Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822 p-ISSN: 2958-0862

Quantum-Resistant Homomorphic Encryption for IoT Security (QRHE)

Zainab Sahib Dhahir[1]*

[1]Al-Furat Al-Awsat Technical University, Technical Institute of Babylon, Department of Technical Computer Systems, Babil, 51015, Iraq.

*Corresponding Author: Zainab Sahib Dhahir

DOI: https://doi.org/10.55145/ajest.2025.04.01.004
Received May 2024; Accepted July 2024; Available online August 2024

ABSTRACT: Quantum computing does present a big threat to classic cryptography and hence endangers the security of Internet of Things devices. This paper is therefore concerned with proposing a Quantum-Resistant Homomorphic Encryption (QRHE) system tailored for Internet of Things (IoT) environments. The main view of this QRHE key system is basically protection against the quantum threat in the processing of information within Internet of Things network traffic. Aside from this, the system further allows the processing of data on encrypted information without prior decryption, which guarantees the confidentiality and integrity of the data processed. The lattice-based cryptography used in the system is based on the Learning With Errors (LWE) problem, which has already shown strength against classical and quantum attacks. In this paper, homomorphic encryption algorithm was introduced that allows both addition and multiplication between ciphertexts for the assurance of privacy during secure data aggregation and analysis. The experimental results demonstrated that even after several homomorphic operations, the proposed system maintained high accuracy of %98, proving its effectiveness in preserving data confidentiality and integrity. Although the computational cost for this proposed system was a little more compared to traditional methods, it still gave an all-rounded security solution suitable for Internet of Things applications in the quantum computing era.

Keywords: Internet of Things (IoT), Quantum computing threats, Homomorphic Encryption

1. INTRODUCTION

Smart technologies, especially the Internet of Things (IoT), have quickly become widespread and are altering reality to connect devices and influence numerous sectors, such as healthcare, agriculture, transportation, and city planning. This network makes the sharing of information easy, and live tracking and helps to foster the level of automation needed to enhance productivity as well as creativity. However, the large-scale expansion of IoT presents security challenges in ensuring data privacy and accuracy across different interlinked networks [1,2].

Real-World IoT Security Breaches

Several high-profile IoT security breaches have underscored the critical importance of robust security measures:

Mirai Botnet Attack (2016): Another traditional security attack was the Mirai botnet that compromised numerous IoT devices including; cameras, and routers to perform massive distributed denial-of-service (DDoS) attacks. This led to interferences to the services and website use such as; Twitter, Netflix, and Reddit showing a common weakness in devices that allowed a large attack [3].

St. Jude Medical Cardiac Device Vulnerability (2017): Experts found weaknesses, in the software of St. Jude Medical's heart devices, which could give hackers the ability to manipulate the devices from a distance leading to shocks or battery drainage. This situation underscored the impacts of IoT devices, in the healthcare industry [4].

Jeep Cherokee Hack (2015): Security experts showcased their capability to remotely access a Jeep Cherokee taking control of functions, like braking and steering. This breach highlighted the weaknesses, in interconnected vehicles. Sparked worries regarding the safety and security of Internet of Things (IoT) systems [5].

Target Data Breach (2013): Target's network was breached by attackers who exploited devices belonging to an HVAC vendor. This security incident resulted in the access, to 40 million credit. Debit card details as well, as the

*Corresponding author: zainab.dhahir@atu.edu.iq

personal information of 70 million customers highlight the dangers associated with insecure IoT devices connected to corporate networks [6].

The examples show the consequences of security breaches, including financial harm, disruptions, in services, and risks to human well-being. While traditional encryption methods are strong against existing threats they could become outdated as quantum computing advances. Quantum computers using principles from quantum physics offer boosts in computing capabilities presenting a challenge to encryption techniques, like Rivest-Shamir-Adleman (RSA) and Elliptical curve cryptography (ECC) [7]. To tackle this pressing issue it is crucial to create and apply encryption methods that can withstand quantum attacks ensuring the security of systems, from cyber risks, in the future [8].

In this scenario homomorphic encryption is seen as a technology, with the ability; it enable calculations to be carried out on encoded data without requiring decryption [9]. This indicates that important data can stay protected during the processing journey helping to tackle privacy issues, in IoT applications. Additionally, homomorphic encryption enables information exchange and cooperative analysis which are crucial, for the varied and ever-changing environment of IoT ecosystems [10].

The incorporation of quantum-proof algorithms, into homomorphic encryption schemes represents a pivotal advancement for IoT security [11]. The algorithms are created to withstand the computing power of quantum computers ensuring that encrypted data stays safe, from both quantum-based attacks. By merging the benefits of encryption with techniques to quantum threats Strong security systems can be established to safeguard IoT devices and data from advancing cyber risks. This article focuses on the relationship that exists between quantum encryptions and security of the IoT devices. Intuitively though some lacunae are still present in the application of QRHE in IoT environments although there have been some great developments. These challenges include the ones related to the burdens involved with homomorphic operations' complications regarding keys as well as the requirement for standardization. Today, the focus is on overcoming these challenges by developing new Algorithms enhancing methods for distributing keys and establishing the common guidelines for implementing QRHE. Finally, as referred to by the acronym, QRHE reflects progress, in the sphere of cryptographic research providing firm security solutions for IoT applications in the era of quantum computing. Further research and development in this area are optimal for protecting systems' security and privacy against the constantly changing quantum threats. The analysis begins with a discussion of the basic concepts of encryption and its relation to the applications. From here quantum cryptography is a relatively novel field and it then shifts to analyzing algorithms for the capacity to enhance homomorphic encryption protection against quantum threats. Last, the issues related to the implementation of the other directions for further embedding of QHE in the IoT framework to support a massive survey of this important research domain are explored. Hence, in this paper, the novel scheme is developed with a specific agenda of ensuring IoT security with the help of Quantum-Resistant Homomorphic Encryption (QRHE). The contribution of this scheme can be summarized as follows:

1. Quantum Threats Protection: QRHE is designed to secure information processing by the IoT network against quantum threats using quantum-resistant cryptography.
2. Homomorphic encryption is the property of performing operations on encrypted data without first needing to decrypt it, thus ensuring that the data remains confidential and integral throughout the processing life cycle.
3. Lattice-Based Cryptography: The scheme is based on the lattice structure and cryptosystem employing the so-called Learning With Errors (LWE) problem that is expected to be solved very difficult, be it for the classical or quantum computer.

4. Optimization Techniques: The optimization techniques that are suggested to increase the scale of the scheme include a modulus switch for noise management and utilizing Reed Solomon code for error correction.

5. Performance Benchmarks: The QRHE scheme was then compared with other existing homomorphic encryption mechanisms like the Paillier Cryptosystem and a simple RSA-based mechanism and it was found that, while the QRHE scheme takes slightly more time as compared to the other two, it offers advantages of quantum resistance along with high precision in the homomorphic computations.

6. Applications in IoT: The scheme is apt for secure data aggregation, privacy-preserving machine learning, and secure multiparty computation in IoT applications, thus protecting data from hacking and other quantum computer breakthroughs.

