



Research Article

Design of A Post Quantum Key Encapsulation Mechanism for Consumer Electronics Using Ring Learning with Layered Errors

Fagbohunmi Griffin Siji* 

Computer Engineering Department, Abia State University, Uturu, Nigeria

Abstract

The rise in technical features and capabilities of consumer electronics (CSE) devices ranging from smartphones, high fidelity (Hifi) radio systems, MP radio sets and IoT-enabled smart devices, have created new possibilities about their use and applications. You tube files can now be downloaded on smartphones and played directly on high fidelity audio radio sets through Bluetooth. There have also been increase in the use of smart phones in diverse areas of life such as in academics, industries etc., where the use of online lectures and video conference meetings have increased in recent times. It is evident that these technologies offer convenience and smart connectivity, on the downside however is the security concern resulting from these communications. Classical cryptography algorithms built into these consumer electronics are prone to quantum computing attacks, and with the steady growth of quantum computing and algorithms, it is only a matter of time before the current classical cryptography algorithms become obsolete. This aim of this paper is to design a light weight Ring Learning With Layered Errors Key Encapsulation Mechanism (RLWLEKEM), a CCA secure lattice based post quantum cryptography algorithm for consumer electronics (CSE) platform. Experiments were conducted on Raspberry Pi 5, which served as a representative testbed for resource constrained CSE devices. Results were gathered using the experimental test bed to compare the performance of RLWLEKEM with some selected cryptography protocols such as code based cryptography (CBC), and classical cryptography algorithms. Performance metrics used for comparison include encryption time, execution time, communication overhead and energy consumption. Experimental results show that RLWLEKEM outperforms the other compared cryptography protocols in terms of time to complete cryptography process, storage/communication overhead and energy consumption and thus was most suitable for deployment as post quantum cryptography key encapsulation mechanism on energy constrained CSE devices.

Keywords

Consumer Electronics, Edge Computing, User Experience, Post-quantum Cryptography, Resource-Constrained Devices, IoT Security, Quantum-Resistant Algorithms

1. Introduction

CSE devices such as wearables, smartphones and smart home assistants, are currently integrated with fifth generation (5G) wireless connectivity and Internet of things (IoT). This makes it possible to intelligently perform data streaming with

*Correspondence: Fagbohunmi Griffin Siji (fagbhume.griffin@abiastateuniversity.edu.ng)

Received: 13 June 2026; Accepted: 8 July 2026; Published: 29 October 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

various other hardware devices as well as control automation in industries [1]. This ability provides improved user experience, efficient and better lifestyle [2]. It is now possible for consumers to have non-interrupted streaming, non-contact accessibility to smart home appliances with updates (real-time) on fitness and health condition irrespective of their location. Due to the different connectivity options available in these CSE devices, the ever increasing demand for enhanced features by consumers and the ever changing technologies, it is becoming difficult to maintain a balance between price and device specifications for manufacturers. Maintaining high level of security for these CSE devices is also a challenge for designers [3]. CSE devices now have embedded in them different technologies that has come to shape the way they are being used [4]. These technologies include migration of CSE communication to 5G and 6G networks, IoT, robotics, autonomous (intelligent) vehicles, virtual reality (VR), drones, augmented realities (AR), and artificial intelligence just to mention but a few [5, 6].

Even though some of these technologies are used in other application areas, their application in CSE devices require a different approach due to the resource constraint nature of embedded systems used for communication in CSE devices. For example the design approach for data communication between different industrial devices may be adapted for use in consumer electronics (CSE) so as to offer satisfactory user experience. This technology adaptability is necessary so that IoT communication with consumer electronics satisfies the requirements for user experience, innovation and user satisfaction. With the recent advancements in quantum computing, it is necessary to design CSE IoT data communication algorithms that are resilient to quantum computing attacks. This is necessary as real life deployment of quantum computers may soon be available, and this may be catastrophic for the current cryptographic techniques used in CSE. The current cryptographic methods used in consumer electronics are classical cryptography techniques, such as elliptic curve cryptography (ECC), and Rivest, Shamir, Adleman (RSA) cryptography. These schemes are susceptible to quantum attacks, thereby exposing the security and confidentiality of CSE devices using these cryptography methods. Shor's algorithm can easily compromise both RSA and ECC algorithms because it can easily factor large primes in finite time, a feat not possible on classical computers [7]. Even though quantum computers have not been fully operational in real network scenarios, the harvest now, decrypt later slogan requires that action must be taken immediately so as to prevent the expected development of quantum computers to take one off guard [8]. Confidential data stored or transmitted by CSE devices can be held in a safe memory and decrypted retroactively when quantum computers become fully operational [9]. In the same vein, as many CSE devices do have a long life span, it is important that cryptography protocols be designed now to protect their security for the future [10]. The need for a secure cryptography algorithm against both classical and quantum computing attacks

have led the US National Institute of Standards (NIST) to develop a standard blueprint for lattice based PQC algorithms [11].

The contributions in this paper are as follows: (i) design a light weight CCA secure lattice based post quantum cryptography key encapsulation mechanism algorithm suited for resource constrained consumer electronics devices. This algorithm is called Ring Learning with Layered Errors Key Encapsulation mechanism (RLWLEKEM), (ii) set up an experimental test bed to show why RLWLEKEM cryptography is best suited for resource constrained CSE devices in comparison to both code based and classical cryptography. (iii) Comparing the performance of the proposed algorithm against other cryptography algorithm techniques using the NIST standardization round 4 key encapsulation procedure as a benchmark. (iv) Test the proposed algorithm on Raspberry Pi 5 test bed which serves as a good representation for resource constrained devices using different performance metrics such as latency, memory consumption and communication of key sizes.

2. Review of Related Literature

There are quite a number of works on post quantum cryptography, some are survey while others are research articles. In most researches present in literature, there is a dearth of post quantum cryptography targeting resource constraint implementation scenario of CSE devices. The performance analysis of post quantum cryptography in non-commutative platform group was discussed with a view determining how effective they were in solving some classical cryptography algorithm problems the analysis were not extended to resource constraint CSE devices [12]. The libogs library was used to provide the minimum performance specification for 80x86 microcomputers based on the NIST post quantum cryptography benchmark [13], however this design cannot also be applied to resource constraint CSE devices. The incorporation of post quantum cryptography (PQC) into TLS 1.3 system was performed on a code based cryptography algorithm, the experimental results from their analysis show the time needed for communication handshake between source and receiver in CSE devices connected to desktop computers [14]. Their experiments did not however test for a full CSE networks i.e. where CSE devices are connected to one another. The hardware implementation of field programmable gate arrays (FPGAs) using very high speed integrated description language (VHDL) on CSE devices was described based on the NIST post-quantum cryptography hardware evaluation [15], however, the design was limited to hardware circuit implementation on CSE devices while the real hardware performance evaluation was not taken into consideration.

The impact of quantum computing threats to classical cryptography algorithms was discussed using post quantum digital certificate, the future direction in protecting data in the post quantum era was also highlighted [16]. The researchers stated

that quantum key distribution (QKD) employs the principle of quantum mechanics to resist quantum computing threats. A code based cryptography (CBC) algorithm was designed based on the recommendations of the NIST's standardization criteria [17]. The authors argued that their CBC algorithm was appropriate in real life systems. A review article was done comprising six types of post quantum cryptography i.e. code based cryptography (CBC), hash based cryptography (HBC), lattice based cryptography LBC, symmetric key cryptography, multiparty computation and identity based cryptography (IBC), this was aimed at comparing the efficiency of the algorithms against quantum based attacks [18]. Amongst these categories of post quantum cryptography, the authors came to the conclusion that lattice based cryptography was the preferred one, and the most consistent among the six categories of PQC algorithms considered. The authors also discussed the difficulties of implementing the PQC algorithms which include implementation of hardware and software, to which they concluded that further research was required. A discussion on the appropriate timing for the implementation switch to post quantum cryptography from the existing classical cryptography was done especially with the continued development of quantum computers [19]. The paper also discussed the various PQC techniques required to protect network devices from quantum based attacks. Methods for combining both classical and post quantum cryptography was also discussed as well as how to properly transit from the existing classical cryptography algorithms used in various organizations to the post quantum cryptography. Experimental setups was designed to test run transiting from classical cryptography to post quantum cryptography. An overview of PQC development's life cycle was provided with a view to ascertaining the efficiency and time completion of a quantum resilient network topology and lastly the standardization requirements stated by various organization which set minimum benchmark for PQC algorithms implementation in different network scenarios was discussed [20]. The authors then discussed future research directions for designing post quantum cryptography for telecommunications network.

