International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Construction of Communication Protocol Using Ring-LWE-Based Homomorphic Encryption in Iot-Cloud Environment

[1]Ch Jayanth Babu, [2]R Padmavathy
[1]Dept.of Computer Sci.& Engg.
National Institute of Technology, Warangal
Telangana, India 506004
jayanth_babu786@student.nitw.ac.in
[2]Dept.of Computer Sci.& Engg.
National Institute of Technology,Warangal
Telangana, Inida 506004
rpadma@nitw.ac.in

Abstract— 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 cryptographic encryption and access control at cloud 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 paper, we proposed a scheme for the security and privacy of the 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 analysed for security, privacy, reduced messaging overhead and computation overhead. The objective of this paper is to Design and Construct a Ring-LWE-based homomorphic encryption(HE) communication protocol for authenticated user message encryption in a IoT cloud computing environment. The evaluation function in holomorphic encryption defined based on Ring-LWE encryption for a practical sharing-enabled cloud storage. Then, formally proving the security of the proposed protocol for classical and quantum attacks in cloud environment like Manin-the-middle (MITM) attack, Denial of Service (DoS) and Replay Attack.

Keywords-Quantum secure cryptography; Internet of Things; Ring-LWE; Homomorphic Encryption; Lattice based cryptography; Man in-the-middle (MITM) attack; Denial of Service (DoS) and Replay Attack

I. Introduction

The Internet of Things (IoT) is evolving as the "Future Internet," where all devices are inter-connected for communication and transfer of data. The sensor device in the physical layer collects data and transfers it to servers via the gateway device (G) for processing and storage. For a secure IoT-based applications, it is necessary to ensure the data source is trustworthy. On the other hand, IoT cloud server must have a robust mechanism to store and share the encrypted data with other nodes and intermediate nodes such as gateway devices. Cryptographic algorithms used in IoT applications have unique challenges due to limited resources. Most encryption schemes are built based on two well-known mathematical hard problems: Integer Factorization (IF) and Discrete Logarithm Problem (DLP). However, Shor and Peter [17][18] solved asymmetric cryptosystems (RSA, DH, ECC) in polynomial time with quantum algorithms. Then after, Grover's quantum algorithm [7] solved symmetric cryptosystems (AES, SHA-2, SHA-3) in polynomial time. The RSA and Diffie-Helman key exchange schemes are proved less efficient for huge volumes of data and easy to break with quantum computers. Researchers have started working towards quantum algorithms for the future, as a result, it is proved that *post-quantum* or *quantum-immune* cryptosystems can withstand quantum computers.

Post-quantum secure protocols have significance in contemporary IoT infrastructure environments. It has been shown that no polynomial time algorithm can solve mathematically complex problems of the type Lattice-based, Hash-based, Code-based, Multivariate, and Isogeny cryptosystems. Lattice-based cryptography (LBC) is one of them, and it is very efficient, safe, and excellent for resource-constrained platforms [4]. Because it uses matrices and vectors for computation in specified rings or fields of small order, lattice-based problems operate on relatively small integers. learning with errors (LWE), Short integer solution (SIS), and variants of these problems make lattice-based cryptosystems more secure. In this paper, we design and develop a Ring-LWE based homomorphic encryption scheme for IoT cloud environment to enhance security against post-quantum attacks. The proposed scheme is designed based on the Ring-LWE problem. In this scheme, IoT nodes will register at the cloud server, then 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 is used for quantum enabled security and privacy. The proposed scheme is analysed for security and compact in presence of quantum attacker.

1.1 Motivation and Contributions

The security and privacy of resource-constrained devices have become an emerging area of research in current wireless communication networks. Many encryption schemes have been proposed for IoT infrastructure networks. However, majority of protocols are vulnerable to quantum attacks. In this paper, the Ring-LWE-based fully holomorphic encryption-based design is proposed for the authenticated IoT user and cloud server environment. It ensures data safety and avoids the leakage of user information. The data verification method is applied a fully homomorphic cryptogram algorithm at the cloud server to encrypt the data sent by the IoT Node in the cloud environment.

The aim of this research paper is to explore the application of fully homomorphic encryption (FHE) in cloud and IoT environments. The primary focus is on developing a data verification technique utilizing FHE, which facilitates secure communication between the server and IoT Node device. 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 encryption scheme presented aims to facilitate registration and key generation within the IoT Cloud environment by facilitating the exchange of identities and random values among the IoT-node user and the cloud server. To ensure secure data verification and management, the IoT node and cloud server verify signature and identity values. The reliability of the generated signature during the registration step and message management process is maintained. Additionally, a communication protocol is developed to establish a secure communication channel that encompasses user verification and the data transfer process.