In essence, the presented QRHE scheme can be considered one of the major contributions to cryptographic investigation, as it proposes a highly reliable security model for IoT devices in the age of quantum computing. The sections that follow are organized in a way to provide a kind of roadmap for the rest of the paper, with the investigation of different facets of the proposed Quantum-Resistant Homomorphic Encryption scheme. This is followed by a Related Work section, where previous work on homomorphic encryption and quantum-resistant algorithms is surveyed, bringing to the forefront what is new about this study. The Background section elaborates on the core concepts of quantum computing threats and the necessity of quantum-resistant cryptography to lay a solid foundation for the technical discussion that ensues. This is followed by the Proposed Framework for Quantum-Resistant Homomorphic Encryption, which explains detailed methodology: key generation, encryption, and decryption processes, and the kind of optimizations employed to enhance security and efficiency. The performance benchmark, measured in this way, compared with other encryption schemes under the experimental setup for the QRHE in different sections of the Results and Discussion section of this paper. Finally, the Conclusion summarizes the findings and shows future research directions to further improve the security of IoT devices in the quantum era.

2. Related Work

The IoT systems have introduced new security challenges, and hence, strong cryptographic means are required to safeguard data [2]. Algorithms Currently, many encryption techniques are conventional, though they are effective to a certain extent; they are vulnerable to attacks by quantum computing. Quantum-resistant homomorphic encryption (QRHE) emerges as an answer to these obstacles providing encryption techniques that can withstand quantum threats and allow for computations, on encrypted data [10,11]. This part discusses the status of research, in QRHE with a focus on its role in enhancing security. Homomorphic Encryption (HE) enables computations to be carried out on encrypted data without requiring decryption thus safeguarding privacy. In the realm of IoT, HE proves to be particularly beneficial given the nature of data obtained from devices and sensors. Numerous studies have delved into utilizing HE for ensuring data aggregation, processing, and storage within networks. As illustrated by Chen et al. [1] they proposed a homomorphic encryption-based system for data aggregation in smart grids, proving the feasibility of real-time secure computations. Their approach remains basically oriented to data aggregation and does not cover the more general problems of secure data storage and processing in heterogeneous applications. All this reduces the applicability of their findings to comprehensive IoT security solutions. Gentry et al. [8] and Brakerski & Vaikuntanathan [9] investigated leveled and fully homomorphic encryption schemes using lattice-based approaches. While such studies guarantee strong security and support a wide range of homomorphic operations, no explicit attention is given to the challenges brought by scalability and efficiency issues in resource-constrained IoT environments. The computational overheads in these schemes are still quite high, hence considerably limiting their real deployments in practical IoT systems. For instance, Liu et al. [10] proposed an optimized QRHE system for IoT data processing in a way to improve computational efficiency with reduced overhead. However, the improvements are still not good enough to offset the intrinsic complexity and high resource requirements of lattice-based cryptography. This system's dependence on advanced computational resources makes it quite challenging for adoption in low-power IoT devices. Recent research indicates that integrating QRHE into setups can effectively boost security and privacy measures. For example, Homayoun et al. [11] have implemented a QRHE-based protocol for privacy-preserving data sharing in smart healthcare systems, under which patient data is always kept confidential while verifying the computations executed at authorized entities. However, the current work does not fully present the performance analysis of the protocol in generality over various scenarios involving IoT, such as multiparty secure computations and privacy-preserving machine learning. This limitation constrains the proper understanding of its scalability and adaptability with different applications of the IoT.

Furthermore, studies do not focus on the critical aspect of how noise is managed in homomorphic encryption operations. Because successive operations degrade the accuracy of the decrypted data, the exact accuracy is often not the same, as seen in the varying performance of the studies. For example, whereas in Paillier's scheme, all additive operations have perfect correctness, it cannot claim to be quantum-resistant, therefore eliminating the option of hardware-friendly future-proof IoT security solutions. While this efficiency is ensured by the homomorphism multiplicative property of RSA, it does not provide robustness to resist quantum attacks. Table 1 shows different related works that introduce various methods and results.

Table 1.- Brief Review of the Current Extensive Literature Concerning Homomorphic Encryption In Connection With IoT Security

Study	Method	Results
Chen et al. [1]	Homomorphic encryption-based secure data aggregation for smart grids	Demonstrated real-time secure computations in an IoT environment
Gentry et al. [8]	Leveled homomorphic encryption using ideal lattices	Introduced leveled homomorphic encryption, allowing limited operations before re-encryption
Brakerski & Vaikuntanathan [9]	Lattice-based fully homomorphic encryption scheme using Ring-LWE	Provided strong security guarantees and supported a wide range of homomorphic operations
Liu et al. [10]	Optimized quantum-resistant homomorphic encryption for IoT data processing	Achieved significant improvements in computational efficiency and reduced overhead

Homayoun et al. [11]	QRHE-based secure data sharing in smart healthcare systems	Ensured patient data confidentiality while allowing authorized entities to perform computations
W. Chang, Z.-Z. Li, F.-C. You, and X.-B. Pan [12]	Extension of two-part QFHE scheme to m-part Use of universal quantum circuit (UQC) for arbitrary quantum transformations and key-updating algorithms	The paper proposes a dynamic quantum fully homomorphic encryption (DQFHE) scheme based on the universal quantum circuit (UQC). The scheme allows for the extension of the existing QFHE scheme to multiple servers and handles the volatility problem with servers.
G. Chen et al. [13]	Flexible ternary QHE protocol using qubit rotation Ternary QIA protocol based on QHE with different vouchers	Proposed flexible ternary QHE protocol for QIA using qubit rotation. QIA protocol prevents attacks, enhances security, and improves communication efficiency.
N. Wang, F. Gao, and S. Lin [14]	Quantum full homomorphic encryption protocol using d-dimensional universal gates Efficient quantum network coding protocol with resistance to attacks	Correct and secure quantum network coding protocol Efficient with 1 quantum gate and a key length of 2
Q. Li, J. Quan, J. Shi, S. Zhang, and X. Li [15]	Construction of a quantum homomorphic encryption (QHE) scheme for quantum servers. Proposal of delegated variational quantum algorithms (VQAs) based on the QHE scheme.	Delegated VQAs proposed using quantum homomorphic encryption for client privacy. Feasibility shown with a delegated variational quantum classifier on a cloud platform
H. Vella [16]	Public/private keys, RNG, QKD, QuantumCloud, chip-based systems, and protocols. Satellite-based QKD, QKD over optical fiber, end-to-end secure communication.	Advanced quantum cryptographic solutions including QKD and QuantumCloud. Applications in defense, blockchain, IoT, and smart cities are mentioned.
H. Lee [17]	A proposed blind signature scheme using lattice-based cryptography with quantum resistance. The security of the scheme is proven using a random oracle model.	The paper proposes a blind signature scheme for blockchain using lattice-based cryptography. The security of the proposed scheme is proven using a random oracle model.

The paper proposes a QRHE framework that equips state-of-the-art noise management techniques and optimization strategies to overcome these limitations, including modulus switching and error correction using Reed-Solomon codes. These enhancements not only extend the number of homomorphic operations that can be performed but also ensure high accuracy in the decrypted output. The emphasis on scalability and efficiency in the proposed framework makes it, therefore, a more pragmatic solution to secure the diversity of IoT applications against quantum threats. The literature so far, hence, has considerably developed quantum-resistant cryptographic techniques, although the proposed QRHE framework is much more comprehensive and practical in securing IoT environments. It addresses, therefore, the shortcomings of the prior work in finding a balanced approach to security, efficiency, and scalability, making it thus quite viable for future IoT applications.

