Ring-LWE based Post-Quantum Authentication Protocols for Internet of Things and Blockchain Applications

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

Submitted by

Ch. Jayanth Babu

(Roll No. 717148)

Under the guidance of

Prof. R. Padmavathy

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA
DECEMBER 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Ring-LWE based Post-Quantum Authentication Protocols for Internet of Things and Blockchain Applications, submitted by Mr. Ch. Jayanth Babu [Roll No. 717148] is approved for the degree of DOCTOR OF PHILOSOPHY at National Institute of Technology, Warangal.

Examiner

Research Supervisor
Prof. R. Padmavathy
Dept. of Computer Science and Engg.
NIT Warangal
India

Chairman
Head of the Department
Computer Science and Engg.
NIT Warangal
India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled, Ring-LWE based Post-Quantum Authentication Protocols for Internet of Things and Blockchain Applications , submitted in partial fulfillment of requirement for the award of degree of DOCTOR OF PHILOSOPHY to National Institute of Technology Warangal, is a bonafide research work done by Mr. Ch. Jayanth Babu [Roll No. 717148] under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor

Prof. R. Padmavathy
Dept. of Computer Science and Engg.
National Institute of Technology, Warangal
India.

Place: NIT Warangal
Date: December, 2023

DECLARATION

This is to certify that the work presented in the thesis entitled “Ring-LWE based Post-Quantum Authentication Protocols for Internet of Things and Blockchain Applications” is a bonafide work done by me under the supervision of Prof. R. Padmavathy and was not submitted elsewhere for the award of any degree.

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

Ch. Jayanth Babu
(Roll No. 717148)
Date: 29-12-2023

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my supervisor Prof. R. Padmavathy, Department of Computer Science and Engineering for her valuable guidance throughout the course. Her technical perception, profound knowledge, sincere effort in guiding a student and devotion to work have greatly charmed and constantly motivated me to work towards the goal. She always gave me ample time for discussions, reviewing my work and suggesting requisite corrections.

I extend my gratitude to all my Doctoral Scrutiny Committee members for their valuable suggestions and faculty members of Computer Science and Engineering Department. I wish to express my sincere thanks to Prof. Ch. Sudhakar, Prof. U.S.N. Raju, and Prof. H.P. Rani for their insightful comments and suggestions during oral presentations.

I am immensely thankful to Prof. Ch. Sudhakar, Prof. R. B. V. Subramaanyam, Prof. P. Radha Krishna, Dr. S. Ravi Chandra and Prof. R. Padmavathy, Heads of Dept. of CSE, during my stay in the department, for providing adequate facilities in the department to carry out the research work. I wish to express my sincere thanks to Prof. N.V. Ramana Rao and Prof. Bidyadhar Subudhi, Directors of NIT Warangal, during my stay at the institute, for their official support and encouragement.

My gratitude to my family for their unconditional love, support and prayers for my success in achieving the goal. Lastly, I am deeply indebted to the almighty for giving the opportunity to pursue this line.

Ch. Jayanth Babu

Dedicated to

My Family & Teachers

ABSTRACT

Authentication and access control protocols must be robust enough to enable security in the communication networks. Authentication verifies the identity of a user, process, or device, often as a prerequisite to allowing access to resources in a system and, therefore, the integrity of data. Several authentication schemes are proposed in the literature based on number-theoretic assumptions such as integer factorization and discrete logarithm problems. These schemes are proved vulnerable to quantum attacks once quantum computer comes into reality. Most importantly, the infrastructures are made up of wireless communication and sensing devices like the Internet of Things. Cryptosystems built on lattice structures is considered as lattice-based cryptography have emerged as a promising and popular domain in the post-quantum cryptography to deal with quantum attacks. Its considered as dominant among other categories due to its high efficiency and strong security. The lattice-based cryptosystems operate relatively small integers because it uses matrices and vectors for computation in specified rings or fields of small order.

A few authentication schemes with lattice assumptions were proposed in the literature. But, most of the schemes are not modeled for resource constrained devices with reduced key size, cipher texts and signature lengths. We propose an authentication and key exchange protocol for a client node -cloud server with a variant of lattice assumptions in ring structure called Ring-Learning With Error(Ring-LWE) for sensor network environment. The scheme is implemented in the IoT-Cloud environment scenario for its performance and compared with relevant protocols. Our scheme has reduced key size and contains fewer operations in key generation, authentication, and verification. We also propose a node-to-node authentication scheme, where two nodes in the network will communicate with lattice assumptions. In addition to two nodes, there exists an intermediatory device gateway device which generates the session key for the communication. We extended the quantum security to cloud level and proposed a lattice-based encryption scheme for the privacy of data in the cloud environment that meets homomorphic properties. The primary goal of the scheme is to employ fully homomorphic data management with efficiency and security. The proposed scheme surpasses the traditional encryption schemes.

All Our schemes are designed based on the Ring-LWE problem in the lattice and its correctness is proved formally and verified with standard protocol verification tool called AVISPA. The informal analysis of these schemes demonstrates security against known attacks in the internet of things environment. All these schemes are proved formally based on hardness of lattice problem in ring structure.

As the discrete logarithm problem in the elliptic curve cryptography(ECDLP) also proved vulnerable to quantum attacks, we proposed a post-quantum blockchain using Ring-LWE signature algorithm . We analysed blockchain vulnerabilities and existing lattice signature techniques. We used a modified Ring-TESLA algorithm to validate blockchain transactions. We also proposed a randomized consensus to avoid the dominant validator problem and maintain the decentralization property among the nodes in the blockchain. We provide comprehensive security proof and analysis of the proposed scheme in the presence of a quantum adversary.

Keywords: Authentication, Post-Quantum cryptography, Lattice-based Cryptography, Internet of Things, Blockchain Technology, Ring-LWE, Digital Signature, Consensus.

Contents

ACKNOWLEDGMENTS i

ABSTRACT iii

List of Figures x

List of Tables xi

List of Algorithms xii

1 Introduction 1

1.1 Lattice-based Cryptography 3

1.2 Post-Quantum Blockchain 5

1.3 Related Work 7

1.3.1 Public Key Cryptography (PKC) based Authentication schemes 7

1.3.2 Lattice-based Authentication schemes 11

1.3.3 Homomorphic Encryption Schemes 13

1.3.4 Analysis of Lattice signature vulnerabilities 16

1.4 Contributions 18

1.5 Organization of the Thesis 22

2 Preliminaries 23

2.1 Notations 23

2.2 Lattices 24

2.2.1 Definition of Lattices 24

2.2.2 Parameters in Lattices 25
2.2.2.1 Fundamental Paralleopiped 26
2.2.2.2 Determinant 26
2.2.2.3 Successive Minima 27
2.2.3 Ideal Lattices 29
2.2.4 Modular Lattices 29
2.3 Classic Lattice Problems 30
2.3.1 Gaussian Measures over Lattices 31
2.3.1.1 Discrete Gaussian Sampling over Lattices 32
2.3.2 Average-case Lattice Problems 33
2.3.3 Ring-SIS and Ring-LWE 35
2.4 Backgroud of IoT 38
2.4.1 Energy Optimization 39

3 Lattice-based authentication and key exchange protocol for Internet of Things 41
3.1 Introduction 42
3.2 Contribution and Protocol Overview 43
3.2.1 Security Requirements 44
3.2.2 System Model 45
3.2.3 Adversary Model 47
3.3 Proposed Model 50
3.3.1 Setup 50
3.3.2 Registration 51
3.3.3 Login and Authentication 52
3.3.4 Verification 53
3.4 Correctness 55
3.4.1 Verification by AVISPA 58
3.5 Security Analysis 60
3.6 Implementation 63
3.6.1 Raspberry Pi 63

3.6.2 Experimentation Overview 64
3.6.3 Results 65
3.6.4 Performance Evaluation 66
3.6.5 Computation Cost 66
3.6.6 Communication Cost 70
3.7 Summary 70

4 Quantum-secure N2N Authentication protocol model for IoT Sensor Networks 72
4.1 Introduction 72
4.2 Contribution and Protocol Overview 73
4.2.1 Security Requirements 74
4.2.2 System Model 74
4.2.3 Adversary Model 75
4.3 Proposed Model 79
4.3.1 Setup 79
4.3.2 Registration 80
4.3.3 Login and Authentication 80
4.3.4 Verification 82
4.4 Security Analysis 84
4.5 Performance Evaluation 87
4.5.1 Computation Cost 87
4.5.2 Security Level and Parameter Selection 88
4.6 Summary 90

5 Construction of communication protocol using Ring-LWE-based Homomorphic Encryption in IoT-Cloud Environment 91
5.1 Introduction 92
5.2 Contribution and Protocol Overview 93
5.2.1 Security Requirements 94
5.2.2 System Model 95
5.2.3 Threat Model 96

5.2.3.1 Assumptions 97

5.3 Proposed Model 98

5.3.1 Setup and Key Generation Phase 98

5.3.2 Data Encryption Phase 99

5.3.3 FHE Verification 101

5.4 Security Analysis 102

5.5 Performance Evaluation 105

5.6 Summary 106

6 Post-Quantum Blockchain with Provable Security 108

6.1 Introduction 109

6.2 Contribution and Construction Overview 111

6.2.1 Quantum Vulnerabilities 112

6.2.2 Quantum Resistant Solutions 113

6.3 Modified Ring-TESLA Algorithm 115

6.3.1 Randomized Consensus 117

6.4 Post Quantum Blockchain Network Model 119

6.4.1 Blockchain Construction 120

6.4.2 Merkle Hash Tree 123

6.5 Security Analysis 123

6.5.1 Blockchain Security Requirements 124

6.5.1.1 Resistance to Quantum Attacker 125

6.5.2 Formal Proof 125

6.5.3 Blockchain and IoT integration 129

6.6 Summary 130

7 Conclusion and Future Directions 131

7.1 The Major Contributions of the Thesis 131

7.2 Future Directions 133

Bibliography 135

List of Publications 146

List of Figures

2.1 A basis of $\mathbb{Z}^2$. 25

2.2 Another basis of $\mathbb{Z}^2$. 25

2.3 A different lattice. 25

2.4 Not a full rank lattice. 25

2.5 Successive Minima: $\mathcal{L}_1(\mathcal{L}) = 1, \mathcal{L}_2(\mathcal{L}) = 2.3$ [1] 28

3.1 System model 46

3.2 Protocol verification using AVISPA tool 59

3.3 Illustration of protocol implementation 64

4.1 Node-to-node communication scenario in IoT network 75

4.2 Proposed Protocol 81

5.1 System Model 96

5.2 Illustration of proposed IoT Cloud Server encryption 100

6.1 Blockchain transaction model 120

6.2 Merkle hash tree structure 123

List of Tables

1.1 Lattice based Protocols 14

3.1 Notations 46
3.2 Rasberrypi specifications 65
3.3 Protocol implementation for different parameters 66
3.4 Notations 67
3.5 Average elapsed times of Lattice-operations(100000 executions) 67
3.6 Comparison w.r.t number of operations 69
3.7 Comparison w.r.t computation time 69

4.1 Notations 79
4.2 Comparison w.r.t. number of operations 87
4.3 Comparative security analysis of protocols 89

5.1 Comparison of time and space complexities 105

6.1 Overview of various blockchain vulnerabilities 114
6.2 IoT vs. Blockchain enabled IoT 130

List of Algorithms

2.1 Gram Schmidt Orthogonalization 27

6.1 Gen$(1^\lambda; a_1, a_2)$ 115

6.2 Sign$(\mu; s, e_1, e_2, t_1, t_2, seed_a)$ 116

6.3 Verify$(\mu; (z, c'), t_1, t_2, seed_a)$ 116

6.4 Unforgeability Experiment $Expt^{ufcma}_{\Pi,A}(1^\lambda)$: 129

Chapter 1

Introduction

Authentication and access control protocols need to possess sufficient strength to ensure security within communication networks. The process of authentication involves confirming the identity of a user or device as a it is a prerequisite for granting access to system resources. This, in turn, safeguards the integrity of the data. Numerous authentication schemes have been suggested in the literature, relying on number-theoretic assumptions like Integer Factorization(IF) and Discrete Logarithm Problems(DLP). However, these schemes are demonstrated to be vulnerable to quantum attacks once quantum computers become into existance. Most importantly, the infrastructures are made up of wireless communication and sensing devices like the Internet of Things(IoT). The IoT is evolving as the "Future Internet" where every device is connected to another device; likewise, it is connected with billions of devices communicating over the internet without human interaction. The data collected in the Internet of things is from various environments and transmitted to servers for storage and processing. For sensitive IoT applications, it is desirable to authenticate IoT nodes to confirm the source of data is trustworthy. On the other hand, IoT devices should have a robust method for authenticating with the server or other communicating devices.

The security of the mejority cryptographic schemes is based on the hardness of computational problems like discrete-logorithm and integer factorization problems. In 1994, Peter Shor[2, 3] in his seminal works developed quantum algorithms that break

IF and finite field DLP in polynomial time. Later, Grover's algorithm [4] solved symmetric cryptosystems (AES, SHA-2, SHA-3) in polynomial time. This means, all the cryptographic schemes built on the above number theoretic assumptions will become insecure once quantum computers are fully realized. The alternative way to withstand quantum computers is quantum-immune or post-quantum crypto-systems. The concept of quantum computers was first used by Richard Feynman in 1982. A Quantum computer is like a modern computer that makes use of quantum mechanics for computation process. Unlike classical computers where information is stored in binary digits 0 and 1 called bits, a quantum computer uses qubits that hold not only 0 or 1 but can hold both simultaneously, known as superposition of states. Due to this feature, several inputs in quantum computers can be evaluated at once through which they are likely to solve computationally hard problems that remained unsolved by classical computers.

The threat of quantum computing capabilities has been well recognized by academic researchers, large organizations, and government agencies.

In 2015, the National Security Agency (NSA) unveiled initiation to shift towards quantum-safe cryptography for safeguarding classified information[1].. Subsequently, in 2016, the National Institute of Standards and Technology (NIST) issued an open call to explore the standardization of quantum-safe cryptographic algorithms[2].. At present the third round of submission and evaluation of schemes are going on and the standardization is expected to be completed tentatively by 2024 [3]. Many big organizations like IBM[4], Rigette[5], Alibaba [6] have already realized quantum computer-based services to clients. Other organizations like Microsoft, Intel, Google, and many more have invested to build quantum computers. Recently in 2020, the Govt. of India announced a National Mission on Quantum Technologies & Applications (NM-

[1] https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm

[2] https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[3] https://csrc.nist.gov/Projects/post-quantum-cryptography/workshops-and-timeline

[4] https://www.ibm.com/quantum-computing/

[5] www.rigetti.com

[6] quantumcomputer.ac.cn/index.html

QTA) [7] with a budget of rupees 8000 crores to work on quantum-computers and quantum-computing, quantum-communication, quantum-key distribution, quantum-encryption, and so on. Different proposals have been made to build quantum-safe cryptographic primitives including hash-based cryptography, multivariate cryptography,code-based cryptography, Lattice-based cryptography and Supersingular elliptic curve isogeny cryptography. These approaches are called post-resistant or quantum-safe cryptography and a major advantage of these approaches is they are compatible with existing crypto infrastructure.

1.1 Lattice-based Cryptography

Lattice-based cryptography stands out as an encouraging alternative to traditional cryptographic approaches. In 1996, Ajtai [5] described the hard problems in lattices from a cryptographic perspective. Since then, research has progressed steadily in this direction. Constructions based on lattices have several advantages in addition to resistance to quantum computers. In brief, constructions based on lattices involve simple matrix operations like addition and multiplication.

Lattice-based cryptosystems rely on worst-case hardness assumptions. That means, if the security of a lattice-based cryptosystem is compromised, it implies the ability to find the solution to any instance of a challenging lattice problem that has been established as NP-complete. This is not desirable in the typical security foundation of cryptographic schemes, because it is designed based on average-case problems. In average-case scenarios, the security is dependent on the ability to solve a randomly selected instance of the problem according to a specific probability distribution, should the cryptosystem's security be compromised.

Almost all cryptographic constructions can be done using lattices. Additionally, one can construct attribute-based encryption revealing the data only to the specific user who has a predicate and fully-homomorphic encryption performing operations on

[7] https://dst.gov.in/budget-2020-announces-rs-8000-cr-national-mission-quantum-technologies-applications

encrypted data. These are not known to be constructed from any other assumptions.

Lattice-based cryptography operates with relatively small integers as it employs matrices and vectors for computations within specific rings or fields of small order. These lattice-based algorithms maintain the balance between confidence of security and computational efficiency with key size, ciphertext, and signature lengths, which are desirable in the communication network.

This is represented in the NISTIR 8105[8]:"Report on Post-Quantum Cryptography" and NISTIR 8309[9]: "Status Report on the Second Round of the Post-Quantum Cryptography Standardization Process" released by National Institute of Standards and Technology (NIST) in 2016 and 2020 respectively. The National Security Agency (NSA)[10] also reviewed post-quantum cryptography algorithms from a cyber security perspective and promoted lattice-based cryptography over elliptic curve cryptography. The advancements in post-quantum cryptography indicate that it is nearing readiness for practical real-world applications. The security of lattice-based cryptosystems relies on two challenging lattice problems: Learning With Errors (LWE) and Short Integer Solution (SIS). The versions of these problems adapted for rings are known as Ring-LWE and Ring-SIS. Ring-SIS and Ring-LWE offers the advantage of greater efficiency and significantly smaller key sizes compared to security schemes built on non-ring variants.[6]. The LWE problem involves discovering a secret vector, s, given polynomial samples $A.s + e$, where e is the error vector chosen from a specified error distribution function, χ, and A is a uniformly generated matrix. The ring variant of LWE, known as $Ring - LWE$, is considered more practical in terms of computation cost and memory storage.

Lyubashevsky et.al. [7] stated and demonstrated that ring variants lattice problems are hard as finding the solution to worst-case problems in a special class of lattices [8].

When formulated over ideal lattices with Ring-LWE as the fundamental hard

[8] https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf

[9] https://www.nist.gov/publications/status-report-second-round-nist-post-quantum-cryptography-standardization-process

[10] https://www.nsa.gov/Cybersecurity/Post-Quantum-Cybersecurity-Resources/

problem, these cryptographic schemes demonstrate competitive key, signature, and ciphertext sizes, along with efficient computing times.[9].It can be used to build algorithms for encryption, signing, and key agreement from the basic building blocks.

The Ring-LWE problem involves the task of distinguishing between a uniform distribution and noisy ring multiplications in polynomial time. To illustrate, consider the challenge of distinguishing A and $< A.s + e >$), where A is a randomly chosen element from a ring, and s and e are sampled from their respective distribution functions.

There are various methods for discrete sampling: Bernoulli, Knuth-Yao, rejection, and cumulative distribution table. The post-quantum secure protocols are essential for current client-server environment scenarios. However, most protocols designed are not suitable for devices with limited computational resources. In this thesis, we proposed Ring-LWE-based post-quantum authentication schemes for IoT environments at the node level, cloud server level, and homomorphic encryption at the cloud server level to enhance performance and security against lattice attacks.

1.2 Post-Quantum Blockchain

Hash functions play a crucial role in blockchains, serving various purposes such as generating user addresses (private/public keys), shortening public addresses, and linking blocks for transactions occurring simultaneously. Widely used hash functions in blockchains, such as $SHA-256$ or *Scrypt*, are chosen for their characteristics of being easy to verify while computationally challenging to forge. This property enables users to create digital signatures, facilitating authentication for both users and their data transactions within the blockchain network.

Bitcoin employs ECDSA for authenticating transactions within the blockchain. Each Bitcoin address represents a cryptographic hash derived from the ECDSA public key.

Any transaction that is sent must be signed, and we can always check the validity of that signature and who has signed it. The current blockchain employs the recommended 192-bit elliptic curve domain parameters, specifically the curve $secp192k1$ proposed by Daniel and R. L. Brown[10]. In view of the latest advancement in quantum computers, the encryption schemes underlying current blockchains are based on intractability assumptions for conventional adversaries that might not always hold for quantum adversaries. In particular, blockchain technology relies on ECDSA for transaction authentication, which will be vulnerable to quantum adversaries.

If Shor's algorithms[2] are employed to compute a user's private key from a given public key, legitimate users face the risk of losing all their assets and privacy. This scenario can lead to the unauthorized signing of various transactions or the forging of a user's signature by malicious attackers. Additionally, Grover's algorithm [4]poses a threat by efficiently searching for data and solving hash functions, uncovering collisions in a hash space of size 'n' with a complexity of $O(\sqrt{n})$, whereas classical searching algorithms have a higher complexity of $O(n)$.Quantum computing invalidates blockchain in two ways.

First, hash inversion is assumed as computational hard problem. If a quantum computer can simplify this process, then the blockchain's authenticity and legitimacy are threatened. Indeed, Grover's search algorithm outperforms traditional brute-force methods in retrieving the preimage of a function value at a considerably faster rate. From the given input, it generates output and compares it with other outputs to isolate the input. Second, a quantum computer could compromise any component of a blockchain implementation that is dependent on private or public key cryptography, such as data communication or signature algorithm.

To address potential threats to blockchain systems, Designing a post-quantum blockchain is imperative in light of emerging threats. Therefore, leveraging the benefits of the $Ring-LWE$ hard problem, we propose a post-quantum blockchain with a lattice-based signature scheme with provable security. The proposed blockchain builds upon the aforementioned $Ring-LWE$ hard problems; it replaces ECC-based operations with lattice-based constructions.

We suggest implementing countermeasures to enhance the security of classical signature schemes. Consequently, we proposed the integration of a post-quantum blockchain featuring a modified $Ring - TESLA$ signature algorithm as a replacement of traditional ECDSA. Hence, this ensures that communications stay secure and highly efficient, even in the presence of large-scale computers of type quantum computation enabled. Additionally, we assess the vulnerability of the current lattice-based signature schemes, such as the GLP scheme (CHES 2012)[11], BLISS (CRYPTO 2013)[12], and the ring-TESLA (AfricaCrypt 2016)[13], as well as their implementations. We examine several attacks, including randomization attacks and zeroing attacks.

1.3 Related Work

In this section, A comprehensive overview of the related work associated with the four contributions outlined. We analyze existing schemes, evaluating their security and efficiency in comparison with the proposed solutions. The related work is organized into the following subsections.

1.3.1 Public Key Cryptography (PKC) based Authentication schemes

In IoT-based networks, verifying node/user identity requires unique and intrinsic properties satisfied to complete the secure authentication process. Therefore, many protocols are proposed based on various properties such as password-based user authentication [14], Token-based transaction authentication [15], One-time password (OTP) based authentication [16], Location-based authentication [17], and Biometric-based authentication [18] in wireless communication networks such as IoTs.

Majid Mumtaz et al. [19] proposed an authentication system for a smart IoT environment based on RSA. In this model, the RSA-based Public Key Infrastructure (PKI) is used to provide authentication. Transparent proxies are used to access applications, while the IDentity Management System (IDMS) stores credentials. In this case, the IDMS server, CA server, and the Policy Decision Point (PDP) authorization server assist the authentication server (AS) in validating node identity, which may lead to server impersonation attacks and communication overhead. In this case, to retrieve the public key from the microSD the secret PIN is used where as for decryption and digital signing, the private key stored inside the card will be used. but, the login device will be known by the attacker that violates the device anonymity and the microSD can be easily tampered, it could be a malware attack vector that causes a man-in-the-middle attack.

Sheetal Karla et.al. [20] proposed a mutual authentication system based on ECC for secure communication between cloud servers and embedded devices via HTTP cookies. They claimed in the security analysis that the method is resistant to a variety of attacks and it satisfies all security requirements. But, Saru Kumari et.al. [21] demonstrated how Kalra and Sood's authentication scheme is vulnerable to insider attacks and off-line password guessing attacks. She also proved the scheme failing to achieve mutual authentication, session key agreement and device anonymity.

The design and development of authentication protocols that are suitable for requirements of IoT environment is a relatively new area of research. In the literature, number of two-factor authenticated key exchange (2-AKE) schemes are more than number of three-factor authentication key exchange (3-AKE) schemes. Wang et al. [22] demonstrated a list of the most recent 2-AKE. However, due to many security flaws, the majority of them are far from satisfactory.

Nam et al. [23] presented a 2-Factor Authenticated Key Exchange Scheme for WSN that provides user anonymity. This scheme makes use of ECC, but only for anonymous user to gateway device authentication and allowing sensor devices to compute only light-weight operations. This protocol ensures the user anonymity in the Random Oracle Model (ROM) under the $ECCDH$ assumptions in the group G, as well as the symmetric encryption scheme's security.

Yang et al. [24] attempted to solve the bottleneck of anonymous credentials and proposed a lightweight Authenticated Key Exchange protocol for IoT applications using a Dynamic Accumulator (DA). A large number of values are combined into a single value known as the accumulator in this scheme. Each accrued value has a witness, which proves that the accumulated value is still present in the accumulator. The proof can be performed with zero knowledge, which means that no clue about the witness or the value is revealed.

Turkanovic et.al [25] proposed a scheme for wireless sensor networks based on the IoTs notations which ensures a user authentication and key agreement. In this, a lightweight computation based secure key agreement allowed to a remote user to negotiate a session key with a sensor node. Although the user never contacts the gateway node, the scheme ensures mutual authentication between the user, gateway node and sensor node. However, it employs basic hash and XOR calculations, resulting in high risks and challenges posed by IoT.

Khan et al. [26] proposed a finger print based remote authentication scheme for mobile devices. It has been demonstrated that this scheme is vulnerable to desynchronization attacks and user impersonation attack. The user anonymity also not ensured by the scheme. Fan Wu et.al. [27] proposed an ECC based three-factor remote authentication scheme and provided formal proof with solid forward protection to solve the drawbacks. It is secure and has the potential to protect the user's privacy.

H.S.Islam et.al. [28] analyzed several problems of Lin's scheme [29] based on the chaotic maps. Then, in 2016, he introduced an dynamic identity-based user authentication system for mobile users. He claimed it is efficient and robust that based on security of extended chaotic maps. He demonstrated the scheme's correctness and security using the chaotic map-based Diffie–Hellman (CDH) problem's hardness assumption. But, the scheme is designed for mobile devices and the computation cost is comparatively very high for the latest trend of protocols for constrained devices.

Debiao He et.al. [30] introduced a bilinear pairing based anonymous mobile user authentication protocol. The proposed protocol includes a bilinear pairing operation and a hash operation to improve user performance. This protocol is analyzed for its security that ensures and it can meet the requirements of security for multi-server architectures. The user must perform eight secure hash operations, two scalar multiplication operations, one point addition operation and two exponentiation operations in this protocol. Therefore, the computation cost for this protocol is high, making it unsuitable for resource-constrained devices.

Challa et. al.[31] proposed an ECC-based authentication scheme. In this case, if both parties mutually authenticate each other, a valid user may access data collected from a sensing system of IoT application. They can generate a secret session key to use in future secure communications. To protect against replay attacks, the proposed scheme uses random numbers and current timestamps. As a result, it is assumed that all entities in the Internet of Things world have their clocks synchronized. However, it has been demonstrated that Challa's 3-factor AKE scheme is vulnerable to a various attacks [32], including quantum attacks.

Li et al. [33] proposed a RFID based AKE scheme with reader cache to address flaws in previous schemes for IOT medical services. The security flaws in Fan et al. scheme [34] LRMAPC are caused by the protocol's relying on r_0, r_1 random values, and k tag key sent through public channel. As a result, an attacker can easily generate $(Query, r_A)$ to mask himself as the valid reader. Furthermore, the reader is not authenticated before accessing the database. The lack of authentication causes tag forgery attacks. The Fan's scheme is redesigned by storing the secret key r_k and reader keys R_K where $\forall r_k \in R_K$ of the each valid user in the back-end database.

jia et.al [32] recently proposed a 3-factor key exchange scheme for the internet of things based on the Challa et. al. (2017)'s [31] authenticated key exchange scheme, which lacks untraceability, anonymity. This scheme also vulnerable to impersonation attacks, stolen smart card attacks, and offline password guessing attacks. In the extended security model, the improved 3-factor authenticated key exchange scheme secures security. This enhanced scheme is also compared with schemes of Challa et al. [31] and Turkanovi et. al. [25].

The majority of current authentication and key exchange schemes are based on ECC,in which key computation part is scalar-point multiplication. The scalar-point multiplication is significantly more expensive than symmetric encryption/decryption, hash function evaluation, and MAC generation/verification[35][36]. Furthermore, despite the use of ECC, the majority of schemes fail to achieve user anonymity,mutual authentication and impersonation attack [37][38]. Some schemes are computationally efficient by employing energy-reducing algorithmic engineering technique, such as Energy Complexity Model (ECM)[36][39] [40], but have the inherent flaws of user anonymity and are vulnerable in presence of quantum attacker.

1.3.2 Lattice-based Authentication schemes

This section describes lattice-based authentication schemes proposed based on the hardness of LWE problem and its variants, $Ring-LWE$ and $Ring-SIS$. At first, J.Ding et al. [41] coined the first lattice-based key exchange protocol based on LWE problem, that is provably secure. His work is the initiation for many key exchange protocols proposed currently using LWE and Ring-LWE. In his work, the property of commutative and approximation equivalence to build key exchange cryptosystems. The reconciliation mechanism designed in the protocol is used to exchange signals between two parties and two close values without errors. For reconciliation of errors with high probability, the norm difference is chosen carefully in such a way that it is modulus q bounded. This protocol differs from the standard Diffie-Hellman protocol in that it cannot provide authentication. As a result, this scheme is vulnerable to MIM attacks, as demonstrated by Zhang et.al [42]. However, the same concept of reconciliation is used to construct other protocols similar to this work. Later, research towards the implementation of lattice-based authentication and key exchange (AKE) protocols was done by Ahmad Boorghany et al. [43] on AVR(8-bit) and ARM(32-bit) processors and they used Fast Fourier Transformation(FFT).

Zhang et al.[42] proposed the first practical 2-pass AKE scheme based on ideal lattices that are provably secure. This scheme is a Ring-LWE variant of the standard HMQV protocol [44]. To demonstrate the security, the Bellare-Rogaway model [45] is used with weak perfect forward secrecy. One-pass variation of the scheme is also demonstrated for specialized purposes and security options provided ranging from 80 bits to 360 bits.

J.W.Bos et al.[46] used a 128-bit secure parameter choice to implement Peikert's RLWE key exchange mechanism [47]. The key exchange described by Peikert is nearly identical to the technique described in Ding et al [41] secure key exchange protocol. Bos et al. demonstrated Peikert's protocol and parameter selection in a proof-of-concept implementation. The protocol is then integrated with OpenSSL, resulting in post-quantum Transport Layer Security cipher suites RingLWE-ECDSA (RSA)-AES128- GCM-SHA256, which includes digital signatures (ECDSA or RSA). To demonstrate the security of proposed cipher suites, they used the authenticated-confidential channel establishment (ACCE) model.

Alkim et al. [9] proposed the enhanced scheme from the Peikert et al. [47] and J.W.Bos et al.[46] scheme. In this scheme, they suggested more optimized parameters and error distribution mechanisms for less expensive sampling and proved the security easily. The efficient error-reconciliation method proposed in this paper helps to defend against backdoor attackers. The security parameter size is increased to double and claimed that the communication overhead is reduced to halve along with computation cost by following the same implementation of the J.W.Bos scheme [46] using the ARM Cortex-M0 processor.

Feng et al.[48] proposed a protocol for authentication of mobile devices based on an ideal lattice with user anonymity, which he claimed was the first of its kind in the post-quantum world. The protocol is analyzed for informal security. The outcome demonstrates that the protocol can satisfy the security requirements of mobile client-server architecture. Then, using the Random Oracle Model(ROM) and the Ring-LWE problem, they demonstrated that it is provably secure. Finally, the protocol is implemented in a client-server architecture created using a mobile device and computer. This protocol is being considered for comparison with our proposed protocol, and the computation overhead is reduced, which is essential for constrained devices. These schemes in the literature ensure that post-quantum cryptography can be applied on resource constraint devices such as IoT and lattice-based constructions can be more strong than traditional PKC schemes.

Guneysu et al. [11] proposed a signature technique (GLP) based on the lattice-primitives. On ARM Cortex-M4F(32-bit) microcontroller, Oder et al.[49] described an implementation of Ducas et al.[12] signature called BLISS. The different gaussian sampling methods are proposed on the microcontroller, such as Bernoulli sampling [12], Knuth-Yao sampling [50], and Ziggurat sampling [51]. The NTT and sparse multiplication approaches were investigated for polynomial arithmetic. Poppelmann et al. [52] optimized the BLISS signature scheme by removing some of the bit-reversal operations and reducing multiplication operations which is named BLISS-B. It is implemented on 8-bit AVR architecture at the same security level of BLISS.

Akleylek et al. [13] proposed first provably secure lattice-based signature method called Ring-TESLA in the year 2016. In this approach, the Ring-LWE problem is used for the tight security reduction that allows to prove the security along with efficient instantiation. On the basis of experimental data from software implementation, it is demonstrated that Ring-TESLA outperforms both GLP and BLISS systems. Later, Barreto et al. [53] enhanced the ring-TESLA scheme. This RLWE-based digital signature technique allows considerably faster key pair generation, signatures, and verification. This scheme outperforms majority of the traditional and lattice-based signature schemes on current processors. The following table 1.1 lists some lattice-based schemes that have been found in the literature.

1.3.3 Homomorphic Encryption Schemes

The term ”privacy homomorphism” and Homomorphic Encryption (HE) concepts were initially used by the researcher Rivest in the year 1978.[61]. This holomorphic encryption method allows encryption or modification of ciphertext directly. The fundamental concept is that when plaintext is subjected to addition or multiplication, it exhibits a comparable behavior to ciphertext after undergoing encryption. Before the formalization of homomorphic encryption, certain encryption systems had already demonstrated a level of homomorphism. One of its type is the Hill Cipher encryption scheme, that relies on the principles of linear algebra, advanced matrix manipulation, and rules of modulo arithmetic. It is a more mathematical cipher compared to other schemes and it meets the additive homomorphic property [62]. Later, with the introduction of a privacy homomorphic scheme, the era of homomorphism has been started and many schemes were proposed. The RSA algorithm is based on an IF problem which follows the homomorphic multiplicative property[61, 63].

In 1984, Goldwasser and Micali introduced the first PKE algorithm with semantic security. A probabilistic encryption technique named GM algorithm is proposed

Table 1.1: Lattice based Protocols

Scheme	Lattice Primitive	Lattice Type	Security	ZK/WH	Orig.Scheme
Lyubashevsky [7]	SVP	Ideal	Concurrent	WH	Authentication
Kawachi et al. [54]	GapSVP	General	Concurrent	WH	Authentication
Xagawa et al. [55]	NTRU	General	Active	ZK	Authentication
Lyubashevsky [56]	SVP	General	Concurrent	WH	Signature
Cayrel et al.[57]	SIS	Ideal	Concurrent	ZK	Authentication
Silva et al. [58]	LWE	General	Active	ZK	Authentication
Guneysu et al.[11]	R-LWE	Ideal	Passive	-	Signature
J.Ding et al. [41]	LWE	General	Active	WH	Key Exchange
Lyubashevsky [59]	R-LWE	Ideal	Concurrent	WH	Signature
L.Ducas et al.[12]	NTRU	General	Passive	-	Signature
T.Oder et al.[49]	NTRU	General	Passive	WH	Signature
Boorghany et al.[43]	NTRU	General	Concurrent	WH	Authentication
Zhang et al.[42]	R-LWE	Ideal	Concurrent	WH	Authentication
Bos et al.[46]	R-LWE	General	Active	WH	Key Exchange
Dousti et al. [60]	SIS	General	Active	ZK	Authentication
v Alkim et al.[9]	R-LWE	General	Active	WH	Key Exchange
Akleylek et al.[13]	R-LWE	General	Passive	WH	Signature
Bareto et al.[53]	R-LWE	General	Passive	WH	Signature
Feng et al. [48]	R-LWE	Ideal	Active	WH	Authentication

Note:ZK :zero knowledge ; WH :witness hiding.

based on quadratic residue modulo and trapdoor function[64]. However, this algorithm showed low efficiency and satisfies only additive holomorphic encryption property. In 1985, the ElGamal Cryptosystem was introduced as an asymmetric encryption algorithm that relied on a combination of elliptic curve cryptosystem and public key cryptosystem [65]. This cryptographic system showcases a multiplicative homomorphic property, making it applicable for both encryption and signature verification purposes. In 1994, Benaloh made enhancements to the probabilistic encryption algorithm [66], enabling it to encrypt a specific number of bits r at once. However, this improved technique lacked full holomorphic capabilities and only supported additive homomorphism.

