Security Analysis of Kyber Cryptographic Algorithm Using Number Theory and Lattice Principles

Syakira Azzahra Rachmania - 13525055
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: arrsyakira@gmail.com , 13525055@std.stei.itb.ac.id

***Abstract*—The advancement of universal quantum computers running Shor's Algorithm poses a critical threat to classical public-key cryptosystems like RSA and ECC. To mitigate these vulnerabilities, the FIPS 203 standard officially standardized the Module Lattice-Based Key Encapsulation Mechanism derived from the CRYSTALS-Kyber framework. This paper provides a comprehensive security analysis of Kyber-512 by evaluating its underlying foundation in number theory and lattice principles. This study analyzes the mathematical transition from modular arithmetic over cyclotomic polynomial rings to high-dimensional geometric structures. To demonstrate the integrity of the Learning with Errors (LWE) framework, a low-dimensional algebraic verification walkthrough is presented across the key generation, encapsulation, and decapsulation phases. The analysis highlights how mapping polynomial coefficients into a dense, 512-dimensional Module Lattice effectively secures the cryptographic space against quantum cryptanalysis by leveraging the NP-hard Closest Vector Problem (CVP). Ultimately, this tight integration of modular ring arithmetic and lattice geometry establishes that Kyber-512 delivers robust post-quantum immunity, confirming its suitability for modern digital security infrastructures.**

Keywords—Kyber; Number Theory; Modular Arithmetic; Post-Quantum Cryptography; Lattice-Based Cryptography;

I. Introduction

The rapid advancement of the digital era and information technology has seamlessly integrated computing into every aspect of human life. Along with this pacing transition, cryptography has significantly progressed, driving the necessity for more complex algorithms in the domain of cybersecurity. Modern digital infrastructures heavily rely on public-key cryptography to guarantee data confidentiality, integrity, and authenticity across open communication networks. However, as cryptographic implementations become more pervasive, the underlying mathematical assumptions that once guaranteed their security are now facing some novel threats.

For decades, widely deployed public-key cryptosystems such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) have been deemed secure, under the assumption that classical computers find integer factorization and discrete logarithm problems computationally intractable. Nevertheless, this foundation contains a critical vulnerability when quantum mechanics is taken into consideration. As demonstrated by Peter Shor, a universal quantum computer utilizing Shor's Algorithm can solve discrete logarithms and factor integers within polynomial time constraints. This capability enables the execution of quantum cryptanalysis that could compromise conventional public-key infrastructures [1].

To mitigate these looming quantum vulnerabilities, the National Institute of Standards and Technology (NIST) initiated a global standardization process to transition toward post-quantum cryptography (PQC). This collective effort culminated in the official publication of the Federal Information Processing Standards (FIPS) 203, which establishes the Module-Lattice-Based Key-Encapsulation Mechanism Standard. FIPS 203 explicitly specifies ML-KEM, a highly optimized variant derived from the CRYSTALS-Kyber algorithm [2], whose security relies on the computational hardness of the Module Learning with Errors (MLWE) problem [3].

Therefore, this paper will focus on the CRYSTALS-Kyber algorithm framework analysis in a more practical context. This includes breaking down the mathematical foundation specified in the FIPS 203 standard such as the application of the number theory concepts, specifically modular arithmetic on polynomials and the basic idea of lattice-based security. Furthermore, this study will incorporate a Python-based experiment to test the algorithm directly. The simulation will measure practical aspects of the algorithm, such as the execution time for the encryption and decryption processes, as well as the size of the generated keys. By combining mathematical theory with a simple programming simulation, this paper aims to provide a clear understanding of how ML-KEM operates as the new standard for post-quantum cybersecurity.

