*International Journal of Applied Science and Engineering*
Citation: *IJASE*: **13**(02): 233-244, December 2025
**DOI:** 10.30954/2322-0465.2.2025.6

RESEARCH PAPER COMPUTER SCIENCE

# Analysis of Ring-LWE and LWR Cryptographic Schemes: Foundations, Implementation, and Performance

**Sandip Kanoo and Prodipto Das***

*Assam University, Silchar, India*

**Corresponding author: prodipto.das@aus.ac.in*

**Received:** 21 Sept., 2025 **Revised:** 03 Nov., 2025 **Accepted:** 22 Nov., 2025

## ABSTRACT

Lattice-based cryptography has emerged as a leading framework for securing communication in the post-quantum era, supported by strong worst-case hardness guarantees. This study focuses on two central constructions, Ring Learning with Errors (Ring-LWE) and Learning with Rounding (LWR), and examines their mathematical foundations, implementation behavior, and performance characteristics. While Ring-LWE benefits from well-established and conservative security reductions, along with broad applicability across cryptographic primitives, its reliance on discrete Gaussian sampling introduces computational overhead and potential side-channel vulnerabilities. In contrast, LWR replaces stochastic noise with deterministic rounding, thereby simplifying implementation and improving efficiency, particularly on resource-constrained platforms, albeit under comparatively newer and less conservative hardness assumptions. Both schemes are implemented and benchmarked across multiple parameter sets, enabling a systematic comparison of key generation, encryption, and decryption costs. The experimental results highlight the complementary strengths of these approaches and suggest that hybrid constructions may effectively combine the strong theoretical guarantees of Ring-LWE with the practical efficiency of LWR.

**Keywords:** Lattice-based cryptography, Ring-LWE, Learning with Rounding, Post-quantum security, Hard lattice problems, Quantum-resistant cryptography

The field of cryptography is undergoing an evolutionary phase, propelled by the dual pressures of surging data security demands and an impending threat from quantum computing. The classical cryptographic schemes, forming a significant portion of today's secure communication infrastructure, are increasingly endangered by the potency of quantum algorithms such as Shor's[1] and Grover's[2]. In this changing scenario, lattice-based cryptography has been one of the front-runners for post-quantum cryptographic

**How to cite this article:** Kanoo, S. and Das, P. (2025). Analysis of Ring-LWE and LWR Cryptographic Schemes: Foundations, Implementation, and Performance. *IJASE.*, **13**(02): 233-244.

**Source of Support:** None; **Conflict of Interest:** None

solutions because it relies on mathematically hard problems such as the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP)[3], which are resistant to both classical and quantum attacks.

Quantum-era communication security can be achieved through two main paths: post-quantum cryptography (PQC) and quantum cryptography. Quantum cryptography relies on the laws of quantum mechanics and offers information-theoretic security through methods such as quantum key distribution, quantum signatures, quantum bit commitment, and quantum teleportation. Although QKD is secure in theory, it lacks built-in authentication and depends on specialized hardware, making it difficult to deploy widely today. In contrast, PQC is based on the hardness of specific mathematical problems and provides a more practical and affordable solution. It supports both confidentiality and authentication while remaining compatible with existing computing systems. Since it does not require special hardware, PQC is easier to integrate and scale within current cryptographic infrastructures[4]. The main post-quantum cryptographic techniques are Hash-Based[5,6], Lattice-Based[3], Code-Based[7], Multivariate[8], and Isogeny-Based[9].

The significant breakthrough of lattice-based cryptography is the Ring Learning with Errors (Ring-LWE) problem, which is the improved version of Learning with Errors (LWE)[4]. Ring-LWE combines both the strength of lattice-based security and computational advantages by operating over polynomial rings, providing practical implementations of encryption, digital signatures, and homomorphic encryption schemes. Its theoretical underpinnings link it to the hardness of problems on ideal lattices, ensuring strong security guarantees. This implies that the practical implementation of Ring-LWE faces obstacles such as increased computational overhead and complicated parameter selection processes, which restrain its applicability in large-scale real-world environments.

## Objectives of the Study

The objective of this work is to compare Ring-LWE and Learning with Rounding (LWR) from both theoretical and practical perspectives. This study examines their mathematical foundations, differences in noise handling mechanisms, and performance under similar parameter settings. By measuring key generation, encryption, and decryption costs, the paper aims to evaluate the trade-off between the strong security guarantees of Ring-LWE and the implementation efficiency offered by LWR. The analysis also provides insight into whether a hybrid approach could balance security and efficiency more effectively.