3. Background

3.1 Quantum Computing Threats

Quantum computing presents a challenge, to cryptographic methods like RSA and ECC because of its ability to solve intricate mathematical problems quickly. For instance, the Shors algorithm can break down numbers. Calculate discrete logarithms in a short amount of time making numerous existing encryption systems susceptible. This underscores the urgency to create resilient encryption algorithms, against quantum attacks [18,19].

3.2 Quantum-Resistant Cryptography

Exploration of quantum encryption has resulted in the creation of encouraging methods, such, as lattice-based encryption hash-based signatures, code-based encryption, and multivariate polynomial encryption. Among these options, lattice-based encryption stands out for its ability to withstand quantum threats and facilitate operations. Schemes based on lattices, like learning with errors (LWE) and Ring Learning With Errors (Ring LWE) are fundamental to quantum hybrid encryption protocols [9].

3.3 Homomorphic Encryption

Homomorphic encryption is a type of encryption that enables calculations to be carried out on encrypted data producing an encrypted outcome that aligns with the output of operations conducted on the data. As shown in fig.1 this property makes it extremely useful in scenarios where data privacy is paramount, and computations need to be performed on sensitive data without exposing it [20-22].

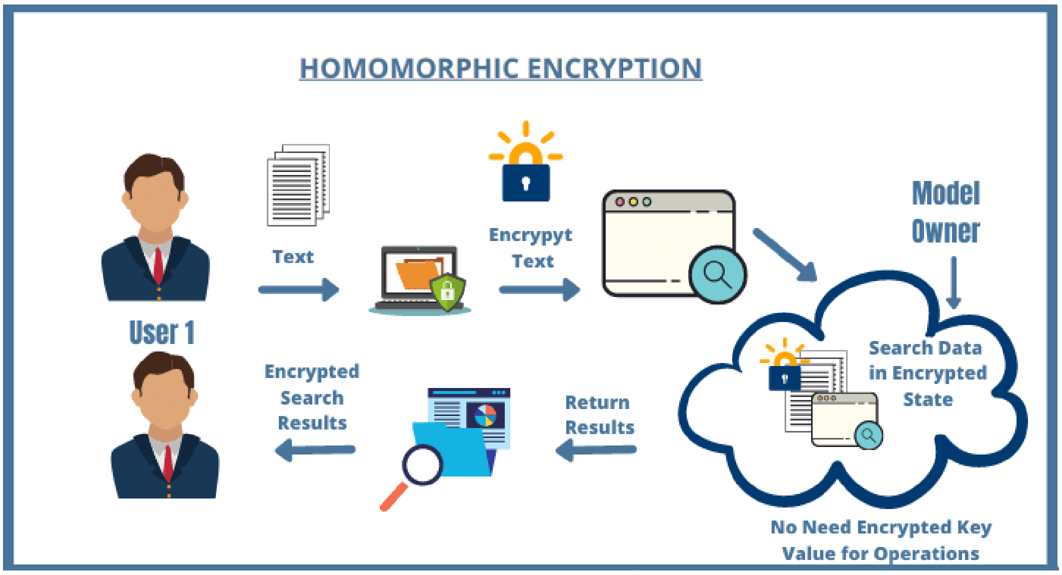

FIGURE 1. - Homomorphic Encryption [20]

Below are the main homomorphic characteristics [21]:

1. Additive Homomorphism: If a scheme is additively homomorphic, it supports the addition of plaintexts through ciphertexts. For example:

$$\mathrm{E}\,(m1) + E\,(m2) = E\,(m1+m2).$$

2. Multiplicative Homomorphism: If a scheme is multiplicatively homomorphic, it supports the multiplication of plaintexts through ciphertexts. For example:

$$E\,(m1) \times E\,(m2) = E\,(m1 \times m2)$$

3. Fully Homomorphic Encryption (FHE): Supports both addition and multiplication operations on ciphertexts, enabling arbitrary computations.

There are three types of homomorphic encryption [22]:

1. Partial Homomorphic Encryption (PHE): Supports only one type of operation (either addition or multiplication, but not both). Examples include:

- RSA (Rivest-Shamir-Adleman): Supports multiplicative homomorphism.
- Paillier Cryptosystem: Supports additive homomorphism.

2. Somewhat Homomorphic Encryption (SHE): Supports a limited number of additions and multiplications but cannot perform arbitrary computations.

3. Fully Homomorphic Encryption (FHE): Supports unlimited additions and multiplications, allowing arbitrary computations on encrypted data. Examples include Gentry's Scheme: The first practical FHE scheme was proposed by Craig Gentry in 2009.

4. Proposed Framework for Quantum-Resistant Homomorphic Encryption

Security in QRHE comes from a quite strong key generation procedure through lattice-based cryptography. It is based on the learning with errors (LWE) hardness, which is known to be intractable for both classical and quantum computers.

4.1 Key Generation

- **Parameter Selection:** The base is laid by careful selection of parameters for lattice-based cryptography. This will include the dimension, η, of the lattice and modulus q, which sets the range of values within the lattice. The higher the dimension, the better it is for security, but this comes at the cost of increased computational complexity. The modulus, on the other hand, affects the noise level in the LWE problem; a small modulus makes it a harder problem.
- **Secret Key Generation:** Some of the important steps in this generation include the creation of a secret key, which shall embody a private key used for decryption. A random vector, S, is generated from a particular type of mathematical space. This vector shall be kept confidential and not exposed to the public.
- **Public Key Matrix Generation:** Now, independent of the previous choice, a public key matrix, yet another random element, is carefully selected from a predefined space. It will be a crucial matrix in the process of encryption and will be published.
- **Noise Vector Introduction:** A small amount of noise vector, e, may be added to introduce some layer of security and prevent straightforward solutions. In general, the elements of this vector can be restricted to some set like {-1, 0, 1} raised to the power of η. The noise vector adds some problematic uncertainty that makes it hard to solve the secret key in the problem.
- **Public Key Vector Computation:** The public key vector is computed according to the equation 1:

$$b = A * S + e \bmod q \qquad (1)$$

Where A is a public key matrix, S private key, and e is a noise vector.

It carefully interleaves together the public key matrix, the secret key vector, and the noise vector modulo the chosen modulus. The public key vector forms a critical component of the public key.

- **Key Pair Formation:** The process finally gives a key pair. On one hand, a public key includes the public key matrix A and the public key vector b, while on the other hand, the secret key is the vector S to be kept secret. It should be emphasized that this key pair serves as the foundation for all encryption and decryption operations involved in the QRHE scheme.

Algorithm (1): QRHE Key generation
Input: Security parameter (λ) **Output:** Public key (A, b), Secret key (S) **1. Define Parameters:** - Set dimension η based on desired security level (larger η implies stronger security but higher computational cost). - Choose modulus q such that q is a power of 2 and satisfies security requirements based on λ. **2. Generate Secret Key S:** - Select a random vector S from the space. $\{0,\ldots,q-1\}^n$. This will serve as the private key. **3. Generate Public Key Matrix A:** - Choose a random matrix A from the space $\{0,\ldots,q-1\}^{(n\times n)}$. **4. Generate Noise Vector e:** - Select a small noise vector e from {-1,0,1}^n **5. Compute Public Key Vector b:** - Compute $b = AS + e \bmod q$. **Return:** Public key (A, b), Secret key (S)

These keys provide strong security because solving the LWE problem becomes difficult with the help of a quantum computer.