In the year 1999, the most popular Paillier encryption scheme was proposed by was proposed by[67] based on the quadratic residue.It is a probabilistic encryption scheme that enables additive homomorphic operations.

In 2005, Boneh et. al., introduced the (Boneh,Goh,Nissim) BGN cryptosystem, leveraging bilinear pairing [68]. The cryptographic algorithm demonstrates both additive and a singular multiplicative property, positioning it as the nearest scheme to the concept of homomorphism. In contrast, other algorithms like GM and Paillier solely satisfy additive homomorphism, whereas RSA, ElGamal, and BGN provide multiplicative homomorphism. However, uniquely satisfies both multiple additions and a single multiplication operation. Due to the support for either additive or multiplicative homomorphism, these algorithms are categorized as either single or partially homomorphic encryption algorithms.

Gentry et.al.[69] introduced the first lattice-based homomorphic encryption technique in 2009 based on ideal lattices. This technique supports full homomorphism with additive and multiplicative homomorphic properties. It means it supports for addition and multiplication of ciphertext with an unlimited number of times. Later, the homomorphic encryption schemes evolved rapidly, and many schemes were proposed by researchers. There are three distinct categories of homomorphic encryption(HE) schemes. The first category encompasses a fully homomorphic encryption scheme proposed by Gentry. This scheme involves constructing a Somewhat Homomorphic Encryption (SWHE) on the ideals of different rings. It employs techniques such as compressing the decryption circuit to reduce polynomials and utilizes bootstrapping technology to achieve fully homomorphic encryption, assuming cyclic security.

The second category comprises an integer-based HE scheme [70]that aligns with

Gentry's concept but eliminates operations based on ideal lattices of the polynomial ring. Instead, all operations are performed using integers.

In the third category, homomorphic encryption methods are based on either fully homomorphic LWE or Ring-LWE. These schemes rely on the hardness of the LWE to attain fully homomorphic encryption capabilities.

This method uses non-linearization to build a fully homomorphic encryption system, similar to the *BGV* encryption scheme and is based on fault-tolerant learning [71].

1.3.4 Analysis of Lattice signature vulnerabilities

This section discusses lattice signature schemes derived from the LWE problem and its variations, including Ring-LWE and Ring-SIS. We assess the vulnerability of current lattice-based signature schemes: the GLP scheme (CHES 2012), BLISS (CRYPTO 2013) [12], and the ring-TESLA (AfricaCrypt 2016) [13]. We consider various attacks, including randomization, zeroing, and skipping. There is need for countermeasures for vulnerabilities to improve the security of lattice-based signature schemes. As a result, we modified the Ring-TESLA signature algorithm to construct the post-quantum blockchain (PQB).

Guneysu et al. [11] proposed *GLP* signature scheme based on the hardness of $Ring - LWE$. In *GLP*, the secret key is computed from secret, error polynomials $s, e \overset{\$}{\leftarrow} R_{q,[1]}$ whose coefficients lies in {-1,0,1}. The public key b computed as $b = a.s + e(\mod q)$ with random polynomial $a \overset{\$}{\leftarrow} R_q$. For given input message μ, the signature algorithm generates $y_1, y_2 \overset{\$}{\leftarrow} R_{q,[k]}$ then computes hash c_1 with $ay_1 + y_2, \mu$. Then, two polynomials z_1, z_2 for signature are computed with the hash value and secrets. The rejection sampling is used to hide the secret and any of the signature polynomials z_1 or z_2 is compressed to return the signature with some probability. For a given hash value c_1 and signature polynomials z_1, z_2, the verification algorithm checks the size of the z_1, z_2 and the equality of the newly computed hash $c_2 = (az_1 + z_2 * -tc, \mu)$ with the hash value c_1. The practical implementation parameter sets GLP-I, GLP-II, and GLP-III are suggested for the efficiency of the scheme. Guneysu et al. in [11].

Ducas et al.[12] introduced a signature scheme called $BLISS$ based on the hardness of $Ring-SIS$ on $NTRU$-like lattices. The key pairs are generated $NTRU$-like which must satisfy rejection sampling. The public key A is generated from two polynomials f, g chosen from the distribution function χ. The public key $A = (a, 1) \in R_q^{1\times2}$ where $a = g/f$. The private key is a short matrix $s \in Z_{2q}^{m\times n}$ and the public key A must satisfy the relation $A.s = qI_n(\mod 2_q)$. In the signature algorithm, to sign on a message μ, a vector y is sampled from the Gaussian distribution function D_σ^m where σ is the std. deviation. The hash value c_1 of the message is computed $(Ay \mod 2_q, \mu)$ then generates the output value $z = y + (-1)^b s.c_1$ by sampling the bit b from $\{0, 1\}$. The rejection sampling is applied to return the signature z, c_1 with some probability. In the verification algorithm, for a given signature z, c_1 it verifies for conditions $||z|| \leq B_2$ where bound $B_2 = \nu\sqrt{m\sigma}$, ν is security parameter and $m, \sigma \in D_\sigma^m$. The new hash value $c_2 = H(Az + qc \mod 2_q), \mu$ is computed and the equality of c_1 and c_2 is verified.

On ARM Cortex-M4F(32-bit) micro-controller, Oder et al.[49] described an implementation of Ducas et al.[12]. The BLISS signature scheme is optimized by Poppelmann et al. [52] by removing some of the bit-reversal operations and reducing multiplication operations, which is named BLISS-B. It is implemented on the 8-bit AVR architecture at the same security level as BLISS.

Akleylek et al. [13] proposed $Ring-TESLA$ signature algorithm based on the hardness of $Ring-LWE$ problem. The secret key sk is the combination of the one secret s and two error polynomials (e_1, e_2) sampled from D_q^m. The public key $pk = (b_1 = a_1.s + e_1, b_2 = a_2.s + e_2)$ computed by randomly sampling polynomials $a_1, a_2 \overset{\$}{\leftarrow} R_q$. To sign a message μ, a random polynomial $y \overset{\$}{\leftarrow} R_{q,[B]}$ sampled to compute the hash value c. The hash c is computed from product of $(a.y_1, a.y_2)$ and the message μ. It is encoded as a polynomial c'. Finally, rejection sampling is applied to return the signature, which consists of the polynomial $z = (y + s.c)$ and encoded polynomial c' i.e. (z, c'). To verify the signature, the equality of the c' is verified with the newly computed hash value $c'' = H(\lfloor a_1.z - b_1.c\rceil_{d,q}, \lfloor a_2.z - b_2.c\rceil_{d,q})$ and the size of the z. On the basis of experimental data from software implementation, it is demonstrated that Ring-TESLA outperforms both *GLP* and *BLISS* systems. Later, Barreto et al. [53] enhanced the ring-TESLA scheme.

1.4 Contributions

This section outlines the contributions made in this thesis, with each subsection providing a summery of the primary objectives and contributions. In this thesis, we introduced four Post-Quantum authentication schemes based on Ring-LWE to enhance the security of IoT and Blockchain Applications. The initial three schemes concentrate on delivering authentication and encryption within an IoT-cloud environment, leveraging the Ring-LWE problem. These schemes aim to secure against known attacks while upholding computational efficiency, rendering them more effective than existing lattice-based authentication schemes in the existing literature. The fourth scheme involves constructing a post-quantum blockchain utilizing a modified Ring-TESLA signature algorithm, offering defense against quantum adversaries in the blockchain network. Additionally, we propose a randomized consensus proof-of-stake (PoS) to address the dominant validator problem and preserve decentralization among the nodes. A concise overview of these proposed schemes follows below.

- Scheme 1: "Lattice-based authentication and key exchange protocol for Internet of things"

  In this, we proposed and validated a lattice-based authentication and key exchange protocol for the IoT environment. The protocol's security is based on LWE in a polynomial ring; shortly, we call Ring-LWE. The advantage of Ring-LWE is the reduced communication overhead and key size by representing the matrix as degree n polynomials in R_q. The protocol correctness is proved formally and verified with the standard verification tool AVISPA for authentication. The informal protocol analysis demonstrates that it is secure against known attacks on the internet of things environment. The protocol's performance is analyzed and compared with relevant protocols. It shows that the communication cost is the same as other protocols, and the computation cost is minimal.

The contributions of this scheme can be summarized as follows:

1. We proposed a design of a lattice-based authentication protocol for the IoT based on the hardness of $Ring - LWE$ problem. The protocol model is suitable for IoT cloud environment scenarios for quantum-safe authentication.
2. We examined the protocol's security against the quantum adversary and demonstrated its correctness, as well as its resistance to known security attacks and threats.
3. We conducted a performance evaluation of the protocol, focusing on computation and communication overhead.
4. We practically implemented the proposed protocol in IoT client-server scenarios and presented results.

- Scheme 2: "Quantum-secure node-to-node authentication protocol model for IoT sensor networks"

  In this, the recently proposed lattice-based authentication scheme is analyzed [72], and we proposed a quantum-secure node-to-node authentication protocol model for IoT sensor networks by making use of the scheme design[73]. The protocol is modeled, and its correctness is proved formally based on Inhomogeneous Short Integer Solution ($ISIS$) problem on lattices. We verified the protocol model's security against known attacks on the IoT infrastructure. Our analysis involved considering the widely recognized three-party protocol model to evaluate the proposed model's performance. It is analyzed for a 100-bit security level with specified security parameters. The average computation cost is computed for the number of hash functions($h(.)$) and polynomial multiplication(PM) operations. The proposed model, required $2h(.)+3PM$ for the IoT node,$1h(.)+2PM$ for the Gateway device, and in total $3h(.)+5PM$ operations are needed. We also compare our protocol model to similar protocols and demonstrate that it is both computationally efficient and quantum-safe. The adversary is unable to extract any information from the communication among IoT nodes. The contributions of the scheme are summarised as follows:

1. We proposed a node-to-node authentication protocol model for IoT infrastructure. It is suitable for IoT cloud environment with quantum-safe authentication of nodes.
2. The protocol ensures the anonymity of the IoT user's identity, which is known only to the gateway device. No adversary Adv can get the identity information.
3. We analyzed the protocol's security against the quantum adversary and proved its correctness and safety from known attacks and threats.
4. We presented the performance evaluation of the protocol in terms of computation and communication overhead.

- Scheme 3: "Construction of communication protocol using Ring-LWE-based homomorphic encryption in IoT-cloud environment"

  In this, we proposed a homomorphic encryption scheme for the security and privacy of user data in a cloud environment. Various types of homomorphic encryption schemes are studied for data privacy in the cloud. The Ring-LWE-based homomorphic encryption scheme is proposed for privacy in the cloud environment which meets the homomorphic properties. In this scheme, IoT nodes will register at the cloud server, then the server authenticates IoT nodes and accepts the encrypted data to share with other nodes whenever requested. It stores the data on a cloud server with quantum-safe encryption. For the encryption of the data at the cloud server, the Ring-LWE based fully homomorphic encryption(FHE) is used for quantum-enabled security and privacy. The proposed scheme is evaluated for its security and compact in the presence of a quantum attacker. By employing FHE-based data management and verification methods, the overall efficiency and security surpass those offered by traditional encryption algorithms. Additionally, the implementation of signature verification minimizes the associated overhead, thus enhancing the efficiency of the verification process when compared to existing methods. computation overhead. The contributions of the scheme are summarised as follows:

1. We constructed a communication protocol for authenticated user message encryption in an IoT cloud computing environment, based on Ring-LWE homomorphic encryption.
2. We proposed the evaluation function in holomorphic encryption is defined based on Ring-LWE encryption for a practical sharing-enabled cloud storage.
3. We conducted a formal analysis and provided security proofs for the proposed protocol against classical and quantum attacks in cloud environments, including Man-in-the-middle (MITM) attacks, Denial of Service (DoS), and Replay Attacks.

- Scheme 4: “Post Quantum Blockchain with Provable Security”

  In this, we provided an overview of modern Blockchain network vulnerabilities to quantum adversaries, as well as some post-quantum mitigation strategies. Then, a post-quantum blockchain is constructed using a modified $Ring{-}TESLA$ signature algorithm that defends the blockchain network against quantum adversaries. The security of the proposed signature scheme is based on $Ring-LWE$, or LWE in a polynomial ring. The advantage of $Ring-LWE$ is the reduced communication overhead and key size by representing the matrix as degree n polynomials in R_q. We also propose a randomized consensus, proof-of-stake (PoS) to avoid the dominant validator problem and maintain the decentralization property among the nodes. We provide comprehensive security proof and analysis in the against a quantum adversary. This scheme provides support for the development of future quantum-resistant blockchain applications. The contributions of the scheme are summarized as follows:

  1. We evaluated the Ring-LWE-based lattice signature schemes and their vul-

nerabilities to quantum attacks and modified the Ring-TESLA signature scheme.

2. We constructed the Post Quantum Blockchain (PQB) architecture. It uses the modified Ring-TESLA signature algorithm. It support for secure blockchain applications that resist quantum attacks.
3. We proposed a randomized consensus, Proof-of-Stake (PoS) to minimize the advantage of a quantum adversary.
4. We analyzed the proposed post-quantum blockchain (PQB) and demonstrated its correctness and security in the presence of an adversary under the assumption of the Ring-LWE hard problem.

1.5 Organization of the Thesis

In Chapter 2, we discuss all the basic definitions, hard problems in lattices, and various cryptographic primitives. Chapter 3 presents the scheme-1 for the IoT node-to-cloud server authentication, Chapter 4 presents scheme-2 for a Node-to-Node authentication, and Chapter 5 presents scheme-3 for cloud data management using Ring-LWE based Homomorphic encryption. In Chapter 6, we present scheme-4, a post-quantum blockchain model using the Ring-LWE-based signature algorithm for transaction authentication along with randomized consensus for the proposed blockchain. Finally, Chapter 7 concludes the thesis providing some directions for further research.

Chapter 2

Preliminaries

This chapter provides a concise overview of preliminaries, explains the concept of lattices and their associated challenging problems and outlines several lattice-based cryptographic primitives used in this thesis.

The organization of this chapter is as follows: Section 2.1 presents the various notations used in the thesis. Section 2.2 presents the lattice, various parameters related to the lattice, and some specific lattices like ideal lattices and modular lattices. Hard lattice problems are also explained in Section ?? in this thesis.

2.1 Notations

Natural numbers, real numbers, and integers and represented by $\mathbb{N}, \mathbb{R}$ and $\mathbb{Z}$ and respectively. Let K be positive integer, the set $\{1, 2, \ldots, N\}$ is represented by $[K]$ and the set $\{-K, \ldots, 0, \ldots, K\}$ is represented by $[-K, K]$. For any $q > 0$, $\mathbb{Z}_q$ represents the set $\{0, \ldots, q-1\}$. All the vectors are assumed to be in column form. Let $\mathrm{a} \in \mathbb{Z}^{n_1}$ and $\mathrm{b} \in \mathbb{Z}^{n_2}$ be two vectors, then $\mathrm{a}||\mathrm{b} \in \mathbb{Z}^{n_1+n_2}$ denotes concatenation of two vectors. For two matrices $\mathrm{A} \in \mathbb{Z}^{n_1 \times m}$ and $\mathrm{B} \in \mathbb{Z}^{n_2 \times m}$, row concatenation is represented by $[\mathrm{A}|\mathrm{B}] \in \mathbb{Z}^{(n_1+n_2)\times m}$. For two matrices $\mathrm{A} \in \mathbb{Z}^{m \times n_1}$ and $\mathrm{B} \in \mathbb{Z}^{m \times n_2}$, column concatenation is represented by $[\mathrm{A}||\mathrm{B}] \in \mathbb{Z}^{m \times (n_1+n_2)}$. Matrix A transpose is represented by $\mathrm{A}^\top$. The l_2 and l_∞ denotes the norm $\|\cdot\|$ and $\|\cdot\|_\infty$, respectively. The norm of a matrix $\mathrm{A} \in \mathbb{Z}^{n \times m}$ that has columns $(\mathrm{a}_i)_{i=1}^m$ is defined as the norm of $A's$longest column (i.e., $\|\mathrm{A}\| = max_i \|\mathrm{a}_i\|$). If the columns of $\mathrm{A} = (\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_m)$ are linearly independent, then $\widetilde{\mathrm{A}} = (\tilde{\mathrm{a}_1}, \tilde{\mathrm{a}_2}, \ldots, \tilde{\mathrm{a}_m})$ denote the Gram-Schmidt orthogonalization of vectors $\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_m$ in the same order. For a finite set S, sampling a value x accoding to distribution D is represented by $x \hookleftarrow D_S$.

2.2 Lattices

In this section, we discuss the lattice definition and its parameters. We briefly discuss types of lattices such as ideal lattices, modular lattices, and other relevant terminology. we also discuss the Gaussian distribution over lattices, a few classic lattice problems, and average case problems in lattices and ideal lattices.

2.2.1 Definition of Lattices

Let $\{\ \mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_m\ \}$ be a set of linearly independent vectors over $\mathbb{R}^n$. The lattice $\mathcal{L}$ formed by basis $\mathrm{B} = [\mathrm{b}_1|\mathrm{b}_2|, \ldots, |\mathrm{b}_m] \in \mathbb{R}^{n \times m}$ is denoted as:

$$\mathcal{L}(\mathrm{B}) = \Big\{ \sum_i \mathrm{b}_i z_i : z_i \in \mathbb{Z}\ \forall\ i \in [m] \Big\} = \Big\{ \mathrm{Bz} : \mathrm{z} = (z_1, z_2, \ldots, z_m)^\top \in \mathbb{Z}^m \Big\}.$$

Thus, $\mathcal{L}(\mathrm{B})$ is the set of all integer combination of m linearly independent vectors over $\mathbb{R}^n$. Integers n and m denote the dimension and rank of a lattice respectively. A lattice is a full-rank when $m = n$. The vectors $\mathrm{b}_1, \ldots, \mathrm{b}_m$ are called as basis of the lattice. We usually write $\mathcal{L}$ for a lattice formed by basis matrix B instead of $\mathcal{L}(\mathrm{B})$ if no ambiguity occurs. The lattice generated by the basis vectors $b_1 = (0,1)^\top$ and $b_2 = (1,0)^\top$ is $\mathbb{Z}^2$ i.e., set of all integer points in two dimension (see Figure 2.1).

The lattice $\mathcal{L}$ will have multiple bases if the rank of $\mathcal{L}$ is greater than 1, for example basis vectors $b'_1 = (1,1)^\top$ and $b'_2 = (2,1)^\top$ generate the same lattice $\mathbb{Z}^2$ (see Figure 2.2). The basis vectors $b''_1 = (1,1)^\top$ and $b''_2 = (2,0)^\top$ generate a different lattice, a set of all integer points whose coordinates sum to an even number (see Figure 2.3). All the lattices defined above are full-rank lattices in which dimension and rank are

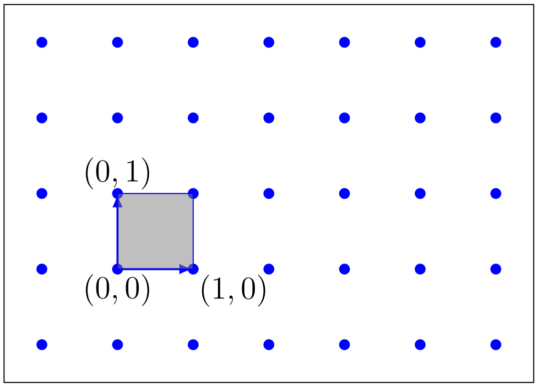


Figure 2.1: A basis of $\mathbb{Z}^2$.

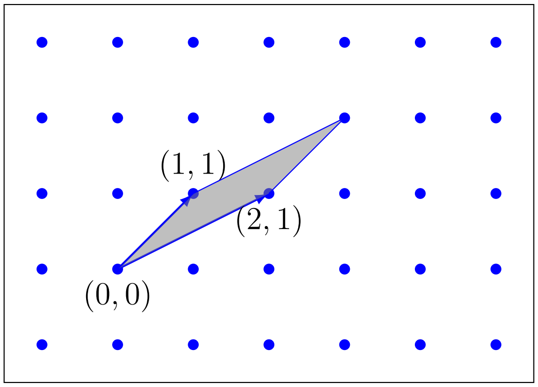


Figure 2.2: Another basis of $\mathbb{Z}^2$.

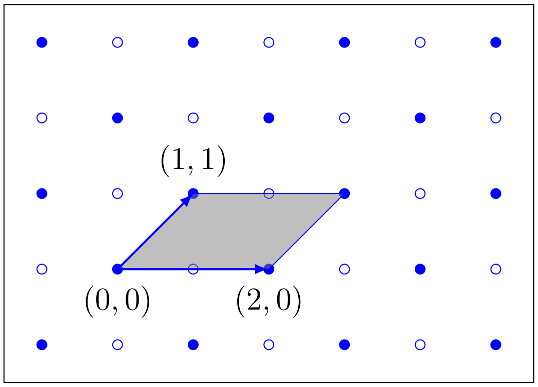


Figure 2.3: A different lattice.

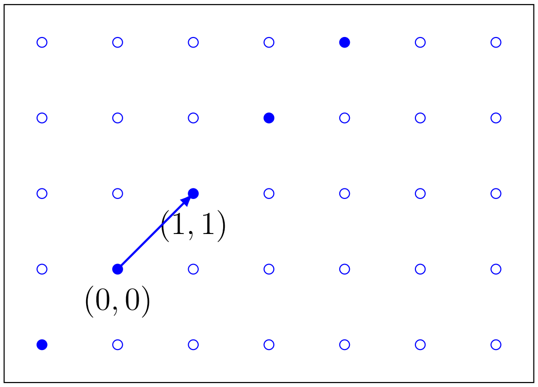


Figure 2.4: Not a full rank lattice.

the same. Figure 2.4 shows an example of a non-full rank lattice generated by basis vector $b = (1, 1)^\top$ in which rank is 1 and dimension is 2. The following lemma gives the condition when two bases generate the same lattice.

Lemma 2.2.1. (**citation**) "Bases B and B$'$ generate the lattice $\mathcal{L}$ iff there exists a unimodular matrix [1] U such that $\mathrm{B} = \mathrm{B}'\mathrm{U}$."

Definition 2.2.2. "The span of a lattice $\mathcal{L}$ generated by basis matrix $\mathrm{B} = [\mathrm{b}_1|\mathrm{b}_2|, \dots, |\mathrm{b}_m] \in \mathbb{R}^{n\times m}$ is the linear space spanned by basis vectors.

$$span(\mathcal{L}(\mathrm{B})) = \left\{ \sum_i \mathrm{b}_i z_i : z_i \in \mathbb{R}\ \forall\ i \in [m] \right\} = \left\{ \mathrm{Bz} : \mathrm{z} = (z_1, \dots, z_m)^\top \in \mathbb{R}^m \right\}\text{''}.$$

2.2.2 Parameters in Lattices

We recall various parameters of lattices that are used in defining lattice problems.

[1] A matrix $\mathrm{U} \in \mathbb{Z}^{n\times n}$ is unimodular if determinant of the matrix U is ±1.

2.2.2.1 Fundamental Paralleoped

Definition 2.2.3. For a lattice $\mathcal{L}$ generated by basis matrix $\mathrm{B} = [\mathrm{b}_1|\mathrm{b}_2|, \ldots, |\mathrm{b}_m]$, the fundamental parallelepiped $P(\mathrm{B})$ is defined as

$$P(\mathrm{B}) = \{x_i \mathrm{b}_i : 0 \leq x_i < 1\}.$$

In Figure 2.1, 2.2, and 2.2, the shaded region denotes the fundamental parallelopiped formed by the corresponding basis. Consider a full rank lattice $\mathcal{L}$ with rank n, not every set of n linearly independent vectors in $\mathbb{Z}^n$ forms the basis of $\mathbb{Z}^n$. For example, in Figure 2.3 vectors $b_1'' = (1,1)^\top$ and $b_2'' = (2,0)^\top$ are linearly independent but they are not the basis for $\mathbb{Z}^2$. The fundamental parallelopiped helps to decide if any set of n linearly independent vectors of $\mathcal{L}$ can act as a basis. A set of linearly independent vectors in the lattice is a basis if the fundamental parallelopiped generated by those vectors does not contain any other lattice point except origin. For example, in Figure 2.2 the fundamental parallelopiped region contains the non-zero lattice point $(1,0)$. Whereas in Figure 2.1, and 2.2 there are no lattice points in the fundamental parallelopiped region except origin.

Lemma 2.2.4. "Let $\mathcal{L}$ be a lattice with rank n and $\mathrm{C} = [\mathrm{c}_1|....|\mathrm{c}_n]$ be a set of linearly independent vectors in $\mathcal{L}$. The matrix C is a basis if and only if $P(\mathrm{C}) \cap \mathcal{L} = \{0\}$."

2.2.2.2 Determinant

The lattice's determinant is denoted as the volume of the fundamental parallelepiped. It can be computed using orthogonal vectors of the basis. Given a basis of the lattice, its orthogonal basis can be created using Gram Schmidt Orthogonalization (GSO) process and it is described in Algorithm 2.1.

We can observe that the vectors obtained using GSO process are pairwise orthogonal.

If B is the basis of lattice $\mathcal{L}$ then, the determinant of $\mathcal{L}$ is computed as

$$det(\mathcal{L}) = \mathrm{vol}(\mathcal{L}) = \Pi_{i=1}^{n} \|\mathrm{b}_i^*\|.$$

Algorithm 2.1 Gram Schmidt Orthogonalization

```
Require: Basis $\mathrm{B} = \{\mathrm{b}_1, \ldots, \mathrm{b}_n\}$
Ensure: Orthogonal Basis $\mathrm{B}^* = \{\mathrm{b}_1^*, ...., \mathrm{b}_n^*\}$
Let $\mathrm{b}_1^* = \mathrm{b}_1$
for $i = 2$ to $n$ do
    Compute $\mu_{ij} = \frac{<\mathrm{b}_i, \mathrm{b}_j^*>}{<\mathrm{b}_j^*, \mathrm{b}_j^*>}$ $\forall\ 1 \leq j < i$
    $\mathrm{b}_i^* = \mathrm{b}_i - \mu_{ij}\mathrm{b}_j^*$
end for
return $\{\mathrm{b}_1^*, \ldots, \mathrm{b}_n^*\}$
```

where $\mathrm{b}_1^*, \ldots, \mathrm{b}_n^*$ are the orthogonalized vectors of the basis B which are generated using GSO process. These orthogonal vectors may not be the points in the lattice. Alternatively, determinant can be computed as $det(\mathcal{L}) = \sqrt{\mathrm{B}^\top \mathrm{B}}$.

Let B and B$'$ generate the same lattice $\mathcal{L}$. The determinant is given by

$$det(\mathcal{L}) = \sqrt{\mathrm{B}^\top \mathrm{B}} = \sqrt{\mathrm{U}^\top \mathrm{B}'^\top \mathrm{B}' \mathrm{U}} = \sqrt{\mathrm{B}'^\top \mathrm{B}'}.$$

where the above equation is obtained using Lemma 2.2.1. Therefore, the determinant is unique for a lattice and it does not depend on any specific basis.

2.2.2.3 Successive Minima

The $\mathcal{L}_1(\mathcal{L})$ denotes the minimum distance in a lattice $\mathcal{L}$. It is defined as the length of the shortest non-zero lattice representing the minimum distance between any two points in the lattice.

$$\mathcal{L}_1(\mathcal{L}) = \min_{\mathrm{x},\mathrm{y} \in \mathcal{L}, \mathrm{x} \neq \mathrm{y}} \|\mathrm{x} - \mathrm{y}\| = \min_{\mathcal{L} \setminus \{0\}} \|\mathrm{x}\|.$$

It is extended to i-th successive minima. The i-th successive minima $\mathcal{L}_i(\mathcal{L})$ is the smallest radius r containing i linearly independent vectors.

$$\mathcal{L}_i(L) = \inf\{r : \dim(\mathrm{span}(\mathcal{L} \cap B(0, r))) \geq i\}.$$

where $B(0,r) = \{\mathrm{x} \in \mathbb{R}^n : \|\mathrm{x}\| < r\}$. For an arbitrary lattice $\mathcal{L}$, first successive minima and second successive minima are given in Figure 2.5.

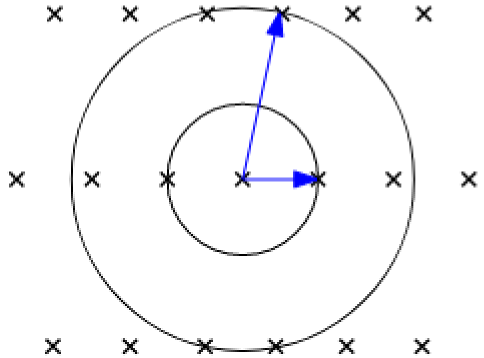


Figure 2.5: Successive Minima: $\mathcal{L}_1(\mathcal{L}) = 1, \mathcal{L}_2(\mathcal{L}) = 2.3$ [1]

Now, we recall the various bounds on the successive minima.

Bounds on Successive Minima

Given a basis B, lower bound on the first successive minima is specified in the lemma given below.

Lemma 2.2.5. "For a basis B, if $(\mathrm{b}_1^*, ..., \mathrm{b}_n^*)$ are the orthogonalized vectors of B, then the first successive minima of $\mathcal{L}(\mathrm{B})$ in l_2 norm satisfies

$$\mathcal{L}_1(\mathcal{L}(\mathrm{B})) \geq \min_i \|\mathrm{b}_i^*\|."$$

Minkowski [74] has given an upper bound on the first successive minima and an upper bound on the product of successive minima

Lemma 2.2.6. "For any full-rank lattice $\mathcal{L}$, a convex symmetrical set S with $Vol(S) \geq 2^n det(\mathcal{L})$ contains a non-zero lattice point and $\mathcal{L}_1(\mathcal{L}) \leq \sqrt{n} det(\mathcal{L})^{\frac{1}{n}}$."

Lemma 2.2.7. "Let B be a basis of full rank lattice $\mathcal{L}$, then successive minima $\mathcal{L}_1(\mathcal{L}), \mathcal{L}_2(\mathcal{L}), \ldots, \mathcal{L}_n(\mathcal{L})$ in l_2 norm satisfies

$$\Pi_{i=1}^n(\mathcal{L}_i(\mathcal{L})) \leq \sqrt{n} det(\mathrm{B})^{\frac{1}{n}}."$$

2.2.3 Ideal Lattices

In addition to the previously discussed general lattices, this thesis focuses on a specific type known as ideal lattices.

Let $f(X)$ denote a monic irreducible polynomial of degree n over the integers, then we define a ring $\mathcal{R} = \mathbb{Z}[X]/f(X)$, where $\mathbb{Z}[X]$ is the polynomial ring with coefficients in $\mathbb{Z}$. This $\mathcal{R}$ is called a ring of integer polynomials modulo $f(X)$. A non-empty set $\mathcal{I} \subset \mathcal{R}$ is an ideal of $\mathcal{R}$ if $\mathcal{I}$ is additive subgroup of $\mathcal{R}$ and for all $r \in \mathcal{R}$ and all $x \in \mathcal{I}$, $r \cdot x \in \mathcal{I}$.

Let τ be an additive isomorphism from $\mathcal{R}$ to some lattice $\tau(\mathcal{R})$ over $\mathbb{R}^n$. Then an ideal lattice defined with respect to the ring $\mathcal{R}$, the isomorphism τ and the ideal $\mathcal{I}$ is simply $\tau(\mathcal{I})$ over $\mathbb{Z}^n$.

For example, Consider the ring $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$, where n is power of 2. For additive isomorphism choose the coefficient embedding function that maps a ring polynomial $v = v_0 + v_1X + \ldots + v_{n-1}X^{n-1} \in \mathcal{R}$ to $\tau(v) = (v_0, v_1, \ldots v_{n-1})^\top \in \mathbb{Z}^n$. For the ideal $\mathcal{I}$ consider it to be $r\mathcal{R}$ for $r = r_0 + r_1X + \ldots + r_{n-1}X^{n-1} \in \mathcal{R}$. We can see that ideal lattice corresponding to the ring $\mathcal{R}$, the isomorphism τ and the ideal $\mathcal{I} = r\mathcal{R}$ is generated by the following matrix/basis,

$$\begin{bmatrix} r_0 & -r_{n-1} & \ldots & -r_1 \\ r_1 & r_0 & \ldots & -r_2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & \ldots & r_0 \end{bmatrix}$$

2.2.4 Modular Lattices

In 1996, Ajtai defined a random class of lattices called modular lattices or $q-$ary lattices [5]. Any lattice $\mathcal{L}$ which belongs to this class satisfies $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$. Numerous cryptographic constructions based on lattices rely on q-ary lattices, as problems formulated on these lattices are generally challenging on average.

Definition 2.2.8. For a prime $q \geq 2$, matrix $\mathrm{A} \in \mathbb{Z}_q^{n\times m}$ where $m \geq n \geq 1$, define the following three lattice,

$$\mathcal{L}_q(\mathrm{A}) = \{\mathrm{e} \in \mathbb{Z}^m : \exists \mathrm{x} \in \mathbb{Z}_q^n \text{ s.t. } \mathrm{A}^{\mathrm{T}}\mathrm{x} = \mathrm{e} \mod q\}$$
$$\mathcal{L}_q^{\perp}(\mathrm{A}) = \{\mathrm{x} \in \mathbb{Z}^m : \mathrm{A}\mathrm{x} = 0 \mod q\}$$

For any $\mathrm{u} \in \mathbb{Z}_q^n$, the coset $\mathcal{L}_q^{\mathrm{u}}(\mathrm{A})$ is,

$$\mathcal{L}_q^{\mathrm{u}}(\mathrm{A}) = \{\mathrm{x} \in \mathbb{Z}^m : \mathrm{A}\mathrm{x} = \mathrm{u} \mod q\}$$

”

Lattices $\mathcal{L}_q(\mathrm{A})$ and $\mathcal{L}_q^{\perp}(\mathrm{A})$ are dual i.e., $\mathcal{L}_q(\mathrm{A})^* = q\mathcal{L}^{\perp}(\mathrm{A})$ and $\mathcal{L}_q^{\perp}(\mathrm{A})^* = q\mathcal{L}_q(\mathrm{A})$.

2.3 Classic Lattice Problems

We recall some computational problems in lattices and their complexity. The two important classical problems in lattices are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP).

Definition 2.3.1. SVP: “Given an arbitrary basis B of lattice $\mathcal{L}$, find a shortest non-zero vector x of the lattice $\mathcal{L}$ i.e., $\|\mathrm{x}\| = \mathcal{L}_1(\mathcal{L})$.”

In 1981, Boas et al. [75] proved that SVP is NP-hard problem in l_∞ norm. But the hardness of SVP in l_p norm for $p < \infty$ was not proved until 1996. In 1996, Ajtai [5] proved that the problem is NP-hard for l_2 norm under randomized reductions. The approximate variant of SVP denoted by SVP$_\gamma$ is defined below.