## Related Work

Lattice-based cryptography has become a strong and adaptable foundation for secure communication, with significant work focused on improving the performance of Ring-LWE schemes. Many earlier implementations were tailored for high-performance systems, which limits their suitability for constrained environments.

Feng-Hao Liu and Zhedong Wang, in paper[10], extend LWR and LWE results to the ring setting, providing reductions for Ring-LWR, a link from Ring-LWE to Module Ring-LWR, and a new ring leftover hash lemma. In paper[11], Thomas Pöppelmann and Tim Güneysu propose lightweight Ring-LWE implementations for resource-limited reconfigurable hardware and achieve practical throughput. Thomas Pöppelmann *et al.* in paper[12], compare Ring-LWE encryption with the BLISS signature scheme on an 8-bit Atmel ATxmega128 platform.

The study by Ahmad Boorghany *et al.* in[13] evaluates lattice-based schemes on devices like smart cards, highlighting improvements in FFT methods, discrete Gaussian sampling, and public-key encryption for constrained settings. Norman Göttert *et al.* in paper[14], implement polynomial-based LWE cryptosystems in hardware and software using quasi-linear FFT multiplication and hardware-friendly Gaussian sampling.

## Preliminaries

### Lattice

A lattice is a regularly arranged set of infinite points in space. It can be defined as the set of all integer linear combinations of n-dimensional vectors. A lattice is described by a basis of vectors, where the basis consists of n linearly independent vectors in n-dimensional space. Importantly, different sets of basis vectors can generate the same lattice, meaning the representation of a lattice is not unique[15].

Mathematically, a lattice $\mathcal{L}$ in $R^n$ can be expressed as:

$$\mathcal{L} = \left\{ v \in \mathbb{R}^n \mid v = \sum_{i=1}^{n} k_i b_i, k_i \in \mathbb{Z}, b_i \in \mathbb{R}^n \right\},$$

where $b_1, b_2, \ldots, b_n$ are the basis vectors of the lattice, and $k_i$ are integers.

For example, in $R^2$, consider the basis vectors $b_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $b_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The lattice generated by this basis is the set of all points with integer coordinates:

$$\mathcal{L} = \left\{ \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \mid k1, k2 \in Z \right\}$$

Alternatively, using a different basis, such as $b_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $b_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, the same lattice can be generated, but the representation of the basis vectors changes. This illustrates that multiple bases can describe the same lattice. In general, the choice of basis affects the representation of the lattice but does not alter the lattice itself.

### Basis

A basis is a collection of vectors that can be used to reproduce any point in a given space[16]. For a 3-dimensional vector space, a basis consists of three vectors that are both linearly independent and span the space. This means that any vector in the space can be expressed as a unique linear combination of the basis vectors. A simple example of a basis in $R^3$ is the standard basis, which includes the vectors:

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

These vectors are mutually perpendicular and form the axes of the Cartesian coordinate system.

For example, the vector can be expressed as: $v = 4v_1 + 5v_2 + 6v_3 = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 5 \end{bmatrix}$.

In general, to generate a basis for $R^3$, one must select three vectors $v_1, v_2, v_3 \in R^3$ such that the determinant of the matrix formed by placing these vectors as columns is non-zero: if $B = [v_1\ v_2\ v_3]$, then det($B$)0.

This ensures that the vectors are linearly independent and span the space. A basis is not unique; other sets of three linearly independent vectors that span $R^3$ would also qualify as a valid basis.

## Good and Bad Basis

In lattice-based cryptography, the distinction between a good basis and a bad basis is fundamental to both the efficiency of algorithms and the security of cryptographic constructions.

$A$ basis $B = [b_1, b_2, \ldots, b_n]$ is said to be good if its vectors are short, nearly orthogonal, and well-conditioned. Let $\{b_1, \ldots, b_n^*\}$ denote the Gram–Schmidt orthogonalization of $B$. The basis is considered good when $||b^*||$ is small for all $i$.

This property implies that the lattice is represented by short and almost orthogonal vectors, which makes problems such as the Shortest Vector Problem (SVP) or Closest Vector Problem (CVP) algorithmically easier. With a good basis, algorithms such as Babai's nearest plane method or Gaussian sampling work efficiently. For instance, if the basis vectors are close to orthogonal and have length close to unity, one can approximate solutions to CVP by simple coordinate rounding in the orthogonal system.