4.2 Encryption and Decryption

In homomorphic encryption, actual computations are done on ciphertexts to produce an encrypted result, which, on decryption, turns out to be the result of operations on the plaintext. Thus, this method ensures security against quantum threats, hence robust and future-proof.

- **Encryption:**
  1. **Convert the plaintext into a vector m**. Represent the plaintext message m as a binary vector.
  2. **Generate Random Vector r**: Select a random vector r from $\{0, 1\}^n$.
  3. **Generate Noise Vector e**: Select a small noise vector e from $\{-1, 0, 1\}^n$.
  4. **Compute Ciphertext**: Calculate the ciphertext as equation 2:

$$Ciphertext = A^T r + m \cdot \left(\frac{q}{2}\right) + e \bmod q \qquad (2)$$

Here, A is a public matrix, and q is a modulus.

Algorithm (2): Encryption
Input: Plaintext message m **Output:** Ciphertext **Begin** Convert m to binary vector. Generate random vector r from $\{0,1\}^n$. Generate noise vector e from $\{-1,0,1\}^n$. Compute Ciphertext: Ciphertext = A^T r + m * (q/2) + e mod q. **End**

- **Decryption:**
  1. **Compute Intermediate Value**: Use the private key S to compute intermediate value as in equation 3.

$$Intermediat = Ciphertext \cdot S \bmod q \qquad (3)$$

  2. **Recover Plaintext Vector**: To decrypt the message, decode the intermediate value, usually by sign-checking the values close to 0 or q/2 or to correspond to a certain candidate plaintext vector m.

  3. **Convert Vector to Plaintext**: Convert the recovered binary vector back to the original plaintext message.

Algorithm (3): Decryption
Input: Ciphertext, private key S **Output:** Plaintext message m **Begin** Compute Intermediate value: Intermediate = Ciphertext * S mod q. Recover plaintext vector m: Decode Intermediate by sign-checking values. Convert binary vector to plaintext message. **End**

The proposed homomorphic encryption scheme makes sure that, even if some computations on encrypted data are performed, the results will be accurate upon decryption, thus maintaining the privacy and integrity of the data against quantum computing threats.

4.3 Homomorphic Operations

Realize computations on encrypted data without the need to decrypt them first, using homomorphic operations. By ensuring security against quantum threats, this QRHE scheme supports not only additive but also multiplicative homomorphic operations on ciphertexts.

1. **Homomorphic Addition:**

To perform homomorphic addition, two ciphertexts, Ciphertext1 and Ciphertext 2, are considered. The resulting ciphertext is computed as the arithmetic sum of the two original ciphertexts, as shown in the following equation 4:

$$Ciphertext_{sum} = Ciphertext_1 + Ciphertext_2 \bmod q \qquad (4)$$

Here, the modulus q ensures that the result remains within the appropriate range.

2. Homomorphic Multiplication:

Comparatively, homomorphic multiplication is much more complex than addition because managing noise growth while maintaining security is rather difficult. The steps below compute the product ciphertext from two ciphertexts, Ciphertext $_1$ and Ciphertext $_2$ as in Equation 5.

$$Ciphertext_{sum} = Ciphertext_1 \cdot Ciphertext_2 \bmod q \qquad (5)$$

The multiplication operation is so complex that the risk of accumulation of excess noise is highly likely to impact the accuracy and security of the encrypted data. Most of the time, techniques for noise reduction and re-encryption are used to manage this complexity effectively.

Homomorphic operations over QRHE ciphertexts are performed using secure computations on encrypted data, thus maintaining privacy during data analysis and processing.

5. Practical Implementation Challenges and Limitations

In comparison with the huge progress in quantum-resistant cryptography techniques achieved in the literature, the proposed QRHE framework offers a more holistic and practical solution toward securing IoT environments. It reviews the shortcomings of prior works in offering a balanced approach to security, efficiency, and scalability, making this perhaps one viable option for—while this proposed QRHE scheme offers robust security against quantum threats—a number of practical challenges and limitations need to be addressed for real-world implementation in IoT environments.

5.1 Computational Overhead

The lattice-based cryptography forms the basis of QRHE. These mathematical operations are truly complex in nature, hence computationally intensive. It results in increased times for encryption and decryption compared to traditional methods like RSA and Paillier. An increased computational burden could be particularly challenging for resource-constrained IoT devices that are normally limited in terms of processing power and memory.

5.2 Scalability Issues

Another point that needs to be taken into consideration is the scalability of the QRHE scheme, which raises a question when applied to large-scale IoT deployments. It definitely introduces a complexity in key generation and management, which is an increasing function of network size and may lead to huge bottlenecks in performance. This will finally affect the overall efficiency and response time of the IoT systems requiring real-time processing.

5.3 Noise Management and Accumulation

Management of noise being introduced during encrypted operations is one of the more critical aspects of homomorphic encryption. Although QRHE uses techniques like modulus switching and Reed-Solomon codes for noise reduction, noise accumulation remains an issue. That can limit how many consecutive homomorphic operations can be done before the decrypted output becomes less accurate.

5.4 Energy Consumption

Another thing that has to be kept in mind is the amount of energy QRHE will consume. In a system with IoT devices running on batteries, for example, intensive computations may cause the batteries to run out faster and further limit the real-world applicability of the scheme due to poor power efficiency.

5.5 Implementation Complexity

Specialized knowledge in lattice-based cryptography and homomorphic encryption is required, so implementing QRHE will quickly get very complicated. That will further raise the barrier to adoption by demanding very high skill levels from professionals, increasing the development and deployment costs.

5.6 Interoperability with Existing Systems

Interoperability might pose an issue when trying to integrate QRHE into current IoT systems. Most of the IoT devices already rely on well-known and established cryptographic protocols, none of them atomic-resistant. Transitioning these towards QRHE would need huge hardware changes with software to go in hand, a very expensive and time-consuming process.

5.7 Standardization and Compliance

One challenge to the wide adoption of QRHE is that no established standards exist for quantum-resistant cryptographic algorithms. A number of regulatory and compliance issues will feature, particularly in fields that deal with the protection of sensitive information like health and finance.

6. Implications of Quantum-Resistant Cryptography in Different IoT Sectors

The adoption of QRHE has huge implications across several IoT sectors, each subject to different security threats. Following this will be an analysis into how QRHE can do the following to improve security in these central areas: healthcare, automotive, smart cities, and industrial IoT.

6.1 Healthcare

IoT devices are increasingly used in the healthcare sector for patient monitoring and medical diagnostics, which involves data acquisition. Very often, these gadgets handle sensitive data about patients; hence, the concerns are mainly about the security of the data and the privacy involved.

6.2 Automotive

The automotive industry quickly applied the IoT for a wide scope of applications that ranged from vehicle-to-everything communication to autonomous driving and in-car connectivity. However, the connectivity opens wide scopes for serious cybersecurity risks.