Definition 2.3.2. (SVP$_\gamma$): “Given a lattice $\mathcal{L}$ and an approximation factor $\gamma \geq 1$, find a non-zero lattice vector x such that $\|\mathrm{x}\| \leq \gamma\mathcal{L}_1(\mathcal{L})$.”

SVP is proved to be NP-hard within the approximation range of $\gamma = 2^{(\log n)^{\frac{1}{2}-\epsilon}}$ where n is the lattice dimension and ϵ is a small arbitrary constant [76].
The decision version of SVP$_\gamma$ denoted by GapSVP$_\gamma$ is defined below.

Definition 2.3.3. GapSVP$_\gamma$:"Given a lattice $\mathcal{L}$ and a target $d > 0$, decide if $\mathcal{L}_1(\mathcal{L}) \leq d$ (YES-instance) or $\mathcal{L}_1(\mathcal{L}) > \gamma d$ (NO-instance)."

From the Definition 2.3.1, we can observe that SVP is related to the first successive minima $\mathcal{L}_1(\mathcal{L})$. Similarly, there is an approximate Shortest Independent vector problem (SIVP$_\gamma$) which relates to the n-th successive minima ($\mathcal{L}_n(\mathcal{L})$) and is defined below.

Definition 2.3.4. SIVP$_\gamma$: "Given a lattice $\mathcal{L}$ with rank n, find n linearly independent vectors $(\mathrm{v}_1, \ldots, \mathrm{v}_n)$ such that $\max_i \|\mathrm{v}_i\| \leq \gamma \mathcal{L}_n(\mathcal{L})$."

Another classic problem in lattice theory is the CVP. The CVP is defined as follows.

Definition 2.3.5. CVP: "Given a lattice $\mathcal{L}$ and a target vector $\mathrm{x} \in \mathrm{Span}(\mathcal{L})$, find the vector $\mathrm{y} \in \mathcal{L}$ such that $\|\mathrm{x} - \mathrm{y}\| = \mathrm{dist}(\mathrm{x}, \mathcal{L})$."

where $\mathrm{dist}(\mathrm{x}, \mathcal{L}) = \min_{\mathrm{v} \in \mathcal{L}}\{\|\mathrm{x} - \mathrm{v}\|\}$. In [75], it is proved that solving the exact version of CVP is NP-hard. The approximate version of CVP is defined as follows:

Definition 2.3.6. CVP$_\gamma$: "Given an approximation factor $\gamma > 1$, a lattice $\mathcal{L}$ and a target vector $\mathrm{x} \in \mathbb{R}^n$, find a vector $\mathrm{y} \in \mathcal{L}$ such that $\|\mathrm{x} - \mathrm{y}\| \leq \gamma \mathrm{dist}(\mathrm{x}, \mathcal{L})$."

It is shown that solving CVP within constant approximation factor and sub-polynomial function in n is NP-hard [77, 78].

2.3.1 Gaussian Measures over Lattices

Gaussian distribution over lattices is an important topic studied extensively in the literature [79, 80, 81]. This section introduces the definitions and foundational outcomes associated with the Gaussian distribution.

Definition 2.3.7. (Continuous Gaussian Distribution) "For any real $\sigma > 0$ and $\mathrm{c} \in \mathbb{R}^n$, Gaussian function $\rho_{\sigma,\mathrm{c}} : \mathbb{R}^n \to \mathbb{R}$ is defined as $\rho_{\sigma,\mathrm{c}}(\mathrm{x}) = e^{-\pi \frac{\|(\mathrm{x}-\mathrm{c})\|^2}{\sigma^2}}$. As $\int_{\mathrm{x} \in \mathbb{R}^n} \rho_{\sigma,\mathrm{c}}(\mathrm{x}) = \sigma^n$, continuous Gaussian distribution $D_{\sigma,\mathrm{c}}$ is obtained by normalizing the Gaussian function i.e., $D_{\sigma,\mathrm{c}}(\mathrm{x}) = \frac{\rho_{\sigma,\mathrm{c}}(\mathrm{x})}{\sigma^n}$."

When $c = 0$, we denote $\rho_{\sigma,0}$ as ρ_σ. Gaussian function can be extended to any countable set. The fundamental element in constructing lattice-based cryptographic protocols involves the discrete Gaussian distribution, the definition is given below.

Definition 2.3.8. (Discrete Gaussian Distribution). "Let K be a countable set. For any $c \in \mathbb{R}^n$ and real $\sigma > 0$, Gaussian distribution over the set K is defined as $D_{\mathcal{L},\sigma,\mathrm{c}}(x) = \frac{\rho_{\sigma,\mathrm{c}}(\mathrm{x})}{\sum_{\mathrm{y}\in K} \rho_{\sigma,\mathrm{c}}(\mathrm{y})}$."

Smoothing Parameter

Micciancio et al [80] introduced a smoothing parameter related to Gaussian distribution over lattices. The importance of the smoothing parameter is, it sets a threshold value after which many properties of continuous Gaussian distributions are carried over to discrete Gaussian distributions.

Definition 2.3.9. The smoothing parameter $\eta_{\epsilon'}(\mathcal{L})$ for a lattice $\mathcal{L}$ is defined as the smallest s that satisfies $\rho_{\frac{1}{\sigma}}(\mathcal{L}^* \setminus \{0\}) \leq \epsilon'$ for any $\epsilon' > 0$."

The primary motivation behind the smoothing parameter is that when a vector is sampled from a Gaussian distribution with a standard deviation equal to or larger than the smoothing parameter, its distribution becomes statistically close to a uniform distribution.

Lemma 2.3.10. "For any $\mathrm{c} \in \mathbb{R}^n$, lattice $\mathcal{L}$ and $\sigma \geq \eta_{\epsilon'}(\mathcal{L})$, the statistical distance between $D_{\sigma,c} \mod P(\mathrm{B})$ and uniform distribution over $P(\mathrm{B})$ is at most $\frac{\epsilon'}{2}$."

2.3.1.1 Discrete Gaussian Sampling over Lattices

A straightforward technique for sampling from a discrete Gaussian distribution involves sampling from a continuous Gaussian distribution and then rounding the obtained sample to the nearest lattice point using the lattice basis. This method yields a distribution that is statistically close to the discrete Gaussian distribution, especially for large values of the standard deviation σ.

In most of the cases, it is enough σ to be a small multiple of basis length. In 2008, Gentry et al [82] introduced a SampleD algorithm that samples according to discrete Gaussian distribution given an arbitrary basis of the lattice $\mathcal{L}$, parameter σ and $\mathrm{c} \in \mathbb{R}^n$.

Lemma 2.3.11. [82] Given a lattice $\mathcal{L}$, a set of k short linearly independent vectors $\mathrm{S} \in \mathbb{Z}^{n \times k}$, $c \in \mathbb{R}^n$ and $\sigma > \|\mathrm{S}\|\omega(\sqrt{\log n})$ there exists SampleD algorithm which outputs a vector x that is statistically close to the distribution $D_{\mathcal{L},\sigma,c}$."

Few of the standard results about Gaussian distribution over lattices are given below.

Lemma 2.3.12. [80] "For any n-dimensional lattice $\mathcal{L}$, $\sigma \geq \eta_{\epsilon'}(\mathcal{L})$, $0 < \epsilon < 1$ and $\mathrm{c} \in \mathbb{R}^n$

$$Pr_{\mathrm{x} \hookleftarrow D_{\mathcal{L},\sigma,\mathrm{c}}}[\|\mathrm{x} - \mathrm{c}\| > s\sqrt{n}] \leq \frac{1-\epsilon'}{1+\epsilon'} \cdot 2^{-n}$$

From the above lemma it is clear that, samples obtained from discrete Gaussian distribution over a lattice $\mathcal{L}$ are short with overwhelming probability, this fact is extensively used in this thesis.

Lemma 2.3.13. [82] "For any matrix $\mathrm{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, $\mathrm{e} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ where $\sigma \geq \eta_{\epsilon'}(\mathcal{L}_q^{\perp}(\mathrm{A}))$ then the distribution of the syndrome $\mathrm{u} = \mathrm{Ae} \mod q$ is statistically close to the uniform distribution over $\mathbb{Z}_q^n$."

2.3.2 Average-case Lattice Problems

The majority of cryptographic primitives/algorithms are based on average-case hard problems. The first worst-case to average-case reduction for a lattice problem was proposed by Ajtai in 1996 [5]. He proposed Short Integer Solution (SIS) problem by providing a reduction from SIVP_γ to SIS, where γ depends on shortness of SIS solution. Later Regev [83, 84] proposed Learning With Errors (LWE) problem and proved its hardness by providing a reduction from SIVP_γ to LWE. The cryptographic algorithms built using lattices are mostly based on these two average-case hard problems, SIS and LWE.

Definition 2.3.14. "$\mathrm{SIS}_{n,m,q,\beta}$ [5, 80]: Given an uniformly chosen matrix $\mathrm{A} \in \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathrm{x} \in \mathbb{Z}^m$ such that $\mathrm{Ax} = 0 \mod q$ and $\|\mathrm{x}\|_\infty \leq \beta$."

In other words, SIS problem is to find a short non-zero lattice vector in $\frac{\perp}{q}(\mathrm{A})$. Parameter β is chosen such that the solution exists and it is much less than q to avoid the trivial solutions ($\mathrm{x} = (q, q, \ldots, q)^\top$).

For any $m, \beta = poly(n)$ and for any $q \geq \sqrt{n}\beta$, solving $\mathrm{SIS}_{n,m,q,\beta}$ problem with non-negligible probability is at least as hard as solving SIVP_γ problem, for some $\gamma = \mathcal{O}(\beta\sqrt{n})$[82, 80].

A variant of SIS problem is Inhomogeneous Short Integer Solution (ISIS) defined below.

Definition 2.3.15. “$\mathrm{ISIS}_{n,m,q,\beta}$: Given an uniformly chosen matrix $\mathrm{A} \in \mathbb{Z}_q^{n\times m}$ and a vector $\mathrm{u} \in \mathbb{Z}_q^n$, find a non-zero vector $\mathrm{x} \in \mathbb{Z}^m$ such that $\mathrm{Ax} = \mathrm{u} \mod q$ and $\|\mathrm{x}\|_\infty \leq \beta$.”

Definition 2.3.16. “$\mathrm{LWE}_{n,q,\chi}$ [83, 84] Let χ be a distribution over $\mathbb{Z}$ and let $n, m \geq 1, q \geq 2$. For a vector $\mathrm{s} \in \mathbb{Z}_q^n, A_{\mathrm{s},\chi}$ is a distribution of $(\mathrm{a}, \mathrm{a}^\top\mathrm{s} + e)$ over $(\mathbb{Z}_q^n, \mathbb{Z}_q)$, where $\mathrm{a} \hookleftarrow \mathbb{Z}_q^n, e \hookleftarrow \chi$. The $LWE_{n,q,\chi}$ problem asks to distinguish m samples chosen according to $A_{\mathrm{s},\chi}$ (s chosen uniformly) and m samples chosen according to uniform distribution over $(\mathbb{Z}_q^n, \mathbb{Z}_q)$”.

Above definition is considered the decision version of LWE. The problem is to distinguish between m samples chosen according to the distribution $A_{\mathrm{s},\chi}$ and m samples chosen from the uniform distribution over $(\mathbb{Z}_q^n, \mathbb{Z}_q)$.

The search version of LWE problem is to obtain a secret $\mathrm{s} \in \mathbb{Z}_q^n$ from polynomial (m) samples of $(\mathrm{a}, \mathrm{a}^\top\mathrm{s} + e) \in (\mathbb{Z}_q^n, \mathbb{Z}_q)$ where a is uniformly chosen over $\mathbb{Z}_q^n$ and e is chosen according to distribution χ. Generally, Gaussian distribution with mean 0 and standard deviation greater than $2\sqrt{n}$ is used for χ.

For a prime power $q, b \geq \sqrt{n}\omega(\log n)$, and distribution χ, solving $\mathrm{LWE}_{n,q,\chi}$ problem is at least as hard as solving SIVP_γ, where $\gamma = \mathcal{O}(\frac{nq}{b})$ [85, 83]

Variant of SIS and LWE are defined for ring setting, they are called RSIS and RLWE. The two variants are defined in below section.

2.3.3 Ring-SIS and Ring-LWE

Let $q \geq 3$ be a non-negative integer and $\mathbb{Z}_q = [-\frac{q-1}{2}, \frac{q-1}{2}]$. Consider the ring $\mathcal{R} = \mathbb{Z}[X]/X^n + 1$, where $\mathbb{Z}[X]$ is the polynomial ring with coefficients in $\mathbb{Z}$, n is power of 2, and $X^n + 1$ is the cyclotomic polynomial of degree n. Any element of $\mathcal{R}$ is represented as a polynomial whose coefficients are in $\mathbb{Z}$ and degree less than n.

The quotient ring $\mathcal{R}/\mathcal{I}$ is the set of equivalence classes $r+\mathcal{I}$ of $\mathcal{R}$. Let the quotient ring be $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[X]/X^n + 1$. Similarly, any element of $\mathcal{R}_q$ is represented as a polynomial whose coefficients are in $\mathbb{Z}_q$ and degree less than n.

For any polynomial $v = v_0 + v_1X + \ldots + v_{n-1}X^{n-1} \in \mathcal{R}$, we define $\|v\|_\infty = max_i(|v|_i)$. Similarly, for any polynomial vector $\mathrm{u} = (u_1, \ldots, u_m)^\top \in \mathcal{R}^m$, we define $\|\mathrm{u}\|_\infty = max_j(\|u_j\|_\infty)$.

Definition 2.3.17. (Ring-SIS$_{n,m,q,\beta}$ [86, 87, 88]). "Given a uniformly random vector $\mathrm{A} = [a_1|\ldots|a_m] \in \mathcal{R}_q^{1\times m}$, find a non-zero ring vector $\mathrm{x} = (x_1, \ldots, x_m)^\top \in \mathcal{R}^m$ such that $\mathrm{Ax} = a_1x_1 + \ldots + a_mx_m = 0$ and $\|\mathrm{x}\|_\infty \leq \beta$."

For $m > \frac{\log q}{\log(2\beta)}, \gamma = 16\beta mn \log^2 n$, and $q > \frac{\gamma\sqrt{n}}{4\log n}$, the Ring-SIS$_{n,m,q,\beta}$ problem is at least as hard as SVP$_\gamma^\infty$ in any ideal in the ring $\mathcal{R}$ [87].

Definition 2.3.18. (Ring-LWE$_{n,m,q,\chi}$ [8]). "Let $n, m \geq 1, q \geq 2$, and let χ be a probability distribution over $\mathcal{R}$. For $s \in \mathcal{R}_q$, let $\mathrm{A}_{s,\chi}$ be the distribution obtained by sampling $a \hookleftarrow U(\mathcal{R}_q)$ and $e \hookleftarrow \chi$, and outputting the pair $(a, a \cdot s + e) \in \mathcal{R}_q \times \mathcal{R}_q$. The Ring-LWE$_{n,m,q,\chi}$ problem is to distinguish m samples chosen according to $\mathrm{A}_{s,\chi}$ and m samples chosen according to the uniform distribution over $\mathcal{R}_q \times \mathcal{R}_q$."

Let $q = poly(n)$ be a prime power, Let $B = \mathcal{O}(n^{5/4})$ be an integer and χ be a $B-$bounded distribution over $\mathcal{R}$, i.e., it outputs samples $e \in \mathcal{R}$ such that $\|e\|_\infty \leq B$ with overwhelming probability in n. Then, for $\gamma = n^2(q/B)(nm/\log(nm))^{1/4}$, the Ring-LWE$_{n,m,q,\chi}$ problem is at least as hard as SVP$_\gamma^\infty$ in any ideal in the ring $\mathcal{R}$, via a polynomial-time quantum reduction ([86, 8, 89]).

The Ring -LWE requires samples $(a, b) \in R \times R$ very small in count. The representation of lattice in Ring-LWE is compact; that is, for each polynomial $a \in R$

formed with n vectors multiplied with corresponding coefficient values $a.x^i$, where $i \in \{0, 1, ..., n-1\}$ i.e., degree of the polynomial up to $(n-1)$. In the pair (a, b) if the value of a is fixed for all users, then b is the public key $b \in R$. This is why small public keys in Ring-LWE and the number of modular values are reduced to n from $\Omega(n^2)$ modular values that exist in LWE.

Let q is an odd prime; $q > 2$. $\mathbb{R}$ refers to a set of real numbers and $\mathbb{Z}$ refers to a set of integers. The $\mathbb{Z}[x]$ and $\mathbb{Z}_q[x]$ refers to a ring of a polynomial over $\mathbb{R}$ and $\mathbb{R}_q$ respectively. Let R be polynomials ring $R = \mathbb{Z}[x]/f(x)$ and $R_q = \mathbb{Z}_q[x]/f(x)$, where $f(x) = (x^n + 1)$, n be the power of 2 and n, q are positive integers $q \in \mathbb{Z}$ and $n \in \mathbb{Z}$.

An element of polynomial ring R_q is denoted by a polynomial with degree $(n-1)$. Let $a \in R_q$ then $a(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + ... + a_{n-1}x^{n-1}$ and we define euclidean norm of a as $||a|| = \sqrt{a_0^2 + a_1^2 + a_2^2 + ... + a_{n-1}^2}$. The norm of the shortest vector in the lattice is λ_1 and λ_∞ norm is defined as $||a_\infty|| = Max\{a_0, a_1, a_2, a_3, ..., a_{n-1}\}$.

The distribution function χ_σ be discrete gaussian on $\mathbb{Z}^n$ where σ is a positive real number and the standard deviation. The distribution χ_σ bounded by β , if $\Pr[\|x\| > \beta : x \leftarrow \chi] \leq negl(n)$.

For uniformly generated random parameter $a \in R_q$, a small, public and fixed parameter $s \leftarrow \chi_\sigma \in R$, $e \leftarrow \chi_\sigma \in R$, The following lemmas are used as in [90]:

Lemma 2.3.19. For any two elements c, d of the polynomial ring R,, then the following equations hold [90].

$$c.d \leq \sqrt{n}.||c||.||d||$$

$$||c.d||_\infty \leq n.||c||.||d||$$

Lemma 2.3.20. For any given β a positive and real number where $\beta = \omega(\sqrt{logn})$, then it holds the inequality as in [91] is:

$$\Pr_{c \leftarrow \chi_\beta}[||c|| > \beta.\sqrt{n}] \leq 2^{-n+1}$$

Let $\mathbb{Z}_q$ be a set. where $\mathbb{Z}_q = \{-\frac{q-1}{2}, ..., \frac{q-1}{2}\}$, then the middle set ε is defined as $\varepsilon = \{-\frac{q}{4}, ..., \frac{q}{4}\}$. The Cha is the complement set of ε. Then for any $y \in \mathbb{Z}_q$, characteristic function Cha is defined as in[42]:

$$Cha(y) = \begin{cases} 0, & if y \in \varepsilon \\ 1, & if y \notin \varepsilon \end{cases}$$

For a given element $c \in \mathbb{Z}$ and $d = Cha(c)$ the modular function is defined as[42]:

$$Mod_2 = Z_q \times \{0,1\} \rightarrow \{0,1\}$$
$$Mod_2(c,d) = (c + d.\frac{q-1}{2}) \mod q \mod 2$$

We have the following lemmas for this function:

Lemma 2.3.21. The equation $Mod_2(c, Cha(c)) = Mod_2(w, Cha(c))$ holds for the given q the odd prime and two elements $c, e \in R_q$ such that $|e| < \frac{q}{8}$, where $w = c + 2.e$.

The ring R_q can be extended with two functions, Cha and Mod_2, as follows: Given an element $c = c_0 + c_1.x + c_2x^2 + ... + c_{n-1}x^{n-1} \in R$, we present it as a vector $c = (c_0, c_1, c_2, c_3...c_{n-1})$. For a vector $v = (v_0, v_1, v_2, v_3...v_{n-1}) \in \{0,1\}^n$, two functions are defined as $Cha(c) = (Cha(c_0), Cha(c_1), cha(c_2), ..., Cha(c_{n-1}))$ and $Mod_2(c, v) = (Mod_2(c_0, v_0), Mod_2(c_1, v_1), Mod_2(c_2, v_2), ..., Mod_2(c_{n-1}, v_{n-1}))$ [42].

Definition 2.3.22. Search Ring-LWE Problem: Let n, q are two positive integers, and distribution functions χ_s and χ_e are (bounded) distributions over the ring R. The Search Ring-LWE problem is defined as: For a given pairs(a, b and $b = a.s + e$) target is to recover a secret vector s, where $a \overset{\$}{\leftarrow} R_q$, a secret vector $s \overset{\$}{\leftarrow} \chi_s$ and an error vector $e \overset{\$}{\leftarrow} \chi_e$.

Definition 2.3.23. Decisional Ring-LWE Problem:. Let n,q be two positive integers, and distribution functions χ_s and χ_e are bounded distributions over the ring R. The Decisional Ring-LWE is to distinguish two distributions of pairs (a, b) and (a, u) with non-negligible advantage, where $< b = a.s + e >$ for any $a \overset{\$}{\leftarrow} R_q$,the secret $s \overset{\$}{\leftarrow} \chi_s$ and $u \overset{\$}{\leftarrow} R_q$.

(Note: If the Decisional $Ring{-}LWE_{n,q,\chi}$ assumption holds, then Search $LWE_{n,q,\chi}$ assumption also holds.)

2.4 Backgroud of IoT

The IoT infrastructure consists of following entities that provide a diverse technology environment.

- Hardware (Arduino Raspberry Pi, Intel Galileo, Intel Edison, ARM mBed, Bosch XDK110, Beagle Bone Black and Wireless SoC)
- Integrated Development Environment (IDE) for developing device software, firmware and APIs
- Protocols [RPL, CoAP, RESTful HTTP, MQTT, XMPP (Extensible Messaging and Presence Protocol)]
- Communication (Powerline Ethernet, RFID, NFC, 6LowPAN, UWB, ZigBee, Bluetooth,WiFi, WiMax, 2G/3G/4G)
- Network backbone (IPv4, IPv6, UDP and 6LowPAN)
- Software (RIOT OS, Contiki OS, Thingsquare Mist firmware, Eclipse IoT)
- Internetwork Cloud Platforms/Data Centre (Sense, ThingWorx, Nimbits, Xively, openHAB, AWS IoT, IBM BlueMix, CISCO IoT, IOx and Fog, EvryThng, Azure, TCS CUP)

The following five entities can be considered for the five levels behind an IoT system

1. Device platform consisting of device hardware and software using a microcontroller (or SoC or custom chip), and software for the device APIs and web applications.
2. Connecting and networking (connectivity protocols and circuits) enabling internetworking of devices and physical objects called things and enabling the internet connectivity to remote servers

3. Server and web programming enabling web applications and web services
4. Cloud platform enabling storage, computing prototype and product development platforms
5. Online transactions processing, online analytics processing, data analytics, predictive analytics and knowledge discovery enabling wider applications of an IoT system.

2.4.1 Energy Optimization

Smart and secure systems implemented upon IoT technology require device interconnectivity for extended time frames, delivering continuous data. Such operations demand constant power supply. In general, IoT realizations also face the challenge of energy optimization i.e., minimizing their energy[39].Optimizing energy usage in Internet of Things (IoT) systems is important for extending the lifespan of devices, reducing operational costs, and minimizing environmental impact.

Designing and developing a new lighe weight cryptographic algorithm requires a focus on three features: security, cost, and performance. These features can be evaluated using using metrics. The cost can be measured using the physical area in terms of Gate Equivalent (GE), memory, and battery power. Performance can be measured using latency and throughput. Finally, security can be measured in terms of security level and various attack models. Different metrics that can be used for lighe weight cryptographic algorithms are security level, hardware technology, throughput, latency, power and energy consumption, RAM/ROM and efficiency.

Lightweight block ciphers can be developed by using smaller block sizes, smaller key sizes, simpler or lesser number of rounds, and simpler key schedules than those used in conventional cryptographic algorithms. Developing new lightweight hash functions can focus on reducing the output size and message sizes.

Roy.et.al [92] studied the extent to which the underlying hash function (SHA256), a principal element of HMAC, can be made more energy efficient. This approach employs an energy-reducing algorithmic engineering technique, based upon an Energy

Complexity Model (ECM) proposed by Roy et al. [40] [93]on the SHA256 encryption algorithm, which is central to HMAC. HMAC implements both integrity checking and authentication of messages using cryptographic hash functions. Any hash function (e.g. MD5, SHA128, SHA256,etc.) can be used in HMAC combined with a shared secret key. HMAC's strength cryptography-wise is dependent on the strength of its underlying hash function. The input to HMAC is a message M containing 1 blocks ($Y(1)Y(1)$), each of size b. A signature S_i is concatenated to the left of M before it is input to the underlying hash function (e.g. SHA256) to produce a temporary message digest MD'. MD is further concatenated with output signature $S_o = K + \oplus PAD$, which is then hashed again using the underlying hash (e.g. SHA256) to produce MD, the final message digest.

The Double Data Rate Synchronous Dynamic Random Access Memory (DDR SDRAM) is the reference architecture for the energy complexity model (ECM), which has applied to HMAC. In post-quantum blockchain architecture, the Merkle Tree construction is a principal element of blockchain computations, can be made more energy efficient by employing an energyreducing algorithmic engineering technique, based upon an Energy Complexity Model (ECM) proposed by Roy et al. on the SHA256 encryption algorithm, which is central to the Merkle Tree.

The ECM developed in is built upon an abstraction of the Double Data Rate Synchronous Dynamic Random Access Memory (DDR SDRAM) architecture [94]. Main memory in DDR is divided into banks, each of which contains a certain number of chunks.Merkle Tree construction performs its hash calculations via repeated use of the SHA256 encryption algorithm.

Chapter 3

Lattice-based authentication and key exchange protocol for Internet of Things

This chapter presents and validates a lattice-based authentication and key exchange protocol for the IoT Environment. Our scheme security is based on LWE problem in a polynomial ring; shortly, we call Ring-LWE. The advantage of this lattice hard problem is the computations are in the ring R_q for for dimension n that results small cipher texts and reduced key sizes. The proposed protocol correctness is proved formally and verified with the standard verification tool AVISPA for authentication. The security analysis of the protocol demonstrates that it is secure against known attacks on the Internet of things environment. The protocol's performance is analyzed and compared with relevant protocols. It shows that the communication cost is the same as other protocols, and the computation cost is minimal.

The chapter is organized as follows: Section 3.1 presents the introduction. Section 3.2 presents the contribution and construction overview of the proposed scheme. Section 3.3 presents the proposed protocol that is used in the construction of the scheme. In Section 3.4, we describe the correctness of the scheme along with its security requirements. Section 3.6 presents the implementation of our scheme, results and performance evaluation of the scheme. We also compare the proposed scheme with the existing schemes and finally, Section 3.7 summarizes the chapter.

3.1 Introduction

The Internet of Things (IoT) is evolving as the ”Future Internet” where every device is connected with another device; likewise, it is connected with billions of devices communicating over the Internet without human interaction. The data collected in the Internet of Things is from various environments and transmitted to servers for storage and processing. For sensitive IoT applications, it is desirable to authenticate IoT devices to ensure the source of data is trustworthy. On the other hand, IoT devices should have a robust method for authenticating with the server or other communicating devices. Cryptographic algorithms used to protect IoT have unique challenges due to these limited resources. The Elliptic Curve Cryptography (ECC) with the proper parameters is being considered as a possible solution to this problem. However, when quantum computers become available, ECC-encrypted devices become insecure. Lattice-based cryptography (LBC) is a category of post-quantum cryptography(PQC) that is highly efficient, firmly secure, and highly suitable for resource-constrained devices [95]. The LBC operates on relatively small integers because it uses matrices and vectors for computation in specified rings or fields of small order. These lattice-based algorithms maintain the balance between confidence of security and computational efficiency with key size, ciphertext, and signature lengths, which are suitable for lightweight cryptosystems for IoT security.

The security of lattice-based cryptosystems relies on two challenging problems associated with lattices: the Short Integer Solution (SIS) and the Learning With Errors (LWE) problems. The ring variant of SIS and LWE problems are referred to as Ring-SIS, and Ring-LWE has the advantage of being more efficient and having a much smaller key size than security schemes designed using non-ring versions [6]. The LWE problem is to find s a secret vector for given polynomial samples$A.s+e$, where e is the error vector chosen from the specified error distribution function χ and A is the uniformly generated matrix.

The Ring-LWE that is more realistic in terms of computation cost and memory for storage. Therefore, it is suitable for authentication in resource-constrained environments such as IoT [6]. Lyubashevsky et al. [7] stated and demonstrated that ring variants lattices problems are as hard as solving worst-case problems in a special class of lattices [8]. Ring-LWE-based key exchange provides much-improved post-quantum security with a smaller key size [9]. The Ring-LWE is, differentiating a uniform distribution and noisy ring multiplications in polynomial time. For instance, differentiate $(A, < A.s + e >)$, where A is an element of a ring chosen randomly and s, e are sampled from the distribution function. There are various methods for discrete sampling are Bernoulli sampling, Knuth-Yao sampling, rejection sampling, and cumulative distribution table sampling. The post-quantum secure protocols are essential for current client-server environment scenarios. However, most protocols designed based on LWE and its variants are not suitable for small scale computing devices. In this chapter, we formulate and construct a novel lattice-based authentication and key exchange protocol for IoT environment. The objective is to improve both performance and security, particularly in guarding against lattice attacks.

3.2 Contribution and Protocol Overview

This section provides an overview of the contributions made in this scheme and overview of the proposed protocol. Our scheme security is based on LWE problem in a polynomial ring; shortly, we call Ring-LWE. The advantage of this lattice hard problem is the computations are in the ring R_q for for dimension n that results small cipher texts and reduced key sizes. The proposed protocol correctness is proved formally and verified with the standard verification tool AVISPA for authentication. The protocol's performance is analyzed and compared with relevant protocols. It shows that the communication cost is the same as other protocols, and the computation cost is minimal. The protocol's performance is analyzed and compared with relevant protocols. It shows that the communication cost is the same as other protocols, and the computation cost is minimal. Contributions of this scheme are summarized as follows:

1. We proposed a design of a lattice-based authentication protocol for the IoT

using the hardness of the $Ring - LWE$ problem. The protocol permits small dimensions of parameters that are resource-constrained.

2. We examined the protocol's security against the quantum adversary and demonstrated its correctness, as well as its resistance to known security attacks and threats.

3. We conducted a performance evaluation of the protocol, assessing computation and communication overhead as key metrics..

4. We practically implemented the proposed protocol in IoT client-server scenarios and presented results.

3.2.1 Security Requirements

The security requirements for communication between IoT nodes and a Cloud server include:

1. Authentication and Message Integrity: The security protocol should verify legal IoT client and message transmitted to them should be received by only authenticated IoT client without any modification, even though message broadcasted to multiple nodes

2. Node Anonymity: The true IoT device identities are kept secret, that is any cyber attacker will be unable to trace the actions of the client or unable to extract any intercepted information.The node anonymity is additional privacy feature along with Confidentiality, Integrity, and Authentication. It protects the identity and presence of IoT device in the network if an attacker intercepts the communication. Node's anonymity is crucial in sensitive IoT scenarios such as healthcare, military, and personal applications.

3. Session key Security: A session key is generated for communication between the IoT client and the Cloud Server after the authentication phase of the protocol. Its purpose is to keep further internet communication safe.

4. Resistance to Attacks: The protocol should provide a secure communication environment that is resistant to man-in-the-middle attacks, alteration attacks, replay attacks, impersonation attacks, stolen verifier attacks, and offline dictionary attacks, among other types of attacks.

5. Resistance to password guessing attacks: The protocol must possess robust capabilities to safeguard password security, rendering it impervious to traceable, untraceable and online,offline password-guessing attacks.

6. Forward secrecy: The protocol should provide forward secrecy, eventhough an IoT client's password is leaked to the adversary i.e. in the presence of the protocol, the adversary must be prevented from acquiring previous session keys.

3.2.2 System Model

A communication network is established using several Internet of Things (IoT) nodes, functioning as clients that connect to the Cloud Server (CS) over the Internet. The CS is assumed as trusted, and the work is to validate the IoT Client (IC_i) and relay messages. It models a communication between IC_i and CS over a network entirely regulated by a Probabilistic polynomial Time (PPT) adversary A. The IC_i and CS seek to create a shared session key that can be authenticated using a key exchange mechanism. Adversary A can listen to the conversation between IC_i and CS, and A can respond, edit, delay, and insert new messages. The PPT adversary A is permitted to access the Oracle model that generates protocol-simulated outputs for A's query. It may enable protocol communications between any number of IoT client instances, the transmission of any message to these instances, and the monitoring of IC_i and CS answers under protocol requirements. Session keys created by IC_i and CS instances may also be revealed. Finally, adversary A can directly obtain values stored in the IC_i or a password through multiple trials. Following are the assumptions for the IoT scenario: The IoT node(IC_i) and Device Gateway (DG) are identified with a Unique ID and One-time handshake. The DG and IoT Cloud Service Gateway (CSG) are mutually authenticated using PKE. The Central Key Repository (CKR) in the network maintains the IDs of Connected devices in the network. An Unauthorised IoT node(Quantum attacker) tries to access the IoT network. The scenario of the IoT network system model is shown in the figure:3.1. A list of notations used in the protocol is described in the table::3.1.

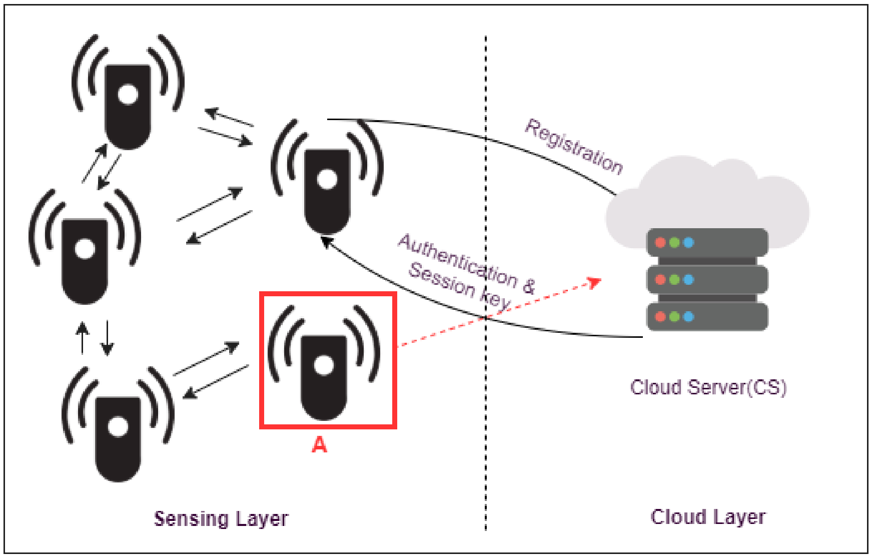


Figure 3.1: System model

Table 3.1: Notations

Symbol	Description
IC_i	IoT Client Device i
CS	IoT Cloud Server
A	Adversary
χ_β	Gaussian distribution function
Id_i	Identity of Device
Pwd_i	Password of user to login to device i
Cha	Characteristic Function
Mod_2	Modular Function
s, e	Secret and error vectors from χ_β
K	Random key for server co-prime to q
$\oplus$	Bit-wise XOR operation
$h(.)$	Hash function
$\|\|$	Concatenation operation

3.2.3 Adversary Model

In the adversary model (like Dolev-Yao model) encryption messages are assumed to be an abstract operation on that algebra. The adversary A is assumed to be a specific (non-deterministic) state machine, and for an adversary, the only way to produce new messages is to perform certain operations on messages it already "known".