II. Theoretical Basis

A. Number Theory and Modular Arithmetic

Number theory is a branch of pure mathematics that studies integers and integer-valued functions. Integers are numbers that do not possess fractional or decimal components. In cryptographic analysis, the main foundation used is the divisibility properties of integers, which are formally defined through Euclidean Theorem on division. The Euclidean Theorem states that for any integers m and n where $n > 0$,

dividing m by n yields a quotient q and a remainder r, which can be expressed in the equation:

$$m = nq + r \text{ where } 0 \leq r < n \text{ [4].}$$

From this concept of remainders emerges modular arithmetic. The operation a mod m yields the remainder r when the integer a is divided by a positive integer m. This concept gives rise to the principle of congruence, where two integers a and b are said to be congruent modulo m, written as:

$$a \equiv b \pmod{m}$$

if and only if m divides the difference $(a - b)$. This congruence relation can also be written equivalently in the form:

$$a = b + km$$

where k is an integer. Within the modular arithmetic system, mathematical operations possess specific constraints; for instance, both sides of a congruence equation can be added, multiplied, or exponentiated by a constant, but the division operation is excluded as it does not consistently satisfy the congruence conditions [4].

The concept of the Greater Common Divisor (GCD) plays a crucial role. Let a and b be non-zero integers. If there exist a largest integer d such that $d \mid a$ and $d \mid b$, it can be denoted as:

$$\text{GCD}(a, b) = d$$

Two integers are considered relatively prime when their GCD is 1 [4]. This condition of being relatively prime is the primary determinant in finding the modular inverse. If a and m are relatively prime and $m > 1$, it is guaranteed that there exist an integer x acting as the inverse of $a \pmod{m}$, satisfying the congruence equation:

$$xa \equiv 1 \pmod{m}$$

The security of many modern cryptographic algorithms is also inextricably linked to the properties of prime numbers. A prime number is a positive integer $p > 1$ whose only divisors are 1 and p itself. This prime property underpins Fermat's Little Theorem [4], which mathematically result in the congruence form:

$$a^{p-1} \equiv 1 \pmod{p}$$

The application of the aforementioned number theory operations is vast, with one of its primary implementations being in the field of cryptography. Cryptography utilizes modular operations and prime numbers as the art of securing messages, encoding data (plain text) into a randomized form (ciphertext) to ensure that the data remains inaccessible to unauthorized parties.

B. *Polynomial Ring*

To manage data structures efficiently without structural growth during encryption operations, post-quantum frameworks utilize polynomial rings over finite fields. Formally, a polynomial ring $\mathbb{Z}_q[x]$ consist of all polynomials whose coefficient are integers evaluated modulo a prime number q. To maintain a strict data boundary, this domain is restricted by a quotient ideal generated by an irreducible polynomial $f(x)$, typically defined as

$$\mathbb{R}_q = \mathbb{Z}_q[x] \,/\, (f(x))$$

To provide a conceptual mapping of this algebraic boundary, Fig. 1 demonstrates the structural anatomy of a basic quotient ring domain

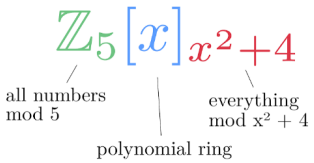


Fig. 1. Notation breakdown and boundary constraints of a toy-model quotient polynomial ring.
(Source: https://xyquadrat.ch/blog/is-polynomial-ring-field/)

As Illustrated in Fig 1, to ensure structure closure within the bounded ring $\mathbb{R}_q$, two essential reduction mechanism are applied during polynomial multiplication:

- Coefficient Reduction
  Every coefficient generated during polynomial addition or multiplication instantly reduced modulo 5, restricting the coefficient space to finite set $\{0, 1, 2, 3, 4,\}$.
- Degree Reduction
  The global structures rule enforces $x^2 + 4 \equiv 0$, which mathematically implies that any polynomial degree equal to or higher than 2 is reduced via the substitution relation $x^2 \equiv -4 \equiv 1 \pmod{5}$.

By implementing this congruence relation, any higher degree expression is mapped back to its representative lower-degree polynomial within the specified domain. This mathematical framework provides the structural foundation for the high-dimensional vector spaces and matrix operations central to lattice-based cryptosystems [3].