In contrast, a bad basis of the same lattice consists of long, skewed, and almost linearly dependent vectors. In this case, the Gram–Schmidt norms $||b^*||$ are large, which increases the orthogonality defect of the basis. As a result, solving SVP or CVP becomes computationally infeasible using this basis. A bad basis often arises from a unimodular transformation of a good basis $B_{pub} = U \cdot B_{good}$, where $U \in Z^{n \times n}$ is unimodular (i.e., det($U$ ) = ±1). The lattice defined by $B_{pub}$ is the same as that defined by $B_{good}$, but computational problems such as CVP remain intractable with respect to $B_{pub}$.

The distinction between good and bad bases forms the basis of the asymmetric key structure in lattice-based cryptography. A good basis acts as the secret key because it enables efficient sampling of short vectors, decryption of ciphertexts, and generation of trapdoors. Since the vectors are short and nearly orthogonal, the holder of the good basis has a computational advantage, enabling polynomial-time solutions to otherwise hard lattice problems. The bad basis, on the other hand, is published as the public key, since it hides this trapdoor structure. To an adversary, the bad basis offers only a computationally hard lattice, making problems such as CVP intractable under worst-case hardness assumptions.

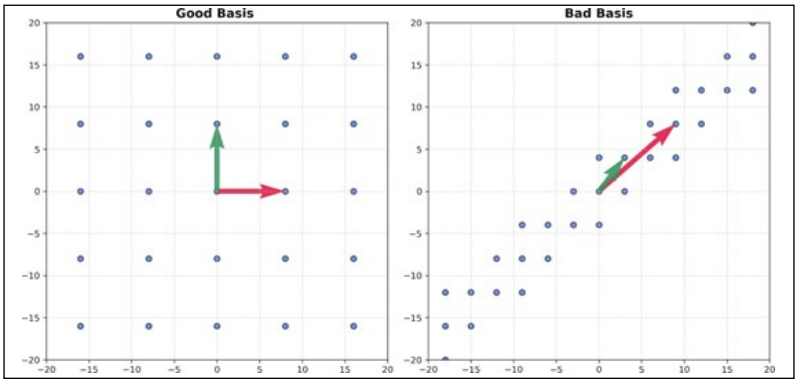


**Fig. 1:** Illustration of lattice bases. A good basis (short and nearly orthogonal vectors) enables efficient algorithms such as decryption, while a bad basis (long, skewed vectors) hides the lattice structure and ensures hardness for adversaries

### Closest Vector Problem

The Closest Vector Problem (CVP) is a well-known problem in computational mathematics and cryptography, specifically in lattice-based cryptography. The problem involves finding the lattice point that is closest to a given target point in a high-dimensional space. More formally, given a lattice L and a target vector $t \in R^n$, the task is to find the lattice point $v \in \mathcal{L}$ that minimizes the Euclidean distance to $t$: $v = arg\ min\ ||v' - t||_2$, $v' \in \mathcal{L}$, where $||v' - t||_2$ is the Euclidean distance between $v'$ and $t$. This problem is particularly challenging in high-dimensional spaces, and its difficulty underpins the security of many lattice-based cryptographic schemes.

### Shortest Vector Problem

The Shortest Vector Problem (SVP) is another fundamental problem in lattice theory, particularly important in lattice-based cryptography. The problem involves finding the shortest non-zero vector in a given lattice. More formally, given a lattice $\mathcal{L}$ in $R^n$, the objective is to find the vector $v \in \mathcal{L}$ such that: $v = arg\ min\ ||v'||_2$, $v' \in \mathcal{L}$, $v' \neq 0$ where $||v'||_2$ is the Euclidean norm (or length) of the vector $v$'. The difficulty of solving the SVP in high-dimensional spaces is what makes it a key problem in lattice-based cryptography, with security relying on the assumption that SVP is computationally hard.

## Learning With Errors and Regev's Encryption Scheme

The Learning With Errors (LWE) problem, introduced by Regev[15], forms the cornerstone of modern lattice-based cryptography. Formally, let $n$ denote the security parameter, $q$ a modulus, and $\chi$ an error distribution over $Z_q$. Given a secret vector $s \in Z_q^n$, the LWE distribution output pairs of the form $(a, b = \langle a,s \rangle + e \bmod q)$, where $a \in Z_q^n$ is chosen uniformly at random and $e \leftarrow \chi$ is an error sampled from the distribution $\chi$. The computational LWE problem requires distinguishing such pairs $(a, b)$ from uniformly random pairs in $Z_q^n \times Z_q$. The security of the LWE assumption relies on the hardness of solving certain worst-case lattice problems, such as the Shortest Vector Problem (SVP) and the Shortest Independent Vectors Problem (SIVP), within polynomial approximation factors.