The remaining sections of the paper are as follows: Section 3 covers the related work of homomorphic encryption(HE) schemes for resource-constrained devices. Section 4 elaborates the construction of the proposed scheme along with the system model and threat models. The protocol is analyzed for several attacks on IoT network on the section in section 5. In Section 6, the performance of the proposed protocol is analyzed in terms of computation cost and communication cost. In the end, Section 7 contains the conclusion of the paper along with future directions of the work.

II. PRELIMINARIES

This section covers lattice hard problems and other variants used in this scheme. It covers the basics of the Ring-LWE problem along with some assumptions

A. Lattice Hard Problems

LWE Distribution: A uniform matrix $A_{s.\chi} \epsilon Z_q^n \times Z_q^n$ for a secret vector $s \in Z_q^n$ and choose uniformly random $a \in Z_q^n$, and choosing $e \leftarrow \chi$ and outputting; $(b =< a, s + e \ mod \ q)$ Note: Error distribution χ over *Z* is usually used for Gaussian distribution or binomial distribution [13].

Ring-LWE Distribution: A ring of polynomials R of degree *n* over *Z*, and defining the quotient ring $R_q = R/qR$. Ring-LWE distribution $A_{s.\chi} \epsilon R_q \times R_q$, secret vector $s \ \epsilon R_q$ and choose uniformly random $a \ \epsilon R_q$ and choosing $e \leftarrow \chi$ and outputting; $(a, b =< a, s + e \ mod \ q)$ [10] [11], The Ring-LWE problem has two versions: search-Ring-LWE and decisional-Ring-LWE.

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 $(a, b =< a, s + e)$ 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 $e \overset{\$}{\leftarrow} \chi_e$.

Decisional 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 Decisional Ring-LWE problem 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 R-LWE $R - LWE_{n,q,\chi}$ assumption holds, then Search-$LWE_{n,q,\chi}$assumption also holds.)

III. RELATED WORK

The concept of Homomorphic Encryption (HE) was first coined by the researcher Rivest in the year 1978 with the term ”privacy homomorphism” [14]. 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. Prior to the formalization of homomorphic encryption, certain encryption systems already achieved a degree of homomorphism. One such example is the Hill Cipher encryption scheme, which 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 [8]. Later, with the introduction of a privacy homomorphic scheme, the era of homomorphism has been started and many schemes are proposed. The RSA algorithm based on an integer factorization problem which follows the homomorphic multiplicative property[14][15].

In 1984, Goldwasser and Micali proposed first public -key encryption algorithm with semantic security. A probabilistic encryption technique named GM algorithm is proposed based on quadratic residue modulo and trapdoor function [16]. 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 public key cryptosystem and elliptic curve cryptosystem [5]. This cryptographic system exhibits a multiplicative holomorphic property and can be utilized for both encryption and signature verification purposes. In 1994, Benaloh made enhancements to the probabilistic encryption algorithm [1], 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 [12] was proposed by was proposed by Paillier based on the quadratic residue. It is random encryption scheme, which allows additive homomorphic operations.

In 2005, Boneh, Goh, and Nissim introduced the BGN cryptosystem, which relies on bilinear pairings [2]. This cryptographic algorithm exhibits the property of addition homomorphism and a single multiplication homomorphism. It stands as the closest scheme to the concept of homomorphism. Other algorithms, such as GM and Paillier, satisfy only additive homomorphism, while RSA, ElGamal, and BGN offer 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.

The first lattice based homomorphic encryption technique proposed in the year 2009 by Gentry [6] based on ideal lattices. This technique supports for the fully homomorphism with additive and multiplicative homomorphic property. It means, it support for addition and multiplication of ciphertext with unlimited number of times. Later, the homomorphic encryption schemes are evolved rapidly, many schemes are proposed by researchers. There are three distinct categories of homomorphic encryption schemes. The first category encompasses an ideal lattice-based fully homomorphic encryption scheme, initially 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 consists of an integer-based homomorphic encryption scheme [19] that follows Gentry's concept but eliminates operations based on ideal lattices of the polynomial ring. Instead, all operations are performed using integers. The third category involves homomorphic encryption methods based on either fully homomorphic LWE (Learning with Errors) or Ring-LWE (Learning With Errors over Ring). These schemes rely on the concept of Learning With Errors to achieve 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 [3].

A. Security Requirements

Security requirements of IoT node data and Cloud server communication is 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).
2. **Confidentiality and Message Integrity:** In 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.

