mathematics*, vol. 11, no. 17, p. 3701, 2023.

[188] J. Cao, P. Yu, X. Xiang, M. Ma, and H. Li, "Anti-quantum fast authentication and data transmission scheme for massive devices in 5g nb-iot system," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9794–9805, 2019.

[189] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das, and N. Saxena, "Lscsh: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 24–32, 2018.

[190] S. Mukherjee, D. S. Gupta, and G. P. Biswas, "An efficient and batch verifiable conditional privacy-preserving authentication scheme for vanets using lattice," *Computing*, vol. 101, no. 12, pp. 1763–1788, 2019.

List of Publications

International Journals Published

1. **Pooja Verma** and Daya Sagar Gupta, "A pairing-free data authentication and aggregation mechanism for Intelligent Healthcare System. ", ***Computer Communications (ELSEVIER) (Q1)***, vol. 198 (2023) (**IF: 6**), DOI:https://doi.org/10.1016/j.comcom.2022.12.009.

2. **Pooja Verma** and Daya Sagar Gupta, "An Improved Certificateless Mutual Authentication and Key Agreement Protocol for Cloud-Assisted Wireless Body Area Networks. network", ***Wireless Personal Communications,(Q2, WoS), Springer***, vol. 147 (2023) (**IF: 2.2**), DOI:https://doi.org/10.1007/s11277-023-10596-8.

Conference Papers Published

1. Pooja Verma and Daya Sagar Gupta, "A Paring-free ID-based Authenticated Key Agreement Protocol for IoT Environment," 12th International Conference on Information Systems and Advanced Technologies "ICISAT 2022" Intelligent Information, Data Science and Decision Support System.

2. Pooja Verma and Daya Sagar Gupta, "SP2P-MAKA: Smart Contract-based Secure P2P Mutual Authentication Key Agreement Protocol for Intelligent Energy System", 12TH International Conference On Intelligent Systems Design and Applications (ISDA 2022).

International Journals Communicated

Communicated Papers

1. **Pooja Verma**, Daya Sagar Gupta, and Kalka Dubey "A Quantum Safe UAV Assisted Blockchain Authentication Protocol for secure Communication in ITS. ", ***Vehicular Communication***, **(IF: 6.4)** (**Revised manuscript submitted**).

2. **Pooja Verma**, Daya Sagar Gupta, and Kalka Dubey, "Quantum-Resistant Anonymous Authentication for Blockchain-Enabled Smart Healthcare ", ***Internet of Things***, **(IF:7.6)** (**under Review**).

3. **Pooja Verma**, and Daya Sagar Gupta, "A Systematic Review of Blockchain Integration in Intelligent Healthcare Architectures ", ***Computer Networks***, **(IF: 4.6)** (**under Review**).