In IoT scenario, every IoT device need to register with a Cloud Server(CS). During the registration process, device may authenticate itself by its identity and receive a public key or other credentials for secure communication. We assume that, the adversary may be impersonate as legal IoT device and get registered at CS and could intercept data being transmitted in the communication. The data could be analyzed, modified, or used to launch further attacks. But the proposed protocol is secure against all possible attacks by adversary who will be proved un-authorized user in the verification step. We define the adversary's capabilities as the ability to register on the Cloud Server (CS) to obtain public keys. Acting on behalf of adversary-controlled entities, the opponent can register any arbitrary public key, even those identical to the keys of specific genuine parties in the system. Adversary A employs *Send* inquiries to dispatch incoming messages to the parties, retrieves the outgoing messages, and exercises control over their delivery, effectively manipulating all communications among the parties.

- Send(msg). The message(msg) can take one of three forms: (ICi; CS), (ICi; CS; out), or (ICi; CS; in; out). As per protocol, the adversary receives the session's response, and the variables *in* and *out* are initialized and changed (through concatenation) accordingly.

To identify potential information leakage, adversary A is authorized to execute the following queries, the results of which are designated as data sets $Secret\_key$, $Session\_state$, and $Session\_key$. We utilize $secret\_key$ to describe a session's secret randomness, and $Session\_State$ refers to all intermediate secret values that are computed or utilized in the memory of the host machine, as well as the $session\_key$. If the session is ended, the $Session\_state$ contains results and $Session\_key$.

- Secret_key leak(s): Adversary A aquires the $Secret\_key$ associated with session s.
- Session_state leak(s): Adversary A aquires the $Session\_State$ of the owner of session s.
- Session_key leak(s): Adversary A aquires the $Session\_key$ in a completed session s.
- Credentials leak(ICi): Adversary A aquires the credentials of ICi by this query.

Definition 3.2.1. Freshness of Session-Id sid: Let us consider a ended session owned by an genuine IoT Client ICi and a trustworthy Cloud Server CS. Assume sid is the matching session of s. If none of the following conditions are met, we claim the session sid is fresh.

1. ICi issues Session_Key leak(s), or Session_Key leak(s) if s exists;
2. ICi issues Session_State leak(s), or Session_State leak (s) if s exists;
3. s exists, and adversary A does one of the following queries: (a) both Credentials leak(ICi) and Secret_Key leak(s), or (b) both Credential leak(CS) and Secret_key leak(s) (s),
4. s does not exist, and adversary A performs either (a) both Credential leak(ICi) and Secret_key leak(s), or (b) Credential leak(CS).

Definition 3.2.2. Freshness of Session s: Let us consider a ended session owned by an genuine IoT Clint ICi and a trustworthy Cloud Server party CS. Assume sid is the matching session of s. Session s is said to be fresh if none of the following conditions are met:

1. A issues Session_key leak(s), or Session_key leak(s) if s exists;
2. A issues Session_State Reveal(s), or Session_State leak (s) if s exists;

3. s exists, and adversary A does one of the following queries: (a) both Credentials leak(ICi) and Secret_key leak(s), or (b) Credentials leak(CS) and Secret_key(s) (s),

4. s does not exist, and adversary A performs either (a) both Credential leak(ICi) and Secret_key leak(s), or (b) Credential leak(CS) before the session s is completed.

Security Experiment: The adversary A objective is to differentiate a session key from a random key. The experiment goes as follows: A is initially furnished with a set of honest parties and is free to pose any sequence of questions as outlined above. A asks the following question during the experiment.
Test_Session(s_t): This query requires that s_t be a new session. A random bit b is chosen to respond to this query. If $b = 0$, the session key of s_t is returned. Otherwise, a random key is generated.

The experiment will be continued until A makes b' estimate. If the test session s_t is still fresh and A's guess is correct, i.e., $b' = b$, A Succeeds the game. The adversary's advantage in the $RLWE - AKE$ protocol experiment is specified as:

$$Adv_{RLWE\_AKE}(A) = Pr[A\_Succeeds] - \frac{1}{2}$$

Definition 3.2.3. Security of AKE: A key exchange protocol π is deemed AKE secure if it is operating against all PPT adversaries A with the given capabilities, and the freshness of the session is determined by the freshness of session id sid it holds:

i. Communication Parties who finish matching sessions compute the same session key.

ii. $Adv_{AKE}(A)$ is negligible for any adversary A with the assumed capabilities.

Definition 3.2.4. Security of AKE with Forward Secrecy:A key exchange protocol achieves AKE security with forward secrecy if, for all PPT adversaries with assumed capabilities, operating against it, the freshness of the session is determined upon the freshness of the session identity sid, it possesses the following:

i. Parties who finish matching sessions compute the same session key.

ii. $Adv_{AKE}(A)$ is negligible for any adversary A with the assumed capabilities.

3.3 Proposed Model

In the proposed protocol, the IoT device is initially registered at a cloud server that serves client IoT devices based on the request. After registration of the IoT device at the server, it operates automatically without the support of human efforts. The proposed protocol is described in the following four distinct phases: setup, registration, login and authentication, and verification.

3.3.1 Setup

The setup phase begins with the Cloud Server (CS), which generates the parameters required for the communication along with its private key. The activities performed by CS are as follows:

Phase	Activities
Setup	i Cloud Server CS chooses n and q, where n is of the form power of 2 and holds $q \mod 2n = 1$. ii Cloud Server CS chooses and random element $a \in R_q$ and the Gaussian distribution function χ_β iii CS selects a random key K which is co-prime to q iv Cloud Server CS computes its public key $CS_p = a.s + K.e$ by sampling $s, e \leftarrow \chi_\beta$. v The hash function h is chosen as $h : \{0,1\}^* \rightarrow \{0,1\}^l$, where l refers the length of the output. vi Cloud Server CS holds the secret key s for confidentiality and other parameters $(n, q, \chi_\beta, a, CS_p, h)$ are published.

3.3.2 Registration

In the registration phase, the IoT client IC_i will be registered for the services from the Cloud Server CS. After completion of the registration process, the IoT Client will get its private key by which it communicates further with CS in a secure channel. The following activities are performed in this phase:

Phase	Activities
Registration	i The IoT Client device IC_i is assigned with its identity and password (Id_i, Pwd_i). The user who initiates the device will login with the given credentials. The identity of the device is static but the password can be changed by the user. After the login into device,the IC_i computes $D_i = h(Id_i\|\|Pwd_i)$ and sends $< Id_i, D_i >$ to Cloud Server CS. ii The CS upon receiving the registration request (Id_i, D_i) from IC_i, CS verifies Id_i then computes IC_i's secret key s and random key K then computes $D_1 = h(Id_i\|\|s\|\|K)$, $D_2 = D_1 \oplus D_i$. It stores D_1 and sends $< D_2 >$ to IC_i. iii The IC_i upon receiving D_2, it determines $D_1 = D_2 \oplus D_i$ then it computes $D_j = h(Id_i\|\|Pwd_i\|\|D_1)$ then stores $< D_2, D_j >$.

3.3.3 Login and Authentication

In this phase, the CS authenticates the IoT client IC_i with its credentials whenever it logins. After the device's authentication, a secure session key will be computed by both IC_i and CS for further communication in the secure communication channel.

Phase	Activities
Login and Authentication	i To login the IoT client device(IC_i), the user will enters the login credentials $\{Id_i, pwd_i\}$. Then it computes $D_i = h(Id_i \|\| Pwd_i)$, $D_1' = D_2 \oplus D_i$ and $D_j' = h(Id_i \|\| Pwd_i \|\| D_1')$ then verifies if $D_j' \overset{?}{=} D_j$. ii Upon the successful login, the IoT client device samples f_i, g_i from χ_β i.e $f_i, g_i, \leftarrow \chi_\beta$ then computes $X_i = a.f_i + K.g_i$ and $K_i = f_i.CS_p$. The characteristic function $C_i = Cha(K_i)$ and modular function $M_i = Mod_2(K_i, C_i)$ is computed. Then it computes $D_3 = Id_i \oplus h(K_i \oplus M_i \oplus X_i)$ and $D_k = h(D_3 \|\| X_i \|\| K_i \|\| M_i \|\| Id_i)$. After computing, the IoT client device sends $< X_i, D_k, D_3, C_i >$ to cloud server CS.

3.3.4 Verification

In this phase, the authentication between IC_i and CS is verified. The following activities are performed in this phase:

Phase	Activities
Verification	i The CS after receiving $< X_i, D_k, D_3, C_i >$ from the IC_i, it computes $K_i' = X_i.s$, $M_i' = Mod_2(K_i, C_i)$. Then it computes $Id_i' = D_3 \oplus h(K_i' \oplus M_i' \oplus M_i)$ and $D_k* = h(D_3\vert\vert X_i\vert\vert K_i'\vert\vert M_i'\vert\vert Id_i')$. Then, it verifies $D_k* \overset{?}{=} D_k$. If the verification is successful, the CS selects the two random samples from the distribution function i.e $f_s, g_s \leftarrow \chi_\beta$ and computes $X_s = a.f_s + K.g_s$, $K_s = f_s.X_i$, $C_s = Cha(K_s)$, $M_s = Mod_2(K_s, C_s)$. Then it computes the session key with the D_1 value stored earlier. The session key $SK = (D_1\vert\vert X_i\vert\vert X_s\vert\vert K_i\vert\vert K_s\vert\vert M_i\vert\vert M_s)$, $D_z = h(SK\vert\vert D_1\vert\vert X_s\vert\vert K_s\vert\vert M_i)$. The CS sends the $< D_z, C_s, X_s >$ to IC_i. ii The IoT client device IC_i receives $< D_z, C_s, X_s >$ from the CS and computes $K_s' = f_i.X_s$,$M_s' = Mod_2(K_s', C_s)$. Then, IC_i determines $D_1 = D_i \oplus D_j$ and computes the session key $SK' = (D_1\vert\vert X_i\vert\vert X_s\vert\vert K_i\vert\vert K_s\vert\vert M_i\vert\vert M_s)$, $D_z' = h(SK'\vert\vert D_1\vert\vert X_s\vert\vert K_s\vert\vert M_i)$ and verifies the $D_z' \overset{?}{=} D_z$. if it holds,the session key is used for further secure communication.

In this phase, we verified authentication of the IC_i and CS similar to the Feng et.al scheme[48]. The first authentication between IC_i and CS is checking if D_k is equal to $h(D_3||X_i||K_i||M_i||Id_i)$. where,

$$K_i = f_i.CS_p$$

$$\implies K_i = f_i.(a.s + K.e)$$

$$\implies a.f_i.s + K.e.f_i$$

$$K'_i = X_i.s = (a.f_i + K.g_i).s = a.f_i.s + K.g_i.s$$

We can write as:

$$K_i = K'_i + K.(f_i.e - g_i.s)$$

from Lemma 2.3.19 and Lemma 2.3.20 :

$$\begin{aligned}|f_i.e - g_i.s| &\leq |f_i.e| + |g_i.s| \\ &\leq \sqrt{n}.||f_i||.||e|| + \sqrt{n}.||g_i||.||s|| \\ &< \sqrt{n}.\beta\sqrt{n}.\beta\sqrt{n} + \sqrt{n}.\beta\sqrt{n}.\beta\sqrt{n} \\ &\implies K.\beta^2.n^{3/2}\end{aligned}$$

From Lemma 2.3.20, $\beta = \omega\sqrt{\log n} \text{and} n < q$:

$$|f_i.e - g_i.s| < K.\beta^2.n^{3/2} < \frac{q}{8}$$

From Lemma 2.3.21, the following relation also holds :

$$M_i = Mod_2(K_i, C_i) = Mod_2(K'_i, C_i) = M'_i$$

Hence, the verification of both the IoT client device IC_i and the cloud server CS is conducted based on the Ring-LWE assumption, ensuring mutual authentication.

3.4 Correctness

Definition 3.4.1. *Decisional* $- RLWE_{n,q,\chi}$ assumption. for any PPT algorithm A, There exists $negl$-a negligible function,such that:

$$|\Pr[A^{O^n_s}(1^n) = 1] - \Pr[A^R(1^n) = 1]| = negl(n)$$

Let n, q be two positive integers, and χ_s, χ_e be bounded distributions over R. Distinguish two distributions of (a, b) and (a, u) , where $(b = a.s + e)$ for $a \overset{\$}{\leftarrow} R_q$,the secret $s \overset{\$}{\leftarrow} \chi_s$ and $u \overset{\$}{\leftarrow} R_q$.

$$|\Pr[A^{O_s^n}(1^n) = (a, b) = 1] - \Pr[A^R(1^n) = (a, u) = 1]| = negl(n)$$

Definition 3.4.2. A key exchange protocol Π is said to be correct if there exists a negligible function $negl$ such that for dimension n.

$$|\Pr[output_{A,\Pi}(1^n, a_i, b_i)] \neq \Pr[output_{B,\Pi}(1^n, a_j, b_j)| \leq negl(n)$$

Definition 3.4.3. A protocol Π for key exchange is secure in the presence of eavesdropping adversaries if for every PPT Adversary Eve, there exists a negligible function $negl$ such that

$$\Pr[RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1] \leq \frac{1}{2} + negl(n)$$

Eavesdropping Key Exchange Experiment $RLWE\_KE^{eav}_{Eve,\Pi}(1^n)$:

1. Parameters $a \leftarrow Z_q^n, s \in Z_q^n, e \leftarrow \chi$ chosen according to protocol Π
2. A random bit r is chosen such that $r \in \{0, 1\}$: if $r = 0$ then value of $b = output_{A,\Pi}(1^n, a_i, b_i)$, and if $r = 1$ then $b \leftarrow \{0, 1\}^n$
3. The security parameter 1^n is given to the Adversary Eve, and the transcript $transcript_\Pi(1^n, a_i, b_i)$ outputs a bit r'.Formally,

$$r' \leftarrow Eve(1^n, transcript(1^n, a_i, b_i)])$$

4. The experiment's output is defined as 1 if $r' = r$ and 0 otherwise. (In the case of $RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1$, we say Eve is successful.)

Theorem 3.4.4. Theorem: Assuming that the Decisional-Ring-LWE problem is hard and construction key exchange protocol is correct and secure in the presence of eavesdropping adversaries.

Proof: From the definition of experiment $RLWE\_KE_{Eve'}$ the adversary Eve receives $(1^n, a_i, b_i)$ and $b = as + e$ if $r = 0$ and $b \leftarrow R_q$ random element if $r = 1$. Distinguishing these two elements is equal to solving $Decisional - Ring - LWE$.

Let ε be a function that holds $\Pr[RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1] \leq \frac{1}{2} + \varepsilon(n)$ and $\Pr[r = 0] = \Pr[r = 1] = \frac{1}{2}$, then we have

$$\begin{aligned} \Pr[RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1] & \\ &= \frac{1}{2}.\Pr[RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1|r = 0] \\ &+ \frac{1}{2}.\Pr[RLWE\_KE^{eav}_{Eve,\Pi}(1^n) = 1|r = 1] \\ &= \frac{1}{2}.\Pr[Eve^{O^n_s}(1^n) = 1] + \frac{1}{2}.\Pr[Eve^{R}(1^n) = 1] \\ \text{Therefore} & \\ &= \frac{1}{2}.\Pr[Eve^{O^n_s}(1^n) = 1] + \frac{1}{2}.\Pr[Eve^{R}(1^n) = 1] = \frac{1}{2} + \varepsilon(n) \\ \text{Equivalently,} & \\ &= \Pr[Eve^{O^n_s}(1^n) = 1] + \Pr[Eve^{R}(1^n) = 1] = 1 + 2\varepsilon(n) \\ \text{That is} & \\ &= \Pr[Eve^{O^n_s}(1^n) = 1] = 1 - \Pr[Eve^{R}(1^n) = 1] \end{aligned}$$

$$1 - \Pr[Eve^{O^n_s}(1^n) = 1] + \Pr[Eve^{R}(1^n) = 1] = 1 + 2\varepsilon(n)$$

$$\Pr[Eve^{O^n_s}(1^n) = 1] - \Pr[Eve^{R}(1^n) = 1] = 2\varepsilon(n)$$

Since it is definition 3.4.1, hardness of Decisional-Ring-LWE is proved, the above difference is negligible and ε is also negligible. Therefore, the Ring-LWE-based protocol is also secure against Adversaries Eve.

3.4.1 Verification by AVISPA

The protocol undergoes validation using the AVISPA tool [96], an acronym for ”Automated Validation of Internet Security Protocols and Applications”. This tool is employed for the examination of Internet security protocols and applications.It verifies the security against various common types of attacks such as replay attacks, man-in-the middle attacks, message integrity, impersonation attacks and Denial of service(DoS) attacks. The security validation of the protocol uses to a High-Level Protocol Specification Language (HLPSL). HLPSL facilitates the development of a straightforward language for simulating transitions through a role-based state-transitions system, along with employing techniques for modular protocol specification. The assessment of the proposed protocol involves the utilization of two backends models: 1. CL-based Attack Searcher (CL-Atse) and 2. On-the-Fly Model-Checker (OFMC).

1. CL-AtSe: Its purpose is to detect attacks on protocol by creating a set of constraints derived from the security protocol outlined in the Intermediate Format (IF).

2. OFMC: It generates a tree based on the specified protocol for analysis and employs various symbols to explore specific states within the state space. This is achieved through both a demand-driven approach and an on-the-fly approach.

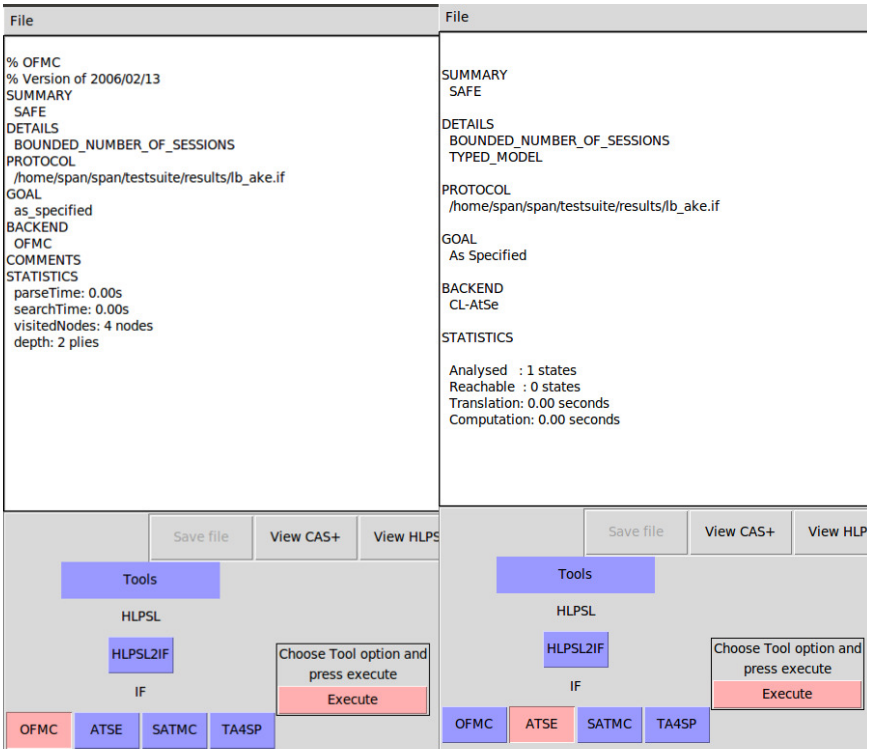
The protocol is analyzed by defining it in HLPSL and then testing it with backends. The following steps are carried out to analyze the protocol with AVISPA:

- Step 1: The HLPSL specification is used to represent the protocol.

- Step 2: The HLPSL code will be translated into IF using the HLPSL2IF translator.

- Step 3: The AVISPA tool’s back-end receives the translated IF specification as input.

There are two basic roles in our HLPSL protocol model: ‘IoT Client device’ and ‘Cloud Server(CS),’ which represent IC_i and CS, respectively. A ’session’ is established between the ’IoT client device’ and the ’cloud server’ after mutual authentication. After defining the basic roles, composed roles are defined to define the protocol’s ’session’.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/lb_ake.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 4 nodes
  depth: 2 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/lb_ake.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 1 states
  Reachable   : 0 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

Figure 3.2: Protocol verification using AVISPA tool

Figure: 3.2 depicts the output of the proposed protocol verification as ”SAFE” in both OFMC and CL-AtSe back-ends. In this scheme, the intruder has complete control over the network, and the intruder granted access to all communications transmitted by the agents assigned to their respective roles. With the acquisition of the required keys, the intruder becomes capable of intercepting, analyzing, and potentially modifying messages. He can assume the identity of any agent and communicate with any other agent via the communication channel. Since the intruder is provided complete control over the communication channel in this way, the backends execute all possible security attacks on the protocol. Simulation results indicate that the proposed protocol is secure. Furthermore, the protocol’s security is substantiated through a mathematical analysis grounded in the attack model. Consequently, it can be confidently used in real-world applications.

3.5 Security Analysis

In this section, a comprehensive security analysis is presented for the proposed authentication scheme designed for IoT devices. The protocol's security is analyzed under the IoT security requirements. Adversary capabilities are defined over the communication network and considered for analysis. An adversary may collect all messages, but he/she cannot extract the secret keys and cannot impersonate the Client IoT device. The node anonymity protects the identity and presence of IoT device in the network if an attacker intercepts the communication. To achieve node anonymity, we may use techniques like pseudo-random number, anonymous communication, group signatures, and selective decryption to address challenges of IoT networks.

For instance, even if an adversary attempts to impersonate someone, the verification process remains incomplete, leading to the rejection of communication. Additionally, the proposed authentication protocol offers security against various other types of attacks.

1. Reply Attack: The replay attack occurs when an adversary A obtains the authentication message from a previous session and uses it as a legitimate user in the current session. In the proposed protocol, every time, the cloud Server CS uses fresh f_s, g_s, and IoT client device IC_i uses fresh f_i, g_i generated from the distribution function χ_β. As a result, When an adversary tries to attempt to login using a random nonce, the login attempt will fail after verification and the server catches a replay attack. Therefore, the proposed protocol can withstand to replay attacks.

2. Clogging Attack: Suppose $< X_i, D_k, D_3, C_i >$ is intercepted by a malicious user M. M replaces D_k with D_m and replays $< X_i, D_m, D_3, C_i >$. Upon receiving it, the CS comes to the step computing D_k^* and subsequently compares it with D_m,

which fails. But by then, the CS has already performed lots of computational-intensive operations. The malicious user M can potentially replay lots of such messages to overload the CS.This type of attack is called clogging attack(a form of DoS attack) In this scenario, If the malicious user M potentially replay lots of such messages to overload the CS, it leads to a form of DoS attack. It is very common attack for any security application. But the proposed protocol enables timestamp to detect and prevent the malicious user with repeated faulty messages in the verification step. The other general solution is to detect and prevent DoS is, to use message counter for the fault messages from malicious user and will be blocked permanently.

3. Mutual Authentication: The cloud server $CS$and the IoT client device IC_i verify each other's authenticity using the conditions $D_k = h(D_3||X_i||K_i||M_i||Id_i)$, and $D_z = h(SK||D_1||X_s||K_s||M_i)$ respectively, where $SK = h(D_1||X_i||X_s||K_i||K_s||M_i||M_s)$,$D_k$, and D_z are computed using the IOT client device secret keys f_i; g_i. A randomly distributed number is required to retrieve Id_i from $D_3 = Id_i \oplus h(K_i||M_i||X_i)$. In the proposed protocol, the IoT client device and cloud server are capable of computing the session key. Therefore, the client and server independently verify one another.

4. Key freshness: The session key SK is compromised if the distribution function is used only once in communication for the generation of the random nonce. But, the Random sampling's uniqueness property ensures that each session will have a unique set of keys to compute session key SK. Our protocol maintains the critical property of key freshness.

5. Man-in-the-middle attack: The attacker with the malicious node may try to obtain communication parameters in this type of attack. In the proposed protocol, lattice-based parameters, and random distribution parameters, the malicious node cannot compute these parameters by solving the Decisional-RLWE problem in real polynomial time [8]. Additionally, the attacker must be aware of the unforgeable random nonce values sampled from error distribution function

χ_β i.e, $s, e \leftarrow \chi_\beta$. Hence, our protocol guarantees security against man-in-the-middle attacks.

6. Offline Dictionary attack: Assume the adversary obtains all data stored on the IoT client smart device such as D_1,D_2, and D_j. The adversary has to construct a D_i to get access; to do so, the adversary guesses the password Pwd_i even if the adversary does not know the Id_i. Without the IoT client identity, it is impossible to verify the correctness. Therefore, offline dictionary attacks on the proposed protocol are not practically possible.

7. Eavesdropping: The impersonation occurs when an attacker forges the genuine IoT node credentials along with the authentication messages and then attempts to modify the login request message of genuine IoT node IC_i credentials from $\{Id_i, pwd_i, X_i\}$ into $\{Id_i^*, pwd_i^*, X_i^*\}$. However, the malicious node cannot obtain the randomly sampled values f_i and g_i from the distribution function χ_β.in the proposed protocol. In addition to this, the secret value s and random key K which are computed by Cloud Server uniquely for the IC_i is can't be forged by the malicious node. Therefore, the eavesdropping attack fails at the first level of authentication in the proposed protocol.

8. Anonymity: The adversary cannot generate authentic messages G_k and D_z because protocol maintains anonymity and K_i, which is protected by random number f_i. During message verification, an adversary who impersonates an open transmitted message will be detected. Therefore, an impersonation attack is not possible.

9. Forward Secrecy: The protocol should not allow to compute previous message or data using the currently transmitted message or data. In this protocol, a malicious user is prohibited from obtaining device data by maintaining secrecy based on a random value of the nonce in each session from the distribution function χ_β. The quantum adversary has no means of knowing the randomly generated numbers within the device due to the protocol's resilience to replay

attacks. As a result, by adding unpredictability into prior communication packets, the protocol provides forward secrecy.

3.6 Implementation

Most commonly used control unit in IoT consists of a Microcontroller Unit (MCU) or a custom chip. A microcontroller is an integrated chip or core in a VLSI or SoC. Popular microcontrollers are ATmega 328, ATMega 32u4, ARM Cortex and ARM LPC. Arduino uses ATmega 328 or ATmega 32u4. Raspberry Pi uses ARM Cortex and ARM LPC microcontroller-based boards. The Cortex-A72 (ARM v8) 64-bit offers more computational power and memory capacity than Arduino making it well-suited for more complex future IoT applications that require higher processing capabilities, such as real-time data analysis. Arduino devices are optimized for tasks which is typically sufficient for the basic data collection and communication. However, implementation of proposed protocol on various types of IoT devices and comparative analysis is future direction for this work. The proposed protocol is implemented in the IoT environment scenario for authenticated communication. It aims to simulate Raspberry Pi as an IoT client and a laptop as a cloud server connected through Wi-Fi to exchange information. In this implementation, a scenario is conducted when a Raspberry Pi is connected to the network through Wi-Fi to compute private and public keys based on the Ring-LWE scheme. The implementation scenario is as shown in the figure:3.3.

3.6.1 Raspberry Pi

In the experiment, the Raspberry Pi (4 Model B) will act as a client, and a laptop will act as a cloud server. The raspberry pi runs a Broadcom BCM2711,quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with 1GB memory and supports wireless connectivity. The device and its specifications are provided in the table:3.2.

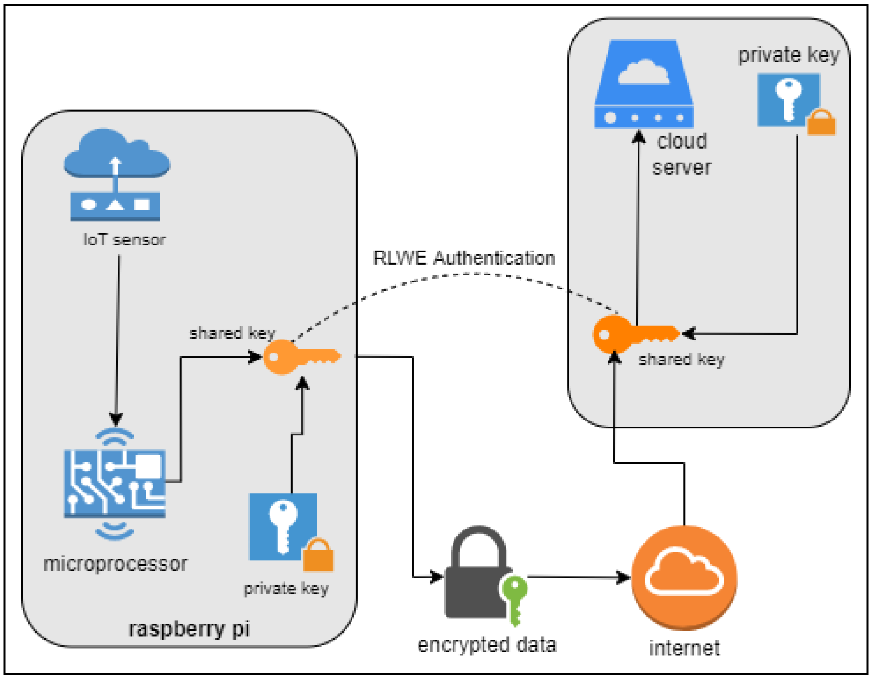
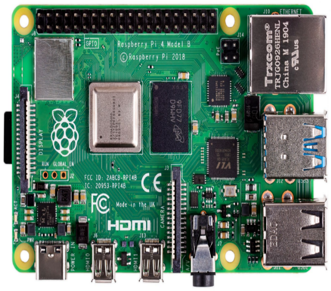


Figure 3.3: Illustration of protocol implementation

3.6.2 Experimentation Overview

The IoT client and cloud server programs are built with Python3 to connect using User Datagram Protocol(UDP). The NumPy library in python is used for mathematical computations which have functions of linear algebra, Fourier transform, and matrices. It supports a faster and easier workflow when testing and building Python applications. Initially, Raspberry Pi is accessed through SSH or the internet to embed the client program in its storage and server program in the laptop. The server and client are connected to the local Wi-Fi of the same network. The server calls $gen\_polyA(n,q)$ function to generate global polynomial A with input parameters n and q then shares with the IoT client device. With the received A, the client computes its public key by sampling secret s and error e from the distribution function. Similarly, The server also computes its public key from the samples s and e. The IoT client calls $iot\_client\_kex()$, and the server calls $cloud\_server\_kex()$ functions to exchange their public keys computed. Next, the client will compute the shared secret and send it to the server for verification. Finally, the server computes its shared secret and verifies the client's shared secret, and authenticates the IoT client device.

Table 3.2: Rasberrypi specifications

Raspberry Pi 4 model B	Specifications
	• Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz • Memory: 1GB LPDDR4 • Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports • SD card support: Micro SD card slot for loading operating system and data storage • Input power: 5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

3.6.3 Results

The Ring-LWE-based authentication protocol is implemented in an IoT client-server scenario by embedding the client program in a Raspberry Pi and the server program in a laptop. The IoT client is placed at a reachable distance of the local Wi-Fi connection for the internet. Similarly, the server is also connected to the same Wi-Fi network through which it can communicate with IoT client device. At first, the server program will run pre-computation and be ready to accept client connection with given parameter values of n and q. Then, the client will bind with the server connection on the server IP and Port number. Both client and server will compute their secrets and exchange public keys to compute shared keys. The client and server elapsed times are recorded separately and given in the following table:3.3.

Table 3.3 demonstrates the average key generation and exchange time for the Raspberry Pi and laptop using the NumPy library in Python. Different parameter values are considered for the experiment; initially, we started with the recommended

Table 3.3: Protocol implementation for different parameters

Parameters	Client time	Server time	Total time
n=256 q=7681	$\approx$ 224.302 ms	$\approx$ 4043.690 ms	4267.992 ms
n=512 q=12289	$\approx$ 442.761 ms	$\approx$ 3533.651 ms	3976.412 ms
n=743 q=2^{12}	$\approx$ 453.392 ms	$\approx$ 4417.312 ms	4870.704 ms
n=1024 q=12289	$\approx$ 233.011 ms	$\approx$ 3341.112 ms	3574.123 ms
n=1024 q=2^{32}	$\approx$ 238.447 ms	$\approx$ 10030.888 ms	10269.335 ms

parameter values of standard LWE [58]. For $n = 256$ bits and $q = 7681$ the total elapsed time is $\approx 4267.992ms$. Whereas, for the recent implementation of Ring-LWE based scheme $NewHope$ recommended parameters are $n = 1024, q = 12289$, and $n = 512$ for its variant scheme results $\approx 3976.412ms$ and $\approx 3574.123ms$, respectively. For the NTRUEncrypt [97] recommended parameters $n = 743, q = 2^{32}$ the total elapsed time is $\approx 4870.704ms$ comparatively higher than parameter values. Finally, for the BCNS [46] recommended parameters,$n = 1024, q = 2^{32}$, the total elapsed time $\approx 10269.335ms$, the highest of all other parameters.

3.6.4 Performance Evaluation

In this section, the proposed authentication protocol evaluated for computational and transmission costs and performance is analyzed with other protocols of the same category. There are several parameter choices for Ring-LWE lattices, as described by Feng et al. in [48]. For the 75 bits security level, we consider parameters as $n = 1024$ bits, odd prime $q = 47$ bits, and $log\beta = 17.1$ for the Gaussian distribution function χ_β, The hash function $h(.)$ is SHA-512 for fixed output length. These parameters are used to implement the proposed protocol in the IoT environment.

3.6.5 Computation Cost

In this section, we examine various operations cost of the proposed protocol. The set of notations used to represent the time complexity is listd in the table:3.4

The cost of the computations in the proposed authentication protocol is assessed with reference to similar work conducted by Feng et al. [48]. They have implemented

Table 3.4: Notations

Symbol	Description
t_{ge}	avg. elapsed time for Gaussian distribution χ_β.
t_{smul}	avg. elapsed time for component-wise scalar multiplication in R_q.
t_{smul}	avg. elapsed time for component-wise scalar multiplication in R_q.
t_{pmul}	avg. elapsed time for component-wise multiplication in R_q.
t_{pma}	avg. elapsed time for component-wise multiplication and addition in R_q.
t_{cha}	avg. elapsed time for characteristic function(Cha) in R_q.
t_h	avg. elapsed time for secure hash function $SHA - 512$.

the protocol using Miracl and LatticeCrypto Library using C/C++ as a programming language. According to their implementation, the average execution time for lattice-related operations is tabulated as in table:3.5.

Table 3.5: Average elapsed times of Lattice-operations(100000 executions)

Lattice operation	Client device(ns)	Server device(ns)
t_{ge}	561.483	73.503
t_{smul}	6.655	0.298
t_{pmul}	13.052	0.307
t_{pma}	29.505	2.549
t_{cha}	35.515	0.689
t_h	180.964	14.09

In the proposed protocol, when a user initiates the IoT client device with login credentials Id_i, Pwd_i, the device verifies credentials using one hash function. If the login is valid then it samples two random values $f_i, g_i \leftarrow \chi_\beta$ then it performs component-wise addition and multiplication operations over R_q. The characteristic function Cha is applied to the value K_i after computing it using component-wise multiplication. The Mod_2 operation is performed only after the AND operation. so, the cost of Mod_2 was neglected. Next, we apply one hash function to compute D_3 and another hash function to compute D_k then send to cloud server CS_i. So, the total computation time of the client device is: $t_{ge} + t_{smul} + t_{pma} + t_{cha} + 3t_h \approx 1176.05$ns.

