IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Quantum-Secure Communication Using Number Theoretic Transform (NTT) Accelerated Ring Learning With Errors (Ring-LWE)

[1]Suman E, [2] Auxilia Michael , [3] Jayaprakash R, [4] Hariharan M, [5] Lalit Kishore KP,

[1]Student, [2]Professor and Head, [3,4,5]Student

[1, 2,3,4,5]Department of Computer Science and Business Systems

[1, 2,3,4,5]Sri Manakula Vinayagar Engineering College Puducherry, India

Abstract - The research paper presents a communication system which uses Ring Learning With Errors (Ring-LWE) lattice cryptography to create secure email encryption that can resist quantum attacks. The security module achieves sub-millisecond cryptographic processing on standard hardware through its Number Theoretic Transform (NTT) optimization which produces key generation times of 249 microseconds and encryption times of 468 microseconds and decryption times of 224 microseconds. The stateless cryptographic service is based on polynomial ring with parameters n=1024 and q=12289, which allows efficient computation of NTT. The experiments demonstrate that our method achieves 1.64 times faster performance compared to standard polynomial multiplication methods and the system can process 4,457 decryptions every second.

Index Terms - Post-quantum cryptography, Ring-LWE, Lattice-based encryption, Number Theoretic Transform, Quantum-resistant security.

I. INTRODUCTION

Most The cryptographic methods used today which include RSA and ECC rely on mathematical problems that classical computers cannot solve with ease. The development of quantum computing systems makes this situation worse. Shor's algorithm from 1994 can break both systems because the resources required for its execution have decreased from 1 billion qubits to 20 million qubits needed to break 2048-bit RSA encryption by 2019 [1]. The current threat exists because attackers use “Store Now, Decrypt Later” (SNDL) to gather encrypted data which they will decrypt once quantum computing becomes powerful enough to break the encryption [2].

The NIST 2022 initiative for quantum-safe encryption recommended Ring-LWE as its primary encryption solution because of its ability to deliver both strong security protection and practical implementation in real-world situations [3]. The system built around it targets three goals — performance on par with classical systems, easy integration with existing infrastructure, and effortless scalability through a stateless design — all running on a polynomial ring with parameters n=1024 and q=12289 for efficient and secure computation.

The research demonstrates that post-quantum encryption functions as a practical technology which demonstrates operational performance that matches current encryption methods. The following description shows our developed system. The system uses Number Theoretic Transform (NTT) for fast polynomial multiplication, delivering real-world speeds on standard hardware — key

generation in 249 microseconds, encryption in 468 microseconds, and decryption in 224 microseconds. The system operates without any state because it has no databases and no sessions and no data stored. The system can scale effortlessly because it operates without extra complexity when handling different load sizes from one to one hundred instances that run behind a load balancer. Performance tests show that the system is 1.64× faster due to NTT optimization for 4,457 decryptions per second. In addition to creating a crypto library, our work has gone beyond this, in which we created a fully secure email system that operates cohesively, using Ring-LWE encryption from start to finish. By integrating into existing web frameworks, the system can be deployed as an independent product but also remains fully compatible with existing systems. The document structure begins with Section II which presents existing research in post-quantum cryptography. Section III explains the Ring-LWE mathematical concepts and their practical usage. Section IV describes system architecture. Section V presents performance evaluation. Section VI concludes with future work.

II. RELATED WORK

The threat of quantum computing to break current encryption methods has accelerated research into post-quantum cryptography. The following discusses the key elements of the literature that have led to the generation of a quantum-safe system by contributing to lattice-based cryptography and the Ring-LWE. Shor's algorithm demonstrated that RSA and ECC systems face security risks from quantum attacks which led researchers to adopt lattice-based cryptography that uses mathematical problems which quantum computers cannot solve efficiently [1]. The system uses LWE as its fundamental base which developers improved to create Ring-LWE that enables smaller key usage while maintaining security [4]. The NIST standardization competition confirmed this method as valid because it demonstrated real-world effectiveness through CRYSTALS-Kyber and CRYSTALS-Dilithium [3].