C. *Lattice Theory*

Post-quantum cryptographic frameworks transition away from classical factoring problems by utilizing the geometric complexities of lattice structures. Mathematically, a lattice $\mathcal{L}$ is a discrete, infinitely repeating grid of points in an n-dimensional Euclidean space. This grid is generated by all possible integer linear combinations of a set of linearly independent basis vectors. The cryptographic robustness of these structures relies heavily on the computational intractability of geometric navigation in extremely high-dimensional spaces [5].

The security foundation is built upon two primary mathematical challenges: the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). The SVP requires finding the shortest non-zero vector within the lattice (the closest valid point to the origin) [5]. Extending this complexity, the CVP requires finding the closest valid lattice point to an arbitrary, randomly positioned off-grid coordinate.

As illustrated in Fig. 2, the mathematical difficulty of solving these problems depends entirely on the quality of the basis vectors provided to the user. A "Good Basis" consists of short, nearly orthogonal vectors (perpendicular to each other), which allows straightforward mathematical rounding to find the closest lattice point. Conversely, a "Bad Basis" consists of long, highly skewed, and nearly parallel vectors. While legitimate users hold the trapdoor (Good Basis) to navigate the grid easily, adversaries are only provided with the Bad Basis [5]. As the dimensions scale up, calculating the closest vector using a bad basis becomes an NP-hard problem, completely resistant to sub-exponential quantum acceleration like Shor's Algorithm.

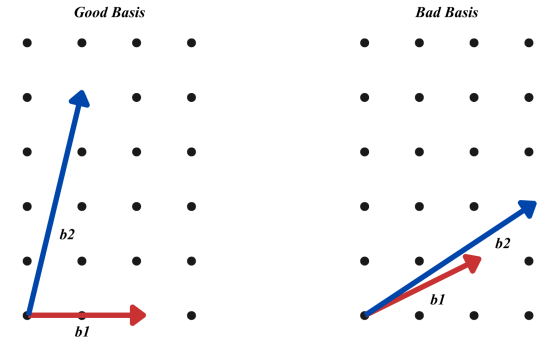


Fig. 2. Geometric visualization of lattice basis vectors. A "Good Basis" provides orthogonal pathways for easy navigation, whereas a "Bad Basis" consists of highly skewed vectors that make identifying the closest lattice point computationally intractable.

D. Learning with Errors

The geometric hardness of lattice problems is mapped into algebraic cryptographic systems through the Learning with Errors (LWE) framework. In a standard algebraic scenario, solving a system of linear equations to find a secret vector **s** given a public matrix *A* and a result vector **t** (where *A***s** = **t**) is trivial using traditional Gaussian elimination.

To prevent this, the LWE framework intentionally disrupts the exactness of the equation by introducing a small, secret error vector e [6]. This transforms the easily solvable linear equation into a noisy, overdetermined system:

$$\mathbf{t} = A\mathbf{s} + e$$

Geometrically, this algebraic perturbation perfectly mimics the Closest Vector Problem (CVP). As shown in Fig. 3, the exact equation As represents a pure, discrete point perfectly aligned on the lattice grid. The addition of the error *e* acts as a spatial "wiggle," shifting the final public vector **t** into a noisy, off-grid position.

To decrypt the message or retrieve the secret key **s**, an attacker must strip away the noise e from the public vector **t**. However, because the attacker only possesses the public "bad basis," finding the original *A***s** point from the off-grid **t** position is fundamentally equivalent to solving a high-dimensional CVP [5], [6]. For the legitimate receiver, the presence of the private key (the good basis) acts as a trapdoor, allowing them to easily round the noisy vector back to the correct lattice point.