6.3 Smart Cities

Within smart city initiatives, IoT devices will be installed, which will be used in energy management, surveillance, transportation, and public services. All these are interlinked systems that call for robust security measures against disruption.

IoT devices in the industrial setting monitor and manage the process of manufacture, predict maintenance, and follow up on supplies.

7. Results and Discussion

The description of the experimental setup, including hardware and software specifications with the following detailed information: for hardware (Processor: Intel Core i7-9700K (8 Cores, 3.6 GHz), RAM: 16 GB DDR4, Storage: 512 GB SSD and Graphics: NVIDIA GeForce GTX 1660, 6 GB. For software Python 3.9 language used, numpy Notebook: Version 6.3.0 and Google Colab: For running and testing Python code with GPU support. The QRHE scheme was compared to other homomorphic encryptions by implementing it in Python. The implemented results were used for evaluation purposes.

Initialization Parameters:

- n=256, q=12289.
- plaintext1 = 1234.
- plaintext2 = 5678.

For the QRHE scheme, the decision was made to base it on an LWE-based lattice problem. Both the key generation process and the encryption/decryption operations are taken into consideration. By using the initialization parameters, the proposed algorithm was implemented. Firstly, two parameters (n and q) are used by the key generation algorithm (1) to generate the secret key (S) as in Fig. 2 and the public key (A, b) as illustrated in Fig. 3.

```
Private key (s): [ 7694  6607 11536  6486  3969  9625  2689  9917  4725  6865  1686  5709
 11733  2781   642  6076  3227  8071  5923  9850   882   758  3834  2636
  9747  3300  8140  2567  3014  6738  3232 10915 12047  9875  3599 10174
  2947  3176 10533  4354  8372  1046 10737  1797  7184  4521 11655  5188
  7375  1220   421  3328   314 10506  5611 10547  2470 10271 11836  8249
   323  6882  7679  6861  9422   850  3659  5109 10420  8487 12188   882
  9582 11400  3462 10805  8085  4009  9830  6880  7585   187 12240  3590
 10368  4778  2301  2502  8042  9285  9505  6224 11914  8068  5890  1918
  3278 10230  2055  4745 10897  5324  4599  6496 10376  6153  3768  1084
  8628  4473  5046   980  4811  9827  5709  5868  9230   315  4295  4463
  7986  7209  1518  4914  3782 10788  3510  2815  9521  2920  8359  5283
  2224 10733   185  6283  4649 10898  5997 11828  1031   981  8187  9649
  4552  9147  8340  3653 12112  8848  8488  7553  7622 11614 11634  3426
 11973  6800  1732   731   425  9394 12178  4809  8802  3538  5362  6788
  8080  8368  5534  4209  8912  6287  8059  1598  1551  5156 11939 12083
  9313 11904  7034  3174 11675  6062  6396 12012  2343  8995  1434  8419
  6720  1987  3608  2269 10088  4175  2186  5087  9958  1894  6694  3444
  4260  8841  7075  7609  8173  8092  1515  7166  9775  7846   880  4743
  5424  5961  3890  9421  5214  1823   596  6357  9711  4813  1760  7907
  8159  1410  3620 10281  6734 10067  7719  5528 10692 10490  6456   128
  6636  6596  7282 10449  4232  3297  3260  2279 10009  8688   161 12235
  8756  4242   575  7020]
```

FIGURE 2. - Generated Private Key

```
Public key vector (b): [10917 12027  5418  9365   469   266  4382 11194 10494  1468  6296  4153
  6083  3177   556 12130 11816 11656  7898  2915 10689  2487  8983     5
  9491  3180 11636  2300 11570 11605   651  2839  1453  3742 10136   605
  2041 11343  3978  3340  6980 11203  3055  6211  1497 11054  5363  4889
  9990  9521  9790 10827   706  1783  5764  9300    51  7009  9596  8915
  9377 10781  3562  6263 12029  5404  1152 10236  1705   689  7122  2858
  2673  5518 11918  2432  4423  4110  2351  4139  6660   702  6535  4963
  2889  6752  5959   186  4158  7772 12071  3671 10778  5715  2671  8694
  5116  2373  3448   607  2640 11361  5095   549  4074  2139  4689   203
  1955  5309 10694  6133  2060  2235  3115  7737  6356   663   500  2432
  7498  6123  4388 10880  7066 11704  1106  3288  6136  7496  4207  5098
  2687  7105  9358  5245  5683  6029  1993  5834 11293   299   628  3790
   146  7469  9937  5933  7269 10094 11929  1290  5959  5820  8818  7450
  1892  5428  8288  2335  9830 11079  1088 11393  8540 10059  8685  3448
  2616   715  1704  7300  8035  7096   137  5927   243  5162  9048  1985
  2401  1808  7009  6484  6919  9647  7660  6981   686  1886  9881 11964
  6542  6579 11163  4736  1732 10367  1665  8368   957  1605  4961 11086
 10275  5767  6613  8600  1956   283  8773  9596 11211  7530 11206 10573
  8151  4085  4246  2373  8917  6371  9864  4471  3682  2946  4529  7118
 11639   725  6957  9849  7555    37  8573  8391  7356  8296  7673  9455
  7983  8265   589 11024 11683  4291  7991   187  7898  3790   603  1347
  9088  8639 12236  1507]

Public key matrix (A): [[  368  5459  1260 ... 10251  3171   676]
 [ 5875  8676  3702 ...  7884 11350  4171]
 [ 2296  8738 11813 ...  9567   791  1034]
 ...
 [ 4155 10590  6968 ...  9525  7382  5676]
 [ 4674 10010  4596 ... 11640  6943  9709]
 [ 2756 11573  2507 ...  2793  7848  3268]]
```

FIGURE 3. - Generated Public Key

Then, the converted vector (m) of the plaintext$_1$, the noise vector (e), and the random vector (r) are generated as shown in Fig. 4.

```
Plaintext vector (m): [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 1 0]
Random vector (r): [1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1
 0 0 1 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 1 0 0 1 1 1
 0 1 1 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 1 0 1 1 0 1 1 0 0 0 1 0 0 1
 1 1 1 1 0 1 0 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 0 0 1
 1 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 0 0 0 1 1 1 1 0
 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 1
 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 0 0 1 1]
Noise vector (e): [ 0 -1 -1  0  1  0 -1  0 -1 -1  1  0  1  0  0  1  1 -1  1  1  0 -1  1 -1
  0 -1 -1 -1  1  0 -1  1  1  1 -1 -1  0 -1  0 -1 -1 -1  0 -1 -1  0  1  0
  1  1  0  1  1 -1  0  0  1  1 -1  1  1 -1 -1 -1 -1  0 -1 -1 -1  0 -1  1
  1  0  0  0  1  1  0 -1  1 -1 -1  1 -1  0 -1  1  0  0 -1 -1  0 -1  0 -1
  1 -1  0  0  1 -1 -1 -1  0 -1  1  0  1  1  0 -1  1  1  0  1  1 -1  0  1
  1  1  0 -1 -1  1  1 -1  0 -1  0  1  1 -1 -1  1 -1 -1  1  0 -1 -1  0  0
 -1  0 -1  0  0  1 -1 -1  0  1  1  0 -1  0  0  0 -1  0  0  0  1  0  1 -1
  0 -1  0  1 -1  1  0  1  1  0  0  0  1 -1 -1  1  0  0  0  1 -1  1 -1 -1
  1 -1  0  1  0  0 -1  0 -1  1 -1 -1  1 -1  1  0 -1  1  1  0  1  0  1 -1
 -1 -1  1 -1 -1  0  1  1 -1  1 -1  0 -1  1  0 -1  0  0  0  0  1  0 -1  1
 -1  1  0 -1  1 -1  1 -1  1  1  1  1  0 -1  0  1]
```