The shortcomings of classical cryptography algorithms in the face of quantum based attacks was discussed using the NIST standardization requirements for PQC, a comparison in the performance of various PQC algorithms on various platforms was also carried out [21]. The future research direction of PQC algorithms was discussed such as hardware and software deployment. A survey of classical and post quantum cryptography algorithms were carried out, using various performance metrics such as time from sending data from source to successful decryption by the receiver [22], the level of security and how various categories of data were encrypted in asymmetric key encryption was also discussed in the paper. A comprehensive survey of various types of adversarial threats and countermeasures in post quantum cryptography algorithm was carried out with a view to satisfying the requirements in NIST's standardization procedure Countermeasures to the

side channel attacks which pose threats to post quantum cryptography was discussed [23].

A review article discussed public key cryptography algorithms and digital signature schemes with a view to addressing how they meet the requirements stated in NIST's PQC standardization procedure, while thereafter proposing the best methods amongst the aforementioned that is resilient to quantum threats [24]. A survey on code based cryptography techniques as well as to how they are ideal as post quantum cryptography protocols in certain network scenarios were highlighted [25]. In the paper, the advantages presented by CBC in overcoming quantum threats were discussed. The application of the CBC schemes in IoT device networks was discussed in addition to its deployment challenges. Lastly the future research direction of CBC schemes in PQC were highlighted.

A comprehensive review of hardware circuit and design for post quantum cryptography was carried using basic arithmetic operations and features of specific algorithm for post quantum cryptography in code based cryptography [26]. A review article on different classifications of post quantum cryptography together with their standard primitives was carried out with a view of assessing their applicability in addressing quantum threats in blockchain IoT networks. [27]. A review article was done on post quantum cryptography in unmanned aerial vehicle (UAV) networks together with their resilience to adversarial threats and the requirements of UAV networks in achieving protection from adversarial threats [28].

A review of latt3ce and code based key encapsulation mechanism implementation on IoT ESP32 and Raspberry Pi network was done in resource constrained devices [29]. Their work described the challenges of PQC implementation on CSE devices, but their design was not tested on the requirements stated in NIST's PQC standardization procedure or on its secure digital signature. The requirement specification for post quantum cryptography key encapsulation mechanism on resource constrained Cortex M4 microcontrollers was developed under chosen ciphertext attack (CCA) [30], however, they did not test the security of the digital signature nor performance on other CSE testbed, such as Raspberry Pi. A comparison of post quantum cryptography digital signature algorithms using performance metrics such as network performance and energy consumption was done with the aim of determining the appropriateness of each algorithm in given network scenario, however the Key encapsulation mechanisms in the application scenario was not tested [31]. The scalability of post quantum cryptography digital signature with varying message sizes was tested on desktop computers [32]. however their test did not include resource constrained CSE devices.

3. Methodology

The protocol proposed in this paper is Ring Learning with Layered Errors Key Encapsulation Mechanism (RLWLE-KEM). It is a variant of the Ring Learning with Errors problem

RLWE problem [33], it should be noted that the original LWE problem was not efficient due to its use of large matrices, while the Ring Learning with Errors (RLWE) consumed large memory space due to its high computational requirements. The aim of this paper is to improve on the memory requirements in RLWE so as to make Ring Learning with Layered Errors key encapsulation mechanism (RLWLEKEM) less computationally intensive while still retaining the security of the protocol against quantum attacks. The concept of lattice will first be described in this section as the hardness of RLWLEKEM is based on the difficulty of lattice based problems.

Lattice based cryptography rely on the perception of the difficulty in solving problems defined by lattices. A lattice can be defined as the repeated pattern of points in either 2 or higher dimensional space. It is a discrete addition of the basis of a lattice. The basis of a lattice is defined as the set of its unit vectors, such that every lattice vector can be defined as integer linear combination of its basis vector set, $S = \{s_1 \dots s_k\}$.

$$L_x(S) = \{\sum_{i=1}^k z_i s_i \mid z_i \in \mathbb{Z}\} \quad (1)$$

There are two basic hard problems under Lattice vectors

Shortest Vector Problem (SVP): This problem is defined by the following, Calculate the shortest non zero vector $n \in M$ in a lattice L . Calculating the correct or estimated solution to this problem is known to be difficult.

Learning With errors (LWE): This idea was invented by Regev [34]. It is defined as follows, In a matrix with the following inputs $a \in \mathbb{Z}_q^n$, a secret vector $s \in \mathbb{Z}_q^k$ and a vector 'b' computed as $b = as + e \pmod{q}$, where e is an error of a small dimension derived from a discrete Gaussian distribution $\mathcal{X}$, the distribution in LWE is defined as $A_{s,\mathcal{X}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ which is given as the set of tuples (a_i, b_i) . In order to solve LWE problem, the solution must be able to differentiate between samples generated from $A_{s,\mathcal{X}}$, from that generated uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. The search LWE problem is about computing the value for s (the secret vector) from a given number of samples selected from $A_{s,\mathcal{X}}$. LWE has been shown to be at least as difficult as worst case lattice problems such as GapSVP.

3.1. Rings and Distributions

A ring R is defined as $\mathbb{Z}[X]/(X^n + 1)$, while R_q is defined as $\mathbb{Z}_q[X]/(X^n + 1)$, where $n = 2n' - 1$, such that $X^n + 1$ is the $2n'$ cyclotomic polynomial. In this paper, n , n' and q have the values of 256, 9 and 7681 respectively.

Compression and Decompression: A compressed function $\text{Compress}_q(x, e)$ is a function which takes an element $x \in \mathbb{Z}_q$, and produce integer outputs in $\{0, 1, \dots, 2e - 1\}$, where $e < \lceil \log_2 q \rceil$. A function Decompress_q is defined as as shown in equation (2)

$$x' = \text{Decompress}_q(\text{Compress}_q(x, e), e) \quad (2)$$

| Where x' is an element whose value is close to x , i.e.

$$|x' - x \pmod{q}| \leq Bq = \lfloor \frac{q}{2e+1} \rfloor \quad (3)$$

These requirements are satisfied by the following functions in equations (4) and (5)

$$\text{Compress}_q(x, d) = (2e/q) \cdot x \pmod{+ 2e} \quad (4)$$

$$\text{Decompress}_q(x, e) = \lfloor (q/2e) \cdot x \rfloor \quad (5)$$

If x' is a function of x as stated in Equation (3), then if $x = \mathbb{Z}_q$ is chosen randomly, the distribution in equation (6) will be uniform over the integers of value with highest magnitude Bq .

$$x' - x \pmod{+ q} \quad (6)$$

Actually, the distribution has the same weight over integers whose highest magnitude is $Bq - 1$, while it has a smaller weight on integer whose magnitude is Bq .

When the functions Compress_q or Decompress_q are used with $x \in \mathcal{R}_q$ or $x \in \mathcal{R}_q^k$. The parameters are used separately on each function. The Compression and Decompression functions are defined in this paper because they are used to remove some bits with lower order from the public key and the ciphertext. These are necessary in order to reduce the cryptographic key sizes (p_k, s_k) , a condition that is required for successful implementation of the RLWLEKEM on resource constrained devices. It should be noted that these operations do not have much implication on the accuracy in the decryption process. As will be shown later in the algorithm pseudocode, the Compress_q function is also used to determine the decryption output to be either 1, whenever the value of the expression $v - s^T U$ is nearer to $\lfloor q/2 \rfloor$ than to 0 or decrypt to zero otherwise.

3.2. Ring Learning with Layered Errors IND-CPA Secure Encryption

If k, et, eu, ev are positive integer parameters, also assume as stated earlier that $n = 512$. Assume $N = \{0, 1\}^{512}$ represent the message space, where all messages $n \in N$ are viewed as polynomial in R , their coefficients can be in the range $\{0, 1\}$. The public key encryption is defined as follows: RLWE CPA = (KeyGen, Enc, Dec) as shown in algorithms 1 to 3. It should be noted that ciphertexts are written as $(u, v) \in \{0, 1\}^{256ku} \times \{0, 1\}^{256eu}$.