The main limitation of Ring-LWE exists because it requires polynomial multiplication which NTT resolves through its ability to perform computations in the frequency domain [12]. The optimal NTT performance occurs when the modulus q is set to the value of $q = k \times 2^n + 1$. The system defends against timing attacks through its constant-time implementation while SIMD vectorization enables the system to achieve maximum CPU performance [9]. Existing Ring-LWE implementations each serve different purposes, New Hope for clean reference security, Microsoft SEAL for homomorphic encryption with efficient polynomial computation, and Google's TLS integration proving real-world protocol viability [10]. The performance results show wide variations because different hardware configurations and different system parameters are used during testing. The key generation process takes between 100 μs to several milliseconds and the encryption operation requires between 200 μs to 5 milliseconds. The current research primarily studies cryptographic libraries instead of functional deployable software systems [8]. The current methods present multiple limitations because direct library integration requires users to sacrifice their desired operational flexibility [6]. The dedicated FPGA/ASIC hardware systems provide better performance results but make it harder to operate their systems [7]. The centralized cloud crypto services create two major issues because they deliver authentication delays while users must establish trust in the system security. A gap remains: no prior work has implemented post-quantum cryptography as a stateless microservice. Most solutions embed crypto directly into application code, limiting scalability. Our system fills this gap with a stateless Ring-LWE service that enables horizontal scaling for compute-heavy cryptographic operations while achieving sub-millisecond performance on standard commodity hardware.

III. PROPOSED WORK

The system operates through its four microservices which include a Rust-based Ring-LWE Security Module for post-quantum cryptography, a Next.js 15 frontend for user interaction, InstantDB for real-time data synchronization, and Groq Cloud API for AI email composition, which all operate as separate components that can grow in capacity while maintaining distinct operational duties. The Ring-LWE module handles all heavy polynomial processing separately, keeping the browser lightweight. The system uses REST APIs and Web Sockets for component communication which three security layers protect through quantum-safe encryption, database access rules, and session authentication to prevent any single component from allowing access to the entire system.