IV. Proposed Protocol

A. System Model

The Internet of Things infrastructure network is made up of several nodes that function as clients and connect to (Cloud Server) 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 the data each other using help of IDP by computing public key and private key pairs to each user. Adversary (*Adv*) can observe the conversation between IN_i and CS, and Adv can reply, modify, delay, and can create new messages. The PPT adversary *Adv* is given access to the Oracle model which generates protocol-simulated outputs for any kind of *Adv*'s query. It may also allow protocol communications between any number of IoT node device 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 device instances may also be revealed. At last, an adversary *Adv* can directly decrypt data stored at CS or a password through multiple trials. The following assumptions are made for the IoT infrastructure scenario: The IoT node (IN_i) and cloud server (CS) are identified with a unique IDentity and One-time handshake. The Cloud Server (CS) and IoT node are mutually authenticated using PKE. The IDP maintains the IDs of connected devices in the network. An Unauthorised IoT node device (Quantum attacker) tries to get access of the IoT infrastructure network. The system model scenario of the IoT infrastructure network is shown in the figure:1.

B. 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.

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

C. Assumptions

In addition to the honest cloud assumption, we assume that 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 section 4.4 and notations used are listed in table 1.

Figure 1 System model of IoT Cloud communication

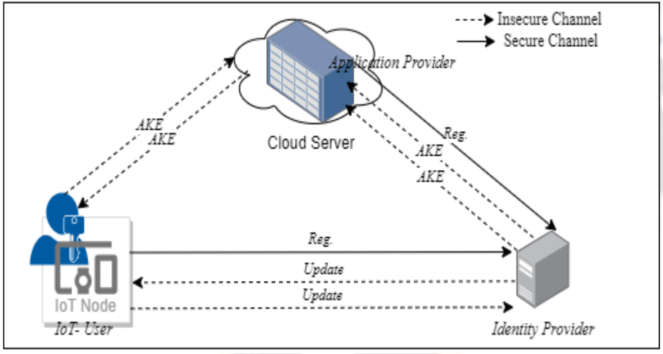


Table 1 Protocol Notations

Notation	Description
IN_i	IoT Node i
CS	Cloud Server
IDP	Identity Provider
s	secret value sampled from χσ
e	random error sampled from χσ
r	random values generated by Nodes
p	prime number
a	Uniform matrix from Z_qn×m
Adv	Adversary/Eavesdropper
H()	Hashing functions
m_i	message i.

D. Construction

The 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. Setup and Key Generation phase: In this phase, IDP setups and involves in the key generation and sharing of public keys between Users and CS with their respective controllers.

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* value is power of 2. This choice ensures that f(x) is irreducible polynomial 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.

Encryption Key Generation: The IDP generates the public-private key pairs for both parties similar to the Ring-LWE [13] scheme. For CS The vector $a_{cs} \in R_q$, two (a_{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 $(b_{cs} = a_{cs} . s_{cs} + e_{cs})$. 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 $(b_{cs} = a_{in} . s_{in} + e_{in})$. 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 [20] is employed for the purposes of signing and verification. 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 $d_{in} = (c_{cs} * s_{in} + p * e_{in}^*)$.

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 sends it 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:

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

$(IN_i\,; D_i;\ T_i)$. Here, IN_i represents the identification of the *i*th 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 , that 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 .

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

$$v_{in} = b_{cs} * r + e_2 + \left(\frac{q}{2}\right) * D_i \, mod \, q$$

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

Decryption $(\boldsymbol{u_{in}}, \boldsymbol{v_{in}}, \boldsymbol{s_{cs}})$: The CS verifies the signature of the user and stores in the database, $v_{in} - u_{in} * s_{cs} = (r * e - s_{cs}\,.\,e_1 + e_2) + \left(\frac{q}{2}\right) * D_i \, mod \, q$. When selecting suitable parameters, it is possible to ensure that the magnitude of $(r * e - s_{cs}\,.\,e_1 + e_2) \in R$ is less than $(q/4)$. As a result, the bits of D_i can be extracted by rounding each coefficient of $(v_{in} - u_{in} * s_{cs})$ 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

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

The CS replies with a 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.