Algorithm 1 RLWLEKEM. CPA. KeyGen (): key generation

1: $p, q \leftarrow \{0, 1\}^{512}$ 2: $A \sim R_q^l \times^l$: =Bob (p) 3. $(s, e_1, e_2) \sim \beta_q^k \times \beta_q^k \times \beta_q^k$: =Bob (q) 4: $t = \text{Compress}_q(As + e_1 + e_2, f_t)$ 5: return $(pk = (t, p), sk = s)$. The flowchart for RLWLEKEM. CPA key generation is shown in Figure 1.

Algorithm 2 RLWLE. CPA. Enc ($pk = (t, \rho)$, $m \in M$): encryption

1: $r \leftarrow \{0, 1\}^{512}$ 2: $t := \text{Decompresssq}(t, f_t)$ 3: $A \sim R_q^{l \times l}$: =Bob (p) 4: $(r, e_1, e_2, e_3) \sim B_{\gamma}^k \times B_{\gamma}^k \times B_{\gamma} \times B_{\gamma}$: =Bob (r) 5: $u := \text{Compresssq}(A \cdot r + e_1 + e_2, f_u)$ 6: $v := \text{Compresssq}(t \cdot r + e_3 + \lfloor \frac{q}{2} \rfloor, m, f_v)$ 7: return $c := (u, v)$

Algorithm 3 RLWLEKEM. CPA. Dec ($sk = s$, $c = (u, v)$): DECAP

$u := \text{Decompresssq}(u, d_u)$ 2. $v := \text{Decompresssq}(v, d_v)$ 3. return $\text{Compresssq}(v - s^T u, 1)$

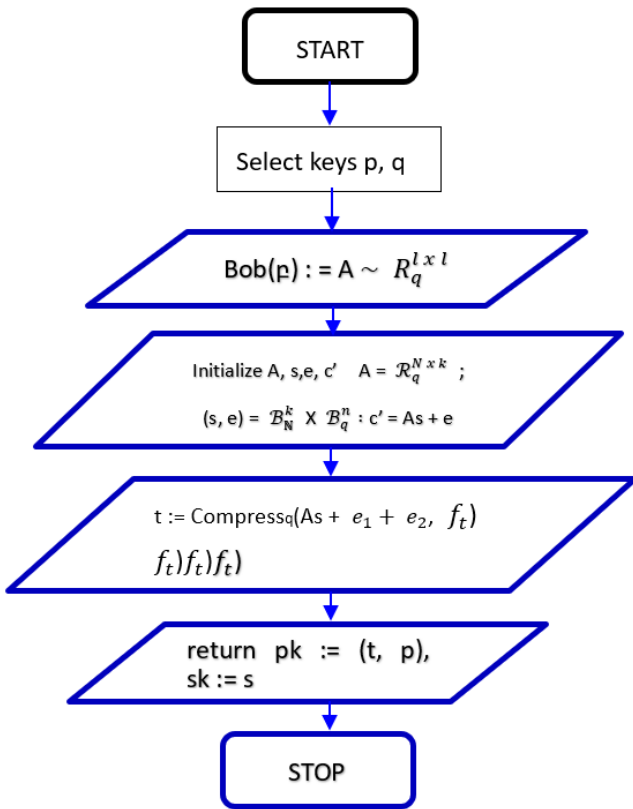


Figure 1. RLWLEKEM. CPA key generation.

While it can be argued that the true shape of data distribution has no effect on the hardness of Ring learning with layered errors key encapsulation mechanism, it will be shown, using a simple Rényi divergence based analysis [34], that it is possible to substitute β_v with an n dimensional rounded Gaussian distribution with standard deviation of $\sqrt{\eta/2}$.

3.3. Ring Learning with Layered Errors

In Ring Learning with Layered Error (RLWLE) problem, a polynomial ring denoted by $R_q = \mathbb{Z}_q[x]/(f)$, where (f) is an irreducible polynomial. is defined as that in which each of the coefficients of each row of the polynomial is a cyclic shift of the rows above it. It has degree n with its coefficients modulus q (i.e. the coefficients wraps around the integer q), a secret

polynomial $s(x)$ is dot multiplied with polynomial $a_i(x)$ whose coefficients are uniformly selected from $\mathbb{Z}_q$. Two error polynomials $e_i(x)$ and $e_j(x) \in R_q$, are generated, the coefficients of the error polynomials are derived from $\mathcal{X}$. The following computation based on polynomial tuples $(a_i(x), b_i(x))$ is now done $b_i(x) = a_i(x) \cdot s(x) + (e_i(x) + e_j(x)) \in R_q$. The ring learning with layered errors (RLWLE) distribution has two difficult scenarios. these are basically what the hardness in RLWLE is built around. The first is the decision ring learning with layered errors problem, which is to differentiate between the $(a_i(x), b_i(x)) \in \mathbb{R}_q^k \times \mathbb{R}_q$ samples where $a_i(x) = \mathbb{R}_q^k$ is uniform, while $b_i(x) = a_i^T s + e_1(x) + e_2(x)$ also $s = \mathcal{B}_q^k$ and $e_1, e_2 = \mathcal{B}_n$, from the same number of samples whose coefficients are generated uniformly from $(a_i(x), b_i(x)) \in \mathbb{R}_q^k \times \mathbb{R}_q$. The second is the search ring learning with layered errors problem is to compute the value of the secret polynomial $s(x)$ from a finite number of samples selected from the ring learning with layered errors {RLWLE} distribution.

Moreover for an algorithm A , the expression in equation (7) are defined as the probability with which an adversarial node can successfully differentiate between a RLWLE generated samples from a randomly generated samples.

$$\left| \Pr c' = 1: A = \mathcal{R}_q^{N \times k}; (s, e) = \mathcal{B}_N^k \times \mathcal{B}_q^n; c' = As + e; c' = A(a, b) \right| - \Pr [c' = 1: A = \mathcal{R}_q^{N \times k};$$

$$c = \mathcal{R}_q^N; c' = A(A, c)] \quad (7)$$

A simple ring learning with layered errors is illustrated in Table 1.

Table 1. Parameters of ring learning with layered errors.

S/N	Polynomial	Notation	Illustration
1	$5 + 2x + 12x^2 + 10x^3$	A	Random polynomial
2	$7 + 8x + 11x^2 + 8x^3$	S	Secret polynomial
3	$0 + x^2 - x^3$	e_1	Secret/layered noise
4	$0.3 - x + x^2 + x^3$	e_2	Secret/layered noise
5	$11 + 9x + 12x^2 + 6x^3$	B	

The search Ring Learning With layered Errors is to compute serial number 2 given serial numbers 1 and 5. In Table 1, the polynomial ring is $\mathbb{Z}_{13}[x]/(x^4 + 1)$. The coefficients of the polynomial is modulo 13. $(x^4 + 1)$ is an irreducible polynomial, it should be noted that the order of the irreducible polynomial is always a power of 2.

In Ring Learning with Layered Errors (RLWLE), shown in

Table 1, the modification made to the original RLWE problem is that two error polynomials $e_1(x)$ and $e_2(x)$ are cascaded on one another to form a layered network of errors. The aim of this is to increase the difficulty in computing the secret vector s , while at the same time reducing the complexity in the key generation /encryption and decryption stage of the algorithm. It is argued here that the cascaded (layered) errors will cause more error ripples in the polynomial equations, which makes it extremely difficult for any oracle to generate, thereby enhancing the difficulty of computing the secret polynomial $(s(x))$. This also makes it possible to reduce the complexity in the key generation /encrypting and decryption stage of the algorithm (the increased complexity in the polynomial equations enables a simpler design for the key generation stage). This reduction/simplification is required to reduce the cryptography key size for secured communication, which makes the algorithm suited for resource constrained consumer electronics devices. This ultimately enhances the resilience of RLWE to quantum based attack. The Gaussian elimination method that could be used to solve the original polynomial equations without errors will now be more complex as the layers of errors cause the addition and multiplication of rows to increase the numbers of errors in each polynomial more than

in the case of the original RLWE. This results in considerable error in the final reduced polynomial. It should be noted that the layered errors in RLWE does not in any way add to the complexity of the original RRLWE problem, as it involves only addition, however it increases the difficulty to decrypt as compared to the original RLWE problem.