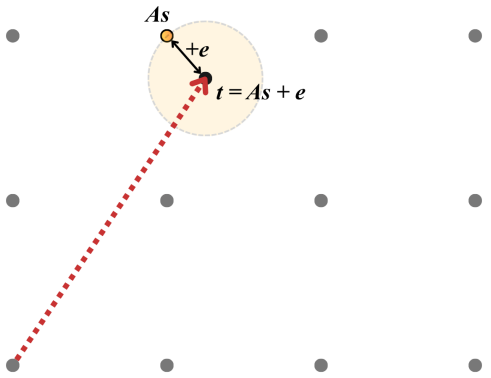


Fig. 3. Geometric visualization of the Learning with Errors (LWE) transformation, where a secret lattice point $A\mathbf{s}$ is perturbed by a small error vector e to produce the noisy public vector $\mathbf{t} = A\mathbf{s} + e$.

E. Kyber

CRYSTALS-Kyber, officially standardized by NIST as the Module Lattice-Based Key Encapsulation Mechanism (ML-KEM) under the FIPS 203 standard, represents a paradigm shift from traditional cryptographic protocols. Unlike the classical Diffie-Hellman Key Exchange where both communicating parties mutually contribute to constructing a shared secret, ML-KEM utilizes an encapsulation approach. This method allows one party to unilaterally generate a symmetric key and securely transport it across an insecure public network using lattice-based cryptography.

As illustrated in Fig. 4, the ML-KEM operational architecture consists of three sequential cryptographic algorithms executed between a receiver and a sender:

1) Key Generation (ML-KEM.Keygen)
The protocol is initiated by the receiver (Tarun), who executes the key generation algorithm to produce a mathematically linked keypair. This consists of a public encapsulation key (*ek*) and a strictly guarded private decapsulation key (*dk*). The public key ek is then transmitted openly over the network to the sender.

2) Encapsulation (ML-KEM.Keygen)
Upon receiving the public key ek, the sender (Aman) executes the encapsulation function. This algorithm generates a fresh, random symmetric Shared Secret key (*K*). Instead of sending *K* directly, the algorithm mathematically locks it inside a cryptographic Ciphertext (*c*) utilizing the receiver's *ek* and bounded error distributions. Only the ciphertext *c* is transmitted back over the public channel, ensuring *K* remains entirely isolated from network exposure.

3) Decapsulation(ML-KEM.Decaps)
Upon receiving the ciphertext c, the receiver utilizes their private key dk to execute the decapsulation function. This algorithm systematically strips away the lattice-based noise integrated during the encapsulation phase, allowing the receiver to independently derive the exact same Shared Secret key (K).

Through this three-phase mechanism, both parties successfully establish an identical symmetric key (K) for subsequent fast data encryption without ever exposing the key itself to potential quantum adversaries monitoring the transmission channel.

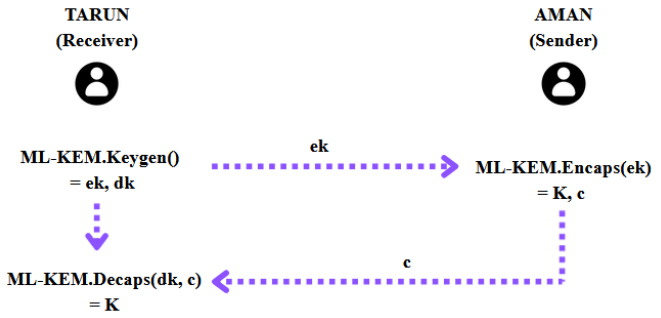


Fig. 4. The execution workflow of the Module Lattice-Based Key Encapsulation Mechanism (ML-KEM), illustrates the secure establishment of a shared secret (K) via ciphertext (c) transmission.

III. Result and Analysis

A. *Security Analysis via Number Theory and Lattice Principles*

Kyber constructs its framework by integrating number theory and lattice cryptography. The security mechanism relies on transforming algebraic polynomial rings into high-dimensional geometric lattices. Following the FIPS 203 standard, the simulation executed in this paper operates over the polynomial ring domain:

$$\mathbb{R}_q = \mathbb{Z}_{3329}[x] / (x^{256} + 1)$$

Under this parameter set, every polynomial consists of 256 coefficients bounded by a prime modulus of 3329. Mathematically, this array of 256 coefficients directly maps to a single coordinate vertex within a 256-dimensional geometric space. By assembling these polynomials into a matrix with a rank of $k = 2$, Kyber expands this structure into a massive 512-dimensional Module Lattice.