After receiving a message from the IoT client device, the server verifies the login by computing one component-wise multiplication and then generates random samples

$f_s, g_s \leftarrow \chi_\beta$. Then, it performs componentwise multiplication and addition operations over R_q. Then it computes the K_s with one more component-wise multiplication and gives as input to the characteristic function Cha. Finally, the session key SK will be computed with a hash function along with two more hash functions, one for encrypting the identity of the client device and secret key of the server, another for the encryption of the session key to sent IoT client device IC_i. Therefore, the total computation time at the cloud server-side is $t_{ge} + 2t_{smul} + t_{pma} + t_{cha} + 3t_h \approx 119.625$ns.

Finally, the IoT client device IC_i receives the message from the cloud server CS then computes the session key SK' using two hash functions along with one component-wise multiplication in R_q used to compute K'_s. So, the total time of the client device is $t_{smul} + 2t_h \approx 368.583$ns. The total time at the client device side is $1176.05 + 368.583 \approx 1544.633$ns. Since the proposed authentication protocol is for an IoT environment, there are few works to compare which are similar but proposed for mobile devices and mobile users.[48],[28],[30].

In [48],the mobile *App* is used to login and connect with the server. One hash function is used to verify the identity, then, After the successful login, the *App* will generate two randomly sampled values, one component-wise scalar, one component-wise multiplication, and addition operation in R_q is used for the authentication process. Therefore, the total computation time is: $t_h + 2t_{ge} + t_{smul} + t_{pma} \approx 1340.09$ns. The server, upon receiving the message from *App* computes two randomly sampled values, one component-wise multiplication, one component-wise multiplication, and addition in R_q, one characteristic function and one hash function is performed. Therefore, the total computation time at the server is $2t_{ge}+t_{smul}+t_{pma}+t_{pmul}+t_{cha} \approx 164.939$ns. Then, the *App* upon receiving a message from the server, it computes one hash function and one component-wise multiplication for authentication. Then, it computes the session key by using four hash functions, one component-wise multiplication, and one characteristic function in R_q. Therefore, the total $t_{pmul} + t_h + 4t_h + t_{pmul} + t_{cha} \approx 966.439$ns. In the last step, the server computes the session key using four hash functions and one component-wise multiplication i.e. $t_h + t_{pmul} \approx 56.667$ns. Overall computation time at *App* side is: ≈ 2306.529 ns and server side is: ≈ 221.606 ns.

In [30], after successful login the mobile user has to compute eight hash functions, once point addition and two scalar multiplications in g_1, two exponentiation operations in g_2. The total computation cost at user side is: $8t_h + t_{sm} + t_{pa} + t_{exp} \approx 44.21$ milliseconds. Then, the server computes five hash functions, one point addition in g_1, two multiplications, and four exponentiation operations in g_2. The total computation cost on the server side is $5t_h + t_{pa} + 2t_{mul} + 4t_{exp} \approx 10.004$ milliseconds. The notations used are:t_{sm}:avg.elapsed time for scalar multiplication in g_1;t_{pa}:avg.elapsed time for point addition in g_1;t_{exp}:avg.elapsed time for exponentiation in g_2 as defined in [30].

In [28], after successful login, the mobile user has to compute five hash functions and three chaotic functions. The total computation time at the user side is $5t_h + 3t_c \approx$ 98.105 milliseconds. Similarly, the server needs to compute three hash functions and one extended chaotic function. The total computation time on the server side is $3t_h + t_c \approx 5.625$ milliseconds. The notations used are:t_h:avg.elapsed time for hash function;t_c:avg.elapsed time for extended chaotic map function as defined in [28]. The comparisons of protocols w.r.t number of operations performed at client-side and server-side is analyzed and presented in the table:3.6.

Table 3.6: Comparison w.r.t number of operations

Protocol	Client operations	Server operations
Islam et al.[28]	$5t_h + 3t_c$	$3t_h + t_c$
Debiao et al.[30]	$8t_h + t_{sm} + t_{pa} + t_{exp}$	$5t_h + t_{pa} + 2t_{mul} + 4t_{exp}$
Feng et.al[48]	$6t_h + t_{ge} + 3t_{smul} + t_{pma} + t_{cha}$	$4t_h + t_{ge} + t_{smul} + t_{pma} + 3t_{pmul} + t_{cha}$
Proposed	$5t_h + t_{ge} + 2t_{smul} + t_{pma} + t_{cha}$	$3t_h + t_{ge} + 2t_{smul} + t_{pma} + t_{cha}$

The comparisons of protocols w.r.t total computation time for client-side operations and server-side operations are analyzed and presented in the table: 3.7.

Table 3.7: Comparison w.r.t computation time

Protocol	Client time	Server time	Total time
Islam et al.[28]	$\approx$ 98105000 ns	$\approx$ 5625000 ms	103730000 ns
Debiao et al.[30]	$\approx$ 44210000 ns	$\approx$ 10004000 ns	54214000 ns
Feng et.al[48]	$\approx$ 1732.252 ns	$\approx$ 134.32 ns	1866.572 ns
Proposed	$\approx$ 1544.633 ns	$\approx$ 119.978 ns	1664.611 ns

3.6.6 Communication Cost

The communication cost of the proposed authentication protocol is assessed with reference to similar work conducted by Feng et al.[48], We consider the polynomial in R_q which is of the size 4096 bits, and the hash function considered is SHA-3 for 512 bits output. The modulus q of the size 1024 bits and timestamp considered 32 bits used in this analysis.

In [48], the user sends the message $(x_i, aid_i, \omega_i, \alpha_i)$ to the server and the server replies with the message $(x_s, \omega_s, \alpha_s)$ to the user. In this, the $x_i, x_s \in R_q$, $\omega_, \omega_s \in 0, 1$ outputs of characteristic functions and $\alpha_i, \alpha_s, aid_i$ are general hash functions. Therefore,, the communication cost is evaluated as: for the message $(x_i, aid_i, \omega_i, \alpha_i)$ is 4096+512+1+512=5121 bits, and for the message $(x_s, \omega_s, \alpha_s)$ is 4096+1+515=4609 bits with a total of 5121 bits+4609 bits=9730 bits.

In the proposed protocol, the transmission message from the IoT client deviceIC_i and the cloud server CS consists of four parameters (X_i, D_k, D_3, C_i). Then, the communication cost is computed as 4096 +512+1+512= 5121 bits. The cloud server CS upon receiving this message, replies with (D_z, X_s, C_s) to IoT client device IC_i. The communication cost of this message is computed as 4096+1+512=4609 bits with a total of 5121 bits+4609 bits=9730 bits. Therefore, the proposed protocol communication overhead is the same as Feng et.al [48] protocol.

3.7 Summary

In the era of the Internet of Things, numerous authentication protocols have been suggested, leveraging modern cryptographic approaches to address challenging computational problems. However, most of them are proved vulnerable to quantum attacks. The chapter described our first contribution scheme, a post-quantum secure authentication and key exchange protocol for IoT. The security analysis demonstrates that the proposed protocol is provably secure and meets the requirements for security in an IoT environment. The security of the protocol is proved formally. The protocol has been verified automatically using the AVISPA tool. The protocol performance is analyzed in terms of communication and computation costs. The protocol is implemented on Raspberry Pi with different parameters. Our scheme has a lower computation cost compared to similar types of protocols. Due to the protocol's high resistance to quantum attacks and various types of attacks, it can be helpful in a wide variety of applications to ensure quantum security at different levels.

Chapter 4

Quantum-secure N2N Authentication protocol model for IoT Sensor Networks

This chapter introduces and validates a quantum-secure node-to-node authentication protocol model designed for the Internet of Things network environment. The security of our scheme relies on the Inhomogeneous Short Integer Solution ($ISIS$) problem. The protocol model undergoes verification against known attacks in the IoT network. Performance analysis and comparison with relevant protocols demonstrate that the proposed protocol is distinctive and quantum-secure.

The chapter is organized as follows: Section 4.1 presents the introduction. Section 4.2 presents the motivation and contribution overview of the proposed scheme. Section 4.3 presents phases of the proposed protocol that are used in the construction of the scheme. In Section 4.4, we describe the correctness of the scheme along with its security analysis. Section 4.5 presents the results and performance evaluation of the scheme. We also compare the computation cost of the proposed scheme with the existing schemes and finally, Section 4.6 summarizes the chapter.

4.1 Introduction

The rapid development of wireless communication and sensor networks is the basis for forming an Internet of Things (IoT) infrastructure. In IoT-based applications, the authentication and access control protocols must be robust to withstand current attacks. The majority of security protocols are based on integer factorization and discrete logarithm problems, which are proved vulnerable to quantum attacks. In this chapter, we propose a quantum-secure node-to-node communication protocol for the Internet of Things infrastructure network. The protocol is designed and proved its correctness formally based on $ISIS$ hard problem in lat- tices. The protocol is verified against known attacks in the IoT network. The protocol's performance is analyzed and compared with relevant protocols. It shows that the proposed protocol is unique and quantum-secure.

4.2 Contribution and Protocol Overview

In this, the recently proposed lattice-based authentication scheme is analyzed [72], and we proposed a quantum-secure node-to-node communication protocol model for IoT sensor networks by making use of the scheme design[73]. The protocol is modeled, and its correctness is proved formally based on the hardness of the Inhomogeneous Short Integer Solution ($ISIS$) problem on lattices. The security of the protocol model is verified against known attacks on the IoT infrastructure. We considered the well-known three-party protocol model to analyze the performance of the proposed model. It is analyzed for a 100-bit security level with specified security parameters. The average computation cost is computed for the number of hash functions($h(.)$) and polynomial multiplication(PM) operations. The proposed model, required $2h(.) + 3PM$ for the IoT node,$1h(.)+2PM$ for the Gateway device, and in total $3h(.)+5PM$ operations are needed. We also compare our protocol model to similar protocols and demonstrate that it is both computationally efficient and quantum-safe. The adversary can not get any information from the communication of IoT nodes. The contributions of the scheme are summarised as follows:

1. We proposed a node-to-node authentication protocol model for IoT infrastructure. It is suitable for resource-constrained devices with smaller parameters.

2. The protocol ensures the anonymity of the IoT user's identity, which is known only to the gateway device. No adversary *Adv* can get the identity information.

3. We analyzed the protocol's security in the presence of an adversary and proved its correctness and safety from known attacks and threats.

4. We presented the performance evaluation of the protocol in terms of computation and communication overhead.

4.2.1 Security Requirements

Security requirements of communication between IoT nodes and Cloud server i) Authentication and Message Integrity ii) Node Anonymity iii) Session key Security iv) Resistance to Attacks v)Resistance to password guessing attacks vi)Forward secrecy.

4.2.2 System Model

In this model, the Gateway Device (GWD) is authenticating IoT nodes and generates the session key for secure communication. This protocol model is not extended to Cloud Server level in this scenario. So, we have not highlighted the Cloud server (CS) interactions in this model. The IoT infrastructure network is created with a number of nodes that operate as clients and connect with the GWD that connects to ($CloudServer$) over the internet. The CS is assumed as trusted, and the Gateway Device(GWD) work is to validate IoT Nodes (IN_1),(IN_2) and relay messages. It models a communication between IN_i to GWD, GWD to IN_2 over a network entirely regulated by a Probabilistic polynomial Time (PPT) adversary *Adv*. The IN_1 and IN_2 seek to communicate with each other using intermediary device GWD by using a shared session key. Adversary *Adv* can listen to the conversation between IN_1 and GWD,IN_2 and GWD, and *Adv* can respond, edit, delay, and insert new messages. The PPT adversary *Adv* is granted access to the Oracle model that generates protocol-simulated outputs for *Adv*'s query. It may enable protocol communications between any number of IoT Node instances, the transmission of any message to these instances, and the monitoring of IN_1, IN_2, and GWD answers under protocol requirements. Session keys created by GWD and IoT node instances may also be revealed. Finally, adversary Adv can directly obtain values stored in the IN_1 and IN_2 or a password through multiple trials.

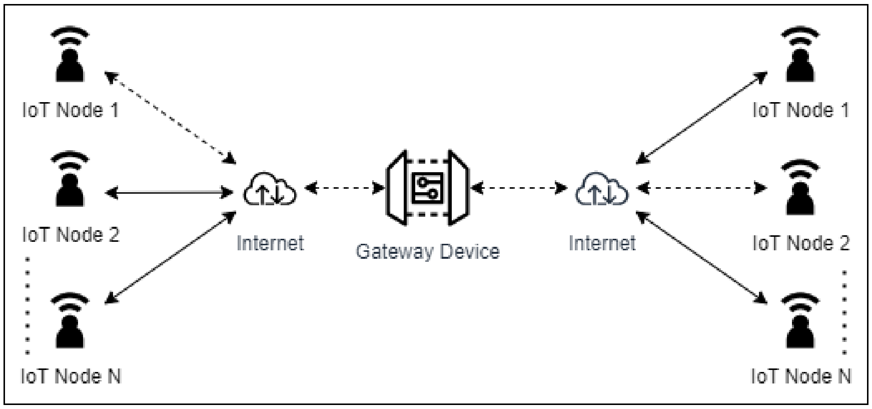


Figure 4.1: Node-to-node communication scenario in IoT network

Following are the assumptions for the IoT infrastructure scenario: The IoT nodes (IN_1),(IN_2) and Gateway Device (GWD) are identified with Unique ID and One-time handshake. The GWD and Cloud Server (CS) are mutually authenticated using PKE. The Central Key Repository (CKR) in the network maintains the IDs of Connected devices in the network. An Unauthorised IoT node(Quantum attacker) tries to access the IoT infrastructure network. The scenario of IoT infrastructure network is shown in the figure:4.1.

4.2.3 Adversary Model

We specify the adversary's capabilities to register at a gateway device GWD to obtain public keys. On behalf of adversary-controlled parties, the opponent can register any arbitrary public key of its choice, including public keys equal to keys of certain honest parties in the system. The Adversary Adv sends incoming messages to the parties via $Send$ inquiries, receives the parties' outgoing messages, and decides on their delivery, effectively controlling all communications between the parties.

- Send(message). The message can take one of three forms: $(IN_i; GWD)$, $(IN_i; GWD; out)$, or $(IN_i; GWD; in; out)$. According to the protocol, the adversary receives the session's response, and the variables *in* and *out* are initialized and changed (through concatenation) accordingly.

To detect information leakage, the adversary *Adv* is permitted to do the following queries, the results of which are designated as data sets $Secret\_key$, $Session\_state$, and $Session\_key$. We utilize $secret\_key$ to describe a session's secret randomness, and $Session\_State$ to denote all intermediate secret values computed or used in the host machine's memory, as well as the $session\_key$. If the session is completed, the $Session\_state$ contains results and $Session\_key$.

- Secret_key leak(s): The adversary *Adv* obtains the $Secret\_key$ associated with session s.
- Session_state leak(s): The adversary *Adv* obtains the $Session\_State$ of the owner of session s.
- Session_key leak(s): The adversary *Adv* obtains the $Session\_key$ in a completed session s.
- Credentials leak(IN_i): The adversary *Adv* obtains the credentials of node IN_i by this query.

Definition 4.2.1. Freshness of Session-Id *sid*:) Let us consider a completed session owned by an honest IoT Node IN_i and a trusted Gateway Device GWD. Assume *sid* is the matching session of s. If none of the following conditions are met, we claim the session *sid* is fresh.

1. IN_i issues Session_Key leak(s), or Session_Key leak(s) if s exists;
2. IN_i issues Session_State leak(s), or Session_State leak (s) if s exists;
3. s exists, and adversary *Adv* does one of the following queries: (a) both Credentials leak(IN_i) and Secret_Key leak(s), or (b) both Credential leak(GWD) and Secret_key leak(s) (s),

4. s does not exist, and adversary Adv performs either (a) both Credential leak(IN_i) and Secret_key leak(s), or (b) Credential leak(GWD).

Definition 4.2.2. Freshness of Session s): Let us consider a completed session owned by an honest IoT Node IN_i and a trusted Gateway Device GWD. Assume sid is the matching session of s. Session s is said to be fresh if none of the following conditions are met:

1. Adv issues Session_key leak(s), or Session_key leak(s) if s exists;
2. Adv issues Session_State Reveal(s), or Session_State leak (s) if s exists;
3. s exists, and adversary A does one of the following queries: (a) both Credentials leak(IN_i) and Secret_key leak(s), or (b) Credentials leak(GWD) and Secret_key(s) (s),
4. s does not exist, and adversary Adv performs either (a) both Credential leak(IN_i) and Secret_key leak(s), or (b) Credential leak(GWD) before the session s is completed.

Security Experiment: The adversary Adv's goal is to distinguish a session key from a random key, hence the experiment goes as follows. Adv is initially provided a set of honest parties and is free to execute any sequence of the questions outlined above. Adv asks the following question during the experiment.

Test_Session(s_t): This query requires that s_t be a new session. A random bit b is chosen to respond to this query. If $b = 0$, the session key of s_t is returned. Otherwise, a random key is generated.

The experiment will continue until Adv makes a b estimate. If the test session s_t is still fresh and Adv's guess is correct, i.e., $b' = b$, Adv Succeeds the game. The adversary's advantage in the $ISIS\_N2N$ protocol experiment is specified as:

$$Adv_{ISIS\_N2N}(Adv) = Pr[Adv\_Succeeds] - \frac{1}{2}$$

Definition 4.2.3. Security of N2N Protocol: A Node-to-Node communication protocol is secure π is deemed $ISIS$ secure if it is operating against all PPT adversaries Adv with the given capabilities, and the freshness of the session is determined by the freshness of session id sid it holds:

i Communication Parties who finish matching sessions compute the same session key.

ii $Adv_{N2N}(Adv)$ is negligible for any adversary Adv with the assumed capabilities.

Definition 4.2.4. Security of N2N with Forward Secrecy : A Node-to-Node communication protocol is considered $N2N$ secure with forward secrecy if, for all PPT adversaries Adv with the assumed capabilities running against it, and the freshness of session is determined by the freshness of session id sid, it possesses the following:

i Parties who finish matching sessions compute the same session key.

ii $Adv_{N2N}(Adv)$ is negligible for any adversary Adv with the assumed capabilities.

Definition 4.2.5. Inhomogeneous Small Integer Solution(ISIS) Problem: The Small Integer Solution(SIS) problem is defined as: In Given a modulus q, m uniformly random vectors $a_i \in Z_q^n$, forming the columns of a matrix $A \in Z_q^{n\times m}$, find a nonzero integer vector $v \in Z^m$ of norm $||v|| \leq \beta$ such that

$$f_A(v) := A.v = \sum_i a_i.v_i = 0 \in Z_q^n$$

The inhomogeneous version of the SIS problem is to find a short integer solution to $A.x = u \in Z_q^n$, where $A \in Z_q^{n\times m}, u \in Z_q^n$ are uniformly random and independent [98].

$$f_A(x) := A.x = \sum_i a_i.x_i = u \in Z_q^n$$

The $ISIS$ problem is a special case of a Small Integer Solution(SIS) problem. Using this problem, we construct the node-to-node authentication protocol. The construction of the protocol is described in the section 4.3 and its notations are listed in the table 4.1.

Table 4.1: Notations

Symbol	Description
IN_1, IN_2	Two IoT Nodes 1 and 2
x_1, x_2	Two unique identification numbers of nodes
s_1, s_2	Secrets keys of Nodes
r_1, r_2	Ramdom values generated by Nodes
GWD	Gateway Device
A	Uniform matrix from $Z_q^{n\times m}$
Adv	Adversary/Attacker
$H_1(.), H_2(.)$	Hash functions
n_1, n_2	Random values generated for Nodes

4.3 Proposed Model

In the IoT infrastructure network, the IoT Node is initially registered at a Gateway Device (GWD) that connects to the cloud server(CS) and other IoT nodes. After registration of the IoT node at GWD, it operates automatically and is controlled by GWD. The proposed node-to-protocol for this IoT infrastructure network has four phases: setup, registration, login and authentication, and verification phase. The set of activities is shown in the figure: 4.2.

4.3.1 Setup

The setup phase begins with the Gateway Device(GWD), which generates the parameters required for the communication along with its private key. The activities performed by GWD are as follows:

i The Gateway Device(GWD) chooses modulus q,n and m uniformly random vectors in Z_q^n. The permutation $\sigma \in P_m$ is a linear operation chosen from set of all permutations P_m, R_σ is associated $m \times m$ binary matrix, and matrix $A \in Z^{n\times m}$ providing $u = A.x(\mod q)$.

ii The Gateway Device(GWD) generates u_i for each IoT node N_i which has a unique identification number x_i in the proposed scheme that satisfies $u_i = A.x_i(\mod q)$. The GWD stores x_i, u_i and R_σ^{-1} in database. The (GWD) also computes $A.R_\sigma^{-1}$ and used for the verification.

iii Each IoT node (IN_i) chooses its secret key $s_i \overset{\$}{\leftarrow} Z_q^m$ that satisfies the $ISIS$ problem. The IoT nodes also randomly chooses $r_i \overset{\$}{\leftarrow} \{0,1\}^m$. All computations are performed in modulo q.

4.3.2 Registration

During this phase, the IoT Node IN_i will be registered at GWD for cloud server CS services and will communicate with other IoT Nodes IN_j. After completing the registration process, the IoT Node will receive its private key, which it will use to communicate with GWD in a secure channel. The following activities are performed in this phase:

i The IoT node IN_i is assigned with its identity and password (Id_i, Pwd_i). The user who initiates the device will login with the given credentials. After the login into device,the IN_i computes $D_i = H(Id_i||Pwd_i)$ and sends $< Id_i, D_i >$ to GWD.

ii The GWD upon receiving the registration request (Id_i, D_i) from IN_i, GWD verifies Id_i then assignes IN_i's new unique identity x_i and computes $u_i = A.x_i($ mod $q)$ then shares A, u_i with the IoT node IN_i to store in it. The GWD also computes $A.R_\sigma^{-1}$ for the verification of the committed values.

iii The IoT node IN_i upon receiving A, u_i, it generates the secret key $s_i \in Z_q^m$ by following the $ISIS$ properties. It also generates a random key $r_i \overset{\$}{\leftarrow} \{0,1\}^m$ before staring the authentication phase.

Similarly, Another IoT node IN_j will be registered at GWD for further communications. The IoT node IN_j upon receiving A, u_j, it generates the secret key $s_j \in Z_q^m$ by following the $ISIS$ properties. It also generates a random key $r_j \overset{\$}{\leftarrow} \{0,1\}^m$

4.3.3 Login and Authentication

In this phase, the GWD validates both IoT Nodes IN_i and IN_j credentials whenever logins. After authentication of both IoT Nodes IN_i and IN_j at GWD, a secure

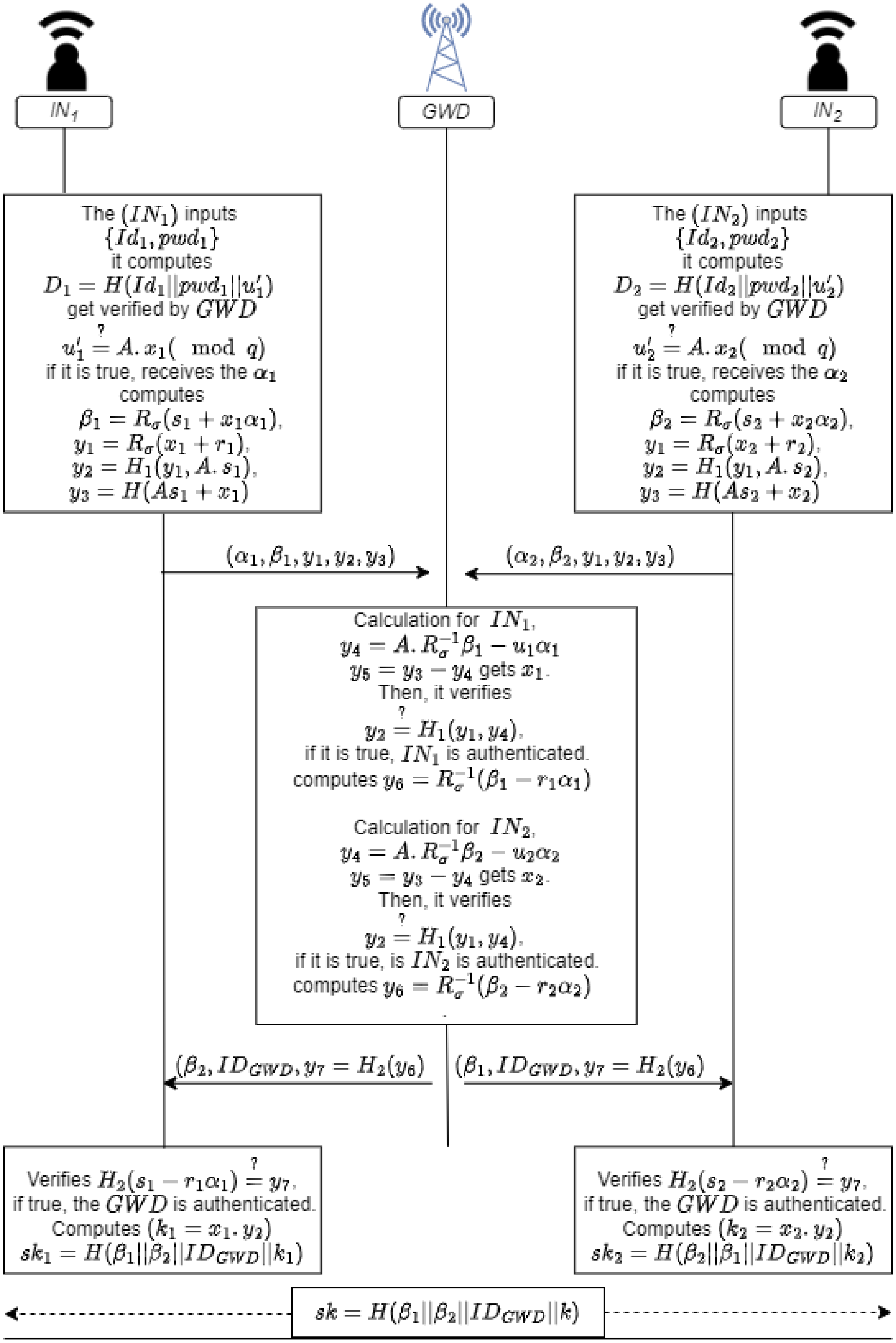

IN_1
GWD
IN_2
The (IN_1) inputs $\{Id_1, pwd_1\}$ it computes $D_1 = H(Id_1||pwd_1||u_1')$ get verified by GWD $u_1' \overset{?}{=} A.x_1(\mod q)$ if it is true, receives the α_1 computes $\beta_1 = R_\sigma(s_1 + x_1\alpha_1)$, $y_1 = R_\sigma(x_1 + r_1)$, $y_2 = H_1(y_1, A.s_1)$, $y_3 = H(As_1 + x_1)$
The (IN_2) inputs $\{Id_2, pwd_2\}$ it computes $D_2 = H(Id_2||pwd_2||u_2')$ get verified by GWD $u_2' \overset{?}{=} A.x_2(\mod q)$ if it is true, receives the α_2 computes $\beta_2 = R_\sigma(s_2 + x_2\alpha_2)$, $y_1 = R_\sigma(x_2 + r_2)$, $y_2 = H_1(y_1, A.s_2)$, $y_3 = H(As_2 + x_2)$
$(\alpha_1, \beta_1, y_1, y_2, y_3)$
$(\alpha_2, \beta_2, y_1, y_2, y_3)$
Calculation for IN_1, $y_4 = A.R_\sigma^{-1}\beta_1 - u_1\alpha_1$ $y_5 = y_3 - y_4$ gets x_1. Then, it verifies $y_2 \overset{?}{=} H_1(y_1, y_4)$, if it is true, IN_1 is authenticated. computes $y_6 = R_\sigma^{-1}(\beta_1 - r_1\alpha_1)$
Calculation for IN_2, $y_4 = A.R_\sigma^{-1}\beta_2 - u_2\alpha_2$ $y_5 = y_3 - y_4$ gets x_2. Then, it verifies $y_2 \overset{?}{=} H_1(y_1, y_4)$, if it is true, is IN_2 is authenticated. computes $y_6 = R_\sigma^{-1}(\beta_2 - r_2\alpha_2)$
$(\beta_2, ID_{GWD}, y_7 = H_2(y_6)$
$(\beta_1, ID_{GWD}, y_7 = H_2(y_6)$
Verifies $H_2(s_1 - r_1\alpha_1) \overset{?}{=} y_7$, if true, the GWD is authenticated. Computes $(k_1 = x_1.y_2)$ $sk_1 = H(\beta_1||\beta_2||ID_{GWD}||k_1)$
Verifies $H_2(s_2 - r_2\alpha_2) \overset{?}{=} y_7$, if true, the GWD is authenticated. Computes $(k_2 = x_2.y_2)$ $sk_2 = H(\beta_2||\beta_1||ID_{GWD}||k_2)$
$sk = H(\beta_1||\beta_2||ID_{GWD}||k)$

Figure 4.2: Proposed Protocol

session key will be computed for both IoT nodes for node-to-node communication in IoT infrastructure network.

i To login the IoT node (IN_i), the user will enters the login credentials $\{Id_i, pwd_i\}$. Then it computes $D_i = H(Id_i||Pwd_i||u'_i)$ and sends it to GWD. The GWD validates $\{Id_i, pwd_i\}$ and verifies the unique identification value assigned to it at the time of registration $u'_i \overset{?}{=} A.x_i(\mod q)$ the IoT nodes from the database.

ii Upon the successful login of IN_i, the GWD chooses a random value$\alpha_i \in Z_q$ and sends it to IN_i.

iii The IN_i receives the α_i and computes $\beta_i = R_\sigma(s_i + x_i\alpha_i)$, $y_1 = R_\sigma(x_i + r_1)$, $y_2 = H_1(y_1, A.s_i)$,$y_3 = H(As_i + x_i)$ and sends $< \alpha_i, \beta_i, y_1, y_2, y_3 >$ to GWD.

iv The GWD after receiving $< \alpha_i, \beta_i, y_1, y_2, y_3 >$ from the IN_i, it computes $y_4 = A.R_\sigma^{-1}\beta_i - u_i\alpha_i$ and $y_5 = y_3 - y_4$ to get the identity x_i. Then, it verifies $y_2 \overset{?}{=} H_1(y_1, y_4)$, if it is true, the IoT node is verified successfully. Then, the GWD sends hash of $y_6 = R_\sigma^{-1}(\beta_i - r_1\alpha_i)$, β_j, ID_{GWD} to the IoT node IN_i.

v The IN_i receives the $y_7 = H_2(y_6), \beta_j,_{GWD}$ value and verifies $H_2(s_i - r_1\alpha_i) \overset{?}{=} y_7$, if it is true, the GWD is authenticated successfully. Then, it computes the $k_i = x_i.u_j$ and session key $sk_i = H(\beta_i||\beta_j||ID_{GWD}||k_i)$.

Similarly, another IoT node IN_j will be authenticated at GWD and IN_j authenticates the GWD. Then,the key $k_j = x_j.u_i$ will be used to compute the shared session key $sk_j = H(\beta_j||\beta_i||ID_{GWD}||k_j)$ for node-to-node communication through GWD device.

4.3.4 Verification

In this phase, the authentication between IoT Node IN_i and GWD is verified. The mutual authentication is verified for both the IoT Node side and Gateway Device side as described in the scheme [72]. In this protocol, the major authentication is done at GWD side. The GWD receives the values β_i, y_1, y_2, y_3 from the IoT node IN_i and computes y_4, y_5, y_6. Then, it verifies the identity of the IN_i by $y_2 \overset{?}{=} H_1(y_1.y_4)$. The proposed protocol is secure if following verification equations are correct.

- Verification-1: The IoT Node IN_i sends the β_i value to the GWD to compute the y_4 value as:

$$\begin{aligned} y_4 &= A.R_\sigma^{-1}.\beta_i - u_i.\alpha_i \\ &= A.R_\sigma^{-1}(R_\sigma(s_i + x_i\alpha_i)) - u_i.\alpha_i \\ &= A.R_\sigma^{-1}R_\sigma.s_i + A.R_\sigma^{-1}R_\sigma.x_i.\alpha_i - u_i.\alpha_i \\ &= A.s_i + A.x_i.\alpha_i - u_i.\alpha_i \\ &= A.s_i \quad (\because u_i = A.x_i) \end{aligned}$$

- Verification-2: After computing the y_4,the GWD computes the unique identification number assigned to it by computing y_5 as:

$$\begin{aligned} y_5 \implies x_i' &= (y_3 - y_4) \\ &= A.s_i + x_i - A.R_\sigma^{-1}\beta_i + u_i.\alpha_i \\ &= A.s_i + x_i - A.s_i(\because y_4 = A.s_i) \\ &= x_i \end{aligned}$$

- Verification-3: After successful authentication of the IoT node IN_i by $Verification-1, 2$,the GWD computes the value y_6 and send its hash value to IoT Node for verification as follows:

$$\begin{aligned} y_6 &= R_\sigma^{-1}(\beta_i - y_1.\alpha_i) \\ &= R_\sigma^{-1}(R_\sigma.(s_i + x_i.\alpha_i) - R_\sigma(x_i + r_1.)\alpha_i) \\ &= R_\sigma^{-1}R_\sigma s_i + R_\sigma^{-1}R_\sigma x_i\alpha_i - R_\sigma^{-1}R_\sigma x_i\alpha_i - R_\sigma^{-1}R_\sigma r_1\alpha_i \\ &= s_i + x_i\alpha_i - x_i\alpha_i - r_1.\alpha_i \\ &= s_i - r_1\alpha_i \end{aligned}$$

Therefore, the IoT node IN_i and the Gateway Node GWD were verified under the ISIS assumptions and mutually authenticated.

4.4 Security Analysis

This section provides a detailed security analysis of the proposed node-to-node communication protocol for the IoT infrastructure network. The protocol's security is analyzed under the IoT security requirements. Adversary capabilities are defined over the communication network and considered for analysis. An adversary may collect all messages, but he/she cannot extract the secret keys and cannot impersonate the legitimate IoT Node. For example, though an adversary Adv can impersonate some node IN_t, it fails in the verification process and the communication will be rejected. Similarly, the proposed authentication protocol is tested for its security against the following attacks..

1. Reply Attack: The replay attack occurs when an adversary A obtains the authentication message from a previous session and uses it as a legitimate user in the current session.In the proposed authentication protocol, the GWD generates the α_i for the IoT node IN_i to computes β_i, y_1, y_2, y_3 and replies to GWD. Now, GWD verifies the α_i generated uniquely. If an adversary tries to reply with these parameters, fails in the verification process. So, he/she can't reply to GWD messages. Similarly, Even if an adversary gets α_i from the previous session, the IoT node IN_i generates random value $r_1 \in \{0,1\}^m$ for computation of committed new values. When an adversary tries to attempt to reply with already used α_i, he/she will be caught in a replay attack. Therefore, the proposed protocol is withstanding replay attacks.

2. Man-in-the-middle attack: The attacker with the malicious node may try to obtain communication parameters in this type of attack. In the proposed protocol, the IoT Node device IN_i uses fresh and random values generated from $r_1 \in \{0,1\}^m$ and the secret key from $s_i \in Z_q^m$. As a result, when an adversary

tries to attempt to log in using a random nonce, the login attempt will fail after verification and the server will catch a Man-in-the-middle attack. The malicious node cannot compute these parameters by solving the $ISIS$ problem in the polynomial time [8]. Therefore, the protocol ensures security against man-in-the-middle attacks.