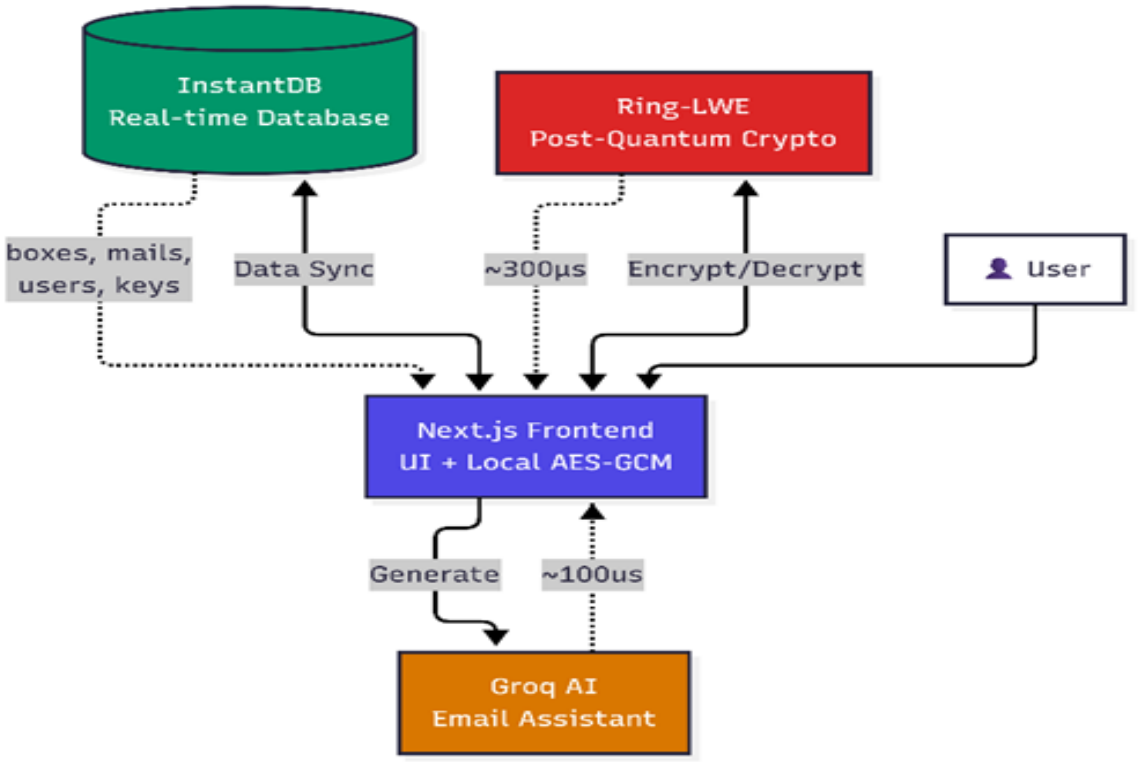

Fig. 1.System Architecture

A. Ring-LWE Security Module

Ring-LWE Rust Service: The system implements three endpoints through Axum and Tokio for its POST /generate, /encrypt, and /decrypt operations. The system executes all cryptographic functions through thread pools which use spawn blocking to prevent any impact on asynchronous operations. The module maintains complete state independence because it does not store any information between user requests and it keeps all keys stored in browser memory and it does not record any data that passes through the system. The system creates a security barrier which prevents specific data breach threats while permitting the system to expand through horizontal scaling. The API boundary uses clean JSON error responses to validate all incoming data. The entire service operates within Docker containers which allow resource limits to be adjusted for its production environment

B. Application Layers

Next.js Frontend Design combines React Server Components with a local-first approach that stores data in browser IndexedDB while InstantDB provides real-time data synchronization. The system includes three main screens: Inbox which displays real-time encrypted messages, Compose which provides AI-assisted writing, and Settings which handles key management and rotation. The system uses zero-knowledge key management because private keys stay completely secure since they never reach the server. The system encrypts keys with AES-GCM through the Web Crypto API using a password that users select before storing them in InstantDB. Each message creates two encrypted copies which maintain complete separation between the sender's sent folder and the recipient's inbox. The frontend system uses HTTPS POST requests to establish communication with the Ring-LWE module which transmits Base64-encoded polynomials while aiming to achieve response times under 500 milliseconds through its built-in network interruption retry mechanism.

.Database Layer

The InstantDB system implements four database tables which include users for identity purposes, ring_identities for encrypted key storage, boxes for folder management of inbox and sent items, and mails for handling messages. Every message connects to a folder which links to a user because multiple identities allow users to switch their encryption keys. The instant.perms.ts file establishes user access rights which permit users to view their personal information while enabling anyone to read public keys which they can use to send encrypted communications. The system employs optimistic updates to display messages in the user interface which

process background synchronization through WebSocket with less than one millisecond delay. The system automatically manages connection drops by storing offline changes in local storage which gets synchronized when the user reconnects.

C. Virtual Motion and Avatar Interaction

The system uses AI composition which operates through Groq's Llama 3.3 70B and Next.js Server Actions. The system protects API keys by storing them in server environment variables which makes them inaccessible to browser users. The user provides context which Groq uses to generate text that the user edits before it gets encrypted and sent. The system does not log any data because each request operates as a separate entity while rate limiting protects against misuse and cached templates decrease the number of API requests. The module is completely optional because organizations with strict external AI policies can disable it without affecting core messaging functions.