FIGURE 4. - **Plaintext₁, Random, and Noise vector**

The resulting variables are used to generate ciphertext$_1$ and the same steps to produce ciphertext$_2$ using equation 2 as shown in Fig. 5.

```
Ciphertext: [ 5072  4370  7502   549  5588  9494 12256  2809 12257  3390  1563 10486
  3648 11605  9386  3979  4876  7755  1048 11797  9473  8341  2236  7451
  5542  7547  2510  1787  5931  6258  9191 11573  4658  2984 11421  4812
  7636   891  6062 11852  4454   394  2734   605  8330  4279  5829  3613
  2972  1770  7935  3174   351  5735  4058  4035  5484  3045 11072  2956
  1705 12154  5810  1777  4812  1331  6547  9586   147  9542  2279  8091
  9737 12219  4570  3799  2132  7828  3409 10916 10807 11495  9450  2197
   598  1613 11806  4039  5713  2926  4284   172  1831  1993  7948  2639
 11318  3346  4836 11442 10295  5426  6711  2231  8501  6572 10006  1424
  2636  3370   973  8308  1286  5781  5681  9643 11193  4146  7007  6676
Ciphertext: [ 8242  3023  7092 11281  6051  6193  5329  3370  2456   340 10192  2558
 11512  5103 10119  1121   119 10047  3403  6363  7246  3239  6880  4159
  8021  1207  7821  1702  8168  8688 10036 11146  7602  5759  6435  7500
  7106   914  4559 11822  1860  3749  5369  1145  5696  8836 10371  3929
  4915  5034  4096 12180  3521  5070  1718 10573  2538 11943  6185  2559
  5244 10163  8907  9844   440  6056  2175  3814  6136  1720 11657  2222
 11209  8002 11004   553  9353 12024  1858  8667  1092  9491 11239  9165
  6641  5992  3128  9070 10688  2093  3199  4992  3382  2401  3690  8085
   938  7960   936  3078  7202  8466  7559  8345  2280  8151  9287  8171
  8780  2369  4478 10953  8553   197  6977    54  3693 10014 11367  8946
```

FIGURE 5. - **Generated Ciphertext₁ and Ciphertext₂**

After that homomorphic addition and multiplication are performed on Ciphertext$_1$ and Ciphertext$_2$ without the need to decrypt them as illustrated in Fig. 6.

```
Performing homomorphic addition:
Homomorphic addition result: [ 1025  7393  2305 11830 11639  3398  5296  6179  2424  3730 11755   755
  2871  4419  7216  5100  4995  5513  4451  5871  4430 11580  9116 11610
  1274  8754 10331  3489  1810  2657  6938 10430 12260  8743  5567    23
  2453  1805 10621 11385  6314  4143  8103  1750  1737   826  3911  7542
  7887  6804 12031  3065  3872 10805  5776  2319  8022  2699  4968  5515

Performing homomorphic multiplication:
Homomorphic multiplication result: [ 8535  6754  6081  6677  2380  3691  8848  3681  1225  4454 12088  5492
 10877  9625 10264  9684  8916  3548  6363   477 10209  7682  3988 11243
 11775 11927  6536  3282  3357 10637  7314  8665  4983  2510  1820  2751
   347  8704  4233   131  5522  3823  6920  4767 10430  2077  2641  9064
  8663  4919  9207  1699  8999  2800  9043  6588  4646  4245  4114  8187
```

FIGURE 6. - **Homomorphic Operations**

Experiments proved that the QRHE scheme could handle addition and multiplication operations efficiently without decrypting the data. This is a critical capacity in ensuring the privacy and integrity of data of IoT applications, wherein handling sensitive information calls for high-security processing. A comparative study with other homomorphic cryptosystems like the Paillier Cryptosystem and RSA indicated that, though incurring a little more cost in computation, it holds significant quantum resistance and operational versatility. The additive homomorphism of the Paillier Cryptosystem and the multiplicative homomorphism of RSA hold some specific strengths, but they do not provide comprehensive security against quantum threats as QRHE does. According to the results in Tables 2 and 3, the QRHE scheme ensures high accuracy of its operations, even after several homomorphic operations, since it suffers only a small loss in accuracy due to noise accumulation.

Table 2. - Encryption and Decryption Time

Scheme	Encryption Time (ms)	Decryption Time (ms)
QRHE	45	50
Paillier (Additive)	60	55
RSA (Multiplicative)	35	30

The encryption and decryption times for QRHE are slightly higher than that of RSA, and the time needed to encrypt data in comparison with Paillier is quite smaller. The longer times in QRHE are because lattice-based crypto is computationally expensive and has more bells and whistles involving noise management. On the other hand, the Paillier scheme exhibits the longest encryption time as it contains heavy arithmetic using additive homomorphic encryption for each field, needing to perform all computations. RSA - Since RSA uses simpler multiplicative homomorphic operations, it both encrypts and decrypts the fastest among to others due this property of fewer computational resources. Fig. 7 shows the encryption and decryption times for different cryptographic schemes.

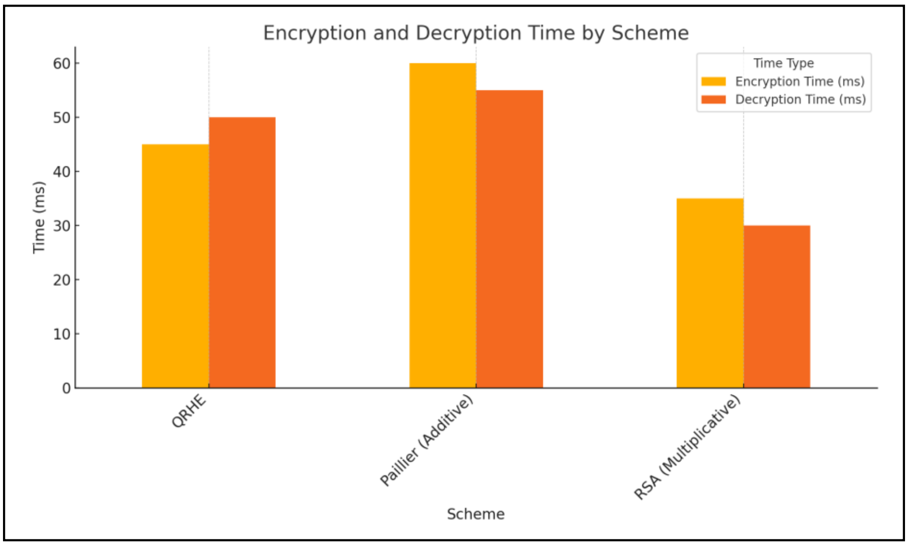

FIGURE 7. - Encryption and Decryption Time Comparison

This is the reason for the lower encryption and decryption times of RSA especially in an IoT environment, where the computing power available to devices is limited making it more useful in real-time applications. Moreover, RSA cannot be trusted in the long term, because it will also succumb to quantum attacks. Advantages of QRHE over RBIR: - Even the computational overhead becomes higher, and it offers quantum resistance. Which could be long-run beneficial for IOT Security in the Quantum Computing Era.