3. Impersonation attack: An impersonation occurs when an attacker tries to act as a genuine IoT node IN_i and eavesdrops on authentication messages. Each IoT Node computes values $r_1 \in \{0,1\}^m, x_i, R_\sigma, s_i$ based on the $ISIS$ problem. If an adversary impersonates an IoT node IN_i, then he/she has to compute committed values $y_1, y_2, y_3,$ and β_i by solving $ISIS$ problem. It is impossible to solve in polynomial time. Similarly, if the adversary tries to eavesdrop on transferred messages to GWD, he/she has to reply IN_i. But, only GWD knows the unique identity number x_i and it can only compute committed values using IN_i node's secret values A, R_σ^{-1}, u. Therefore, the impersonation or eavesdropping attack fails at the first level of authentication in the proposed protocol.

4. Mutual Authentication: The Gateway Device GWD and the IoT Node IN_i verify each other's authenticity by computing $y_5 \implies x_i' = (y_3 - y_4)$ for verify unique identity number assigned to IN_i and The IN_i receives the $y_7 = H_2(y_6)$ value and verifies $H_2(s_i - r_1\alpha_i) \overset{?}{=} y_7$ to authenticate GWD. Similarly, another IoT node IN_j will be authenticated at GWD and IN_j authenticates the GWD. Then, the key $k = x_1.u_j$ or $k = x_2.u_i$ will be used to compute the shared session key $sk = H(x_i||x_j||ID_{GWD}||k)$ for both IoT nodes for node-to-node communication through GWD device. In the proposed protocol, the IoT Node IN_i and GWD are capable of computing the session key. Therefore, mutual authentication is achieved.

5. Forward Secrecy: The protocol should not allow to compute of previous messages or data using the currently transmitted message or data. In this protocol, an adversary is prohibited from obtaining the IN_i's secret key s_i and shared values A, u between GWD and IN_i. Even if an adversary gets the transferred

messages, he/she can't get the secret key and shared parameters. The only way to get these values is to solve $ISIS$ problem, which has proved hard. Because of this, the protocol gives forward secrecy by making each session of computation less predictable.

6. Node Privacy: The IoT node replies to GWD by computing committed values β_i, y_1, y_2, y_3, if IN_i computes the same values for each session, its easy to trace the node. In the proposed protocol, each value is random and independently generated from a pseudo-random generator and one-way hash functions. The IoT node IN_i information is not included in computed committed values to avoid tracing. Therefore, an adversary cannot identify whether the transferred messages are from the same tag or not. Therefore, the proposed protocol achieves the Node privacy property.

7. Scalability: The Gateway Device(GWD) generates u_i for each IoT node N_i which has a unique identification number x_i in the proposed scheme that satisfies $u_i = A.x_i(\mod q)$. The GWD stores x_i, u_i and R_σ^{-1}in the database. If the new active IoT nodes are added to the network or the number of IoT nodes are increased in the network, it will not lead to extra computation cost to GWD.

8. Offline Dictionary attack: Assume the adversary obtains all data stored on the IoT Node device such as A,u_i. The adversary has to construct a β_i, y_1, y_2, y_3 to get the access; to do so, the adversary guesses the password pwd_i even if the adversary does not know the Id_i. Without the IoT node identity number x_i issued by GWD, it is impossible to verify the correctness. Therefore, offline dictionary attacks on the proposed protocol are not practically possible.

Similar to previous chapter, the protocol scenario is changed from Node-to-Server authentication to Node-to-Node authentication in this chapter. However, both protocols are designed based on hardness of Ring-LWE problem in lattices. So, protocol verification with AVISPA will be similar and verifies common types of attacks such as replay attacks, man-in-the middle attacks, message integrity, impersonation attacks and Denial of service(DoS) attacks.

4.5 Performance Evaluation

In this section, the proposed authentication protocol is evaluated for computational costs, and performance is compared with other protocols that fall under a similar category.

4.5.1 Computation Cost

In this section, we scrutinize the computational overhead incurred by the participating IoT nodes in the proposed protocol. The computational cost of the authentication protocol is assessed, drawing insights from analogous work conducted by Shafiq et al. [73]. We considered the same implementation process for the analysis of the proposed protocol. According to the specified protocol implementation, the average execution time for related operations are listed as in the table: 4.2.

Table 4.2: Comparison w.r.t. number of operations

Protocol	IoT Node Operations	GWD Operations	Total Operations
Proposed	2h(.)+3PM	1h(.)+2PM	3h(.)+5PM
[73]	5h(.)+3PM	4h(.)+2PM	9h(.)+5PM
[30]	6h(.)+4PM	8h(.)+8PM	14h(.)+12PM
[31]	8h(.) +5PM	4h(.)+4PM	12h(.)+9PM
[99]	4h(.)+3PM	9h(.)+6PM	13h(.)+9PM
[100]	9h(.)	6h(.)	15h(.)
[101]	11h(.) +4PM	6h(.)+4PM	17h(.)+8PM
[102]	8h(.)+4PM	6h(.)+1PM	14h(.)+5PM
[103]	13h(.)+1PM	10h(.)	23h(.)+1PM

In the proposed protocol, when a user initiates the IoT node with login credentials Id_i, pwd_i, the device verifies credentials using one hash function. If the login is valid then it samples one random value $r_i \overset{\$}{\leftarrow} \{0,1\}^m$ and $s_i \overset{\$}{\leftarrow} Z_q^m$ then it performs two additions and three multiplications operations to compute committed values. Next, we apply two hash functions to send the values to GWD. As the time for addition operations and random value generation is negligible, we can consider the total computation time of the IoT node device is Time for 2 hash functions and Time for 3 multiplication operations.

After receiving a login request from the IoT Node, the GWD verifies the login by computing one multiplication operation $u'_i = A.x_i(\mod q)$ then generates the $\alpha_i \overset{\$}{\leftarrow} Z_q$ and sends to IoT Node IN_i. The GWD receives committed values from the IN_i then it performs two multiplication operations, two addition operations, and one hash function. Finally, the session key sk will be computed with one more hash function. As the time for addition operations is negligible, the total computation time at the Gateway device is Time for two multiplications and Time for Two hash functions. Since, the proposed node-to-node communication protocol IoT network, there are few works for comparative analysis. We considered most of the communication protocols listed in the scheme [73] to compare the proposed protocol computation time.

4.5.2 Security Level and Parameter Selection

The proposed protocol is access for the 100-bit level security and a related set of parameters similar to the protocol by Akleylek et al.[72]. The adversary has to solve the ISIS problem to find the x_i from $A.x_i = u \mod q$. For this reason, it it important to choose appropriate parameters. We use parameters defined in the protocol [57], which are bounded to the theorem proved in [91]. For the 100-bit security level, we consider $m = 2048$, m-bit binary vectors are defined as the private key and hamming weight of the private key is $m/2$. The parameters are $n = 64, q = 257, H(.) = 256$ bits.

The gateway device GWD performs are two matrix-vector operations that lies in Z_q.The space complexity and time complexity is $\mathcal{O}(n \times m)$ in Z_q. It also performs the multiplication operation $A.R_\sigma^{-1}$ and stores it in the database. Each IoT node performs three matrix-vector products, the time and space complexity is $\mathcal{O}(n \times m), \mathcal{O}(n \times m)$ respectively. The proposed protocol is compared with other node-to-node authentication protocols in terms of security properties in the table:4.3. The proposed protocol satisfies the requirements of an IoT infrastructure network. It also shows the node-to-node interaction process through the gateway device. In the table 4.3, notations refers to IN_i, IN_j:Users, S_i, S_j:Sensor nodes, TA:Trusted Authority, CS: Cloud Server IN_i. The proposed protocol is analyzed based on the implementation of Shafiq et al.[73].

Table 4.3: Comparative security analysis of protocols

Protocol	Interactions	1	2	3	4	5	6	7	8	9
Proposed	$IN_i \rightarrow GWD \rightarrow IN_j$	✓	✓	✓	✓	✓	✓	✓	✓	✓
[73]	$IN_i \rightarrow GWD \rightarrow IN_j$	✓	✓	✓	✓	✗	✓	✓	✓	✗
[30]	$IN_i \rightarrow S_i \rightarrow GWD \rightarrow S_j \rightarrow IN_j$	✓	✓	✗	✓	✓	✓	✗	✗	✗
[31]	$IN_i \rightarrow TA \rightarrow S_i$	✓	✓	✓	✓	✓	✓	✗	✓	✗
[99]	$IN_i \rightarrow S_i \rightarrow CS \rightarrow S_j \rightarrow IN_j$	✓	✓	✓	✓	✓	✓	✗	✓	✗
[100]	$IN_i, S_i \rightarrow GWD \rightarrow IN_j, S_j$	✓	✓	✓	✓	✓	✓	✗	✓	✗
[101]	$IN_i \rightarrow GWD \rightarrow S_i \rightarrow GWD \rightarrow IN_j$	✓	✓	✓	✓	✓	✓	✗	✓	✗
[102]	$IN_i \rightarrow GWD \rightarrow S_i \rightarrow IN_j$	✓	✓	✗	✓	✗	✗	✓	✗	✗
[103]	$IN_i \rightarrow GWD \rightarrow S_i \rightarrow GWD \rightarrow IN_j$	✓	✓	✓	✓	✓	✓	✗	✓	✗

1:Reply Attack, 2:Man-in-the-middle attack:, 3:Impersonation attack:, 4:Mutual Authentication:, 5:Forward Secrecy, 6:Node Privacy, 7:Scalability, 8:Offline Dictionary attack, 9:Quantum Secure

The node-to-node interaction is created with a minimum number of messages in the proposed protocol. Multiple IoT nodes can register at GWD devices independently in the IoT infrastructure network. If any two nodes want to communicate, the gateway device generates a secure session key sk after the authentication process.

4.6 Summary

This chapter described a quantum-secure node-to-node authentication protocol model for the Internet of Things network environment. Our scheme security is based on the Inhomogeneous Short Integer Solution problem; shortly, we call$ISIS$ problem. The protocol model is verified against known attacks in the IoT network. The protocol's performance is analyzed and compared with relevant protocols. It shows that the proposed protocol is unique and quantum-secure. The protocol's performance is analyzed and compared with relevant protocols.The implementation of proposed protocol on various types of IoT devices and comparative analysis is future direction of this work.

Chapter 5

Construction of communication protocol using Ring-LWE-based Homomorphic Encryption in IoT-Cloud Environment

This chapter presents a Ring-LWE-based homomorphic encryption scheme for the security and privacy of user data in a cloud environment. Various types of homomorphic encryption schemes are studied for data privacy in the cloud. The Ring-LWE-based encryption scheme is presented for privacy protection in the cloud which meets the homomorphic properties. The scheme is analyzed for security, privacy, reduced messaging overhead, and computation overhead. This chapter aims to formulate and build a communication protocol based on Ring-LWE homomorphic encryption. This protocol is designed for authenticated user message encryption within an IoT cloud computing environment. The evaluation function in holomorphic encryption is defined based on Ring-LWE encryption for practical sharing-enabled cloud storage. Then, formally proving the security of the proposed protocol for classical and quantum attacks in a cloud environment like Manin-the-middle (MITM) attacks, Denial of Service (DoS), and Replay Attack.

The chapter is organized as follows: Section 5.1 presents the introduction. Section 5.2 presents the contribution and construction overview of the proposed scheme. Section 5.3 presents the phases of the proposed protocol that are used in the construction of the scheme. In Section 5.4, we describe the correctness of the scheme along with its security analysis. Section 5.5 presents the analysis and performance evaluation of the scheme. We also compare the proposed scheme with the existing traditional homomorphic encryption schemes and finally, Section 5.6 summarizes the chapter.

5.1 Introduction

The objective of conventional modern encryption is primarily to provide the confidentiality of stored or transmitted data. Effective encryption ensures that even if an unauthorized party gains access to the encrypted data, they are unable to comprehend its contents. In the current digital era, a fundamental principle is that encrypted data should appear indistinguishable from random data. The more obscure a message is, the less information it reveals. The perfectly encrypted data discloses no information, and in a robust conventional encryption system, extracting meaningful information from encrypted data is unfeasible without the corresponding decryption key. Homomorphic encryption (HE), on the other hand, pursues a different goal. It aims to enable computations on encrypted data without the need for decryption, and crucially, without access to the decryption key for the encrypted data. The objective of this contribution is to Design and Construct a Ring-LWE-based homomorphic encryption communication protocol for authenticated user message encryption in an IoT cloud computing environment. The evaluation function in holomorphic encryption is defined based on Ring-LWE encryption for practical sharing-enabled cloud storage. Then, formally proving the security of the proposed protocol for classical and quantum attacks in cloud environments like Man-in-the-middle (MITM) attacks, Denial of Service (DoS) and Replay Attacks.

The first lattice-based homomorphic encryption technique was proposed in the year 2009 by Gentry [69] based on ideal lattices. This technique supports full homomorphism with additive and multiplicative homomorphic properties. It means it supports for addition and multiplication of ciphertext with an unlimited number of times. Later, the homomorphic encryption schemes evolved rapidly, and many schemes were proposed by researchers. There are three distinct categories of homomorphic encryption schemes. The first category encompasses a fully homomorphic encryption scheme based on ideal lattices, initially proposed by Gentry in 2009. This scheme involves constructing a Somewhat Homomorphic Encryption (SWHE) on the ideals of different rings. It employs techniques such as compressing the decryption circuit to reduce polynomials and utilizes bootstrapping technology to achieve fully homomorphic encryption, assuming cyclic security.

The second category comprises an integer-based homomorphic encryption scheme [70]that aligns with Gentry's concept but eliminates operations based on ideal lattices of the polynomial ring. Instead, all operations are performed using integers.

In the third category, homomorphic encryption methods are based on either fully homomorphic Learning With Errors (LWE) or Ring-LWE (Learning With Errors over Ring). These schemes rely on the concept of Learning With Errors to attain fully homomorphic encryption capabilities.

This method uses non-linearization to build a fully homomorphic encryption system, similar to the *BGV* encryption scheme and is based on fault-tolerant learning [71].

5.2 Contribution and Protocol Overview

In this, we proposed a scheme for the security and privacy of user data in a cloud environment. Various types of homomorphic encryption schemes are studied for data privacy in the cloud. The Ring-LWE-based encryption scheme is presented for privacy protection in the cloud which meets the homomorphic properties. In this scheme, IoT nodes will register at the cloud server, then the server authenticates IoT nodes and accepts the encrypted data to share with other nodes whenever requested. It stores the data at cloud server with quantum-safe encryption. For the encryption of the data at cloud server, the Ring -LWE based fully homomorphic encryption(FHE) is used for quantum-enabled security and privacy. The proposed scheme is analyzed for security and compact in the presence of a quantum attacker. By employing FHE-based data management and verification methods, the overall efficiency and security surpass those offered by traditional encryption algorithms. Additionally, the implementation of signature verification minimizes the associated overhead, thus enhancing the efficiency of the verification process when compared to existing methods. The objectives of this scheme are:

1. We designed and constructed a Ring-LWE-based homomorphic encryption communication protocol for authenticated user message encryption in a IoT cloud computing environment.

2. We proposed the evaluation function in homomorphic encryption is defined based on Ring-LWE encryption for practical sharing-enabled cloud storage.

3. We, formally analyzed and proved the security of the proposed protocol for classical and quantum attacks in cloud environments like Man-in-the-middle (MITM) attacks, Denial of Service (DoS) and Replay Attack.

5.2.1 Security Requirements

Security requirements of IoT node data and Cloud server communication are as follows:

1. Privacy: User data must remain confidential and should not be disclosed to any third party. It should only be accessed and managed by the cloud server (CS) and the respective service provider (SP). The IoT node typically equipped with various sensors, the data collected by IoT node should remain strictly confidential, under the management of only the authorized entities and not be shared with any unauthorized third parties. However, the user retains control over their data strictly following privacy policy.

2. Confidentiality and Message Integrity: In the presence of a potential eavesdropper, it is desired that the content of any message remain hidden, preventing the adversary from accessing the actual information in the data.

3. Availability: An adversary could potentially launch a Denial of Service (DoS) attack with the intention of obstructing access to the Cloud Server (CS). Therefore, it is crucial for the CS to remain accessible to all parties whenever it is needed.

5.2.2 System Model

The IoT infrastructure network is created with a number of nodes that operate as clients and connect to ($CloudServer$) over the internet. The Identity Provider (IDP) monitors the identities of users to certify the public key. The CS is assumed to be trusted, and the IDP work is to validate the IoT-Node IN_i, and relay messages. It models communication between IN_i to CS, CS to IN_i through IDP over a network entirely regulated by a Probabilistic polynomial Time (PPT) adversary Adv. The IN_i and CS seek to communicate and exchange data with each other using the help of IDP by computing public key and private key pairs to each user. Adversary Adv can listen to the conversation between IN_i and CS, and Adv can respond, edit, delay, and insert new messages. The PPT adversary Adv is granted access to the Oracle model that generates protocol-simulated outputs for Adv's query. It may enable protocol communications between any number of IoT Node instances, the transmission of any message to these instances, and the monitoring of IN_i, and CS answers under protocol requirements. The encryption keys generated by CS and IoT node instances may also be revealed. Finally, an adversary Adv can directly decrypt data stored at CS or a password through multiple trials.

Following are the assumptions for the IoT infrastructure scenario: The IoT nodes (IN_i) and cloud server (CS) are identified with a unique ID and One-time handshake. The Cloud Server (CS) and IoT node are mutually authenticated using PKE. The IDP in the network maintains the IDs of Connected devices in the network. An Unauthorised IoT node(Quantum attacker) tries to access the IoT infrastructure network. The scenario of the IoT infrastructure network is shown in the figure:5.1.

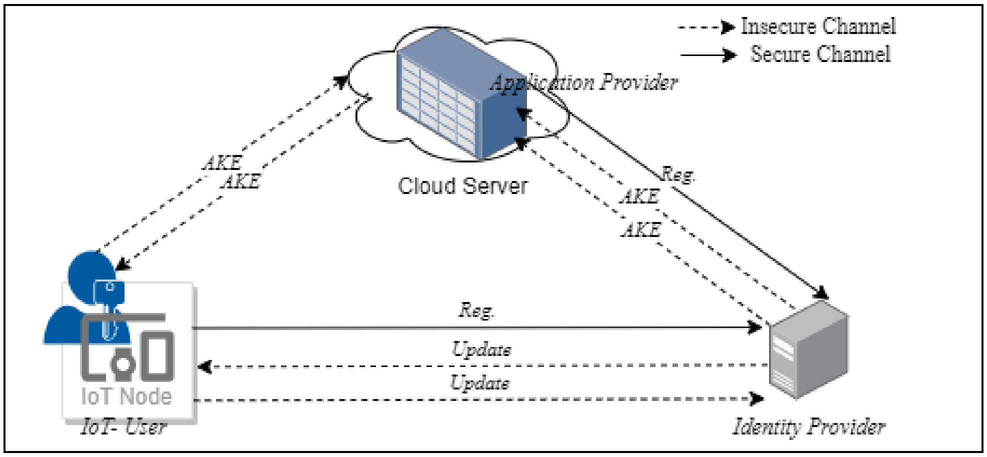


Figure 5.1: System Model

5.2.3 Threat Model

We consider the storage of application data on third-party clouds in a cloud-based environment. The cloud application involves three key entities: the Cloud Server (CS) responsible for storage, the IoT Node User (with apps), and an Identity Provider (IDP) tasked with certifying the public key of each user.

- The Cloud Server (CS) is required to maintain integrity by faithfully adhering to the protocol while simultaneously attempting to maximize knowledge extraction from the stored data.

- The Service provider (SP) should be genuine, if protocol violations are detected, SP could be penalized.

- The Adversary *Adv* is motivated to acquire additional insights into user data, acting passively to gain access to encrypted information while avoiding detection.

- Threats: We consider cloud-side threats as well as client-side threats such as data leakage, unauthorized access, data privacy, user privacy, and malicious insider attacks.

5.2.3.1 Assumptions

As we focused on encryption of data in the scheme, the other communications are assumed to be strong. But, In the reality, there might not be state-of art security mechanism for end to end secure communication. Constructing end-to-end post quantum communication is our future work, by combining proposed protocols. In this model, In addition to the honest cloud assumption, we assume that the Identity Provider(IDP) correctly verifies generates, and verifies CS and IN key pairs for encryption and signature. We make the following assumptions in addition to the honest cloud assumption.

- We assume that the Identity Provider (IDP) accurately verifies the identity-key pairs of users. The IDP can either be an external entity that is well-known and trustworthy, or an internal unit within the system.
- We consider the members of shared data as semi-trusted. This means that they do not collaborate with the cloud provider to expose member data or keys.
- The protocol assumes that the applications involved behave correctly and do not disclose user keys to malicious entities.
- Additionally, we assume that state-of-the-art security mechanisms are implemented to ensure device security and all communication between parties takes place over secure channels.

We also outline the capabilities of the Adversary who enrolls at the cloud server CS with the intention of gaining data access. The Adversary, acting through controlled entities, possesses the ability to register any public key of its choosing, even if it matches the keys of legitimate parties within the system. The protocol's design and implementation are elaborated in the section ?? and its notations are listed in the table ??.

5.3 Proposed Model

The proposed protocol is comprised of two primary phases: 1. Setup and Key Generation Phase and 2. Data Encryption phase. During the First phase, the initial setup of IDP and key generation is performed for IoT node IN Users and CS takes place, and mutual authentication is performed. In the Data Encryption Phase, the transmission of dynamic messages to upload encrypted data in the cloud, as well as corresponding queries on the data is performed.

5.3.1 Setup and Key Generation Phase

In this phase, IDP is set up and involved in the key generation and sharing of public keys between Users and CS with their respective controllers.

1. Setup: It produces the protocol parameters to generate (public key, private key) for both the IoT Node IN User and CS. The IDentity Provider IDP establishes a ring $R_q = Z_q(x)/ < f(x) >$, where $f(x) = x^n + 1 \in Z(x)$, and n is a power of 2. This choice ensures that $f(x)$ is irreducible over the rational numbers. R_q represents the ring of integer polynomials modulo. Additionally, the IDP selects a prime number, denoted as $p \in Z_q^*$. All parties involved are provided with information regarding the ring and the prime number selected.

2. Encryption Key Generation: The IDP generates the public-private key pairs for both parties similar to the Ring-LWE [59] scheme. For CS The vector $a_{cs} \in R_q$, two (s_{cs}, e_{cs}) small elements from error distribution χ_σ for is the std. deviation σ. In this case, s_{cs}- is a secret key and computes $(b_{cs} = a_{cs} \cdot s_{cs} + e_{cs})$. Now $< s_{cs}, (a_{cs}; b_{cs}) >$ are the private and public key pairs. Similarly,For IoT Node IN user, The vector $a_{in} \in R_q$, two (s_{in}, e_{in}) small elements from error distribution χ_σ for is the std. deviation σ. In this case, s_{in}- is a secret key and computes $(b_{in} = a_{in} \cdot s_{in} + e_{in})$. Now $< s_{in}, (a_{in}; b_{in}) >$ are the private and public key pairs.

3. Signature key Generation: The Ring-LWE-based digital signature scheme [104]is

employed for the purposes of signing and verification. A similar encryption key pair, IDP generates private and public key pairs for both parties. It uses $c_{cs} \in R_q$ and two random elements (s_{cs}, e^*_{cs}) from χ_σ and computes $d_{cs} = (c_{cs} * s_{cs} + p * e^*_{cs})$ for CS where p prime number. For IoT Node IN, it generates $d_{in} = (c_{cs} * s_{in} + p * e^*_{in})$.

5.3.2 Data Encryption Phase

In this phase, the data encryption and exchange takes place between CS and IoT Node IN User with its signature verification.

1. The IoT Node IN collects the data from the sensor and requests CS to accept the encrypted data D_i. It encrypts the data D_i using the public key of CS, and signs it, then sent to CS.

2. Upon receiving the user's request, CS creates an entry in its database with the data index and a corresponding timestamp. The entry is structured as follows: $(IN_i; D_i; T_i)$. Here, IN_i represents the identification of the ith IoT user, D_i is the data uploaded from the user and T_i timestamp.

3. To encrypt the data D_i using Fully Homomorphic Encryption (FHE), a process is employed. For an n-bits of Data D_i, which employs polynomials with binary coefficients (0/1), three random elements $(r, e_1, e_2 \in R)$ are generated from the error distribution χ_σ. Subsequently, the pair $(u_{in}, v_{in}) \in R_q^2$ is computed as the encryption of data D_i.

$$u_{in} = a_{cs} \cdot r + e_1 \mod q$$

$$v_{in} = b_{cs} \cdot r + e_2 + (q/2) * D_i \mod q$$

The IoT node sends the encrypted data to CS as $(u_{in}, v_{in}, h(D_i), T_i)$

Decryption(u_{in}, v_{in}, s_{cs}): The CS verifies the signature of the user and stores it in the database,

$$v_{in} - u_{in} * s_{cs} = (r * e - s_{cs}.e_1 + e_2) + (q/2) * D_i \mod q$$

When selecting suitable parameters, it is possible to ensure that the magnitude of $(r \cdot e - s_{cs} \cdot e_1 + e_2 \in R)$ is less than $q/4$. As a result, the bits of D_i can be extracted by rounding each coefficient of $(v_{in} - u_{in} \cdot s)$ to either 0 or $q/2$, depending on which value is closest modulo q.

When a data access request comes to cloud server CS by another IoT user IN_j, it encrypts the data and adds CS signature value. It generates a new random value r_{new} and computes the signature along with the time stamp T_{new}. It chooses new A_{cs},$e_1 \in \chi_\sigma$ and computes

$$(B_{cs} = (A_{cs} + h(r_{new}|T_{new}) \cdot ss_{cs} + p * e_1)$$

The CS replies with message $< h(r_{new}|T_{new}), A_{cs}, B_{cs} >$ to the requested node IN_j. It receives and verifies the signatures of CS as.

$$(-c_{cs} \cdot B_{cs} + d_{cs} \cdot A_{cs}) \mod p == -d_{cs} \cdot h(r_{new}|T_{new}) \mod p$$

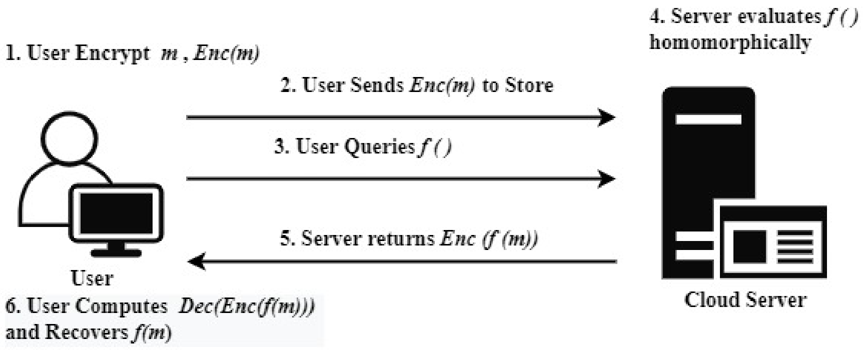

1. User Encrypt *m* , *Enc(m)*
2. User Sends *Enc(m)* to Store
3. User Queries *f ( )*
4. Server evaluates *f ( )* homomorphically
5. Server returns *Enc (f (m))*
User
6. User Computes *Dec(Enc(f(m)))* and Recovers *f(m)*
Cloud Server

Figure 5.2: Illustration of proposed IoT Cloud Server encryption

5.3.3 FHE Verification

Let's consider the encryption parameters as p, q, and r, where p is a positive odd number and q is a large positive integer. During the key generation phase, both p and q are determined. Here, p serves as the encryption key, while r is a randomly chosen number used for encryption.
Given data d, the encrypted data is calculated as follows:

$$Enc_d = d + 2.r + p \cdot q$$

The recovered data value

$$Dec(Enc_d) = (Enc_d \mod p) \mod 2$$

since the $p \times q$ is less than $(2r + d)$,then

$$(Enc_d \mod p) \mod 2 = (2r + d) \mod 2 = d$$

Additive Property verification: Let's consider d_1 and d_2 are two data values. By applying encryption, we can transform these data values to encrypted form as follows:

$$\begin{aligned}
Enc_{d_1} &= d_1 + 2.r_1 + pq_1 \\
Enc_{d_2} &= d_2 + 2.r_2 + pq_2
\end{aligned}$$

In the above equations, Enc_{d_1} and Enc_{d_2} represent the encrypted data. Additionally, r_1 and r_2 are random values chosen for encryption, and pq_1 and pq_2 denote the product of a prime number p and a quadratic residue q.

After the addition operation: $d_3 = (d_1 + d_2)$; the resulting expression for Enc_{d_3} is:

$$\begin{aligned}
Enc_{d_3} &= Enc_{d_1} + Enc_{d_2} \\
&= (d_1 + d_2) + 2(r_1 + r_2) + p(q_1 + q_2)
\end{aligned}$$

If $(d_1+d_2)+2(r_1+r-2)$ is significantly smaller than p, we can simplify the expression for Enc_{d_3} as follows:

$$\begin{aligned} Enc_{d_3} &= (Enc_{d_1} + Enc_{d_2}) \mod p \\ &= (d_1 + d_2) + 2(r_1 + r_2) \end{aligned}$$

Hence, the Additive Homomorphic Encryption (AHE) condition is satisfied.

Multiplicative Property Verification: Let's consider the multiplication equation: $d_4 = (d_1 + d_2)$. In this case,

$$\begin{aligned} d_4 &= d_1 \times d_2 \\ &= (d_1 + 2r_1 + pq_1) \times (d_2 + 2r_2 + pq_2) \\ &= d_1d_2 + 2(2r_1r_2 + r_1d_2 + r_2d_1) \\ &+ p[pq_1q_2 + q_2(d_1 + 2r_1) + q_1(d_2 + 2r_2)] \end{aligned}$$

If $(d_1d_2) + 2(2r_1r_2 + r_1d_2 + r_2d_1)$ is significantly smaller than p, then we can express Enc_{d_4} as:

$$\begin{aligned} Enc_{d_4} &= (Enc_{d_1} \times Enc_{c_2}) \mod p \\ &= d_1d_2 + 2(2r_1r_2 + r_1d_2 + r_2d_1) \end{aligned}$$

Therefore, the Multiplicative Homomorphic Encryption (MHE) property is satisfied.

5.4 Security Analysis

This section provides a detailed security analysis of the proposed node-to-node communication protocol for the IoT infrastructure network. The protocol's security is analyzed under the IoT security requirements. Adversary capabilities are defined over the communication network and considered for analysis. An adversary may collect all messages, but he/she cannot extract the secret keys and cannot impersonate the legitimate IoT Node. For example, though an adversary *Adv* can impersonate some node IN_t, it fails in the verification process and the communication will be rejected. Similarly, the proposed authentication protocol is tested for its security against the following attacks..

1. Reply Attack: The replay attack occurs when an adversary *Adv* obtains the authentication message from a previous session and uses it as a legitimate user in the current session. In the proposed scheme, the CS generates the random value r for the IoT node IN_i used to compute (u, v).If an adversary tries to reply by decrypting these parameters, it fails in the verification process. So, he/she can't reply to CS messages. Similarly, Even if an adversary gets r from the previous session, the IoT node IN_i generates random value $r_i \in \{0, 1\}^m$ for computation of committed new values. When an adversary tries to attempt to reply with already used r, he/she will be caught in a replay attack. Therefore, the proposed protocol is withstanding replay attacks.

2. Man-in-the-middle attack: The attacker with the malicious node may try to obtain communication parameters in this type of attack. In the proposed protocol, the IoT Node device IN_i uses fresh and random values generated from $r_i \in \{0, 1\}^m$ and the secret key from $s_i \in Z_q^m$. As a result, when an adversary tries to attempt to log in using a random nonce, the login attempt will fail after verification and the server will catch a Man-in-the-middle attack. The malicious node cannot compute these parameters by solving the $ISIS$ problem in the polynomial time [8]. Therefore, the protocol ensures security against man-in-the-middle attacks.

3. Impersonation attack: An impersonation occurs when an attacker tries to act as a genuine IoT node IN_i and eavesdrops on authentication messages. Each IoT Node computes values $r_i, s_i, e_i \in \{0, 1\}^m$ based on the $Ring - LWE$ problem. If an adversary impersonates an IoT node IN_i, then he/she has to solve $Ring - LWE$ problem. It is impossible to solve in polynomial time. Similarly, if the adversary tries to eavesdrop on transferred messages to CS,

he/she has to reply IN_i. But, only CS knows the unique identity number and it can only compute committed values using IN_i node's secret values a, r_i, u. Therefore, the impersonation or eavesdropping attack fails at the first level of authentication in the proposed protocol.

4. Mutual Authentication: The cloud server CS and the IoT Node IN_i verify each other's authenticity. The Identity providerIDP verifies its identity before giving access to data at the server. If the signature value isn't verified, then it denies access to the communication. Therefore, mutual authentication is achieved.

5. Node Privacy: The IoT node replies to CS by computing committed values, if IN_i computes the same values for each session, it's easy to trace the node. In the proposed scheme, each value is random and independently generated from a pseudo-random generator and one-way hash functions. The IoT node IN_i information is not included to computed committed values to avoid tracing. Therefore, an adversary cannot identify whether the transferred messages are from the same tag or not. Therefore, the proposed protocol achieves the Node privacy property.

6. Scalability: The CS generates u_i for each IoT node N_i which has a unique identity in the proposed scheme that satisfies $u_i = a.r_i(\mod q)$. The CS stores IN_i identity and u_i.

7. Offline Dictionary attack: Assume the adversary obtains all data stored on the IoT Node device such as a,u_i. The adversary has to construct a u_i, v_i to get access; to do so, the adversary has to guess the random secret even if the adversary does not know the unique identity of IN_i. Without the IoT node identity number issued by IDP, it is impossible to verify the correctness. Therefore, offline dictionary attacks on the proposed protocol are not practically possible.

5.5 Performance Evaluation

In this section, the proposed Ring-LWE-based homomorphic scheme is evaluated for its computational costs, and performance is compared with traditional protocols. We examine the computation overhead of the participating IoT user and cloud server of the proposed protocol. We adopted the design of the protocol to evaluate the computation cost by[105]. We considered a similar implementation setup to compare the performance analysis of the proposed protocol. The comparative performance is tabulated as in the table: 5.1.

Table 5.1: Comparison of time and space complexities

Specification	RSA-HE	Elgamal-HE	ECC-HE	Proposed-HE
TC$_{enc}$	$\mathrm{m}_i.O(n)$	$\mathrm{m}_i.O(n) + m_i.O(1)$	$\mathrm{m}_i.O(1) + m_i.O(1)$	$\mathrm{m}_i.O(1).O(2n)$
TC$_{dec}$	$\mathrm{C}(\mathrm{m}_i).O(n)$	$\mathrm{C}(\mathrm{m}_i).O(n)+ C(m_i).O(1)$	$\mathrm{C}(\mathrm{m}_i).O(1)+ C(m_i).O(1)$	$\mathrm{C}(\mathrm{m}_i).O(1)$
SC$_{enc}$	$\omega_i.O(n)$	$\omega_i.O(n) + \omega_i.O(n)$	2 $\omega_i.O(1)$	$\omega_i.O(1)$
Decryption difficulty	IFP	DLP	ECDLP	Ring-LWE

The analysis of the scheme's efficiency in terms of computation cost and communication overhead is conducted on message encryption and decryption between the IoT user and the cloud server. The time complexity of the traditional homomorphic encryption schemes designed for the IoT Cloud environment is compared. Especially, we considered RSA with Triple-DES, RSA with AES, and ECC with AES encryption schemes for comparison with the proposed scheme, it is shown in table ??.

We considered the implementation setup of [105] for analysis of the performance. According to that, the proposed scheme protocol showed improved speeds of 30 ms and 6.1 ms compared to RSA with Triple-DES encryption and RSA with AES encryption respectively. Then, it showed 0.2 ms encryption speed and 0.4 ms are recorded when compared to the ECC-based encryption scheme.