## Why Noise Matters in LWE

From a geometric perspective, the addition of noise corresponds to perturbations around exact lattice points, thereby obscuring the underlying linear relations. This property provides a direct link between LWE and the hardness of lattice problems, since recovering the secret **s** from noisy inner products becomes as hard as approximating SVP or SIVP in high-dimensional lattices. Thus, noise not only prevents trivial algebraic attacks but also ensures post-quantum security reductions.

## Regev's Encryption Scheme

Regev's seminal encryption scheme[15] is a public-key cryptosystem that demonstrates the practicality of the LWE problem for cryptographic purposes. The scheme is defined as follows:

***Key Generation-*** Choose a secret vector $s \in Z_q^n$. Generate a public matrix $A \in Z_q^{m \times n}$ uniformly at random and sample an error vector $e \leftarrow \chi^m$. Compute $b = As + e \pmod{q}$. The public key is $(A, b)$, while the secret key is $s$.

***Encryption.*** To encrypt a bit $\mu \in \{0, 1\}$, select a random binary vector $r \in \{0, 1\}^m$. The ciphertext is given by $(u, v) = (r^T A, r^T b + [q/2] \cdot \mu) \pmod{q}$.

***Decryption.*** Given ciphertext $(u, v)$ and secret key $s$, compute $v - \langle u, s \rangle \pmod{q}$. If the result is closer to 0, output $\mu = 0$; if it is closer to $[q/2]$, output $\mu = 1$.

The Learning with Errors (LWE) problem, while foundational for lattice-based cryptography, comes with several drawbacks that limit its practical deployment. First, the scheme requires relatively large key sizes and ciphertexts compared to classical cryptosystems, which increases storage and communication overhead. Second, the computational cost of encryption and decryption is high due to operations over large matrices, making it less efficient for constrained devices. Third, ensuring correctness depends on carefully balancing the modulus q and error distribution; if the noise grows too large, decryption errors can occur. Fourth, parameter selection is subtle and critical; weak parameters can compromise security or correctness, while conservative choices further inflate sizes and computation. Finally, although LWE has strong theoretical security guarantees, its performance limitations motivate the use of structured variants like Ring-LWE and Module-LWE, which improve efficiency while maintaining hardness assumptions. Lattice problems position it as a strong candidate for quantum-resistant cryptography.

### Ring

In abstract algebra, a ring is an algebraic structure formed by a set equipped with two binary operations: addition and multiplication[3]. Formally, a ring is denoted by $(R, +, \cdot)$, where $R$ is a set with the following properties:

1. $(R, +)$ is an abelian group.
2. Multiplication is associative, i.e., for all $a, b, c \in R$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. The distributive property holds; for all $a, b, c \in R$: $a \cdot (b + c) = a \cdot b + a \cdot c$, $(b + c) \cdot a = b \cdot a + c \cdot a$.

## Ring Learning with Errors (Ring-LWE)

The Ring Learning with Errors problem is an adaptation of the LWE problem extended to polynomial rings in order to increase computational efficiency and usability in cryptographic applications. It was first studied by Oded Regev[3], which led to a more detailed ring-specific work later on by Lyubashevsky, Peikert, and Regev. Ring-LWE is based on the same principle of hardness extraction from lattices as LWE but differs in that whereas LWE works with vectors, Ring-LWE operates on polynomials. It operates over polynomial rings, commonly in the shape of $R_q = Z_q[x]/(x^n + 1)$, where n is a power of two, $q$ is a prime modulus, and $x^n + 1$ is an irreducible polynomial. The Ring-LWE problem, described in detail in the algorithm 1, is to distinguish between two distributions:

1. Samples chosen randomly from $R_q$
2. Examples of the form $(a, b = a \cdot s + e)$, where $a, s \in R_q$ are polynomials, and e is an error term picked from a Gaussian or other noise distribution.