Within this 512-dimensional geometric grid, the security relies on the LWE protocol, which executed across three sequential operations:

1) Key Generation (ML-KEM.Keygen)

$$dk = \mathbf{s}$$

$$ek = (A, \mathbf{t}), \text{ where } \mathbf{t} = A\mathbf{s} + e$$

The receiver generates a private key dk as a secret vector $\mathbf{s}$which represents the "Good Basis". Then, the system computes an exact lattice point $A\mathbf{s}$ and perturbs it by adding a small error e, producing the public key ek.

2) Encapsulation (ML-KEM.Encaps)

- Input: $ek = (A, \mathbf{t})$
- Output: c, K

The sender utilizes the received public key ek to encrypt a newly generated symmetric shared secret K into a ciphertext c. By injecting transient randomness and noise over the public "Bad Basis," the system forces any eavesdropper to solve the NP-hard CVP across a 512-dimensional space to reverse-engineer the keys.

3) Decapsulation(ML-KEM.Decaps)

- Input: $ek = (A, \mathbf{t})$
- Output: c, K

Upon receiving the ciphertext c, the receiver applies their private key dk. The secret vector $\mathbf{s}$ acts as a geometric trapdoor that instantly filters out the accumulated LWE noise from the ciphertext components, extracting the identical shared secret K efficiently without performing complex geometric searches.

Cryptanalytically, this extreme dimensional scaling acts as the primary defense mechanism of Kyber. Because Shor's Algorithm only accelerates period-finding tasks in one-dimensional algebraic structures (such as integer factorization in RSA or discrete logarithms in ECC), it provides no mathematical advantage when searching for hidden spatial coordinates across a fragmented, 512-dimensional geometric lattice. Consequently, the tight integration of polynomial rings and LWE operations successfully establishes a mathematically sound barrier against sub-exponential quantum computing threats.

B. *Low-Dimensional Algebraic Demonstration of Kyber*

To provide an explicit algebraic demonstration of how the polynomial ring and LWE mechanism works in unison, this section outlines a simplified simulation. For this computational walkthrough, the systems parameters are restricted to a localized ring domain defined as:

$$\mathbb{R}_5 = \mathbb{Z}_5[x] / (x^2 + 1)$$

Within this bounded domain are evaluated modulo 5, restricting the coefficient space to $\{0, 1, 2, 3, 4,\}$. Furthermore, any polynomial expanding to a degree equal to or higher than 2 undergoes reduction:

$$x^2 + 1 \equiv 0 \quad \Rightarrow \quad x^2 \equiv -1 \equiv 4 \pmod 5$$

First, the key generation phase is initiated by the receiver (Tarun) to construct a private trapdoor and a public verification vector. Let the public matrix base be $A = 2x + 1$, the secret key vector (private trapdoor) be

$s = x + 2$, and the small bounded error vector be $e = 1$. The public encapsulation key vector $\mathbf{t}$ is mathematically compiled through the noisy linear relation.

$$\mathbf{t} = (A\mathbf{s} + e) \pmod{x^2 + 1} \pmod 5$$

First, the core polynomial multiplication is executed.

$$(2x + 1)(x + 2) = 2x^2 + 5x + 2$$

Applying the coefficient modular reduction (mod 5) eliminates the linear term $5x \equiv 0$.

$$2x^2 + 5x + 2 \equiv 2x^2 + 2 \pmod 5$$

Next, apply the degree boundary condition ($x^2 \equiv -1$) reduces the quadratic term.