The short integer solution that can be applied to LWE will not work with LLWE equation (3) because the assumption that the error range $C \ll q/2$ does not hold. The addition of cascaded errors increases the range of error range C . This adds to the difficulty in solving RLWE problem.

Correctness of the RLWEKEM: The correctness of the RLWEKEM will be shown below. Parameters will be selected in section 3.5 which will make the decryption error negligible, in other words RLWEKEM. CPA is $(1 - \delta)$

Theorem 1. Assume k is a positive integer parameter. Let s, e, r, e_1, e_2, e_3 be random variables having the same distribution as in algorithms 1 and 2. In addition let $c_j = \psi_d^k, c_u = \psi_{d_u}^k, c_v = \varphi_{d_v}$ be distributed as shown below:

Assume ψ_d^k to be the following distribution over $\mathbb{R}$

Select $y = \mathbb{R}^k$ uniformly and randomly. 2. Return $(y - \text{Decompress}_q(\text{Compress}_q(y, d), d) \bmod q$

$$\text{Now } \delta = \Pr [\| e^T r + e_3 + c_v - s^T e_2 + c_t^T r - s^T e_1 + c_t^T r - s^T c_u \|_\infty \geq [q/4]] \quad (8)$$

This shows that RLWEKEM. CPA is $(1 - \delta)$ correct.

A python script was written to calculate a tight upper bound on δ , the parameter that was set for the proposed algorithm

gives δ as 2-285.

Proof: In order to determine the value of t from algorithm 2, the following is done

$$t = \text{Decompress}_q(\text{Compress}_q(As + e_1 + e_2, dt), dt) = As + e_1 + e_2, c_u \quad (9)$$

For some $c_t \in \mathbb{R}_k$. The value of u in algorithm 3 will be

$$u = \text{Decompress}_q(\text{Compress}_q(A^T r + e_1 + e_2, d_u), d_u) = A^T r + e_1 + e_2, c_u, \quad (10)$$

For some $c_u \in \mathbb{R}^k$. The value of v will be

$$v = \text{Decompress}_q(\text{Compress}_q(t^T r + e_2 + e_3 + [q/2].m, d_v), d_v) = t^T r + e_2 + e_3 + [q/2].m + c_v = As + e + c_t^T r + e_2 + e_3 + [q/2].m + c_v + c_t^T r. \quad (11)$$

Now for some $c_v \in \mathbb{R}$ in equation 11 it can be assumed that the values of c_t, c_u and c_v are distributed as defined in ψ in the theorem. Statement This is so because, the parameters c_t, c_u and c_v are in the form shown in equation (12)

$$(y - \text{Decompress}_q(\text{Compress}_q(y, e), e) \bmod q \quad (12)$$

Where y is a pseudo-random variable which determines the hardness of RLWEKEM. After applying equation (18), the expression in equation (13) is obtained.

$$V - s^T u = e^T r + e_3 + e_3 + [q/2].m + c_v + c_t^T r - s^T e_1 - s^T e_2 - s^T c_u \quad (13)$$

Now if $\| e^T r + e_3 + c_v + c_t^T r - s^T e_1 - s^T e_2 - s^T c_u \|_\infty < [q/4]$, then, the following expression, $v - s^T u = w + [q/2].m$ will be obtained, where $\| w \|_\infty < q/4$. Given

the definition $m' = \text{Compress}_q(v - s^T u, 1)$, it can be shown that

$$q/4 \geq \| v - s^T u - [q/2].m \|_\infty = \| w + q/2.m - q/2.m' \|_\infty \quad (14)$$

Using the triangle Inequality together with the fact that $\|w\|_\infty < [q/4]$, results in the expression in equation (15)

$$\left\| \left\lfloor \frac{q}{2} \right\rfloor \cdot (m - m') \right\|_\infty < 2 \cdot [q/4] \quad (15)$$

In which case, all odd values of q will result in $m = m'$, this proves the correctness of RLWLEKEM. CPA.

3.4. The CCA Secure RLWLEKEM

Assume $G: \{0, 1\}^* \rightarrow \{0, 1\}^{2 \times 512}$ and $H: \{0, 1\}^* \rightarrow \{0, 1\}^{512}$ be hash functions. Let the public key key encapsulation mechanism $RLWLE = (\text{KeyGen}, \text{Encaps}, \text{Decaps})$ be as shown in Algorithms 1, 4 and 5, where KeyGen was same as the one used in RLWLE, CPA from section 3.5, the only difference is that sk includes $pk = (t, Y)$ in addition to a secret 512-bit random value y . This was arrived at by using a KEM variant with implicit rejection of the Fujisaki-Okamoto transform to the RLWLE. CPA encryption scheme [35]. It should be noted that the randomness of r was made explicit in the Enc algorithm. it should be noted here that RLWLE. Decaps does not output $\perp$, in place of this, whenever the encryption fails, a pseudorandom key $K := H(z, d)$ where z is a random secret seed is output

Algorithm 4 RLWLE. Encaps ($pk = (t, Y)$)

1. $N = \{0, 1\}^{256}$ 2. $(K, r) := G(H(PK), n)$ 3. $(u, v) := RLWLE; CPA. Enc((t, Y), n; r)$ 4. $c := (u, v)$ 5. $K := H(K, H(c))$ 6. return (c, K)

Algorithm 5 RLWLEKEM. Decaps (s, z, t, Y), $c = (u, v)$)

1. $n' := RLWLEKEM. CPA. Dec(s, (u, v))$ 2. $(K', R') := G(H, pk), n'$ 3. $(u', v') := RLWLEKEM. CPA. Enc((t, Y), n; r')$ 4. If $(u', v') = (u, v)$ then 5. return $K := H(K, H(c))$ 6. Else 7. return $K := H(z, H(c))$ 8. endif

Correctness of RLWLEKEM. CCA: If RLWLEKEM. CPA is $(1 - \delta)$ correct and G is a random oracle, then RLWLEKEM is $(1 - \delta)$ correct [36].

Security of the Algorithm. The statement stated above shows that RLWLEKEM CCA is secured when the hash functions G and H are designed as random oracles.

Theorem 3. Given any classical adversary E , that queries

$$Adv_{RLWLEKEM}^{CCA}(E) \leq 8q_t^2 \times \delta + 4q_t \times \sqrt{Adv_{RLWLEKEM}^{PR}}(F). \quad (18)$$

However, the security bound shown in equation (18) is non-tight, which means that it can only be used as an asymptotic representation of RLWLEKEM CCA security in a quantum random oracle model. A tightly secure key encapsulation technique was employed to get a tight security bound in the quantum random oracle model [38], this implies that a deterministic variant of RLWLEKEM. CPA is sparse pseudorandom in the quantum random oracle model. It should be noted that the randomness r employed in the deterministic variant of RLWLEKEM. CPA denoted as RLWLEKEM. CPA is generated in a deterministic way from a message n using $r := G(n)$.

random oracle H and G at most q_k times and makes q_f queries to the decryption oracle, then the presence of an adversary F can be deduced in such a way that

$$Adv_{RLWLE}^{CCA}(E) \leq 4 Adv_{RLWLE}^{CPA}(F) + q_k \cdot \delta + \frac{4q_k}{2^{256}}. \quad (16)$$

It is also shown here that, an alternative security reduction is possible from the weaker notion of ONE-WAY CPA security of RLWLEKEM. CPA, however this variant is not as accurate as RLWLEKEM. CCA as the multiplicative factor of q_k is not included.