Ring-LWE has several important advantages within modern cryptography in terms of its efficiency, security, and usability. It mainly relies on the ring structure and reduces the key and ciphertext size, which supports faster operations than other standard LWE schemes. The security of Ring-LWE is based on its correspondence to hard lattice problems, including SVP and the LWE problem, so it is hard against both kinds of attacks, both classical and quantum. In addition, the flexible framework of Ring-LWE encompasses several applications in state-of-the-art cryptographic schemes such as homomorphic encryption, key exchange protocol, and postquantum cryptographic algorithms, and therefore, serves as a backbone of future secure communication systems. The Ring-LWE has some weaknesses that must be carefully taken into account. The use of polynomial rings introduces extra structure into the cryptosystem, which, although enabling efficiency, may compromise security. This theoretical threat of attacks exploiting such structures is present, even if they have not yet materialized in practice. Also, appropriate parameter values, such as the degree n of the polynomials, modulus $q$, or noise distribution parameters, play important roles in fine-tuning for security versus efficiency. Poor parameters

**Algorithm 1** Ring-LWE Encryption Scheme

**Input:** $n, q, p, \sigma$
**Output:** Public key $(a, b)$, secret key $s$

**KeyGen():**
$s \leftarrow$ RandomPolynomial($q$)
$a \leftarrow$ RandomPolynomial($q$)
$e \leftarrow$ GaussianPolynomial($\sigma$)
$b \leftarrow (a \cdot s + e) \bmod q$
**return** $(a, b), s$

**Encrypt**$(pk, msg)$ :
**if** *msg is empty* **then**
**return** empty ciphertext
$(a, b) \leftarrow pk$
$m \leftarrow$ BytesToPoly($msg$); pad and split into degree-$n$ blocks
$e_1, e_2 \leftarrow$ GaussianPolynomial($\sigma$)
ciphertext $\leftarrow \emptyset$

**foreach** *block* $m_i$ **do**
$r \leftarrow$ RandomPolynomial($q$)
$u \leftarrow (a \cdot r + e_1) \bmod q$
$v \leftarrow (b \cdot r + e_2 + m_i) \bmod q$
append $(u, v)$ to ciphertext
**return** ciphertext

**Decrypt**$(s, ct)$ :
**if** *ct is empty* **then**
**return** empty message
$M \leftarrow \emptyset$
**foreach** $(u, v)$ *in ct* **do**
$x \leftarrow (v - u \cdot s) \bmod q$
recover coefficients of $x$ mod $p$ and append to $M$
**return** PolyToBytes($M$)

can make vulnerabilities or inefficiency appear, negating the aim of the proposed system. A further challenge lies in the repeated operations, in which the error term e accumulates, thereby threatening the correctness of decryption in practice. Ring-LWE has been made resistant to quantum attacks, but the known quantum hardness remains quite an open question. Quantum attacks could potentially reveal unknown weaknesses once future quantum computing and algorithms evolve[3].

## Learning with Rounding (LWR)

The Learning with Rounding is a cryptographic hardness assumption that was suggested as a certain variant of LWE. A complete LWR encryption scheme has been described in algorithm 2. A deterministic rounding version of LWR instead of providing Gaussian noise presents similar security and simplifies in some implementations related to LWE. LWR is a modification of the Learning with Errors (LWE) problem where instead of adding a random noise term, a deterministic rounding operation is performed. In LWR, assuming there is a secret vector **s** and random vector $a$ over $Z_q$, it generates samples of the form $(a, \lfloor a \cdot s \rceil_p)$, where $\lfloor \cdot \rceil_p$ denotes the rounding operation mapping elements from $Z_q$ into a smaller modulus $p$ such that $p < q$. The heart of LWR is the distinction of such structured samples from uniformly random pairs, which would be the very basis of the cryptographic security in this scheme.

**Algorithm 2** Learning with Rounding (LWR) Encryption Scheme

**Input:** $n, p, q$, message string
**Output:** Ciphertext $(C_1, c_2)$

**KeyGen():**
$A \leftarrow \text{RandomMatrix}(n \times n, q)$
$s \leftarrow \text{RandomBinaryVector}(n)$
$b \leftarrow (A \cdot s) \bmod q$
$B_{\text{rnd}} \leftarrow \text{round}((p/q) \cdot b) \bmod p$
**return** $A, s, B_{\text{rnd}}$

**Encrypt**$(A, B_{\text{rnd}}, msg)$ :
$M \leftarrow \text{StringToBinary}(msg)$
**if** *length(M) mod* $n \neq 0$ **then**
  pad $M$ with zeros
reshape $M$ into column vector
$r \leftarrow \text{RandomBinaryVector}(n)$
$C_1 \leftarrow (A^T \cdot r) \bmod q$
$c_2 \leftarrow (B_{\text{rnd}}^T \cdot r + \lfloor p/2 \rfloor \cdot M) \bmod p$
**return** $(C_1, c_2)$