$$2(-1) + 2 = -2 + 2 = 0$$

Finally, incorporating the secret noise vector $e = 1$ yields the final public vector component.

$$t = 0 + 1 = 1$$

Second, the Encapsulation Phase is executed by the sender (Aman) upon receiving the public key component to encrypt a discrete message payload. Let the message payload be a scalar value $m = 2$, the transient random polynomial $r = 2x$, and the two independent noise vector be $e_1 = x$ and $e_2 = 2$. To protect the message against direct algebraic scaling analysis, the payload m is scaled by a factor $\lfloor q/2 \rfloor$. Operating under $q = 5$, the scaled message vector becomes $skala(m) = 2 \times 2 = 4$. Aman then constructs the dual-component ciphertext tuple $c = (c_1, c_2)$.

$$c_1 = (Ar + e_1) \pmod{x^2 + 1} \pmod 5$$

First the intermediate polynomial multiplication is evaluated.

$$Ar = 4x^2 + 2x \equiv 2x - 4 \pmod{x^2 + 1}$$

Reducing the negative coefficient then yields

$$2x - 4 \equiv 2x + 1 \pmod 5$$

Injecting the first error $e_1 = x$ completes the first artifact.

$$c_1 = (2x + 1) + x = 3x + 1$$

Component c_2 is compiled by embedding the scaled message onto the public target.

$$c_2 = (tr + e_2 + m \cdot \lfloor q/2 \rfloor) \pmod{x^2 + 1} \pmod 5$$

$$c_2 = (1 \cdot 2x) + 2 + 4 = 2x + 6 \equiv 2x + 1 \pmod 5$$

The resulting encrypted ciphertext packet transmitted over the public network is $c = (c_1 = 3x + 1, c_2 = 2x + 1)$.

Lastly, the encapsulation phase allows the receiver (Tarun) applies the private trapdoor key $\mathbf{s}$ to filter out the structural noise and isolate the raw signal vector $\mathbf{v}$ from the received ciphertext tuple c,. Let the inputs received be the ciphertext components as resulted before, processed using the private key vector $\mathbf{s} = x + 2$. The signal extraction is computed via the relation:

$$\mathbf{v} = (c_2 - \mathbf{s} \cdot c_1) \pmod{x^2 + 1} \pmod 5$$

First, compute the inner-product.

$$\mathbf{s} \cdot c_1 = (x + 2)(3x + 1) = 3x^2 + 7x + 2$$

Then, reduce the coefficient with modulo 5.

$$3x^2 + 7x + 2 \equiv 3x^2 + 2x + 2 \pmod 5$$

Applying the degree reduction rule ($x^2 \equiv -1$) reduce the quadratic form to a linear form.

$$3(-1) + 2x + 2 = 2x - 1 \equiv 2x + 4 \pmod 5$$

Finally, subtracting this resulting polynomial from the second ciphertext component isolates the signal

$$\mathbf{v} = (2x + 1) - (2x + 4) = -3 \equiv 2 \pmod 5$$

The isolated raw signal yields a scalar constant $\mathbf{v} = 2$. This decrypted output matches the sender's original message ($m = 2$), demonstrate how the low-dimensional algebraic constraints successfully maintain the integrity of the framework through transmission.

C. Simulation Environment and Experimental Parameter

To evaluate the practical viability of these mathematical principles, a localized computational simulation was executed. The simulation environment and underlying software parameters are structured as follows:

- *Hardware Specifications:*

  The benchmarks were conducted on an HP EliteBook 840 G8 Notebook PC with an 11th Gen Intel® Core™ i5-1145G7 processor base clocked at 2.60 GHz, and 16.0 GB of installed RAM (3200 MT/s). The simulation ran within a 64-bit operating system architecture.

- *Software Architecture:*

  The cryptographic simulation was compiled and executed utilizing Python 3.13.

- *Cryptographic Configurations:*

  The experiment follows the baseline ML-KEM-512 parameters specified in the FIPS 203 standard, setting

the matrix rank to 2, the modular base to 3329, and the polynomial degree to 256.