Now considering the security of RLWLEKEM. CCA in a quantum oracle model, it can be shown that RLWLEKEM is IND-CCA secure in a quantum random oracle model, as long as RLWLEKEM. CPA is ONE-WAY CPA secure and sparsely pseudo-random [37]. It should be noted that sparse pseudo-randomness is a little more stronger security wise than IND-CPA security it states that (i) a ciphertext generated from the algorithm is pseudo-random (i.e. it cannot be distinguished computationally from a random high entropy bit string) and (ii) there is a high probability that a random bit string cannot be generated from RLWLEKEM ciphertext. The proof in theorem 2 shows that RLWLEKEM. CPA' (i.e. RLWLEKEM. CPA where t is not compressed) is tightly pseudo-random under the hardness assumption of RLWLEKEM. In fact, its pseudo-randomness advantage is bounded by

$$Adv_{RLWLEKEM.CPA}^{PA}(E) \leq 3 \cdot Adv_{K+1,K,\lambda}^{Kyber.CPA}(F). \quad (17)$$

It can be further shown that the same bound is applicable for RLWLEKEM. CPA. It should be noted however that the sparseness property is roughly achieved for RLWLEKEM. CPA, because the ciphertexts generated from the algorithm is a sparse subset of the ciphertext space. $\{0, 1\}^{(512kd_u)} \times \{0, 1\}^{d_v}$.

Theorem 4. Given any quantum adversary E , which makes at most q_t queries to a random query oracle L and M , while it makes at most q_f queries to the decryption oracle, a quantum adversary F exist such that

In the classical variant of the random oracle model, the assumption stated here is tightly bound by the IND. CPA security in the RLWLEKEM. CPA, however the reduction is non-tight in the quantum random oracle model, this was why the term $\sqrt{Adv_{RLWLEKEM}^{PR}}(F)$ was included in theorem 4.

4. Experimental Setup

The performance analysis in this paper was performed on Raspberry pi 5. The focus of this paper is comparing energy

consumption from the various cryptography protocols considered. Raspberry Pi 5 platform was used for the experiments, using standardized cryptographic libraries to achieve a fair basis for comparison. The purpose is to ascertain the suitability of RLWLEKEM algorithm (a lattice based key encapsulation mechanism) as a preferred scheme for post quantum cryptography in resource constrained consumer electronics devices. The specification of the Raspberry Pi 5 is explained below:

Raspberry Pi 5: This represents resource constrained consumer electronics with the following specification; 2.4 GHz quad-core ARM Cortex-A76 CPU, 16GB LPDDR4 XRAM, 64 bit SoC, running Kali Linux operating system (64 bit Debian bookworm). This platform represents resource constrained consumer electronics devices e.g. smart homes, automotive devices, smartphones and set up boxes. The detailed specification of the experimental setup is shown in Table 2.

Table 2. Specification of Raspberry Pi 5.

Component	Specification
Processor	Quad core Cortex a76 (arM v8) 64-bit SoC @ 2.4 GHz
RAM	4GB LPDDR4
Storage	SSD m. 2.256GB
Operating System	Kali Linux (64-bit)
Deployment use case	Embedded System IoT security (computer Electronics)

4.1. Libraries and Benchmarking Frameworks

The framework used for benchmarking the simulation was Python OS 6. The source code used for analysing the scripts and benchmarking can be found at <https://github.com/griffin/analyze>.

The RLWLEKEM lattice cryptography to be used in this paper will be run on liboqs (quantum safe library) an open source C library (oqs/oqs.py and oqs/rand.pv on a python 0.8.0 module wrapper). The implementation of the RLWLEKEM lattice cryptography will be performed with these libraries. The other post quantum cryptography algorithms used in comparison was implemented on liboqs open source C library (version 0.8.0 using 'oqs-python' 0.8.0 wrapper). It should be noted that the algorithm implementation proposed in this paper is compatible with the NIST standardization process. The compilation of liboqs on the Raspberry Pi 5 platform was performed on the Linux/ARM64, using standard toolchains (xcode-select on Linux OS), Pi 4/400/5/CM4 using aarch64-rpi 3-linux-gnu.

For the classical algorithms to be used in performance comparison, the evaluation was measured using python cryptography (version 3.8+) library which provides high level recipes

application programming interface for easy symmetric encryption, available as wrapper around C libraries i.e. OpenSSL (version 7.3.11+). These libraries are available on the Raspberry Pi platform and was used for the implementation of Rivest, Sharmir, Adleman, (RSA) and Elliptic Curve Diffie Hellman Cryptography (ECDH),

The following Python timer modules was used for measuring the execution times of the cryptography algorithm used in this paper,

- 1) time.perf_counter(): this provides high resolution performance counter.
- 2) Time(): this returns the current time in seconds since the last epoch.

In order to maintain a stable statistical reading, each operation was repeated 100 times while a lower number of repetition was used for algorithms with long execution times such as classic McEliece key generation. The first five runs will not be used in the simulation and will therefore be regarded as start-up values, while the average and standard deviation of the remaining runs will be computed.

4.2. Performance Metrics

The following performance metrics particularly suited to consumer electronics devices was used in the analysis:

- 1) Time of Execution: The mean and standard deviation of the time consumed by each algorithm will be computed in milliseconds. Execution time is defined as the time elapsed between the sender's encrypted data to the time the receiver successfully decrypts the cipher text.
- 2) Communication costs: This refethes overhead (in bytes) incurred during data transmission from sender to the destination. These include key sizes (public and private), shared secret key sizes and computational overhead.
- 3) Trade-offs in Resource requirement: This will be used to compare and analyse the computational cost (time to complete an operation) with the communication / storage overhead (cryptography key / cipher text). This is necessary to compare the relationship of algorithm performance efficiency to changing communication costs and message sizes especially as it concerns resource constrained consumer electronics devices.

4.3. Results and Analysis

The performance of the RLWLEKEM protocol proposed in this paper was compared with other state of the art protocols using the performance metrics outlined in section 4.2. The analysis was with particular reference to how these protocols are suited for use as post quantum cryptography in resource constrained consumer electronics (CSE) devices. The protocols used for comparisons were selected from the NIST standardized protocols, i.e. (ML-KEM, ML-DSA, BIKE, McEliece and some classical cryptography algorithms such as RSA and ECDH), The Raspberry Pi 5 (8GB) was used as the

platform for comparing these protocols as it is representative of modern resource constrained devices used as benchmark for this paper. This is due to its improved performance over its predecessors i.e. Raspberry pi 4.

The execution times for key encapsulation mechanism operations is shown in Table 8. From the table it can be seen that lattice based cryptography key encapsulation mechanism protocols outperform the code based schemes by about 65% in terms of key decapsulation, and about 15% in terms of key encapsulation and 21% in terms of key generation. Similarly

lattice based key encapsulation mechanism protocols outperform classical key encapsulation mechanism algorithms by 34%, however the performance of RSA was particularly poor in terms of key generation due to many iterations involved in its key generation before arriving at a given key pair for each round of operation. The ECDH key encapsulation was however the most efficient in terms of total operation time (shortest time) and lowest communication overhead, however it is not resistant to quantum attacks.

Table 3. Performance of Key Encapsulation Mechanism and Cryptography key sizes.

Algorithm	Family	Secr lev	Key Gen (ms)	Encap (ms)	Decap (ms)	Pub key (bytes)	Cipher (bytes)	Share key (bytes)
BIKE L1	Code based	1	14.49	16.24	15.23	1.542	1574	32
Classic-Mc-Eliece 348864f	Code based	1	37.72	21.03	28.72	261122	97	32
HQC 128	Code based	1	25.33	27.67	36.01	2250	4435	64
BIKE L3	Code based	3	51.04	37.68	15.05	3084	3116	4435
Classic-Mc-Eliece 460896F	Code based	3	91.04	36.06	42.58	564161	157	156
HQC 192	Code based	3	40.97	37.94	22.91	4524	8980	64
BIKE L5	Code based	5	119.35	48.68	61.06	5124	5156	32
Classic-Mc-Eliece 8192128f	Code based	5	185.94	43.12	85.11	1357826	210	32
HQC 256	Code based	5	92.42	43.56	45.29	7247	14422	64
FrodoKEM 640 AES	Lattice	1	21.04	30.27	40.25	9618	9722	16
FrodoKEM 976 AES	Lattice	3	215.75	210.51	221.49	15634	15745	24
FrodoKEM 1344 AES	Lattice	5	370.75	250.87	200.68	21521	21634	32
ML-KEM 512	Lattice	1	47.98	50.98	51.98	801	770	32
ML-KEM 768	Lattice	3	48.22	50.22	51.22	1185	1089	32
ML-KEM 1024	Lattice	5	48.01	51.01	51.11	1570	1569	32
RLWLEKEM	Lattice	5	47.95	50.25	50.97	1286	1268	32
ECDH-256	Classical	1	9.42	12.06	11.06	92	92	32
ECDH-384	Classical	3	11.62	14.52	14.50	122	122	48
ECDH-521	Classical	5	13.90	14.48	16.47	160	160	66
RSA-2048	Classical	1	76.86	12.07	15.74	295	258	32