**Decrypt**$(C_1, c_2, A, s)$ :
$u \leftarrow (C_1^T \cdot s) \bmod q$
$C_{2\_new} \leftarrow (c_2 - \lfloor (p \cdot u)/q \rceil) \bmod p$
$C_2 \leftarrow \text{Threshold}(C_{2\_new}, \lfloor p/2 \rfloor)$
$bits \leftarrow \text{Flatten}(C_2)$
$msg \leftarrow \text{BinaryToString}(bits)$
**return** msg

LWR has several advantages over LWE: it is noise-free in its construction, so implementation and theory both get simplified since noise need not be added explicitly; it is tied to the hardness of worst-case lattice problems similar to LWEs, thus ensuring strong cryptographic robustness; and finally, since LWR is deterministic, parameterization is further streamlined because selection of a noise distribution does not pose the same complexities. This deterministic approach also cuts down on the computational overhead, making LWR an efficient alternative in many applications. However, the rounding operation in LWR is also deterministic, and this introduces possible limitations. For example, vulnerabilities may arise when the rounding function is not properly chosen or the parameters are badly configured. This determinism in rounding, also makes LWR less flexible than LWE, especially for cryptographic schemes that take advantage of the noise being stochastic. Also, the LWR's security analysis is relatively

newer compared to that of LWE. Although believed to be quantum-resistant, it has not been analyzed as intensively as that of LWE, and quantum algorithms may become stronger in the future. Similar to LWE, LWR may also suffer from error amplification in repeated computations, which may impact the correctness of decryption in real-world instantiations[16].

## Experimental Methodology and Result Evaluation

Both Ring-LWE and LWR were implemented using comparable parameter sets to ensure fair evaluation. Experiments were conducted for lattice dimensions with fixed modulus settings. For Ring-LWE, discrete Gaussian noise was applied, while LWR used deterministic rounding. The execution time of key generation, encryption, and decryption was measured and recorded in seconds. All tests were performed under identical computational conditions, and the results were analyzed to compare efficiency and scalability between the two schemes.

**Table 1:** Execution Times for Different Ring-LWE Parameters

| Ring LWE Parameters | KeyGen (s) | Encryption (s) | Decryption (s) | Total Execution Time (s) |
|---|---|---|---|---|
| $n = 1024$, $p = 256$, $q = 49157$ | 0.003171 | 0.008993 | 0.000999 | 0.015161 |
| $n = 512$, $p = 256$, $q = 24577$ | 0.000998 | 0.006227 | 0.000998 | 0.011221 |
| $n = 256$, $p = 256$, $q = 12289$ | 0.000897 | 0.006244 | 0.001001 | 0.010241 |

**Table 2:** Execution Times for Different LWR Parameters

| LWR Parameters | KeyGen (s) | Encryption (s) | Decryption (s) | Total Execution Time (s) |
|---|---|---|---|---|
| $n = 1024$, $p = 256$, $q = 49157$ | 0.026172 | 0.011141 | 0.001000 | 0.038312 |
| $n = 512$, $p = 256$, $q = 24577$ | 0.007995 | 0.003177 | 0.000999 | 0.011171 |
| $n = 256$, $p = 256$, $q = 12289$ | 0.002995 | 0.002997 | 0.000998 | 0.006991 |

Table 1 presents the execution time of the Ring-LWE scheme for different lattice dimensions. As the polynomial degree increases from 256 to 1024, the overall execution time increases, mainly due to higher computational cost in polynomial multiplication and Gaussian sampling. Encryption consistently consumes more time than decryption, while key generation time grows with dimension size. The results indicate that larger parameters improve security but introduce additional computational overhead.

Table 2 shows the execution time of the LWR scheme under similar parameter settings. Compared to Ring-LWE, LWR generally demonstrates competitive or lower total execution time for smaller dimensions due to the absence of Gaussian sampling. However, for larger dimensions, key generation time increases noticeably. The results highlight the efficiency advantage of deterministic rounding while maintaining scalability across parameter sizes.

Fig. 2 visually illustrates the performance trend of Ring-LWE across different parameter sizes. The graph confirms that execution time increases with lattice dimension, with encryption being the most computationally intensive operation. The trend reflects the impact of polynomial arithmetic and noise sampling on runtime complexity.