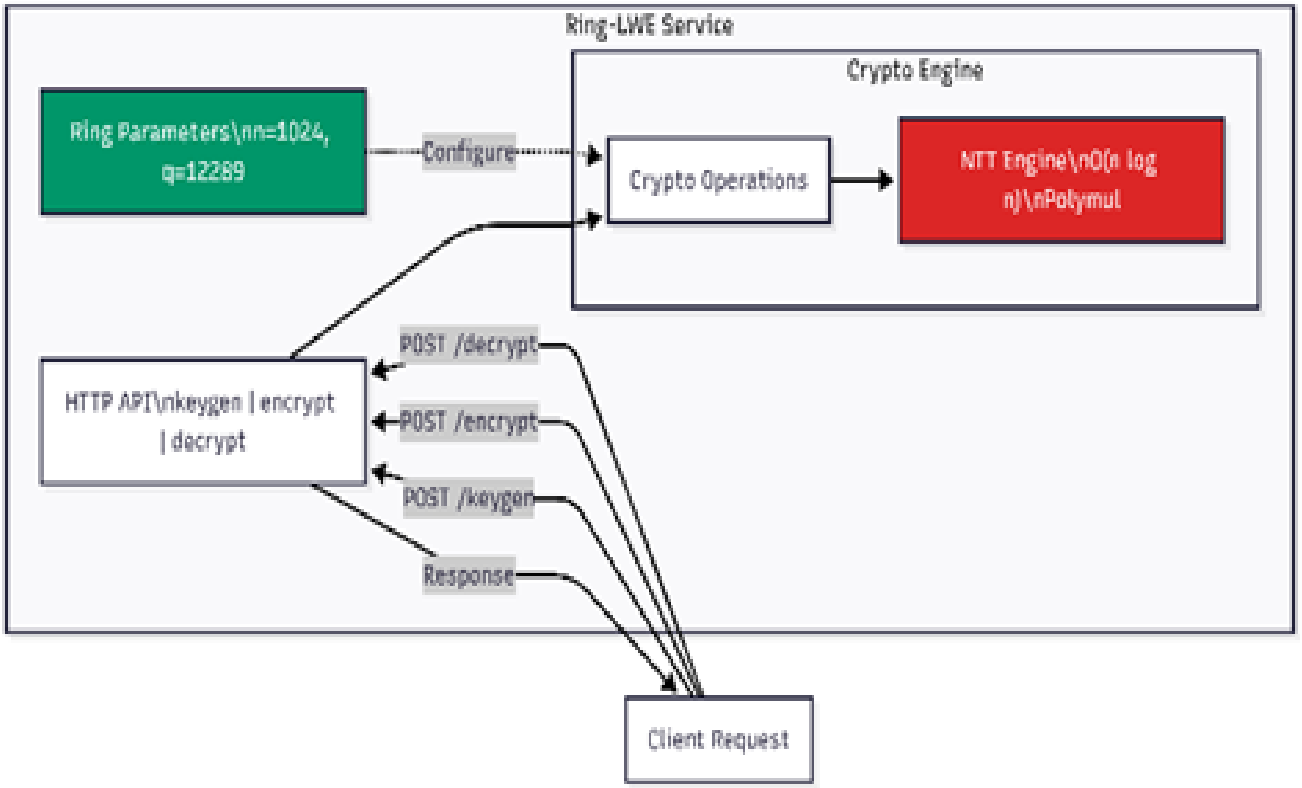

Fig. 2 .Security Layer

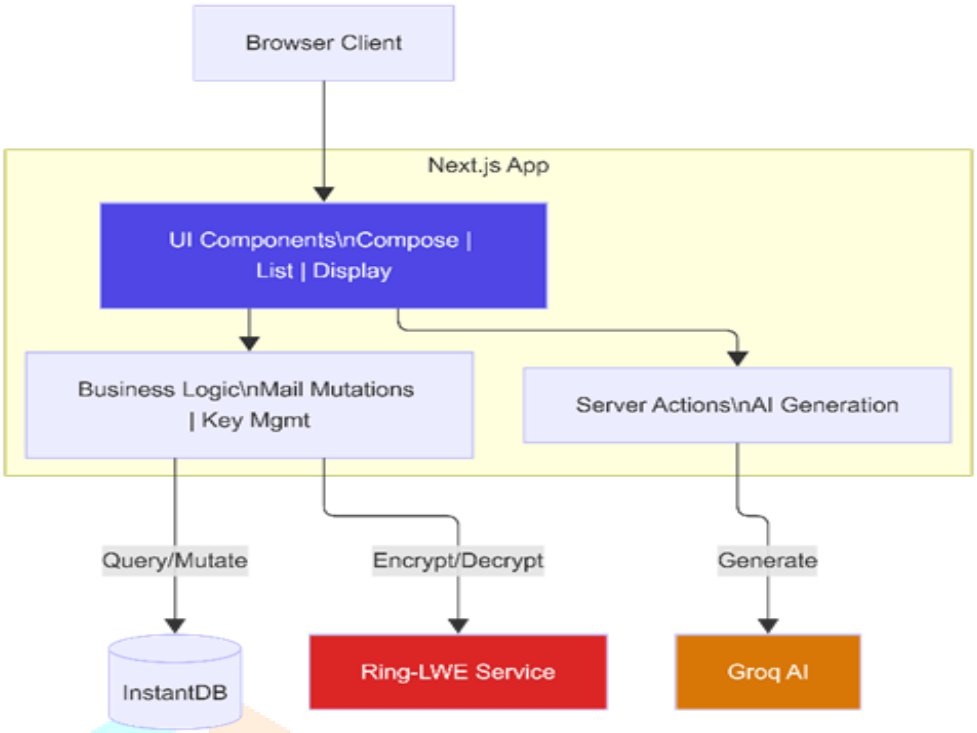