Algorithm	Family	Secr lev	Key Gen (ms)	Encap (ms)	Decap (ms)	Pub key (bytes)	Cipher (bytes)	Share key (bytes)
RSA-3072	Classical	3	182.77	13.12	13.85	423	386	32
RSA-4096	Classical	5	418.15	13.18	21.97	551	514	32

Table 3 also shows the comparison on the resources required by the various cryptography algorithms, i.e. time for key encapsulation mechanism (KEM) among the cryptographic protocols considered. From the table, it will be observed that KEM usually results in the slow down of the encryption process, howbeit at different rates. KEMs result in slow down of the lattice based key encapsulation mechanism by about 35 times, while the reduction in code based key encapsulation mechanism (generation/encapsulation) was about 40 times, however the reduction caused by decapsulation was about 10 times, therefore it wasn't evenly spread as in both lattice and code based KEMs. The reduction caused by KEMs in FrodoKEM was substantial i.e. about 100 times. Classical KEMs on the other hand exhibited lower slow down of about 15 times. It should be noticed that KEMs results in increased latency of protocol operation, it is therefore necessary to cut down on the time required for this, as it causes more energy to be consumed by the protocol, which isn't good for resource constrained consumer electronics devices. On the average, lattice based cryptography had be best overall performance in terms of KEMs overhead, classical KEMs has better perfor-

mance but are grossly inadequate in securing data from quantum threats.

Detailed performance comparison show that RLWLEKEM lattice cryptography outperforms other lattice based key encapsulation mechanism protocols i.e. ML-KEM by 18%, while it outperforms the FrodoKEM variants by 45%. The table also shows that RLWLEKEM outperforms ML-KEM in terms of time to complete cryptography process (i.e. key generation + key encapsulation + key decapsulation) by 28%, while It outperforms classical KEM (ECDA AND RSA) by 74% The RLWLEKEM lattice KEM outperforms FrodoKEM algorithm in terms of time to complete cryptography process by 38% These results show that apart from lattice based cryptography, other types of Post Quantum Cryptography protocols such as Code based PQC, and FrodoKEM PQC are not ideal for resource constrained devices such as consumer electronics. Also the RLWLEKEM lattice cryptography showed improved performance over other lattice based cryptography protocols due to its reduced complexity in key generation and mapping of the errors to the message matrix as explained in section 3.2.

Table 4. Total cryptography operation time.

Algorithm	Family	Secr lev	Key Gen (ms)	Encap (ms)	Decap (ms)	Cryptography time Total (ms)
BIKE L1	Code based	1	14.49	16.24	15.23	45.96
Classic-Mc-Eliece 348864f	Code based	1	37.72	21.03	28.72	87.47
HQC 128	Code based	1	25.33	27.67	36.01	89.01
BIKE L3	Code based	3	51.04	37.68	15.05	103.77
Classic-Mc-Eliece 460896F	Code based	3	91.04	36.06	42.58	169.68
HQC 192	Code based	3	40.97	37.94	22.91	101.82
BIKE L5	Code based	5	119.35	48.68	61.06	229.09
Classic-Mc-Eliece 8192128f	Code based	5	185.94	43.12	85.11	314.17
HQC 256	Code based	5	92.42	43.56	45.29	181.27
FrodoKEM 640 AES	Lattice	1	21.04	30.27	40.25	91.56
FrodoKEM 976 AES	Lattice	3	215.75	210.51	221.49	647.45
FrodoKEM 1344 AES	Lattice	5	370.75	250.87	200.68	822.30

Algorithm	Family	Secr lev	Key Gen (ms)	Encap (ms)	Decap (ms)	Cryptography time Total (ms)
ML-KEM 512	Lattice	1	47.98	50.98	51.98	150.94
ML-KEM 768	Lattice	3	48.22	50.22	51.22	148.66
ML-KEM 1024	Lattice	5	48.01	51.01	51.11	150.13
RLWLEKEM	Lattice	5	47.95	50.25	50.97	149.17
ECDH-256	Classical	1	9.42	12.06	11.06	32.54
ECDH-384	Classical	3	11.62	14.52	14.50	40.64
ECDH-521	Classical	5	13.90	14.48	16.47	44.85
RSA-2048	Classical	1	76.86	12.07	15.74	104.67
RSA-3072	Classical	3	182.77	13.12	13.85	209.74
RSA-4096	Classical	5	418.15	13.18	21.97	453.30

In Table 4, the total time (key generation + encapsulation + decapsulation) required by the key encapsulation mechanism operation in comparison to the communication overhead (public key size + ciphertext size) was shown. It should be noticed that in order to optimize the performance of cryptography algorithm for resource constrained devices like CSE, it is pertinent to balance the computational complexity with memory consumption/ communication cost i.e. cryptographic key sizes (public, private). Memory consumption refers to the total memory occupied by the public + private + shared key, while the communication cost refers to the energy expended in transferring the message alongside the cryptography keys needed for the cryptography operation. It should be realized here that the total energy consumed (communication cost) is directly proportional to the combined sizes of the keys. From the table, in terms of public key size, lattice based key encapsulation mechanism protocols performs best, outclassing code based key encapsulation mechanism protocols by 78% and has relatively same performance with classical key encapsulation mechanism algorithms. The performance with the public key sizes was similar to that of cipher text sizes and shared key sizes. It can also be seen that Classic McEliece code based cryptography requires excessive communication/storage size. This is due to the computational requirements needed in its generator matrix with the resulting large public key sizes and slow key generation procedure. Even though the ECDH cryptography shows a comparatively good performance in terms resource usage/ communication/storage overhead, its greatest shortcoming is its being non-resistant to quantum based attacks. However among the lattice based cryptography, RLWLEKEM cryptography has the best performance, improving on ML-KEM by 23%, code based cryptography by 47% classical cryptography by 40% and Mc Eliece by 62%.

From Table 5, it can also be noticed that the higher the NIST level of security for a given algorithm, the higher the computational requirement for communication as well as memory

overhead. This usually translates to higher times required for the cryptography process (key generation + key encapsulation + key decapsulation). In general post quantum key encapsulation mechanism seem to moderately increase with security levels, however, the increase with higher security level for RSA is less dramatic. RLWLEKEM was only defined for NIST security level 5 because it is designed for optimum security in constrained consumer electronics devices.

4.4. Throughput Over Different Levels of Security

Table 4 also shows the performance levels of the cryptography protocols across the various security levels defined by NIST. In this experiment, from the table, it can be seen that all the cryptography algorithms will require more resources to meet up with higher levels of security in the NIST standardization. In other words the time to complete a cryptography cycle will be longer as well as the sizes of the cryptography keys (public and private).

The increase in the time to complete a cryptography cycle was least in RLWLEKEM protocol (with the exception of classical cryptography algorithms), next to it are the lattice based cryptography, and code based cryptography algorithm. The increase in the sizes of the cryptography key and ciphertext sizes is remarkably higher compared to that of cryptography cycle completion. It will be noticed that all the cryptography algorithms incur substantial increase over these parameters. However RLWLEKEM outperforms others. The size scaling percentage of RLWLEKEM was 21%, while for the other lattice based cryptography was 24% and code based cryptography has 32% size scaling. From Table 4, it should be noted that, for CSE devices requiring long lifetime expectancy in handling confidential data, protocols on levels 3 or 5 of the NIST standardization should be favoured. It is pertinent that designers select appropriate trade-off between the need for

high computational and communication efficiency in CSE devices to the level of security that can be attained. This will help

provide a proper balance between operational efficiency and security.

Table 5. Key Encapsulation Mechanism overhead (Public key size vs Ciphertext key size).