D. *Framework Performance Metrics: Time and Memory Analysis*

To evaluate the practical performance of Kyber-512, this study measures the execution time and memory requirements of the framework. The simulation was executed ten times to obtain the average execution time, and the complete results are summarized in Table I.

TABLE I. PERFORMANCE METRICS AND DATA OVERHEAD OF KYBER-512

Cryptographic Phase	***Mean Execution Time (ms)***	***Standard Artifact Size (Byte)***
Key Generation	0.0663	800 (Public key *ek*)
Encapsulation	0.0967	768 (ciphertext *c*)
Decapsulation	0.0251	-

As shown in the table, the Kyber-512 framework demonstrates highly efficient operational metrics. Both the encapsulation and decapsulation phases operate at sub-milisecond speeds, averaging 0.0967 ms and 0.0251 ms, respectively. Even with the complex matrix-vector multiplication required to introduce LWE noise during encapsulation, the system executes the process rapidly.

In addition, the framework maintains a very compact memory profile. The public encapsulation key requires only 800 bytes, and the encrypted ciphertext is optimized to 768 bytes. Compared to classical primitives like RSA where keys must scale exponentially to withstand quantum threats, this sub-kilobyte footprint is remarkably efficient. These results prove that lattice-based cryptography can provide robust quantum-resistant security while remaining lightweight and fast.

IV. CONCLUSION

In conclusion, Kyber as a response to the vulnerabilities of classical cryptosystems like RSA and ECC against quantum computing, Kyber-512 successfully provides a robust post-quantum cryptographic solution by combining number theory and lattice principles. The theoretical analysis confirms that mapping modular polynomial rings into high-dimensional geometric spaces effectively secures the key encapsulation mechanism by utilizing the computational hardness of the Closest Vector Problem (CVP) and the Learning with Errors (LWE) framework.

Furthermore, the localized computational simulation validates the framework's practical efficiency by operating at sub-millisecond execution speed and maintaining a compact sub-kilobyte memory footprint. These results proved that Kyber delivers an optimized balance between security and performance, confirming that lattice-based cryptography does not sacrifice network throughput for quantum immunity and solidifying Kyber as an efficient and secure standard for digital communication infrastructures.

APPENDIX

The complete implementation program can be accessed at this link https://github.com/araended/kyber-simulation

ACKNOWLEDGMENT

The author would like to thank Mr. Rinaldi Munir as the lecturer for his insight and lessons throughout the semester. The author also extends heartfelt thanks to her family and friends who have accompanied her until now.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994, pp. 124-134.

[2] J. Bos et al., "CRYSTALS – Kyber: A CCA-secure module-lattice-based KEM," in 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 2018, pp. 353–367.

[3] National Institute of Standards and Technology (NIST), "Module-Lattice-Based Key-Encapsulation Mechanism Standard," Federal Information Processing Standards (FIPS) Publication 203, Aug. 2024.

[4] R. Munir, "Teori Bilangan (Bagian 1, 2, dan 3)," Bahan Kuliah IF1220 Matematika Diskrit, Program Studi Teknik Informatika, STEI-ITB, 2026. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/matdis25-26.htm. [Accessed: 17-June-2026].

[5] O. Regev, "On lattices, learning with errors, random-linear codes, and cryptography," Journal of the ACM (JACM), vol. 56, no. 6, pp. 1-40, 2009. https:/[5] O. Regev, "On lattices, learning with errors, random-linear codes, and cryptography," Journal of the ACM (JACM), vol. 56, no. 6, pp. 1-40, 2009, doi: 10.1145/1568318.1568324.

[6]

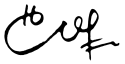
[7] [6] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange - a new hope," in 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, 2016, pp. 327-343./doi.org/10.1145/1568318.1568324

[8] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange - a new hope," in 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, 2016, pp. 327-343.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2025



Syakira Azzahra Rachmania
13525055