After generating the key pairs, mutual authentication was performed using the unique ID value of the IoT node, denoted as IN_i, and the identity value of the cloud server provided by the IDendentity Provider represented by IDP. This resulted in improved performance, reducing the processing time by 21 ms compared to traditional RSA-based encryption and 2 ms compared to ECC-based encryption. Signature verification was carried out using the respective public key values generated for both the cloud server and the IoT node. The verification process exhibited improved performance, reducing the processing time by 23 ms and 10 ms compared to RSA-based and ECC-based signature verification.

Furthermore, the study conducted a comparative analysis on memory usage in message encryption, considering the performance limitations required by devices operating in an IoT environment. It was noted that in the context of recent system specifications, the space complexity was not initially taken into account due to the sufficient performance of volatile memory. However, the proposed encryption protocol addresses this issue by incorporating the learning with the error problem. This problem introduces an error value during the coding process using other keys, excluding the secret key, thus enhancing security against differential attacks. The proposed encryption scheme is based on the hardness of Ring-LWE, making message decryption more challenging.

5.6 Summary

Traditional homomorphic encryption techniques used in the Internet of Things (IoT) and cloud computing applications face vulnerabilities when subjected to quantum attacks. To address this issue, adopting a protocol with robust resistance against quantum attacks and various other forms of attacks becomes crucial to ensure quantum security across different levels. In the coming years, the protocol will undergo extensive testing to evaluate its processing speed, storage capabilities, and compatibility with different programming languages. This comparative analysis will aid in assessing its suitability for diverse applications. The proposed Ring-LWE-based homomorphic encryption scheme for a cloud-IoT environment establishes the quantum secure communication channel between the user and the cloud server. The proposed scheme considered the flexible network scenario in an IoT-cloud environment with a variety of IoT node devices. The IoT user will register at the cloud server, it generates public key pairs for the encryption and signature verification. Fully holomorphic encryption is used to prevent data leakage or damage and provide privacy from attacks. The proposed scheme is compared with traditional homomorphic encryption schemes for performance analysis and security evaluation. The performance analyzed w.r.t., scheme's time complexity, space complexity, security against attacks, and privacy. The protocol's high resistance to quantum attacks, is useful in a wide variety of applications to ensure quantum security at different levels.

Chapter 6

Post-Quantum Blockchain with Provable Security

This chapter provides an overview of modern blockchain network vulnerabilities to quantum adversaries, as well as some post-quantum mitigation strategies. Then, a post-quantum blockchain is constructed using a modified $Ring - TESLA$ signature algorithm that defends the blockchain network against quantum adversaries. The security of the proposed signature scheme is based on the hardness of $Ring - LWE$,or learning with errors in a polynomial ring. We also propose a randomized consensus, proof-of-stake (POS) to avoid the dominant validator problem and maintain the decentralization property among the nodes. We provide comprehensive security proof and analysis in the presence of a quantum adversary. This chapter provides support for the design and development of future quantum-resistant blockchain applications.

The chapter is organized as follows: Section 6.1 presents the introduction. Section 6.2 presents the contribution and construction overview of the proposed scheme. Section 6.3 presents the proposed modified Ring-TESLA signature algorithm that is used in the construction of the scheme. In Section 6.4, we describe the construction of the post-quantum blockchain along with the randomized consensus technique. Section 6.5 presents the security analysis of the proposed blockchain. We also present formal security proof of the proposed blockchain. finally, Section 6.6 summarizes the chapter.

6.1 Introduction

The fundamental concept of blockchain technology revolves around employing a decentralized and distributed block storage structure, along with point-to-point transmission, to enable users to reach consensus without the need for a central authority. This technology has sparked a significant technological and industrial revolution in the digital realm. In a trustless environment, a consensus algorithm could bring about significant change[106] The first functional blockchain was proposed in the year 2008 by Nakamoto et al.[107], which is the backbone of the Bitcoin cryptocurrency. In recent years, many research and industrial organizations have been attracted to constructing a variety of decentralized applications using blockchain technology. If the blockchain has been applied to the banking industry, cloud computing, and other distributed applications, it is difficult to attack its security mechanism and business model [108]. This means that studies of the security of blockchains need to look at both current and future threats, like quantum attacks.

The blockchain uses hash functions and digital signatures for consensus and transaction authentication. Consensus algorithms ensure all honest nodes in the network maintain a distributed ledger of all valid transactions, preventing double-spending. Bitcoin and other blockchain-based apps use Proof-of-Work (PoW) to establish distributed consensus[109]. The PoW is a decentralized consensus challenge that enables network participants to solve a mathematical function. It is used for transaction validation and token mining in cryptocurrency mining. To minimize the computational power, several modern blockchain applications are attempting to replace Proof-of-Work (PoW) with a different block delegation mechanism called Proof-of-Stake (PoS) [110] and Byzantine Fault Tolerance (BFT) that is based on Byzantine General Problem(BGP) [111]. When compared to PoW-based blockchain networks, it can minimize the cost of adding new blocks.

Hash functions are used in blockchains to generate user addresses (private/public keys) or shorten public addresses. Hash functions are also used to link blocks for transactions that occur at the same time. For example, hash functions like $SHA-256$ or *Scrypt* are used in blockchains because they are easy to check yet hard to forge, allowing users to generate digital signatures to authenticate themselves or their data transactions. The Elliptic curve digital signature algorithm (ECDSA) is used in Bitcoin for transaction authentication in the blockchain. Every Bitcoin address is a cryptographic hash of the ECDSA public key. Any transaction that is sent must be signed, and we can always check the validity of that signature and who has signed it. The current blockchain employs the recommended 192-bit elliptic curve domain parameters, specifically the curve $secp192k1$ proposed by Daniel and R. L. Brown[10]. In light of the recent advancement of quantum computers, the encryption schemes underlying current blockchain networks are based on intractability assumptions for conventional adversaries that might not always hold for quantum adversaries. In particular, blockchain technology relies on ECDSA for transaction authentication, which will be vulnerable to quantum adversaries. Legitimate users will lose all of their assets and privacy if someone uses Shor's algorithms[2] to generate a user's private key from a public key to sign a variety of unauthorized transactions, or if an attacker forges a user's signature. Similarly, Grover's algorithm[4] can be used to search for data and solve hash functions by finding collisions in a hash space of size n, with a complexity of $O(\sqrt{n})$, but classical algorithm complexity is $O(n)$. Quantum computing invalidates blockchain in two ways. First, hash inversion is assumed to be a computationally hard problem. If a quantum computer can simplify this process, then the blockchain's authenticity and legitimacy are threatened. The Grover's search algorithm[4] gets the pre-image of a function value significantly more quickly than the conventional brute-force search. From the given input, it generates output and compares it with other outputs to isolate the input. Second, a quantum computer could compromise any component of a blockchain implementation that is dependent on private or public key cryptography, such as data communication or signature algorithm.

From considering attacks on the blockchain, designing the post-quantum blockchain is essential. Therefore, leveraging the benefits of the Ring-LWE hard problem, we propose a post-quantum blockchain with a lattice-based signature scheme with provable security. The proposed blockchain builds upon the aforementioned Ring-LWE hard problems; it replaces ECC-based operations with lattice-based constructions. We propose attack countermeasures to improve the security of lattice-based signature schemes. As a result, we modified the Ring-TESLA signature algorithm to create the post-quantum blockchain. Therefore, it ensures communications remain secure and highly efficient even if large quantum computers become a reality. In this scheme, we also examine the vulnerability and resistance of the currently most efficient lattice-based signature schemes, such as the GLP scheme (CHES 2012)[11], BLISS (CRYPTO 2013)[12], and the ring-TESLA (AfricaCrypt 2016)[13], as well as their implementations. We examine a number of attacks, including randomization attacks and zeroing attacks.

6.2 Contribution and Construction Overview

In this, we proposed a post-quantum blockchain is constructed using a modified $Ring-TESLA$ signature algorithm that defends the blockchain network against quantum adversaries. The security of the proposed signature scheme is based on the hardness of $Ring-LWE$,or learning with errors in a polynomial ring. The advantage of $Ring-LWE$ is the reduced communication overhead and key size by representing the matrix as degree n polynomials in R_q. We also propose a randomized consensus, proof-of-stake (POS) to avoid the dominant validator problem and maintain the decentralization property among the nodes. We provide comprehensive security proof and analysis in the presence of a quantum adversary. This scheme provides support for the design and development of future quantum-resistant blockchain applications. The contributions of the scheme are summarized as follows:

1. We evaluated the Ring-LWE-based lattice signature schemes and their vulnerabilities to quantum attacks and modified the Ring-TESLA signature scheme.

2. We constructed the Post Quantum Blockchain (PQB) architecture. It uses the

modified Ring-TESLA signature algorithm with blockchain to provide a secure cryptocurrency scheme that can resist quantum attacks.

3. We proposed a randomized consensus, Proof-of-Stake (PoS) to minimize the advantage of a quantum adversary.

4. We analyzed the proposed post-quantum blockchain (PQB) and demonstrated its correctness and security in the presence of an adversary under the assumption of the Ring-LWE hard problem.

6.2.1 Quantum Vulnerabilities

A quantum computer can solve classical mathematical hard problems over a finite abelian group with super-polynomial speed. The RSA algorithm is based on the finite abelian group Z_n, and the $ECDSA$, used in Bitcoin, is based on an elliptic curve-based finite abelian group structure. A quantum computer can solve classical mathematical hard problems over a finite abelian group with super-polynomial speed. Furthermore, using the quantum Fourier transform, Shor's quantum algorithm [3] can solve Integer Factorization (IF) and Discrete Logarithm Problems (DLP) at an exponential speedup. As a result, the digital signature algorithms used in most existing blockchain networks will be broken and adversaries will get access to users' sensitive information.

The solution for the searching problem is the foundation for PoW systems. Grover's search algorithm[4], when combined with a quantum computer, can yield a quadratic speedup for all searching problems. It is significantly faster than a classical brute force search in $O(n)$ time because it can compute the pre-image of a function value within $O(\sqrt{n})$ time. There are two possibilities to attack blockchain-enabled systems using Grover's search approach. First, it may search for hash collisions that are subsequently used to replace blocks without affecting the blockchain's integrity. Second, it can reduce nonce production time in mining, making chain reconstruction faster from a altered block onwards, potentially opening the door to a regenerative chain attack. As a result, it may easily change the history of transaction records and control the generation of new blocks by mining faster. Therefore, these vulnerabilities should be given more attention, and appropriate countermeasures should be proposed as soon as possible.

In the case of signature algorithms, the current blockchain structure uses ECDSA [112] [113] on $secp256k1$ curve. It starts with a randomly generated private key k, and multiplies with a randomly generated point G on the curve to get the associated public key $K = k \times G$. This computation is a one-way function, it can be computed in one direction; we can compute K from known k, but the reverse is hard. The user address in the blockchain can be obtained from the seed value multiplied by private key k. The public key$K$ is distinct computed from distinct k. The address is a hash value generated from $SHA256$, $RIPEMD160$, and a series of code hashing algorithms. The K is shared with the user who wants to perform a transaction. The advantage in this scheme is that you can use the seed value to generate multiple addresses, but the wallet contains only the seed value. The wallet can be consistent with the property of the deterministic wallet. However, the blockchain with ECDSA will be vulnerable to quantum attacks in the future. An adversary can use Shor's algorithm [2] to find a private key derived from an elliptic curve public key, by which he can sign unauthorized transactions or forge a user's valid signatures. The current blockchain vulnerabilities are summarized in the table 6.1:

6.2.2 Quantum Resistant Solutions

Many researchers have worked hard in recent years to develop defenses against quantum attacks. Overall, there are certain visions that appear to be quite promising in addressing these challenges, as follows:

- Post Quantum Cryptography(PQC): It is more practical for the existing blockchain networks to develop quantum-resistant or post-quantum cryptographic techniques, such as lattice-based cryptography.

- Post Quantum Blockchain (PQB): It's the informational vision system, that combines a traditional blockchain with post-quantum cryptography or a tradi-

Table 6.1: Overview of various blockchain vulnerabilities

Blockchain	Attack	Effect	Vulnerability
Bitcoin [107]	Signature	High	Duplicate transaction, Computes private key using public key and Shor's algorithm .
Ethereum[114]	Signature	High	Forge Signature using public key and Shor's algorithm
LiteCoin[115]	Signature	High	Duplicate transaction, Computes private key using public key and Shor's algorithm
BitcoinGold[116]	Signature	High	Duplicate transaction, Computes private key using public key and Shor's algorithm
BitcoinCore[117]	Signature	High	Duplicate transaction, Computes private key using public key and Shor's algorithm
BitcoinCash[118]	Signature	High	Duplicate transaction, Computes private key using public key and Shor's algorithm
Menero[119]	Consensus	Moderate	Attack on EdDSA to remove User and transaction anonymity,solving RondomX consensus using Grover's algorithm
BEAM[120]	Consensus	Moderate	Intercept transactions and removing anonymity using Grover's algorithm
Grin[121]	Consensus	Moderate	Intercept transactions and removing anonymity using Grover's algorithm
ZCash[122]	Signature, Consensus	Very High	Gains private key and generates tokens by attacking $Zk - SNARK$ zero-knowledge protocol

tional blockchain storage structure with quantum communication.

- Quantum Hashing(QH): Quantum hashing, which uses the same intermediate hash values as binary hashing, is expected to be more resilient against various distortions. [123].

6.3 Modified Ring-TESLA Algorithm

As per the attack analysis of the lattice signature algorithms, the $Ring - TESLA$ algorithm is secure against randomization and zeroing attacks, except for the zeroing of the randomness and zeroing of hash polynomial attacks. To avoid these attacks, the randomness needs to be increased by including more random parameters and existing random parameters such as secret s and error e polynomial to be protected. In the existing $Ring - TESLA$ scheme, the s, e are filtered for their coefficients and should not cross the constant upper-bound limit of L. In the modified $Ring - TESLA$ signature algorithm, the filtering functions $checkE()$ and $checkS()$ are removed to hide the coefficient upper limit and to increase the speed of the key generation. Instead of sampling random polynomials $(a_1, a_2 \leftarrow R_q)$, a pseudo-random generator $PRG(seed_a)$ with input seed $seed_a$ is used for every signature to increase the randomness. For each distinct fresh pair (a_1, a_2), the attacker needs to challenge a distinct lattice to attack the signature. The modified signature algorithm is defined with the following parameters: $(n, q, d, \omega, k, U, B, L)$ are integers.n is power of two and q is prime,$q = 1(\mod 2n)$. The quotient ring of polynomials $R_q = Z_q[x]/(x^n + 1)$ is defined with $degree \leq (n - 1)$ and coefficient in $(-\frac{q}{2}, \frac{q}{2})$. The gaussian distribution χ_σ is defined with std. deviation σ. The F,encoding function that maps vector length n and weight ω to output of the Hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$. The pseudo-random generator $PRG(seed_a)$ with input seed $seed_a$.

Algorithm 6.1 Gen($1^\lambda; a_1, a_2$)

```
Require: $n, \sigma, q, \omega$ and $a_1, a_2$: two ring elements from $PRG(seed_a)$
Ensure: key pairs: $(pk, sk)$
  $s, e_1, e_2 \leftarrow \chi_\sigma^n$
  $t_1 = a_1 * s + e_1(\mod q); t_2 = a_2 * s + e_2(\mod q)$
  $sk \leftarrow (s, e_1, e_2), pk \leftarrow (t_1, t_2, seed_a)$
  return $(pk, sk)$
```

Algorithm 6.2 Sign($\mu; s, e_1, e_2, t_1, t_2, seed_a$)

```
Require: $n, \sigma, q, d, B, U, k, \omega$, and $H : \{0,1\}^* \to \{0,1\}^k$, message $\mu \in \{0,1\}^*$
    $PRG(seed_a) : \{0,1\}^* \to R_q^2$, Mapping Function $F : \{0,1\}^k \to B_\omega^n$
Ensure: Sign:$(z, c)$
$(a_1, a_2) \leftarrow PRG(seed_a)$
$y \overset{\$}{\leftarrow} [-B, B]^n$
$v_1 \leftarrow a_1 * y(\mod q); v_2 \leftarrow a_2 * y(\mod q)$
$c' \leftarrow H(\lfloor v_1 \rceil_{d,q}, \lfloor v_2 \rceil_{d,q}, \mu)$
$c \leftarrow F(c')$
$z \leftarrow y + s * c$
$w_1 \leftarrow v_1 - e_1 * c(\mod q), w_2 \leftarrow v_2 - e_2 * c(\mod q)$
if $|[w_1]_{2^d}|, |[w_2]_{2^d}| \notin R_{2^d}$ or $z \notin R_{B-U}$ then
    Restart
    return $(z, c')$
```

Algorithm 6.3 Verify($\mu; (z, c'), t_1, t_2, seed_a$)

```
Require: $n, \sigma, q, d, k, \omega$, and $H : \{0,1\}^* \to \{0,1\}^k$, message $\mu \in \{0,1\}^*$, $PRG(seed_a)$ :
    $\{0,1\}^* \to R_q^2$, Mapping Function $F : \{0,1\}^k \to B_\omega^n$
Ensure: Accept, Reject: $(1, 0)$
$(a_1', a_2') \leftarrow PRG(seed_a)$
$c \leftarrow F(c')$
$w_1' \leftarrow a_1' * z - t_1 * c(\mod q)$
$w_2' \leftarrow a_2' * z - t_2 * c(\mod q)$
$c'' \leftarrow H(\lfloor w_1' \rceil_{d,q}, [w_2']_{d,q}, \mu)$
if $c' = c'' \wedge z \in R_{B-U}$ then
    return 1 else 0
```

For polynomial multiplication in the quotient ring $R_q = Z_q[x]/(x^n + 1)$, the Number Theoretic Transform (NTT) [124] is used for efficient encryption time in lattice-based cryptography. The time complexity for the polynomial multiplication using

NTT is quasi-linear $O(nlogn)$. Therefore, input parameters should be selected in such a way that NTT is applicable. The n chosen as power of two and $q = 1($ mod $2^n)$. For the inverse NTT operations, Barrett reduction is preferred over coefficient reduction because of the modular design. The sparse and hybrid multiplications require more NTT operations since the sparse multiplication can be applied only to the integer domain.

6.3.1 Randomized Consensus

The well-known consensus algorithm used in the blockchain is Proof of Work (POW) proposed by Nakamoto et.al. [107]based on the *hashcash* problem. In this, the miner has to prove his resources were spent on computation to add a new block to the network. There are similar consensus algorithms that have been found in the literature, such as memory-intensive PoW that don't allow its acceleration using application-specific integrated circuits(ASIC). The PoW named Momentum proposed by Larimer et.al. [125], is based on the detection of hash collision problems. another PoW Cuckoo Cycle proposed by Tromp et.al. [126], based on a constant-sized subgraph finding problem in a given random graph, and PoW called Equihash proposed by biryukov et.al. [127], based on the extended birthday problem. These schemes all use the same template and are based on *hashcash* model proof-of-work. Let $H_c : \{0,1\}^* \rightarrow \{0,1\}^n$ be a cryptographic hash function and The block header hash:$H_b = h_c(header)$, The PoW problem that finds the random nonce r on predicate P can be defined as:

$$P(H_b, r) \text{ and } H_c(H_b||r) \leq k$$

The Proof-of-work in this form allows to increase the difficulty level by changing the target parameter k.

Bitcoin and its underlying cryptographic schemes are vulnerable to possible quantum attacks since PoW mainly depends on hashing. With the help of Grover's search algorithm [4], quantum miners will get more hashing advantage and dominate the network by 51% attack. However, The proposed post-quantum blockchain is designed with a randomized consensus algorithm that does not include hashing.

Randomized Proof-of-Stake: The Proof-of-Stake(PoS) is the variant of the consensus algorithm that depends on the stake of the validator (miner) in the public blockchain. Proof-of-stake(PoS) is energy-efficient and provides fast transaction processing and resistance to quantum miners since it doesn't contain the hashing operation. The quantum miner can not get any additional advantage over the network nodes even if he uses Grover's search algorithm [4]. In PoS, the membership and selection of validators are based on the minimum amount they have in their accounts. So, the disadvantage of the dominant validator problem exists in a blockchain network. Whoever stakes more currency in their account will dominate the network. To solve this problem, we randomized PoS process by randomly assigning tokens like the lottery method, which provides an equal probability for each node to become the winner node in the network. In a PoS-based blockchain network, every node consists of some amount of currency as a stake and the chance of winning a node is proportional to its total stake in the network. The selection of the winning validator node begins from root node 1 to node n. Let p be the pointer to the current node 1, and the stake of the current node is stored in temporary variable t. At this point a random value r is generated between 0 to $Total_{sv}$.(where $Total_{sv} = \sum_{i=1}^{i=n} stakevalue_i$). If the random value is less than the value in the temporary variable ($r < t$), then the corresponding node to the t will be declared as a winner.

Example: Let us assume the blockchain consists of 5 nodes with stake values 15,10,20,8,25 sequentially. Then, the total stake value will be ($15+10+20+8+25 = 78$). Traditionally, the node that holds more stake will have more chances to become the winning node. In the given example, the node with the highest stake value 25 has a high chance of winning, and the node with the lowest stake value 8 has fewer chances to win, In the proposed approach, a random number $r = 51$ $r(0 < r < 78)$ will be generated. The first node stake value is stored in the temporary variable ($t = 15$). If ($r < t$) then the corresponding node to t is the winner, otherwise the pointer moves to the next node and adds its stake value with the existing t value.

Then, it compares with r until it returns the winner node. In the given example, the node with the lowest stake value 8 will have more chance 67% and the winner node. In the proposed approach, the probability of a particular node will be computed as follows:

$$P(CurrentNode) = \frac{(PreviousNode_{sv} + CurrentNode_{sv})}{Total_{sv}}$$

Let random value $r = 51$, the probability of each node with stake values 15,10,20,8,25 are computed as:

$$P(\text{Node holds 15 stake value}) = (0 + 15)/78 = 0.19$$

$$P(\text{Node holds 10 stake value}) = (15 + 10)/78 = 0.32$$

$$P(\text{Node holds 20 stake value}) = (15 + 10 + 20)/78 = 0.57$$

$$P(\text{Node holds 8 stake value}) = (15 + 10 + 20 + 8)/78 = 0.67$$

$$P(\text{Node holds 25 stake value}) = (15 + 10 + 20 + 8 + 25)/78 = 1$$

At the node with 8 stake value, the temporary variable holds ($t = 53$) since (15+10+20+8 = 53). The condition ($r < t$) will be met, therefore node that holds an 8 stake value will be the winner.

6.4 Post Quantum Blockchain Network Model

This section describes the Post-Quantum blockchain network model using the lattice digital signature scheme. The scenario is similar to a Bitcoin peer-to-peer network formed by a chain of blocks. Blocks are organized by the Merkle tree: a tamper-evident binary tree. Every block stores the transactions authenticated using the quantum-resistant signature algorithm. In the signature scheme, the public key pk is used for identity and the secret key sk is used to prove identity. The transaction Tx is signed by the sender and sent to the receiver's address. Once the receiver agrees to the transaction, it will be authorized by other nodes and added to the block in the network. The block corresponding to the transaction will be sent to other nodes in the chain. The randomized Poof-of-Stack (PoS) consensus algorithm is used to validate the transaction, and the winner will add the block to the longest chain connected through hash pointers in the network. Finally, the distributed chain network is updated with the new block, and it is considered a completed transaction.

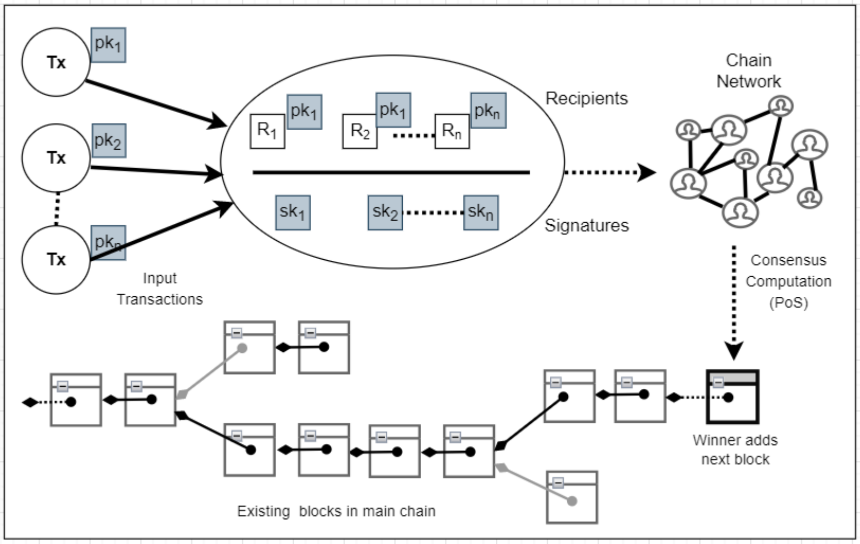


Figure 6.1: Blockchain transaction model

6.4.1 Blockchain Construction

The post-quantum primitive Ring-LWE[56] is the base for the construction of the blockchain. The concept is to use a modified Ring-TESLA signature algorithm in place of ECDSA for blockchain creation. To avoid the quantum miner advantage with Grover's algorithm[4] on Proof-of-Work(PoW) consensus, the Proof-of-Stake(PoS) algorithm is used by removing the dominant validator problem with random parameters. From a set of parameters defined in the algorithm in section 6.3, Post Quantum Blockchain(PQB) can be constructed by following steps:

1. $PQBSetup(1^\lambda)$: The setup function outputs set of parameters $n, q, d, \omega, k, \sigma$ for given a security parameter λ. The recommended standard deviation $\sigma = 2.828$ as in braker et.al scheme [85].

2. $PQBKeyGen(1^\lambda; a_1, a_2)$: This function returns the secret and public key pairs(sk, pk) from the given input parameters. One secret vector and two error vectors

are generated from distribution function $s, e_1, e_2 \xleftarrow{\$} \chi_\sigma^n$ for secret key $sk \leftarrow (s, e_1, e_2)$. Two random elements are generated from $PRG(seed_a)$ for given input seed value to compute the public key $pk \leftarrow (t_1, t_2, seed_a)$ where $t_1 = a_1 * s + e_1 (\mod q); t_2 = a_2 * s + e_2 (\mod q)$.

3. $PQBGenesisSetup()$: The first block of the chain is called genesis or block 0. As there is no preceding hash value in this block, we use all 0s of length n i.e $\{0\}^n$ as the previous hash.

$$random_i \xleftarrow{\$} \{0,1\}^n; nonce \xleftarrow{\$} \{0,1\}^n;$$
$$time\_stamp \leftarrow currenttime;$$

4. $PQBGenesisMerkle()$: After the genesis block setup, we build a data structure for a chain of blocks. The Merkle hash tree is used to organize blocks with random numbers $random_i$, $time\_stamp$, $H()$ - hash function, Lattice signature $Sign()$. For user i, We define identity $i = ID_i, Username_i$ and public key pk_i. For each i the pk_i defined as:

$$pk_i = random_i || H(i) || time\_stamp || Sign(i)$$

The Merkle tree and its subtrees will be constructed using the pk_i for the user node i as:

$$H_{\frac{i-1}{2},..,j} = \begin{cases} H_{\frac{i-1}{2},...,1} = H(pk_i) \text{ if } i = even \\ H_{\frac{i-1}{2},...,0} = H(pk_i) \text{if } i = odd \end{cases}$$

After the formation of the left and right sub-tree leaf nodes, the root node hash value H_{root} will be computed.If the depth of the Merkle tree is k then, number of nodes in the tree $0 \leq i \leq 2^k$

5. $PQBGenesisConstruct()$: This function crates block 0. PQBChain requires a hash value from the genesis block to create a prior hash of block 1. The previous

hash value will be computed as follows:

$$H_{Block0} = H(\{0\})^n||nonce||time\_stamp||H_{root})$$

6. $PQBUserSetup(pk_i, H_{root})$: This function setups the user similar to the block setup process. The following algorithm is configured to set up the user:

$$nonce \xleftarrow{\$} \{0,1\}^n;$$
$$H_{root} : H_{block1} \leftarrow H_{Block0};$$
$$pk_i \in \{0,1\}^n; \text{where}, 0 \leq i \leq 2^k;$$
$$time\_stamp_i \leftarrow \text{current time};$$

7. $PQBUserSign(sk_i, pk_i)$: This function adds user details to the chain along the computed signature. The signature will be computed for its identity using secret key and public key pairs. It returns the signature $Sign(ID_i) = (z_i, c_i')$ for user identity ID_i.

$$(t_{1,i}, t_{2,i}) \leftarrow pk_i;$$
$$z_i, c_i' \leftarrow Sign(ID_i);$$
$$Sign(ID_i, sk_i, t_1, t_2);$$

The output signature is (z_i, c_i'), where z_i will be computed after applying the encoding function. Then, the Merkle hash tree is constructed for block i same as the construction of the genesis block not exceeding to 2^k.

8. $PQBUserVerify(ID_i, pk_i, Sign(ID_i))$: The user verification algorithm verifies the user signature $Sign(ID_i)$ and user public key pk_i.

$$(t_{1,i}, t_{2,i}) \leftarrow pk_i;$$
$$a_{1,i} \leftarrow t_{1,i}; a_{2,i} \leftarrow t_{2,i};$$
$$z_i, c_i' \leftarrow Sign(ID_i);$$
$$Verify(ID_i, z_i, c_i', t_{1,i}, t_{2,i});$$

The constructed blockchain with a set of functions provides security from the quantum adversary as it uses $Ring - LWE$ based signature algorithm. Users can use encryption and decryption functions using $Ring - LWE$ problem to communicate application data.

6.4.2 Merkle Hash Tree

In the Merkle hash tree, the hash of transactions is stored in the leaf node, while the hash of the leaf nodes is stored in the non-leaf nodes in the next level. It continues till it reaches a Merkle root. If a block contains four transactions that are hashed as: $H_{0,0}, H_{0,1}, H_{1,0}$ and $H_{1,1}$ in the leaf node. Then, the non-leaf node contains the hash of left and right hash values as $H_0 = H(H_{0,0}||H_{0,1})$ and $H_1 = H(H_{1,0}||H_{1,1})$. These two hashes are again hashed to compute the merkle root H_{root}. $H_{root} = H(H_0||H_1)$. The structure of the Merkle tree is shown in the figure:6.2. The number of nodes in the tree depends of the parameter k. The Merkle hash tree contains $2^k - 1$ nodes. As a result, $2^k - 1$ hash operations are required to generate the H_{root} hash value. In the typical case of each block, the complexity of searching pk_i is $O(log_2(n))$.

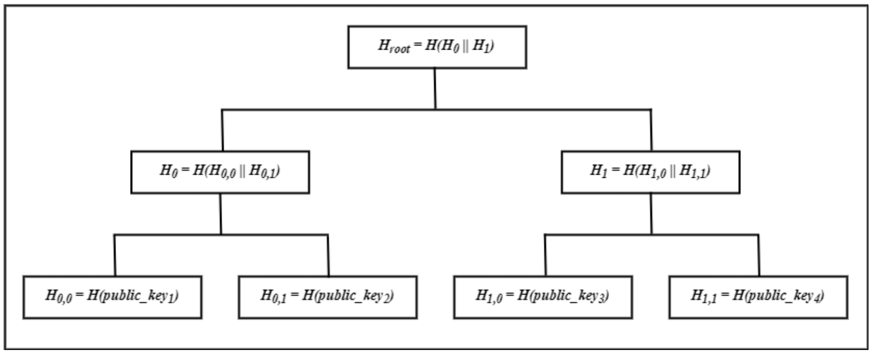

$H_{root} = H(H_0 || H_1)$
$H_0 = H(H_{0,0} || H_{0,1})$
$H_1 = H(H_{1,0} || H_{1,1})$
$H_{0,0} = H(public\_key_1)$
$H_{0,1} = H(public\_key_2)$
$H_{1,0} = H(public\_key_3)$
$H_{1,1} = H(public\_key_4)$

Figure 6.2: Merkle hash tree structure

6.5 Security Analysis

In this section, the proposed post-quantum blockchain(PQB) construction is analyzed for its security. The Ring-LWE-based signature algorithm is checked for its resistance to quantum attackers. The signature algorithm also formally proved for security with an unforgeability experiment in the presence of an attacker.

6.5.1 Blockchain Security Requirements

There are some security requirements to be fulfilled by any blockchain design and implementation. Garay et.al [128] investigated the backbone protocol of Bitcoin and formalized the PoW consensus protocol's security requirements, such as chain quality and common prefix. later, Kiayias et al. [129] demonstrate the Ouroboros PoS methodology and defined security qualities such as $liveliness$ and $safety$. These security requirements are defined as follows:

1. chain quality: Chain quality refers to the number of blocks created by genuine users in the chain. If there are n total number of nodes in the chain and k number of malicious nodes, then the number of malicious players is bounded by $(\frac{k}{n-k})$.

2. common prefix: The blockchain maintained by honest players will have a substantial common prefix. More specifically, if an honest player prunes t blocks from the end of the chain, the probability that the resulting pruned chain is not a prefix of another honest player's chain. That is if two honest players prune t numbers of blocks from their chain, they will get the same prefix.

3. liveness: It guarantees that all validators will complete consensus on a value. If honest validators attempt to include a particular transaction, then after some time (transaction confirmation time) that is equivalent to u slots, the transaction is said to be stable if and only if, for each query on a node, received a response honestly.

4. safety: It determines transaction stability in the chain. A transaction is considered as *stable* if it is k blocks deeper in the chain. k is the security parameter. Once a system node declares a particular transaction t_i as stable, It will report

t_i at the same position in the ledger if is queried by the remaining nodes. if any transaction conflicts with t_i, it will not report stability.

The above security requirements can be met because the proposed blockchain implements PoS consensus algorithms in the public blockchain platform. The security of the blockchain is proved with an attack model in the presence of a quantum attacker. The Ring-LWE-based signature algorithm is a post-quantum primitive, making it resistant to attack by a quantum adversary.

6.5.1.1 Resistance to Quantum Attacker

The proposed post-quantum blockchain is secure against Grover's search algorithm [4]. The vulnerability of Proof-of-Work with Grover's search algorithm is removed with Proof-of-Stake which doesn't contain hashing. Attacking a lattice-based cryptosystem with n-bit security key dimension with a sphere-sieve to solve SVP needs $2^{0.268n+O(n)}$-bit complexity [130]. Solving $Ring-LWE$ is as difficult as solving the worst-case hardness. Therefore, even with existence of a quantum computer, the attack effort decreases with square root complexity. Shor's algorithm [3], on the other hand, is incapable of attacking our PQB design because encryption and digital signature are not based on IFP or DLP problems. Therefore, the proposed blockchain is resistant to Shor's quantum algorithm. The generic attack with the help of a quantum computer will have attack complexity is $min(O(2^{\frac{n_1}{2}}), 2^{0.268n_2+O(n_2)})$. If the post-quantum blockchain utilizes the $SHA3$, the hash function is secure using the classical computing attack. Therefore, we can assume that the hash function's complexity is $O(2^{n_1})$. The signature attack complexity is $2^{0.298n_2+O(n_2)}$ [130].Then,total attack complexity is $min(O(2^{n_1}), 2^{0.298n_2+O(n_2)}$.

6.5.2 Formal Proof

The correctness and completeness of the proposed lattice digital signature scheme are proved using trapdoor functions with preimage sampling[82]. Trapdoor functions defined set of PPT algorithms $(GenTrap, DomSam, PreSam)$ defined as follows:

1. $GenTrap(1^n)$:For given input parameter n, it returns computing function information and trapdoor information pair (s, k). where, $f_s : Domain_n \to Range_n$, the function maps the Domain and Range based on n and k is trapdoor information of f_s.