Algorithm	Security level	Family	Public key size (bytes) log scale	Ciphertext size (bytes) log scale
ML-KEM 1024	5	Lattice	1026	1106
ML-KEM 768	3	Lattice	963	1034
ML-KEM 512	1	Lattice	876	906
RLWLEKEM	5	Lattice	905	980
RSA 2048	1	Classical	467	446
RSA 3072	3	Classical	563	786
RSA 4096	5	Classical	743	836
ECDH 521	5	Classical	224	252
ECDH 384	3	Classical	149	147
ECDH 256	1	Lassical	94	84
BIKE L5	5	Code based	6326	7042
BIKE L3	3	Code based	4463	5042
BIKE L1	1	Code based	2044	6136
Frodo KEM 1344 AES	5	Lattice	21428	21468
Frodo KEM 976 AES	3	Lattice	10896	14728
Frodo KEM 640 AES	1	Lattice	9879	10146
Classic Mc Eliece 8192128f	5	Code based	100346	348
Classic Mc Eliece 460896f	3	Code based	626348	236
Classic Mc Eliece 348864f	1	Code based	386442	113

4.5. Communication Overhead

In Table 5, a comparison of the public key size with respect to the ciphertext key size was shown. (It should be noted that the public and private key sizes are the same). This was used to demonstrate the communication overhead resulting from the transmission of encrypted messages based on the key encapsulation mechanism procedure employed. The average sizes of both public key and ciphertext sizes required by the different cryptography algorithms was shown in the table. From the table, it can be seen that RLWLEKEM cryptography protocol has comparatively lower public key and ciphertext sizes to other compared cryptography protocols (with the exception of ECDH). The sizes of public key and ciphertext in RLWLEKEM protocol was around 905 bytes and 980 bytes respectively compared to 1.003MB and 348B in level 5 security McEliece. ML-KEM 1024 uses around 1.026 kB and

1.106kB respectively for public key and ciphertext key sizes. FrodoKEM 1344 AES uses around 21.42 kB and 21.468 kB for public key and ciphertext key sizes respectively.

It would be noticed that with the exception of ECDH cryptography protocol, in terms of public key sizes, lattice based cryptography has about 40% improved performance to BIKE, 145% improved performance to Classic MC Eliece protocol and 95% improved performance to Frodo KEM protocol. The performance analysis in terms of ciphertext sizes show that RLWLEKEM outperforms ML-KEM, BiKE and Frodo KEM by an average of 3%, 82% and 128% respectively. ECDH protocol has the best performance together with Classic Mc Eliece protocol, however they are not resilient to quantum attack and has high computational requirements for computing the public/private key sizes respectively. Overall RLWLEKEM has the best combination of public/ciphertext key sizes for optimum performance on a resource constrained CSE devices.

Table 6. Average key encapsulation mechanism across cryptography algorithms.

Algorithm	Family	Type	Key Gen	Encap	Decap
BIKE	Code based	PQC	3214	62.45	367
ML-KEM 1024	Lattice	PQC	1784	24.5	225
Frodo-KEM 1344 AES	Lattice	PQC	1823	22.8	235
RLWLEKEM	Lattice	PQC	1654	18.45	202
Classic Mc Eliece	Code based	PQC	2767	58.25	56.34
HQC 256	Code based	PQC	2854	52.73	32.24
ECDSA-521	ECDSA	Classical	2734	2.56	32.20
RSA-4096	RSA	Classical	2245	2.45	32.20

Table 6 shows the average time required for the key encapsulation mechanism by the different categories of cryptography algorithms. From the table, it will be noticed that lattice based cryptography outperforms code based cryptography in terms of time to complete key generation by an average of 34%, while it betters classical cryptography algorithm by 48%. In terms of time to complete encryption, classical cryptography algorithms returned the best times, outperforming lattice based cryptography by 51% and code based cryptography by 121%. In terms of time to complete decapsulation, classical based cryptography also performs best, outperforming lattice based cryptography by 42% and code based cryptography by 92%. The shortcoming of classical cryptography algorithms is that they are not resilient to quantum attacks and so can't be considered as PQC algorithm.

A detailed analysis involving only lattice based KEM algorithms show that RLWLEKEM outperforms ML-KEM in terms of time to complete key generation by 9%, time to complete encryption by 7% and time to complete decapsulation by 6%. RLWLEKEM also outperforms Frodo KEM 1344 AES in terms of time to complete key generation by 15%, time to complete encryption by 17% and time to complete decapsulation by 14%. It can therefore be seen from the above analysis that RLWLEKEM is the best cryptography option for key encapsulation mechanism in resource constrained consumer electronics devices.

Table 7. Communication overhead performance.

Algorithm	Ciphertext size (bytes)	Public key size (bytes)
Code based	1026	124675
Frodo KEM	21448	22566
ML-KEM	1847	1862
RLWLEKEM	1465	1482

Algorithm	Ciphertext size (bytes)	Public key size (bytes)
RSA	898	874
ECDH	284	292

Table 8. Key encapsulation mechanism performance.

Algorithm	Key Gen (msec)	Encap (msec)	DecapS (msec)
Code based	102	37.04	23.52
Frodo KEM	194.23	119.45	35.87
ML-KEM	48.26	51.05	51.07
RLWLE-KEM	46.65	47.24	47.28
RSA	169.42	12.94	23.76
ECDH	15.57	14.24	15.36

4.6. Overall Cryptography Algorithms Analysis

Tables 7 and 8 provide analysis using only level 5 key encapsulation mechanism protocols used for comparison with the Lattice RLWLEKEM cryptography. The analysis provided here serves as a high level representation of the overall performance of the cryptography algorithms. The comparisons of the various key encapsulation mechanisms using public key and ciphertext sizes on Raspberry Pi 5 is shown in Table 7. Table 8 shows the times required to complete the key encapsulation mechanism process, i.e. key generation, encapsulation and decapsulation. From both tables, it will be noticed that in terms of ciphertext sizes, Lattice RLWLEKEM outperforms ML-KEM and Frodo KEM by 8% and 56% respectively. ECDH had the lowest ciphertext sizes, followed by RSA, but

they are both not resistant to quantum attacks. Code based cryptography algorithm generally has better ciphertext sizes to lattice based, however their shortcoming is in the complexity of generating the public and private key in addition to their very large public/private key sizes.

5. Conclusion

From the results of experiments carried out in this paper, it was noticed that RLWLEKEM and indeed most lattice based cryptography had the best performance in terms of speed of operation i.e. how fast the cryptography key generation was, size of cryptographic keys i.e. public and private, and also it was most secured to quantum based attack. In fact RLWLEKEM cryptography had a slight advantage over NIST standardized lattice schemes such as ML-KEM and ML-DSA in terms of operational speed and efficiency of key encapsulation mechanism. RLWLEKEM also had performance edge in terms of cryptographic key sizes over other lattice based cryptography, mainly due to its simple encryption/decryption procedure. RLWLEKEM is therefore the best option as far as parameters related to deploying cryptography algorithms on resource constrained consumer electronics is concerned.

RLWLEKEM cryptography also had the best results in terms memory consumption and communication overhead. Recommendation for Cryptography Algorithm Deployment for Consumer Electronics.

In summary to the results obtained from the experiments conducted in this paper, the following recommendations are made for the deployment of cryptography protocols in various categories of consumer electronics (CSE) devices. RLWLEKEM provides a balanced performance in terms of energy efficient communication overhead, a moderate computational requirements and memory consumption. These performance characteristics result in lower energy consumption compared to other lattice based and PQC KEM algorithms. This ultimately leads to enhanced life time expectancy of the batteries used in most resource constrained electronics devices.

In future, a PQC signature based scheme will be designed, this will be a complement to the RLWLEKEM, designed in this paper, the two will combine to form a full post quantum cryptography algorithm, which will satisfy the requirements of data security, confidentiality and integrity.

Acknowledgments

The author acknowledges Department Computer Engineering, Abia State University for making available the use of their laboratories.