Fig. 3 presents the performance behavior of LWR. The graphical representation shows smoother growth in execution time compared to Ring-LWE, particularly in encryption and decryption stages. The absence of stochastic noise sampling contributes to improved efficiency, especially for moderate parameter sizes.

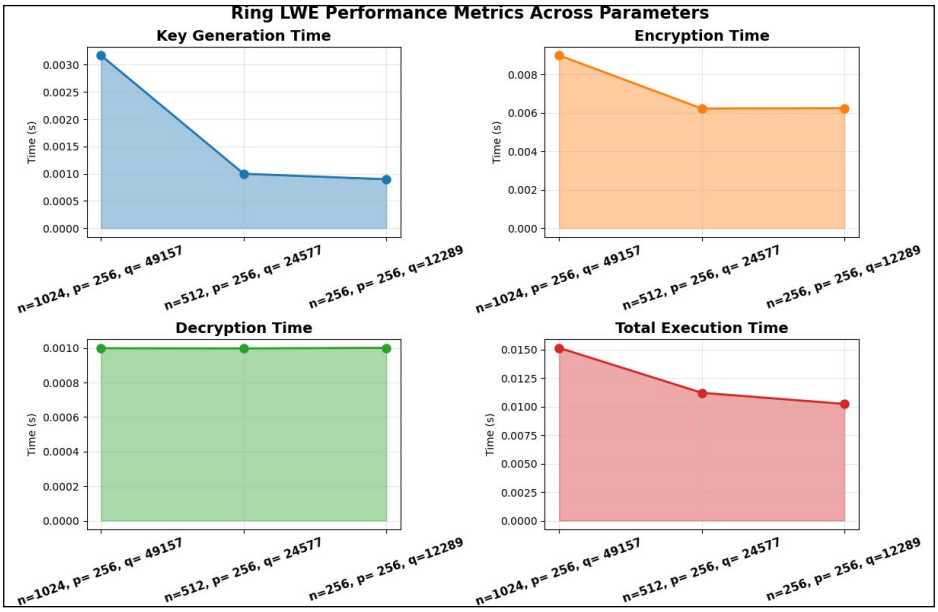


**Fig. 2:** Performance of the Ring-LWE algorithm with different parameter settings, showing key generation, encryption, and decryption metrics and Total Execution Time in Seconds

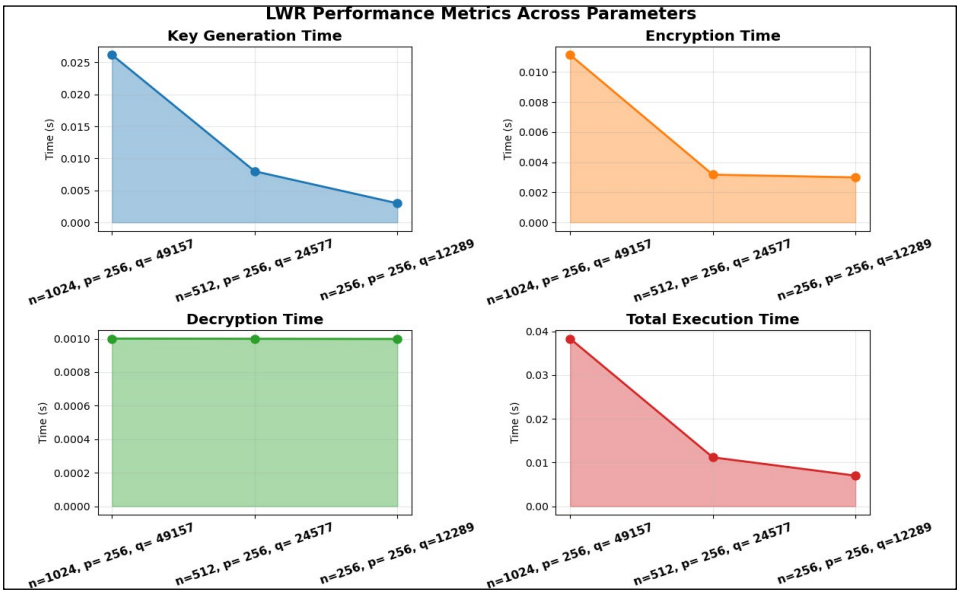


**Fig. 3:** Performance of the LWR algorithm with different parameter settings, showing key generation, encryption, and decryption metrics and Total Execution Time in Seconds

## CONCLUSION AND FUTURE WORK

Ring-LWE has become the central hardness assumption for lattice-based cryptography, only because of its strong worst-case to average-case reductions from ideal lattice problems. This solid foundation supports many applications, including key exchange and fully homomorphic encryption. Despite its advantages, Ring-LWE has practical drawbacks: it depends on discrete Gaussian sampling, which is expensive to implement, hard to secure against side-channel attacks, and a potential source of decryption errors when parameters are not carefully tuned. High-quality Gaussian samplers also demand substantial randomness and add further implementation complexity.