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

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:

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

The recovered data value

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

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

$$Enc_d \, mod \, p) mod 2 = 2.r + d) mod \ 2 = d$$

Additive Property verification: Let's consider d1 and d2 are two data values. By applying encryption, we can transform these data values to encrypted form as follows:

$$Enc_{d1} = d_1 + 2.r_1 + p\ .q_1)$$

$$Enc_{d2} = d_2 + 2.r_2 + p\ .q_2)$$

In the above equations, Enc_{d1} and Enc_{d2} represent the encrypted data. Additionally, r_1 and r_2 are random values chosen for encryption, and $p\ .q_1$ and $p\ .q_2$ 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_{d3} is:

$$Enc_{d3} = Enc_{d1} + Enc_{d2}$$
$$= (d_1 + d_2) + 2(r_1 + r_2) + p(q_1 + q_2)$$

If $(d_1 + d_2) + 2(r_1 + r_2)$ is significantly smaller than *p*, we can simplify the expression for Enc_{d3} as follows:

$$Enc_{d3} = (\, Enc_{d1} + Enc_{d2}\,) mod \, p$$
$$= (d_1 + d_2) + 2(r_1 + r_2)$$

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,

$$d_4 = (d_1 \times d_2)$$

$$= (d_1 + 2r_1 \times p.q_1) \times (d_2 + 2r_2 \times p.q_2)$$
$$= d_1.d_2 + 2(2r_1r_2 + r_1d_2 + r_2d_1)$$
$$+ p[pq_1q_2 + q_2(d_1 + 2r_1) + q_1(d_2 + 2r_2)]$$

If $d_1.d_2 + 2(2r_1r_2 + r_1d_2 + r_2d_1)$ is significantly smaller than p, then we can express Enc_{d4} as:

$$Enc_{d4} = (\, Enc_{d1} + Enc_{d2}\,) mod \, p$$
$$= d_1.d_2 + 2(2r_1r_2 + r_1d_2 + r_2d_1)$$

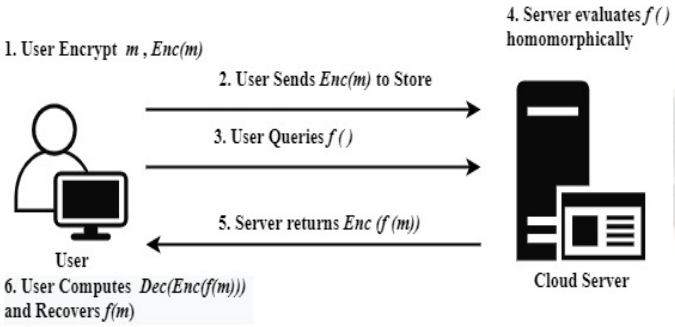
Therefore, the Multiplicative Homomorphic Encryption(MHE) property is satisfied.

V. Security Analysis

This section presents a comprehensive security analysis of the proposed node-to-node communication protocol for the Internet of Things infrastructure network. The security of the protocol is examined in light of the IoT security requirements. Over the communication network, adversary capabilities are defined and examined for analysis. An adversary *Adv* may collect all messages, but he or she cannot obtain the secret keys or impersonate as the authorized IoT Node. For example, even if an adversary *Adv* can impersonate some node IN_t , the verification procedure fails and the corresponding message is refused. Similarly, the proposed authentication mechanism is verified for security against the attacks listed below.

1. Reply Attack: When an adversary *Adv* obtains the authenticated message from the previous session and uses that message as a legitimate user in the current session then we say it a reply attack. 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 *Adv* tries to reply by decrypting these parameters, it fails in the verification process. So, he/she cannot 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 new committed values. When an adversary *Adv* 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.

Figure 2 Proposed IoT Cloud Server encryption Scenario


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

2. Man-in-the-middle attack: In this attack, an adversary *Adv* with the malicious node may try to obtain communication parameters. In the proposed scheme, the IoT node device IN_i uses fresh and random value generated from $r_i \in \{0,1\}^m$ and the secret key from $s_i \in Z_q^m$. As a result, if an adversary tries to login using a random nonce, the login attempt will fail after verification, and the server will detect a Man-in-the-middle attack. The malicious node cannot compute these parameters in polynomial time by solving the ISIS problem [10]. As a result, the protocol protects against man-in-the-middle attacks.

3. **Impersonation attack:** An impersonation happens when an adversary *Adv* pretends to be a legitimate IoT node IN_i and listens in on authentication messages. Based on the Ring LWE problem, each IoT Node computes values $r_i, s_i\ e_i \in \{0,1\}^m$. If an adversary pretends to be an IoT node IN_i, he or she must solve the Ring-LWE problem. It cannot be solved in polynomial time. Similarly, if the adversary tries to listen in on transferred communications to CS, he or she must respond IN_i. However, only CS is aware of the unique identification number and may compute committed values using the IN_i node's secret values a, r_i, u. As a result, the impersonation or eavesdropping attack fails at the proposed protocol's initial level of authentication.