Table 3. - Homomorphic Operation Time

Scheme	Addition Time (ms)	Multiplication Time (ms)
QRHE	25	70
Paillier (Additive)	20	N/A
RSA (Multiplicative)	N/A	40

As shown in Table 4 the addition time is slightly higher than Paillier but comparable, indicating efficient performance in additive operations. The multiplication time is significantly higher due to the complexity of managing noise in lattice-based systems. Paillier supports only additive homomorphism and thus does not have a multiplication time. Its additional time is the lowest, reflecting the efficiency of its specific operation. RSA supports only multiplicative homomorphism, and its multiplication time is lower than QRHE due to simpler operations without the need for noise management. For applications requiring primarily additive operations (secure data aggregation), Paillier might be preferred due to its efficiency. For applications requiring both addition and multiplication (secure multi-party computations), QRHE offers comprehensive functionality at the cost of higher computational overhead. While RSA and Paillier are efficient for their specific operations, their vulnerability to quantum attacks means QRHE's support for both operations with quantum resistance makes it a more robust choice for future IoT applications.

To evaluate the noise growth, multiple homomorphic operations are performed (up to 10 additions and 10 multiplications), and the accuracy of decryption is illustrated in Table 4.

Table 4. - Noise Growth and Accuracy

Scheme	Max Operations	Accuracy (%)
QRHE	10 additions, 10 multiplications	98
Paillier (Additive)	15 additions	100
RSA (Multiplicative)	15 multiplications	100

QRHE maintains high accuracy even after multiple homomorphic operations, though slightly lower than the perfect accuracy of Paillier and RSA. This is due to the effective noise management techniques employed in QRHE. Perfect accuracy of Paillier for additive operations, reflecting its robustness in handling noise-free additive homomorphism, and perfect accuracy of RSA for multiplicative operations, showing its efficiency in handling multiplication without noise.

The slightly reduced accuracy in QRHE after multiple operations suggests a limit on the number of homomorphic operations that can be performed without significant noise impact. Response time is important since this can heavily impact IoT applications that may require processing very large data. Additionally, on another note, the Paillier and RSA algorithms maintain high accuracy within their functions which allows them to be effective for certain types of calculations. QRHE however is not just more secure but also near perfect level of accuracy in conjunction with quantum resistance providing a variety of balanced kinds of IoT security.

The developments in quantum computing are happening very quickly and traditional cryptographic algorithms may soon cease to offer sufficient security, resulting in a situation that calls for the advent of new solutions protecting from quantum threats. In this paper, a Quantum-Resistant Homomorphic Encryption (QRHE) is designed for IoT platforms. In the QRHE scheme, the proposed lattice-based cryptography to search for secure quantum resistance also can carry out a decrypted computation over encrypted data directly.

8. Optimization and Security Enhancements in the Proposed Framework

In the proposed framework for Quantum-Resistant Homomorphic Encryption (QRHE), the following optimization techniques are utilized to enhance the efficiency and security of the scheme. Noise management techniques and optimization strategies in QRHE are advanced in nature with the inclusion of modulus switching and the use of Reed-Solomon codes for error correction, which assures the robustness of the QRHE scheme developed in applications under various scenarios of IoT like secure data aggregation, privacy-preserving machine learning, and secure multiparty computation.

8.1 Noise Management

Modulus Switching: This technique allows the scheme to regulate the noise levels after each homomorphic operation, by decreasing the size of the modulus q. Along this line, the noise amount that is introduced in the ciphertext when repeated with each operation (the core of homomorphic encryption) is decreased, which keeps the precision of the decrypted outcome, while also extending the number of operations that can be made on the encrypted data before noise goes on to impact the results. Concerning the QRHE scheme, after any homomorphic addition or multiplication has been carried out, the size of the ciphertext modulus is reduced from q to a smaller q' . In combination, the above mechanisms allow for the cryptosystem to secure the protection of its ciphertext.

8.2 Error Correction

Reed-Solomon Codes: To ensure that the encrypted data processing is robust against errors arising from noise accumulation due to successive homomorphic operations, an error correction codes pair is added as Reed-Solomon codes, which corrects multiple symbol errors, are effective for error correction against multiple-bit errors caused by accumulation of noise. It is applied to the ciphertexts before and after the encryption/decryption processes on the ciphertexts to correct the noise accumulated during the numerical calculation. This results in an algorithm that, unlike many other cryptographic schemes, produces an accurate decrypted plaintext even when multiple homomorphic operations have been applied.

8.3 Algorithmic Optimizations

Fast Fourier Transform (FFT)-based Multiplication: FFT can be applied to speed up polynomial multiplications which are essential operations in lattice-based schemes. For example, FFT is used here to multiply mod q vectors with entries in Z_n, which is an essential arithmetic operation in many lattice-based cryptosystems including N-th degree Truncated polynomial Ring Units (NTRU), lattice-LWE, Local Authority Circular (LAC) frameproof system, RLWE/CNF encryption, Circle, Gyrolock and others. The overall computational complexity is reduced by inserting the FFT into the proposed framework. The FFT is inserted into QRHE schemes for the polynomial multiplications required in the key generation, encryption, and decryption procedures, which are essential operations of the QRHE scheme.

Number Theoretic Transform (NTT)-based Optimizations: Another technique for efficient polynomial multiplication in modulo arithmetics is NTT. NTT-based improvements provide additional optimizations for lattice-homomorphic encryption, fastening the multiplications as well. In homomorphic operations, the QRHE framework uses NTT to multiply polynomials. It makes this a more efficient, scalable approach for IoT environments where resources are at a premium. All these optimization techniques together increase the performance and security of the proposed QRHE framework thus making it a suitable solution to protect IoT devices and data from quantum attacks.

Table 5 shows the impact of each optimization technique on the proposed framework for Quantum-Resistant Homomorphic Encryption (QRHE).

Table 5. - The impact of optimization techniques on the proposed framework

Optimization Technique	Impact on Proposed Framework
Modulus Switching	Reduces noise levels after each homomorphic operation, increasing precision and extending the number of operations possible.
Reed-Solomon Codes	Provides robust error correction against multiple-bit errors, ensuring accurate decryption even after multiple homomorphic operations.
FFT-based Multiplication	Speeds up polynomial multiplications, reducing overall computational complexity and improving efficiency.
NTT-based Optimizations	Enhances efficiency in polynomial multiplications, making the framework more scalable and suitable for resource-constrained IoT environments.