2. $DomSam(1^n)$: It returns a sample p from the random distribution in the $Domain_n$. But, $f_s(p)$ distribution is uniform in $Range_n$.

3. $PreSam(k, q)$: For given $f_s(p) = q$ and $q \in Range_n$,it returns the conditional distribution sample $p \leftarrow DomSam(1^n)$.

Definition 6.5.1. Minimum entropy: For all $q \in Range_n$, the minimum entropy of sample $p \leftarrow DomSam(1^n)$ takes atleast $\omega(logn)$ complexity for given function $f_s(p) = q$.

Definition 6.5.2. Distinctness:The probability of any PPT algorithm $A(1^n, s)$ that outputs distinct samples $p, p' \in Domain_n$ that satisfies $f_s(p) = f_s(p')$ is negligible.

We construct the Full Domain Hashing (FDH)[45] signature scheme with trapdoor collision-resistant preimage sampleable functions (PSF). It is based on an assumption of hardness of $Ring-LWE$ problem on chosen parameters. We construct the scheme with set of functions $(GenTrap, DomSam, PreSam)$. These functions are relative to the random oracle model: $H = H_n : \{0, 1\}^* \to Range_n$. Then,the tuples in the signature scheme is defined with a set of functions as follows:

- $Gen(1^n)$: It generates a pair of keys $(s, k) \leftarrow GenTrap(1^n)$. It uses s as the signing key and k as the verification key, where s is the description of the function f_s and k is the trapdoor information.

- $Sign(k, \mu)$: It returns the signature σ_μ from stored (μ, σ_μ). Else it generates signature $\sigma_\mu \leftarrow PreSam(k, H(\mu)$ and returns σ_μ.

- $Ver(s, \mu, \sigma)$; It verifies the signature σ, if it belongs to domain $\sigma \in Domain_n$ and $f_s(\sigma) = H(\mu)$,it accepted, else it is rejected.

Definition 6.5.3. ($Decisional - RLWE_{n,q,\chi}$ assumption. for any PPT algorithm A, There exists $negl$-a negligible function,such that:

$$|\Pr[A^{O_s^n}(1^n) = 1] - \Pr[A^R(1^n) = 1]| = negl(n)$$

Let $n, q > 0$, and (χ_s, χ_e) are two bounded distributions over R. Probability to distinguish (a, b) and (a, u) , where $(b = a * s + e)$ for $a \overset{\$}{\leftarrow} R_q$,the secret $s \overset{\$}{\leftarrow} \chi_s$ and uniformly random $u \overset{\$}{\leftarrow} R_q$.

$$|\Pr[A^{O_s^n}(1^n) = (a, b) = 1] - \Pr[A^R(1^n) = (a, u) = 1]| = negl(n)$$

Statement: The signature scheme constructed based on the hardness of Ring-LWE is strongly unforgeable under chosen message attack(UFCMA).

Proof: Let us assume, An adversary $\mathcal{A}$ succeeds in forging the signature with some probability ϵ. A polynomial-time simulator Π is constructed that computes the shortest vector in the lattice and solves the learning with errors problem in polynomial ring $Ring - LWE$ with negligible probability close to ϵ. The procedure is described in the following steps:

1. Let s is the pubic parameter describes the function f_s, the Π runs adversary $\mathcal{A}$ on s, simulates signature oracle and random oracle H. Assume that adversary $\mathcal{A}$ first queries hash of the message μ before signature query on μ.

2. For every distinct message $\mu \in \{0, 1\}^*$, an adversary $\mathcal{A}$ performs a hash query to H, then Π generates the $\sigma_\mu \leftarrow DomSam(1^*)\ m^*$ and stores (μ, σ_μ) in the local. The Π replies with $f_s(\sigma_\mu) = s.\sigma_\mu$ to the $\mathcal{A}$.

3. If an adversary $\mathcal{A}$ performs signature inquiry on μ, then simulator Π search for corresponding signature (μ, σ_μ) in the local data and returns σ_μ to $\mathcal{A}$.

4. When an adversary $\mathcal{A}$ provides a fake signature (μ^*, σ^*), the simulator Π finds $(\mu^*, \sigma_\mu*)$ in the local storage and outputs $\sigma^* - \sigma_{\mu*}$ as a solution to the $Ring - LWE$ hard problem..

Let us look into the reduction method described above. From the perspective of an adversary $\mathcal{A}$, the simulator Π works as a genuine system, and the real chosen message attack is identical. The properties of trapdoor functions ensure this point:

- For every distinct hash query, the Π outputs $H(\mu) = f_s(\sigma_\mu)$, where σ_μ is the signature is generated uniformly random from domain $\sigma_\mu \leftarrow DomSam$. This output is distribution is similar to real output from the genuine system.

- After computing $H(\mu)$, the simulator Π returns a message μ for every concerning signature query on σ_μ. And given the constraint $f_s(\sigma_\mu) = H(\mu)$, where $\sigma_\mu \leftarrow DomSam$.

The output for the signature query on a message μ and the signature sampled from the pre-image trapdoor function has the same distribution. As a result, the simulator Π precisely simulates the genuine signature scheme. An adversary $\mathcal{A}$ generates the fake signature for the message (μ^*, σ^*) with probability ϵ. We have $\sigma^* \in Domain_n$ that satisfies $f_s(\sigma^*) = H(\mu^*) = f_s(\sigma_{\mu*})$ because message μ^* is the locally stored and the signature σ^* is a valid signature. In this case, we consider the solution of the $(\sigma_{\mu*} - \sigma^*)$ as a solution to the $Ring - LWE$ hard problem. To show that $\sigma_{\mu^*} \neq \sigma^*$, there are two cases to be considered:

- case 1: If an adversary $\mathcal{A}$ queries the signature on message μ^* and gets the output σ_{μ^*} as an answer. Since the σ^*, μ^* is forged signature, both can not be equal $\sigma_{\mu^*} \neq \sigma^*$.

- case 2: If an adversary $\mathcal{A}$ queries only a hash query for the message μ^*, the simulator Π stores (μ^*, σ_{μ^*}) in the local because signature $\sigma_{\mu^*} \leftarrow DomSam$ during hash query and it returns $f_s(\sigma_\mu)$ to the adversary $\mathcal{A}$. According to the min-entropy pre-image, the min-entropy(σ_{μ^*}) is the smallest value and hard to find for given the condition of $f_s(\sigma_{\mu^*}) = H(\mu^*)$. In view of adversary $\mathcal{A}$,hash query and σ_{μ^*} are independent of each other, we determine min-entropy of σ_{μ^*} is $\omega(logn)$.

From the above cases, the maximum probability of $(\sigma_{\mu^*} = \sigma^*)$ is $2^{-\omega(logn)}$. Therefore, the simulator Π that solves the $Ring - LWE$ hard problem has a probability close to $\epsilon(n)$.

We can also present with tuples of a signature scheme by conducting an analogous experiment to $unforgeabilityunderchosen - messageattack(ufcma)$, which involves an attacker $\mathcal{A}$ against a signature scheme Π. We repeat a matching $ufcma$ experiment that provides $\mathcal{A}$ access to a random oracle H because we verify the security of the $Ring - LWE$ scheme in the random oracle model. We claim that a signature scheme Π is (t, q_s, q_h, ϵ) -$unforgeableunderchosen - messageattack$ if an adversary $\mathcal{A}$ runs t time to post at most q_s queries to the signature oracle and q_h queries for the random oracle, then adversary $\mathcal{A}$ has the following advantage:

$$Adv_{\Pi}^{ufcma}(\mathcal{A}) = Pr[Expt_{\Pi,\mathcal{A}}^{ufcma} = 1] \leq \epsilon$$

Algorithm 6.4 Unforgeability Experiment $Expt_{\Pi,A}^{ufcma}(1^{\lambda})$:

```
Require: $KeyGen(1^{\lambda}); Sign(sk, \mu); Verify(pk, \mu, \sigma)$
Ensure: $\Pi \leftarrow \mathcal{A}$
(sk, pk) ← KeyGen(1^λ)  →  $(sk, pk) \leftarrow KeyGen(1^{\lambda})$
$(\mu^*, \sigma^*) \leftarrow A(1^{\lambda}, pk)^{\mathcal{O}_{sign}(.),H(.)}$
If $Verify(pk, \mu^*, \sigma^*) = 1 \wedge \mu^* \notin \mathcal{Q}_S$;
Retrun 1
else:Return 0
if  then$\mathcal{A}$ Queries $\mathcal{O}_{Sign}(\mu)$ : $\mathcal{Q}_S \leftarrow \mathcal{Q}_S \cup \{\mu\}$
    $\sigma \leftarrow Sign(sk, \mu)$
    Return $\sigma$ to $\mathcal{A}$
```

6.5.3 Blockchain and IoT integration

The integeration of Blockchain and Internet of Things address many open issues of IoT architecture. Especifically, it enhances many aspects such as: Providing secure communication in IoT networks, node-to-node architectures, Data privacy in transmission and exchange and Compatability with new devices.

The Blockchain changes the execution environment of IoT architectures of modern applications.It also enables the interconnection of IoT nodes. The one of the main advantage is ,only legal nodes that registered on Blockchain will be allowed by avoiding impersonation attack. Moreover, the blockchain enable Iot network supports for multiple devices and easy management without additonal efforts. The open issues to focus when we integerate Blockchain and IoT are: Transaction rate, Energy Optimization, Processing capacity,Lack of skills and regulation.

Parameter	IoT	Blockchain+IoT
Network Model	Centralized	Distributed/De-centralized
Security level	Low	High
Compatibility	Low	High
Node Privacy	High	High
Node Identity	Non-transferable	Transferable

Table 6.2: IoT vs. Blockchain enabled IoT

6.6 Summary

In our description, we have outlined a high-performance, high-security digital signature scheme known as Ring-TESLA within a blockchain structure. The security of this scheme is substantiated under the assumption of Ring-LWE. To enhance the blockchain's efficiency, we have implemented the Proof-of-Stake (PoS) consensus algorithm, effectively addressing the dominant validator problem. This blockchain structure is suitable for quantum-resistant blockchain-based applications. The proposed blockchain with specific parameter sets suitable for both pre-quantum and post-quantum applications that outperform alternative conventional or other signature schemes. We also suggested NTT operations in polynomial multiplication and addition for efficient computations. We additionally modified the $Ring-TESLA$ signature algorithm with the $PRG(seed_a)$ to generate random values for each signature generation. Finally, the security of the proposed blockchain is proved in the presence of a quantum attacker.

Chapter 7

Conclusion and Future Directions

7.1 The Major Contributions of the Thesis

This thesis presented the construction of Ring-LWE-based post-quantum protocol models for the authentication of devices in IoT network environments.In Chapter 2, we described about lattices, some cryptographic primitives built using lattices/ideal lattices, and also described some authentication schemes along with the analysis of those schemes.

In Chapter 3, we proposed a lattice-based authentication and key exchange protocol for the Internet of Things Environment. Our scheme security is based on learning with errors problem in a polynomial ring; shortly, we call Ring-LWE. The advantage of Ring-LWE is the reduced communication overhead and key size by representing the matrix as degree n polynomials in R_q. The protocol correctness is proved formally and verified with the standard verification tool AVISPA for authentication. The informal analysis of the protocol demonstrates that it is secure against known attacks on the internet of Things environment. The protocol's performance is analyzed and compared with relevant protocols. It shows that the communication cost is the same as other protocols, and the computation cost is minimal. A detailed comparison of our scheme with the existing schemes is presented in Tables 3.6 and 3.7.

In Chapter 4, we proposed and validated a quantum-secure node-to-node authentication protocol model for the Internet of Things network environment. Our scheme security is based on the Inhomogeneous Short Integer Solution problem; shortly, we call$ISIS$ problem. The protocol model is verified against known attacks in the IoT network. The protocol's performance is analyzed and compared with relevant protocols. It shows that the proposed protocol is unique and quantum-secure. The protocol's performance is analyzed and compared with relevant protocols. A detailed comparison of our scheme with the existing schemes is presented in Table 4.2 and analysis is presented in Table 4.3.

In Chapter 5, we proposed a scheme for the security and privacy of user data in a cloud environment. Various types of homomorphic encryption schemes are studied for data privacy in the cloud. The Ring-LWE-based encryption scheme is presented for privacy protection in the cloud which meets the homomorphic properties. In this scheme, IoT nodes will register at the cloud server, then the server authenticates IoT nodes and accepts the encrypted data to share with other nodes whenever requested. It stores the data on a cloud server with quantum-safe encryption. For the encryption of the data at the cloud server, the Ring-LWE-based fully homomorphic encryption(FHE) is used for quantum-enabled security and privacy. The proposed scheme is analyzed for security and compact in the presence of a quantum attacker. By employing FHE-based data management and verification methods, the overall efficiency and security surpass those offered by traditional encryption algorithms. Additionally, the implementation of signature verification minimizes the associated overhead, thus enhancing the efficiency of the verification process when compared to existing methods. A detailed comparison of our scheme with the existing schemes is presented in Table 5.1.

In Chapter 6, we proposed the construction of a post-quantum blockchain using a modified $Ring-TESLA$ signature algorithm that defends the blockchain network against quantum adversaries. The security of the proposed signature scheme is based on the hardness of $Ring-LWE$, or learning with errors in a polynomial ring. We also propose a randomized consensus, proof-of-stake (PoS) to avoid the dominant validator problem and maintain the decentralization property among the nodes. We provide comprehensive security proof and analysis in the presence of a quantum adversary.

We proposed three Post-Quantum authentication schemes for IoT environment scenarios and one scheme for the construction of a post-quantum blockchain for transaction authentication using the Modified Ring-TESLA algorithm in place of ECDSA. These contributions are mainly focused on providing security from quantum attackers and for future cryptography applications, especially in the field of Internet of Things and Blockchain Technology.

7.2 Future Directions

Post-quantum cryptosystems have various applications. Some of the future directions of proposed schemes in the thesis include:

- Most commonly used control unit in IoT consists of a Microcontroller Unit (MCU) or acustom chip. A microcontroller is an integrated chip or core in a VLSI or SoC. Popularmicrocontrollers are ATmega 328, ATMega 32u4, ARM Cortex and ARM LPC. Arduino uses ATmega 328 or ATmega 32u4.Raspberry Pi uses ARM Cortex and ARM LPC microcontroller-based boards. The Cortex-A72 (ARM v8) 64-bit offers more computational power and memory capacity than Arduino making it well-suited for more complex future IoT applications that require higher processing capabilities, such as real-time data analysis. Arduino devices are optimized for tasks which is typically sufficient for the basic data collection and communication. However, the implementation of proposed protocol (Contribution-1) on various types of IoT devices and comparative analysis is the interesting future direction. This helps to build real-time, future IoT architectures with energy optimazation. Similarly, the practical implementation of the protocol model for N2N authentication (Contribution-2) helps to test the compatibility of various IoT devices.

- Constructing end-to-end post quantum communication by combining proposed protocols is one of the future direction for state-of art scheme. Some of the current sensitive applications can be changed to quantum-safe applications which provides node to cloud level security.

- The implementation of the proposed post-quantum blockchain model can be taken as future work with personalized blockchain platforms. The Comparison of the Ring-TESLA algorithm with other new signature algorithms like CRYSTALS-Dilithium and FALCON signatures made it through to the final round of the NIST competition.

- The Blockchain is the powerful distributed ledger that provides tranparent and immutabilitle transactions. But, in recent times some of the applications are proved vulnerable to attacks. Adaption of quantum-secure blockchain network for these applications is interesting future direction. Another intresting future work is, integeration of quantum-safe blockchain and Internet of Things which address many open issues of IoT architecture. Especially, it enhances many aspects such as: providing secure communication in IoT networks, node-to-node architectures, Data privacy in transmission and exchange and Compatability with new devices.

Bibliography

[1] Oded Regev. Leture 1: Introduction of lattices in computer science, 2014. Lecture Notes, Tel Aviv University, 2014.

[2] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science, pages 124–134. Ieee, 1994.

[3] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2):303–332, 1999.

[4] Lov K Grover. Quantum mechanics helps in searching for a needle in a haystack. Physical review letters, 79(2):325, 1997.

[5] Miklós Ajtai. Generating hard instances of lattice problems. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 99–108, 1996.

[6] Chi Cheng, Rongxing Lu, Albrecht Petzoldt, and Tsuyoshi Takagi. Securing the internet of things in a quantum world. IEEE Communications Magazine, 55(2):116–120, 2017.

[7] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In International workshop on public key cryptography, pages 162–179. Springer, 2008.

[8] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 1–23. Springer, 2010.

[9] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange—a new hope. In 25th {USENIX} Security Symposium ({USENIX} Security 16), pages 327–343, 2016.

[10] Daniel RL Brown. Sec 2: Recommended elliptic curve domain parameters. Standars for Efficient Cryptography, 2010.

[11] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In International Workshop on Cryptographic Hardware and Embedded Systems, pages 530–547. Springer, 2012.

[12] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Annual Cryptology Conference, pages 40–56. Springer, 2013.

[13] Sedat Akleylek, Nina Bindel, Johannes Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. An efficient lattice-based signature scheme with provably secure instantiation. In International Conference on Cryptology in Africa, pages 44–60. Springer, 2016.

[14] Vishal M Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. IEEE Signal Processing Magazine, 33(4):49–61, 2016.

[15] Mike Lindelsee, Olivier Brand, James Dimmick, and Benedicto Dominguez. Token based transaction authentication, January 1 2013. US Patent 8,346,666.

[16] Piyush Bhatnagar and Sridnar Reddy. System, design and process for strong authentication using bidirectional otp and out-of-band multichannel authentication, June 24 2014. US Patent 8,763,097.

[17] James Ashfield, David Shroyer, and Douglas Brown. Location based authentication of mobile device transactions, October 23 2012. US Patent 8,295,898.

[18] Rudolf Maarten Bolle, Sharon Louise Nunes, Sharathchandra Pankanti, Nalini Kanta Ratha, Barton Allen Smith, and Thomas Guthrie Zimmerman. Method for biometric-based authentication in wireless communication for access control, November 16 2004. US Patent 6,819,219.

[19] Majid Mumtaz, Junaid Akram, and Luo Ping. An rsa based authentication system for smart iot environment. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pages 758–765. IEEE, 2019.

[20] Sheetal Kalra and Sandeep K Sood. Secure authentication scheme for iot and cloud servers. Pervasive and Mobile Computing, 24:210–223, 2015.

[21] Saru Kumari, Marimuthu Karuppiah, Ashok Kumar Das, Xiong Li, Fan Wu, and Neeraj Kumar. A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers. The Journal of Supercomputing, 74(12):6428–6453, 2018.

[22] Ding Wang and Ping Wang. Two birds with one stone: Two-factor authentication with security beyond conventional bound. IEEE transactions on dependable and secure computing, 15(4):708–722, 2016.

[23] Junghyun Nam, Kim-Kwang Raymond Choo, Sangchul Han, Moonseong Kim, Juryon Paik, and Dongho Won. Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation. Plos one, 10(4):e0116709, 2015.

[24] Yanjiang Yang, Xuhua Ding, Haibing Lu, Jian Weng, and Jianying Zhou. Self-blindable credential: towards anonymous entity authentication upon resource constrained devices. In Information Security, pages 238–247. Springer, 2015.

[25] Muhamed Turkanović, Boštjan Brumen, and Marko Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. Ad Hoc Networks, 20:96–112, 2014.

[26] Muhammad Khurram Khan, Saru Kumari, and Mridul K Gupta. More efficient key-hash based fingerprint remote authentication scheme using mobile device. Computing, 96(9):793–816, 2014.

[27] Fan Wu, Lili Xu, Saru Kumari, and Xiong Li. A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. Computers & Electrical Engineering, 45:274–285, 2015.

[28] SK Hafizul Islam, Mohammad S Obaidat, and Ruhul Amin. An anonymous and provably secure authentication scheme for mobile user. International Journal of Communication Systems, 29(9):1529–1544, 2016.

[29] Han-Yu Lin. Chaotic map based mobile dynamic id authenticated key agreement scheme. Wireless Personal Communications, 78(2):1487–1494, 2014.

[30] Debiao He, Sherali Zeadally, Neeraj Kumar, and Wei Wu. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE transactions on information forensics and security, 11(9):2052–2064, 2016.

[31] Sravani Challa, Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Alavalapati Goutham Reddy, Eun-Jun Yoon, and Kee-Young Yoo. Secure signature-based authenticated key establishment scheme for future iot applications. IEEE Access, 5:3028–3043, 2017.

[32] Xiaoying Jia, Debiao He, Li Li, and Kim-Kwang Raymond Choo. Signature-based three-factor authenticated key exchange for internet of things applications. Multimedia Tools and Applications, 77(14):18355–18382, 2018.

[33] Chun-Ta Li, Cheng-Chi Lee, Chi-Yao Weng, and Chien-Ming Chen. Towards secure authenticating of cache in the reader for rfid-based iot systems. Peer-to-Peer Networking and Applications, 11(1):198–208, 2018.

[34] Kai Fan, Yuanyuan Gong, Chen Liang, Hui Li, and Yintang Yang. Lightweight and ultralightweight rfid mutual authentication protocol with cache in the reader for iot in 5g. Security and Communication Networks, 9(16):3095–3104, 2016.

[35] Chanchal Khatwani and Swapnoneel Roy. Security analysis of ecc based authentication protocols. In 2015 International conference on computational intelligence and communication networks (CICN), pages 1167–1172. IEEE, 2015.

[36] Priyanka D Harish and Swapnoneel Roy. Energy oriented vulnerability analysis on authentication protocols for cps. In 2014 IEEE International Conference on Distributed Computing in Sensor Systems, pages 367–371. IEEE, 2014.

[37] Dinh Duc Nha Nguyen, Keshav Sood, Yong Xiang, Longxiang Gao, and Lianhua Chi. Impersonation attack detection in iot networks. In GLOBECOM 2022-2022 IEEE Global Communications Conference, pages 6061–6066. IEEE, 2022.

[38] Swapnoneel Roy and Chanchal Khatwani. Cryptanalysis and improvement of ecc based authentication and key exchanging protocols. Cryptography, 1(1):9, 2017.

[39] Swapnoneel Roy, Sanjay P Ahuja, Priyanka D Harish, and S Raghu Talluri. Energy optimization in cryptographic protocols for the cloud. In Applications of Security, Mobile, Analytic, and Cloud (SMAC) Technologies for Effective Information Processing and Management, pages 24–48. IGI Global, 2018.

[40] Cesar Castellon, Swapnoneel Roy, Patrick Kreidl, Ayan Dutta, and Ladislau Bölöni. Energy efficient merkle trees for blockchains. In 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 1093–1099. IEEE, 2021.

[41] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptol. ePrint Arch., 2012:688, 2012.

[42] Jiang Zhang, Zhenfeng Zhang, Jintai Ding, Michael Snook, and Özgür Dagdelen. Authenticated key exchange from ideal lattices. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 719–751. Springer, 2015.

[43] Ahmad Boorghany, Siavash Bayat Sarmadi, and Rasool Jalili. On constrained implementation of lattice-based cryptographic primitives and schemes on smart cards. ACM Transactions on Embedded Computing Systems (TECS), 14(3):1–25, 2015.

[44] Hugo Krawczyk. Hmqv: A high-performance secure diffie-hellman protocol. In Annual International Cryptology Conference, pages 546–566. Springer, 2005.

[45] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security, pages 62–73, 1993.

[46] Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the tls protocol from the ring learning with errors problem. In 2015 IEEE Symposium on Security and Privacy, pages 553–570. IEEE, 2015.

[47] Chris Peikert. Lattice cryptography for the internet. In international workshop on post-quantum cryptography, pages 197–219. Springer, 2014.

[48] Qi Feng, Debiao He, Sherali Zeadally, Neeraj Kumar, and Kaitai Liang. Ideal lattice-based anonymous authentication protocol for mobile devices. IEEE Systems Journal, 13(3):2775–2785, 2018.

[49] Tobias Oder, Thomas Pöppelmann, and Tim Güneysu. Beyond ecdsa and rsa: Lattice-based digital signatures on constrained devices. In 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pages 1–6. IEEE, 2014.

[50] Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. High precision discrete gaussian sampling on fpgas. In International Conference on Selected Areas in Cryptography, pages 383–401. Springer, 2013.

[51] Johannes Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete ziggurat: A time-memory trade-off for sampling from a gaussian distribution over the integers. In International Conference on Selected Areas in Cryptography, pages 402–417. Springer, 2013.

[52] Thomas Pöppelmann, Tobias Oder, and Tim Güneysu. High-performance ideal lattice-based cryptography on 8-bit atxmega microcontrollers. In International Conference on Cryptology and Information Security in Latin America, pages 346–365. Springer, 2015.

[53] Paulo SLM Barreto, Patrick Longa, Michael Naehrig, Jefferson E Ricardini, and Gustavo Zanon. Sharper ring-lwe signatures. IACR Cryptol. ePrint Arch., 2016:1026, 2016.

[54] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In International Conference on the Theory and Application of Cryptology and Information Security, pages 372–389. Springer, 2008.

[55] Keita Xagawa and Keisuke Tanaka. Zero-knowledge protocols for ntru: Application to identification and proof of plaintext knowledge. In International Conference on Provable Security, pages 198–213. Springer, 2009.

[56] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In International Conference on the Theory and Application of Cryptology and Information Security, pages 598–616. Springer, 2009.

[57] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. Improved zero-knowledge identification with lattices. In International Conference on Provable Security, pages 1–17. Springer, 2010.

[58] Rosemberg Silva, C de A Antonio, and Ricardo Dahab. Lwe-based identification schemes. In 2011 IEEE Information Theory Workshop, pages 292–296. IEEE, 2011.

[59] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. Journal of the ACM (JACM), 60(6):1–35, 2013.

[60] Mohammad Sadeq Dousti and Rasool Jalili. An efficient statistical zero-knowledge authentication protocol for smart cards. International Journal of Computer Mathematics, 93(3):453–481, 2016.

[61] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. Foundations of secure computation, 4(11):169–180, 1978.

[62] Lester S Hill. Cryptography in an algebraic alphabet. The American Mathematical Monthly, 36(6):306–312, 1929.

[63] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120–126, 1978.

[64] Goldwasser Shafi and Micali Silvio. Probabilistic encryption. Journal of computer and system sciences, 28(2):270–299, 1984.

[65] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4):469–472, 1985.

[66] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, pages 544–553, 1994.

[67] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18, pages 223–238. Springer, 1999.

[68] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In TCC, volume 3378, pages 325–341. Springer, 2005.

[69] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing, pages 169–178, 2009.

[70] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29, pages 24–43. Springer, 2010.

[71] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings, pages 868–886. Springer, 2012.

[72] Sedat Akleylek and Meryem Soysaldı. A new lattice-based authentication scheme for iot. Journal of Information Security and Applications, 64:103053, 2022.

[73] Akasha Shafiq, Muhammad Faizan Ayub, Khalid Mahmood, Mazhar Sadiq, Saru Kumari, and Chien-Ming Chen. An identity-based anonymous three-party authenticated protocol for iot infrastructure. Journal of Sensors, 2020, 2020.

[74] Daniele Micciancio and Shafi Goldwasser. Complexity of lattice problems: a cryptographic perspective, volume 671. Springer Science & Business Media, 2012.

[75] Peter van Emde Boas. Another np-complete problem and the complexity of computing short vectors in a lattice. Tecnical Report, Department of Mathmatics, University of Amsterdam, 1981.

[76] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In Proceedings of the fifteenth annual ACM symposium on Theory of computing, pages 193–206, 1983.

[77] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science, pages 724–733. IEEE, 1993.

[78] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating cvp to within almost-polynomial factors is np-hard. Combinatorica, 23(2):205–243, 2003.

[79] Dorit Aharonov and Oded Regev. Lattice problems in np$\cap$ conp. Journal of the ACM (JACM), 52(5):749–765, 2005.

[80] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing, 37(1):267–302, 2007.

[81] Oded Regev. New lattice-based cryptographic constructions. Journal of the ACM (JACM), 51(6):899–942, 2004.

[82] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the fortieth annual ACM symposium on Theory of computing, pages 197–206, 2008.

[83] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pages 84–93, 2005.

[84] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM), 56(6):1–40, 2009.

[85] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pages 575–584, 2013.

[86] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography, 75(3):565–599, 2015.

[87] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. computational complexity, 16(4):365–411, 2007.

[88] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Theory of Cryptography Conference, pages 145–166. Springer, 2006.

[89] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 35–54. Springer, 2013.

[90] Jiang Zhang, Zhenfeng Zhang, Jintai Ding, Michael Snook, and Özgür Dagdelen. Authenticated key exchange from ideal lattices. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9057:719–751, 2015.

[91] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing, 37(1):267–302, 2007.

[92] Cesar E Castellon, Swapnoneel Roy, O Patrick Kreidl, Ayan Dutta, and Ladislau Bölöni. Towards an energy-efficient hash-based message authentication code (hmac). In 2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC), pages 1–7. IEEE, 2022.

[93] Cesar E Castellon, Tamim Khatib, Swapnoneel Roy, Ayan Dutta, O Patrick Kreidl, and Ladislau Bölöni. Energy-efficient blockchain-enabled multi-robot coordination for information gathering: Theory and experiments. Electronics, 12(20):4239, 2023.

[94] SU Yu, QU Yugui, and LIN Zhiting. Hmac: An energy efficient mac protocol for wireless sensor networks. Journal of University of Science and Technology of China, 40(10):1054, 2010.

[95] Johannes Buchmann, Florian Göpfert, Tim Güneysu, Tobias Oder, and Thomas Pöppelmann. High-performance and lightweight lattice-based public-key encryption. In IoTPTS 2016 - Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, Co-located with Asia CCS 2016, pages 2–9. Association for Computing Machinery, Inc, may 2016.

[96] A Armando, D Basin, Y Boichut, Y Chevalier, and L Compagna. The AVISPA Tool for the Automated Validation. Computer Aided Verification, 3576:281–285, 2005.

[97] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium, pages 267–288. Springer, 1998.

[98] Chris Peikert et al. A decade of lattice cryptography. Foundations and trends® in theoretical computer science, 10(4):283–424, 2016.

[99] Mimi Ma, Debiao He, Huaqun Wang, Neeraj Kumar, and Kim-Kwang Raymond Choo. An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks. IEEE Internet of Things Journal, 6(5):8065–8075, 2019.

[100] Bahaa Hussein Taher, Sheng Jiang, Ali A Yassin, and Hongwei Lu. Low-overhead remote user authentication protocol for iot based on a fuzzy extractor and feature extraction. IEEE Access, 7:148950–148966, 2019.

[101] Preeti Chandrakar and Hari Om. A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ecc. Computer Communications, 110:26–34, 2017.

[102] Yanrong Lu, Guangquan Xu, Lixiang Li, and Yixian Yang. Anonymous three-factor authenticated key agreement for wireless sensor networks. Wireless Networks, 25(4):1461–1475, 2019.

[103] Jiaqing Mo and Hang Chen. A lightweight secure user authentication and key agreement protocol for wireless sensor networks. Security and Communication Networks, 2019, 2019.

[104] Yanfang Wu, Zheng Huang, Jie Zhang, and Qiaoyan Wen. A lattice-based digital signature from the ring-lwe. In 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, pages 646–651. IEEE, 2012.

[105] Byung-Wook Jin, Jung-Oh Park, and Hyung-Jin Mun. A design of secure communication protocol using rlwe-based homomorphic encryption in iot convergence cloud environment. Wireless Personal Communications, 105:599–618, 2019.

[106] Andreas M Antonopoulos and David A Harding. Mastering bitcoin. " O'Reilly Media, Inc.", 2014.

[107] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, page 21260, 2008.

[108] Umesh Bodkhe, Sudeep Tanwar, Karan Parekh, Pimal Khanpara, Sudhanshu Tyagi, Neeraj Kumar, and Mamoun Alazab. Blockchain for industry 4.0: A comprehensive review. IEEE Access, 8:79764–79800, 2020.

[109] Ben Laurie and Richard Clayton. "Proof-of-Work" Proves Not to Work. (May):1–9, 2004.

[110] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014):1–32, 2014.

[111] L Lamport, R Shostak, and M Pease. The byzantine generals problem acm transactions on progamming languages and syetems, vol. 4 no. 3 pp. 382-401, 1982.

[112] Neal Koblitz. Elliptic curve cryptosystems. Mathematics of computation, 48(177):203–209, 1987.

[113] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). International journal of information security, 1:36–63, 2001.

[114] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. white paper, 3(37):2–1, 2014.

[115] Jeff Reed. Litecoin: An introduction to litecoin cryptocurrency and litecoin mining, 2017.

[116] Anne Haubo Dyhrberg. Bitcoin, gold and the dollar–a garch volatility analysis. Finance Research Letters, 16:85–92, 2016.

[117] Linxiang Cai and Binjun Wang. Research on tracking and tracing bitcoin fund flows. In 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), pages 1495–1499. IEEE, 2018.

[118] Bitcoin Cash. Bitcoin cash. Development, 2, 2019.

[119] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, et al. An empirical analysis of traceability in the monero blockchain. arXiv preprint arXiv:1704.04299, 2017.

[120] BEAM: Confidential DeFi & Crypto — beam.mw. https://beam.mw/. [Accessed 01-01-2024].

[121] Suyash Bagad and Saravanan Vijayakumaran. On the confidentiality of amounts in grin. In 2020 Crypto Valley Conference on Blockchain Technology (CVCBT), pages 78–82. IEEE, 2020.

[122] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. An empirical analysis of anonymity in zcash. In 27th USENIX Security Symposium (USENIX Security 18), pages 463–477, 2018.

[123] Minho Jin and Chang D Yoo. Quantum hashing for multimedia. IEEE transactions on information forensics and security, 4(4):982–994, 2009.

[124] Patrick Longa and Michael Naehrig. Speeding up the number theoretic transform for faster ideal lattice-based cryptography. In International Conference on Cryptology and Network Security, pages 124–139. Springer, 2016.

[125] Daniel Larimer. Momentum–a memory-hard proof-of-work via finding birthday collisions. Technical report, Tech. Rep., Oct. 2013.[Online]. Available: http://invictus-innovations. com ..., 2014.

[126] John Tromp. Cuckoo cycle: a memory bound graph-theoretic proof-of-work. In International Conference on Financial Cryptography and Data Security, pages 49–62. Springer, 2015.

[127] Alex Biryukov and Dmitry Khovratovich. Equihash: Asymmetric proof-of-work based on the generalized birthday problem. Ledger, 2:1–30, 2017.

[128] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Annual international conference on the theory and applications of cryptographic techniques, pages 281–310. Springer, 2015.

[129] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Annual International Cryptology Conference, pages 357–388. Springer, 2017.

[130] Thijs Laarhoven, Michele Mosca, and Joop Van De Pol. Finding shortest lattice vectors faster using quantum search. Designs, Codes and Cryptography, 77(2):375–400, 2015.

Author's Communications

1. Ch Jayanth Babu, R. Padmavathy, "Lattice-based authentication and key exchange protocol for Internet of things", The Journal of Supercomputing, Springer(SCI-Under Review)

2. Ch. Jayanth Babu, R. Padmavathy, "Quantum-secure n2n authentication protocol model for iot sensor networks," Arabian Journal for Science and Engineering, pp. 1–12, Springer, September 2023.(DOI: https://doi.org/10.1007/s13369-023-08242-5). (SCI-Published)

3. Ch Jayanth Babu, R. Padmavathy, "Construction of communication protocol using Ring-LWE-based homomorphic encryption in IoT-cloud environment", International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 9, pp. 2268–2275, Elsevier,November 2023.(Scopus-Published).

4. Ch Jayanth Babu, R. Padmavathy,"Post Quantum Blockchain with Provable Security", IEEE Transactions on Dependable and Secure Computing, (IEEE Trans.,-Comments Received)