Learning with Rounding (LWR) takes a different approach by replacing noise sampling with deterministic rounding. This avoids Gaussian sampling entirely, leading to simpler and more efficient implementations, particularly on constrained devices. Rounding operations can also be implemented in constant time, offering better resistance to timing and side-channel attacks. However, the underlying hardness guarantees for LWR are less conservative than those of Ring-LWE, relying on modulus-ratio–based reductions that are still less mature. These two approaches therefore complement one another: Ring-LWE offers strong theoretical security, while LWR provides efficiency and simplicity. A promising direction is to combine them by designing a hybrid Ring-LWE scheme that incorporates deterministic rounding, aiming to retain Ring-LWE's worst-case guarantees while mitigating the practical costs of Gaussian sampling.

As future work, we intend to formalize and study such a hybrid construction, examining whether deterministic rounding can be integrated without weakening Ring-LWE's hardness. This could lead to post-quantum primitives that strike a better balance between efficiency and conservative security, helping narrow the gap between theory and practical deployment.

## REFERENCES


1. Shor, P.W. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science (pp. 124-134). IEEE.

2. Grover, L.K. 1996. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 212-219).

3. Lyubashevsky, V., Peikert, C. and Regev, O. 2010. On ideal lattices and learning with errors over rings. In Annual international conference on the theory and applications of cryptographic techniques (pp. 1-23). Berlin, Heidelberg: Springer Berlin Heidelberg.

4. Wang, X., Xu, G. and Yu, Y. 2023. Lattice-based cryptography: A survey. *Chinese Annals of Mathematics, Series B,* **44**(6): 945-960.

5. Bernstein, D.J. 2025. Post-quantum cryptography. In Encyclopedia of Cryptography, Security and Privacy (pp. 1846-1847). Cham: Springer Nature Switzerland.

6. Kanoo, S. and Das, P. 2025. Performance and Security Analysis of Hash-Based Digital Signature Schemes. In 2025 7th International Conference on Computer Communication and the Internet (ICCCI) (pp. 141-146). IEEE.

7. McEliece, R.J. 1978. A public-key cryptosystem based on algebraic. *Coding Thv.*, **4244**(1978): 114-116.

8. Ding, J., Gower, J.E. and Schmidt, D.S. 2006. Multivariate public key cryptosystems. Boston, MA: Springer US.

9. Jao, D. and De Feo, L. 2011. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In International workshop on post-quantum cryptography (pp. 19-34). Berlin, Heidelberg: Springer Berlin Heidelberg.

10. Liu, F.H. and Wang, Z. 2020. Rounding in the rings. In Annual International Cryptology Conference (pp. 296-326). Cham: Springer International Publishing.

11. Pöppelmann, T. and Güneysu, T. 2014. Area optimization of lightweight lattice-based encryption on reconfigurable hardware. In 2014 IEEE international symposium on circuits and systems (ISCAS) (pp. 2796-2799). IEEE.

12. Pöppelmann, T., Oder, T. and Güneysu, T. 2015. High-performance ideal lattice-based cryptography on 8-bit ATxmega microcontrollers. In International conference on cryptology and information security in Latin America (pp. 346-365). Cham: Springer International Publishing.

13. Boorghany, A., Sarmadi, S.B. and Jalili, R. 2015. On constrained implementation of lattice-based cryptographic primitives and schemes on smart cards. *ACM Transactions on Embedded Computing Systems (TECS),* **14**(3): 1-25.

14. Göttert, N., Feller, T., Schneider, M., Buchmann, J. and Huss, S. 2012. On the design of hardware building blocks for modern lattice-based encryption schemes. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 512-529). Berlin, Heidelberg: Springer Berlin Heidelberg.

15. Regev, O. 2009. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM),* **56**(6): 1-40.

16. Banerjee, A., Peikert, C. and Rosen, A. 2012. Pseudorandom functions and lattices. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 719-737). Berlin, Heidelberg: Springer Berlin Heidelberg.