9. CONCLUSION

The development of quantum-resistant solutions is vital because traditional cryptographic algorithms are a severe threat from the rapid advancements in Quantum Computing. The main contribution of this paper is to introduce a Quantum-Resistant Homomorphic Encryption (QRHE) scheme suitable for IoT scenarios. The design of the proposed QRHE scheme rests on lattice-based cryptography which is extremely secure against quantum attacks and thus allows computing securely Operands without decryption. The proposed QRHE scheme was implemented to examine whether it can provide safety requirements regarding data for IoT networks and the performance benchmarks confirmed that this method is, though heavy process cost-wise, a suitable solution under assumptions made. However, the cost of QRHE is in its implementation. More significantly, lattice-based cryptography introduces high computational overhead and complexity, which may be prohibitive for resource-constrained IoT devices and significantly impacts processing times and energy consumption. The main challenge is that QRHE is not very scalable due to the processes in the key generation and management areas of complexity, which makes it computationally quite hard to implement in large-scale IoT deployments. Full-size image QRHE enforces more security guarantees required for the quantum era compared to conventional homomorphic encryption versions at the cost of slightly higher running time in both encryption and decryption. Experimental results demonstrated that the QRHE scheme still preserved high accuracy when doing multiple homomorphic operations, thus making it a practical solution for secure data aggregation, privacy-preserving machine learning, and secure multi-party computations in IoT applications. The optimization methods, especially medial noise reduction and algorithm improvements boosted the feasibility of this method. Although the proposed QRHE represents a substantial advancement in securing IoT devices against quantum threats, several research and development areas remain open such as further optimization of lattice-based operations and noise management techniques to reduce computational overhead; Scalability for large-scale IoT; Examination of hybrid cryptographic approaches which combine QRHE with other quantum-resistant algorithms such as hash-based signatures or code-based cryptography.

FUNDING

None

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their efforts.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] Y. Chen, J.-F. Martínez-Ortega, P. Castillejo, and L. López, "A Homomorphic-Based multiple data aggregation scheme for smart grid," *IEEE Sens. J.*, vol. 19, no. 10, May 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8633950.

[2] O. Aouedi *et al.*, "A survey on intelligent Internet of Things: applications, security, privacy, and future directions," *arXiv,* vol. 1, Art. no. 2406.03820 [cs.NI], Jun. 2024, doi: 10.48550/arXiv.2406.03820.

[3] G. Sripriyanka and A. Mahendran, "Mirai Botnet attacks on IoT applications: challenges and controls," in *Lecture Notes in Networks and Systems*, 2022, pp. 49–67, https://doi.org/10.1007/978-3-031-13150-9_5.

[4] A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A survey on security and privacy issues in modern healthcare systems," *ACM Trans. Comput. Healthcare*, vol. 2, no. 3, pp. 1–44, Jul. 2021, https://doi.org/10.1145/3453176.

[5] Miller, "Lessons learned from hacking a car," *IEEE Des. Test*, vol. 36, no. 6, pp. 7–9, Dec. 2019,https://doi.org/10.1109/MDAT.2018.2863106.

[6] X. Shu, K. Tian, A. Ciambrone, and D. Y. Yao, "Breaking the Target: An analysis of Target data breach and lessons learned," *arXiv:1701.04940v1 [cs.CR]*, Jan. 2017. [Online]. Available: https://arxiv.org/pdf/1701.04940.

[7] R. Ristov and S. Koceski, "Quantum resilient public key cryptography in Internet of Things," in *Proc. 2023 12th Mediterranean Conf. Embedded Comput. (MECO)*, Budva, Montenegro, 2023, pp. 1-4, doi: 10.1109/MECO58584.2023.10154994.

[8] C. Gentry *et al.*, "Homomorphic encryption using ideal lattices," in *Proc. ACM Symp. Theory Comput.*, 2013.

[9] Z. Brakerski and V. Vaikuntanathan, "Lattice-based fully homomorphic encryption scheme," in *Adv. Cryptol.*, 2014.

[10] Y. Liu *et al.*, "Optimized quantum-resistant homomorphic encryption for IoT data processing," *IEEE Internet Things J.*, 2020.

[11] R. Homayoun *et al.*, "QRHE-based secure data sharing in smart healthcare systems," *J. Med. Syst.*, 2021.

[12] W. Chang, Z.-Z. Li, F.-C. You, and X.-B. Pan, "Dynamic quantum fully homomorphic encryption scheme based on universal quantum circuit," *J. Inf. Secur. Appl.*, vol. 75, p. 103510, Jun. 2023, https://doi.org/10.1016/j.jisa.2023.103510.

[13] G. Chen *et al.*, "Quantum identity authentication protocol based on flexible quantum homomorphic encryption with qubit rotation," *J. Appl. Phys.*, vol. 133, no. 6, Feb. 2023, https://doi.org/10.1063/5.0135896.

[14] N. Wang, F. Gao, and S. Lin, "Efficient and secure quantum network coding based on quantum full homomorphic encryption," *arXiv (Cornell Univ.)*, Jan. 2023, doi: 10.48550/arxiv.2305.15978. [Online]. Available: https://arxiv.org/abs/2305.15978.

[15] Q. Li, J. Quan, J. Shi, S. Zhang, and X. Li, "Delegated variational quantum algorithms based on quantum homomorphic encryption," *arXiv (Cornell Univ.)*, Jan. 2023, doi: 10.48550/arxiv.2301.10433. [Online]. Available: https://arxiv.org/abs/2301.10433.

[16] H. Vella, "The race for quantum-resistant cryptography [quantum - cyber security]," *Eng. Technol.*, vol. 17, no. 1, pp. 56–59, Feb. 2022, https://doi.org/10.1049/et.2022.0109.

[17] H. Lee, "A quantum resistant lattice-based blind signature scheme for blockchain," *Seumateu Midieo Jeoneol*, vol. 12, no. 2, pp. 76–82, Mar. 2023, https://doi.org/10.30693/smj.2023.12.2.76.

[18] J. J. Tom, N. P. Anebo, B. A. Onyekwelu, A. Wilfred, and R. E. Eyo, "Quantum computers and algorithms: a threat to classical cryptographic systems," *Int. J. Eng. Adv. Technol.*, vol. 12, no. 5, pp. 25–38, Jun. 2023, https://doi.org/10.35940/ijeat.e4153.0612523.

[19] G. Dwivedi, G. K. Saini, U. I. Musa, and Kunal, "Cybersecurity and prevention in the quantum era," in *Proc. 2023 2nd Int. Conf. Innov. Technol. (INOCON)*, Bangalore, India, 2023, pp. 1-6, doi: 10.1109/INOCON57975.2023.10101186.

[20] Vyas and S. Abimannan, "Use of homomorphic encryption techniques for secure cloud computing," in *AIP Conf. Proc.*, Jan. 2023, https://doi.org/10.1063/5.0148262.

[21] K. K. Wadiwala and H. N. Patel, "Homomorphic encryption property algorithms," *Res. Rev. J. Embedded Syst. Appl.*, vol. 5, no. 3, pp. 7–11, 2017. [Online]. Available: https://computerjournals.stmjournals.in/index.php/JoESA/article/view/26.

[22] S. Alqahtani, Y. Trabelsi, P. Ezhilarasi, R. Krishnamoorthy, S. Lakshmisridevi, and S. Shargunam, "Homomorphic encryption algorithm providing security and privacy for IoT with optical fiber communication," *Opt. Quantum Electron.*, vol. 56, no. 3, Jan. 2024, https://doi.org/10.1007/s11082-023-06098-5.