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

5. Node Privacy:

The IoT node responds to CS by computing committed values; if IN_i computes the same values for each session, the node may be easily tracked. Each value in the proposed scheme is generated at random and independently using a pseudo-random number generator and one-way hash algorithms. To avoid tracing, the IoT node IN_i information is not included in computed committed values. As a result, an adversary can not determine whether or not the transferred messages are from the same tag. As a result, the proposed protocol passes the Node privacy property.

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

7. Offline Dictionary attack: Let us assume the adversary *Adv* gains complete access to all data stored on the IoT Node device, such as a, u_i . To gain access, the opponent must build an (u_i, v_i); to do so, the adversary must guess the random secret, even if the adversary does not know the unique identity of IN_i. It is impossible to verify the validity without the IoT node identity number supplied by IDP. As a result, offline dictionary attacks on the proposed protocol are impractical.

VI. Evaluation of Performance

In this section, the computational costs of the proposed Ring-LWE-based homomorphic approach are evaluated and its performance is compared to that of 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 Jin u. a. (2019). We considered the similar implementation setup to compare the performance analysis of the proposed protocol.

According to the Jin [9] implementation, the comparative performance is tabulated as in the table: 2. 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 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 considered for comparison with the proposed scheme, it is shown in table 2. The notations *TC* is for Time Complexity, *SC* stands for Space Complexity and *enc* and *dec* stands for encryption and decryption, respectively. We considered the implementation setup of Jin [9] 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 cloud server provided by 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 ECCbased 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.

Specification	**RSA-HE**	**Elgamal-HE**	**ECC-HE**	**Proposed-HE**
TCenc	O(n).m_i	m_i.O(n) + m_i.O(1)	m_i.O(1) + m_i.O(1)	m_i.O(1).O(2n)
TCdec	C(m_i).O(n)	C(m_i).O(n) +C(m_i).O(1)	C(m_i).O(1) + C(m_i).O(1)	C(m_i).O(1)
SCenc	ω_i.O(n)	ω_i.O(n) + ω_i.O(n)	2 ω_i.O(1)	ω_i.O(1)
Decryption difficulty	IFP	DLP	ECDLP	Ring-LWE

However, the proposed encryption protocol addresses this issue by incorporating the learning with 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.

VII. Conclusion

Traditional homomorphic encryption techniques used in 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, it is useful in a wide variety of applications to ensure quantum security at different levels.

References

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

[2] Boneh, Dan ; Goh, Eu-Jin ; Nissim, Kobbi: Evaluating 2-DNF Formulas on Ciphertexts. In: TCC Bd. 3378 Springer, 2005, S. 325–341

[3] Brakerski, Zvika: 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 Springer, 2012, S. 868–886

[4] Buchmann, Johannes ; G¨opfert, Florian ; G¨uneysu, Tim ; Oder, Tobias ; P¨oppelmann, Thomas: 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, Association for Computing Machinery, Inc, may 2016. – ISBN 9781450342834, S. 2–9

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

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

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

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

[9] Jin, Byung-Wook ; Park, Jung-Oh ; Mun, Hyung-Jin: A design of secure communication protocol using RLWE-based homomorphic encryption in IoT convergence cloud environment. In: Wireless Personal Communications 105 (2019), S. 599–618

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

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

[12] Paillier, Pascal: Publickey 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 Springer, 1999, S. 223–238

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

[14] [Rivest u. a. 1978a] Rivest, Ronald L. ; Adleman, Len ; Dertouzos, Michael L. u. a.: On data banks and privacy homomorphisms. In: Foundations of secure computation 4 (1978), Nr. 11, S. 169–180

[15] [Rivest u. a. 1978b] Rivest, Ronald L. ; Shamir, Adi ; Adleman, Leonard: A method for obtaining digital signatures and public-key cryptosystems. In: Communications of the ACM 21 (1978), Nr. 2, S. 120–126

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

[17] Shor, Peter W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th

annual symposium on foundations of computer science Ieee, 1994, S. 124–134

[18] Shor, PeterW.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: SIAM review 41 (1999), Nr. 2, S. 303–332

[19] Van Dijk, Marten ; Gentry, Craig ; Halevi, Shai ; Vaikuntanathan, Vinod: 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 Springer, 2010, S. 24–43

[20] [Wu u. a. 2012] Wu, Yanfang ; Huang, Zheng ; Zhang, Jie ; Wen, Qiaoyan: A lattice-based digital signature from the Ring-LWE. In: 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content IEEE, 2012, S. 646–651