Author Contributions

Fagbohunmi Griffin Siji: Conceptualization, Resources

Data Availability Statement

The Data for this work can be gotten from the author.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Javeed D, Saeed MS, Ahmad I, Kumar P, Jolfaei A and Tahir M. (2024) An intelligent intrusion detection system for smart consumer electronics network. *IEEE Transaction Consumer Electronics.*; vol 69, issue 4 pp 906–913.
- [2] Rani S, Gupta D, Garg S, Piran MJ and Hossain MS. (2025) Consumer electronic devices: evolution and edge security solutions. *IEEE Consumer Electronics Magazine.* Vol 11 issue 2 pp 15–20.
- [3] Shruti, Rani S. (2025) Towards security of consumer electronic devices: an advanced attribute-based encryption approach multimodal fog computing. *IEEE Trans Consumer Electronics.* pp 1 - 10.
- [4] Sahu AK, Sahu M, Patro P, Sahu G and Nayak SR. (2024) Dual image-based reversible fragile watermarking scheme for tamper detection and localization. *Pattern Analysis Application.* Vol 26 issue 2 pp 571–590.
- [5] Patro P, Kumar K, Kumar G. S and Sahu A. K. (2024) Intelligent data classification using optimized fuzzy neural network and improved cuckoo search optimization. *Iran Journal of Fuzzy System,* vol 20 issue 6 pp 155–69.
- [6] Patro P, Kumar K, Suresh-Kumar G and Swain G. (2022) Similarity and wavelet transform based data partitioning and parameter learning for fuzzy neural network. *Journal of King Saud University of Computing Information Science.* Vol 34 issue 6 pp 3424–3432.
- [7] Wang C, Li X, Xu H, Li Z, Wang J and Yang Z (2023) Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds. *NPJ Quantum Information.* Vol 8 vol 1 issue 3.
- [8] Yang Z, Zolanvari M and Jain R. (2024) A survey of important issues in quantum computing and communications. *IEEE Communication Survey Tutor.* Vol 25 issue 2 pp 1059–1094.
- [9] Mulholland J, Mosca M and Braun J. (2021) The day the cryptography dies. *IEEE Security Privileges .* vol 15 issue 4 pp 14–21.
- [10] Kaur J and Ramkumar K. (2023) The recent trends 3n cyber security: a review. *Journal of King Saud University of Computing Information Science.* Vol 34 issue 8 pp 5766–5781.
- [11] Ramesh R. K, Dodmane R, Shetty S, Aithal G, Sahu M and Sahu A. K. (2025) A novel and secure fake-modulus based Rabin-cryptosystem. *Cryptography.* Vol 7 issue 3 pp 44- 53.

- [12] Kanwal S and Ali R. (2020) A cryptosystem with noncommutative platform groups, in *Journal of Neural Computer Applications*. Vol 29 pp 1273–1278.
- [13] Sauerwein K, Butler JM, Reed C and Reczek KK. (2023) A NIST scientific foundation review. 2023.
- [14] Baldi M, Santini P and Cancellieri G. (2020) Post-quantum cryptography based on codes: state of the art and open challenges. In: 2020 AEIT international annual conference. IEEE, pp. 1–6.
- [15] Basu K, Soni D, Nabeel M and Karri R, (2021) “Nist post-quantum cryptography-a hardware evaluation study,” *Cryptology ePrint Archive*, 2019. [Online]. Available: <https://eprint.iacr.org/2021/047>
- [16] Raavi M, Chandramouli P, Wuthier S, Zhou X, and Chang S (2022) “Performance characterization of post-quantum digital certificates,” in 2022 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2022, pp. 1–9. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9522179/>
- [17] Septien-Hernandez J, Arellano-Vazquez M, Contreras-Cruz M. A and Ramirez-Paredes J. P, (2022) “A Comparative study of post-quantum cryptosystems for Internet-of-Things applications,” *Sensors*, vol. 22, no. 2, pp. 489 - 498, [Online]. Available: <https://www.mdpi.com/1424-8220/22/2/489>
- [18] Kumar A, C. Ottaviani C, Gill S, and Buyya R, (2023) “Securing the future internet of things with post-quantum cryptography,” *SECURITY AND PRIVACY*, vol. 5, no. 2, pp 200-213, Mar. 2023, _preprint: 2206.10473.
- [19] Bernstein D. J, (2023) “Analyzing the complexity of reference post-quantum software: the case of lattice-based KEMs,” *Cryptology ePrint Archive*, 2023. [Online]. Available: <https://eprint.iacr.org/2023/1924>
- [20] Satria G. B, Agus Y. M, and Mnaouer A. B, (2023) “A comparative study of post-quantum cryptographic algorithm implementations for secure and efficient energy systems monitoring,” *Electronics*, vol. 12, no. 18, pp. 3824 - 3835. [Online]. Available: <https://www.mdpi.com/2079-9292/12/18/3824>
- [21] Daniel C, Benjamin A, Griffith S. K, Winful B, and James D. G (2025) “Performance Analysis and Deployment Considerations of Post-Quantum Cryptography for Consumer Electronics,” in 2025 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2025, pp. 1–7.
- [22] Lakhan A. S, (2023) “A Comparative Study on Post-Quantum Cryptographic Digital Signature Algorithms: Network Performance, Key Robustness, and Energy Consumption.” PhD Thesis, Carleton University, 2023. [Online]. Available: <https://repository.library.carleton.ca/concern/etds/3197xn336>
- [23] Xin G, Han J, Yin T, Zhou Y, Yang J, Cheng X, and Zeng X, (2021) “VPQC: A domain-specific vector processor for post-quantum cryptography based on RISC-V architecture,” *IEEE transactions on circuits and systems I: regular papers*, vol. 67, no. 8, pp. 2672–2684, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9061149/>
- [24] Fritzmann T, Sigl G, and Sepúlveda J, (2020) “RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 239–280, 2020. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/8683>
- [25] Pomerance C, (1998) “A tale of two sieves,” *Notices of the American Mathematical Society*, vol. 43, no. 12, pp. 1473–1485.
- [26] Nielsen M. A and Chuang I. L, (2012) *Quantum computation and quantum information*. Cambridge university press.
- [27] Grover L (1998) “A fast quantum mechanical algorithm for database search,” in *Proceedings of the 30th annual ACM symposium on Theory of computing - STOC '98*. ACM Press, 1998, pp. 212–219.
- [28] Mina M. Z and Simion E, (2022) “Information Security in the Quantum Era. Threats to modern cryptography: Grover’s algorithm,” *Cryptology ePrint Archive*, 2022. [Online]. Available: <https://eprint.iacr.org/2021/1662>
- [29] Lyubashevsky V, Peikert C, and Regev O (2015) “A Toolkit for Ring- LWE Cryptography,” in *Advances in Cryptology – EUROCRYPT 2015*.
- [30] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck J. M, Schwabe P, G. Seiler G, and Stehlé D, (2020) CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM,” in 2020 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 202, pp. 353–367. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8406610/>
- [31] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G and D. Stehlé D, (2022) “Crystals-dilithium: A lattice-based digital signature scheme,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/839>
- [32] Fouque P. A, Hoffstein J, Kirchner P, Lyubashevsky V, Pornin T, Prest T, Ricosset T, Seiler G, Whyte W, and Zhang Z, (2018) “Falcon: Fast-Fourier lattice-based compact signatures over NTRU,” *Submission to the NIST’s post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018. [Online]. Available: <https://www.di.ens.fr/~prest/Publications/falcon.pdf>
- [33] Regev O, (2009) “On lattices, learning with errors, random linear codes, and cryptography,” *Journal of the ACM*, vol. 56, no. 6, pp. 1–40, Sep. 2009.
- [34] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz (2022) A modular analysis of the Fujisaki-Okamoto transformation. *IACR Cryptology ePrint Archive report 2017/604*, 2017. <https://eprint.iacr.org/>
- [35] Shi Bai, Adeline Langlois, Tancrede Lepoint, Damien Stehlé and RonSteinfeld (2020) Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In Tetsu Iwata and Jung Hee Cheon, editors, ASIACRYPT 2015, Part I, volume 9452 of LNCS, pages 3–24. Springer, Heidelberg.

- [36] Eiichiro Fujisaki and Tatsuaki Okamoto (2005) Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, CRYPTO'99, volume 1666 of LNCS, pages 537–554. Springer, Heidelberg.
- [37] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. (2022)Tightly- secure key-encapsulation mechanism in the quantum random oracle model. IACR Cryptology ePrint Archive report 2022/1005, <https://eprint.iacr.org/>
- [38] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. (2021) A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, TCC 2017, Part I, volume 10677 of LNCS, pages 341–371. Springer, Heidelberg.