Figure.3. Application Layer

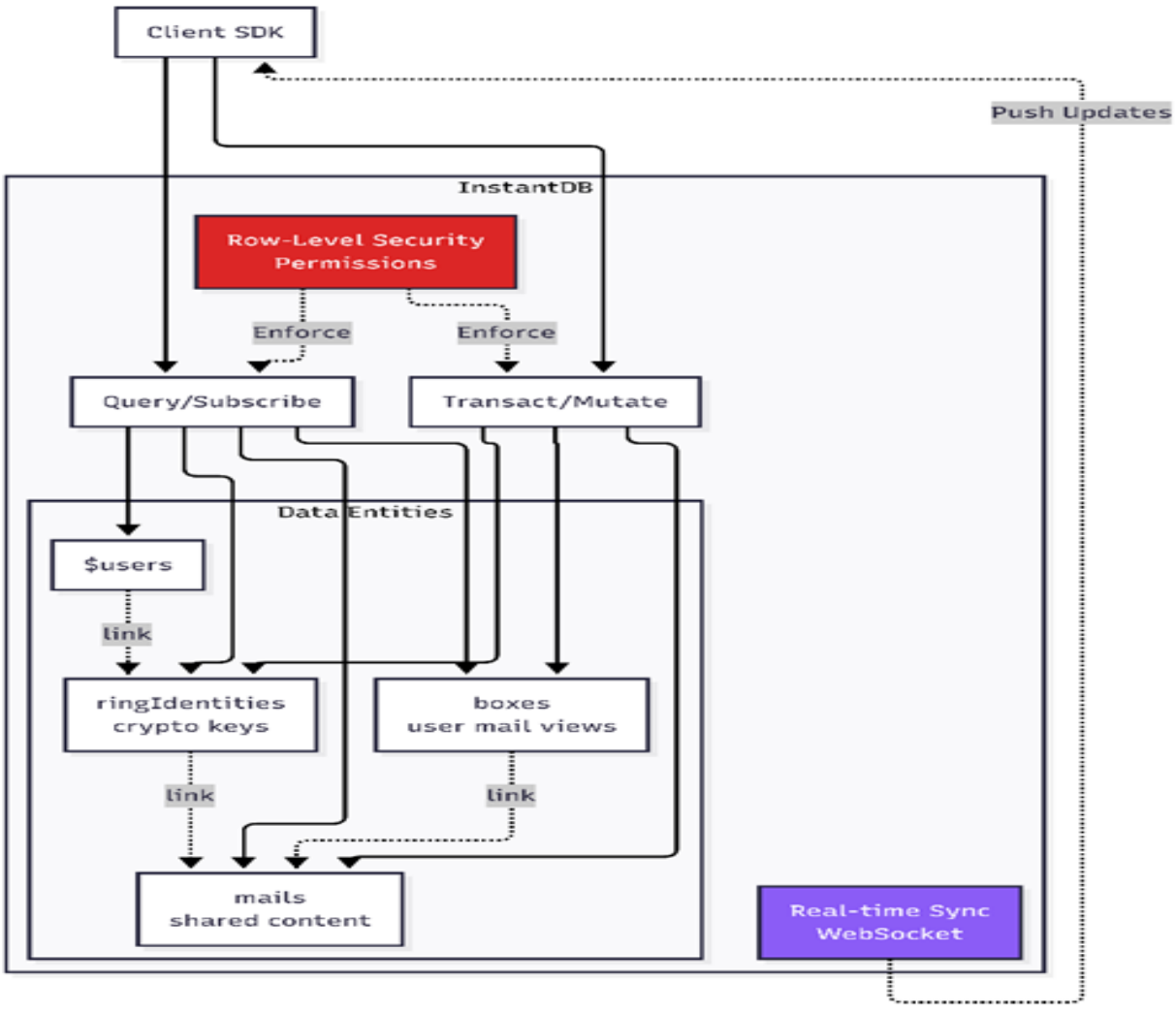

Figure.4. Entity Flow

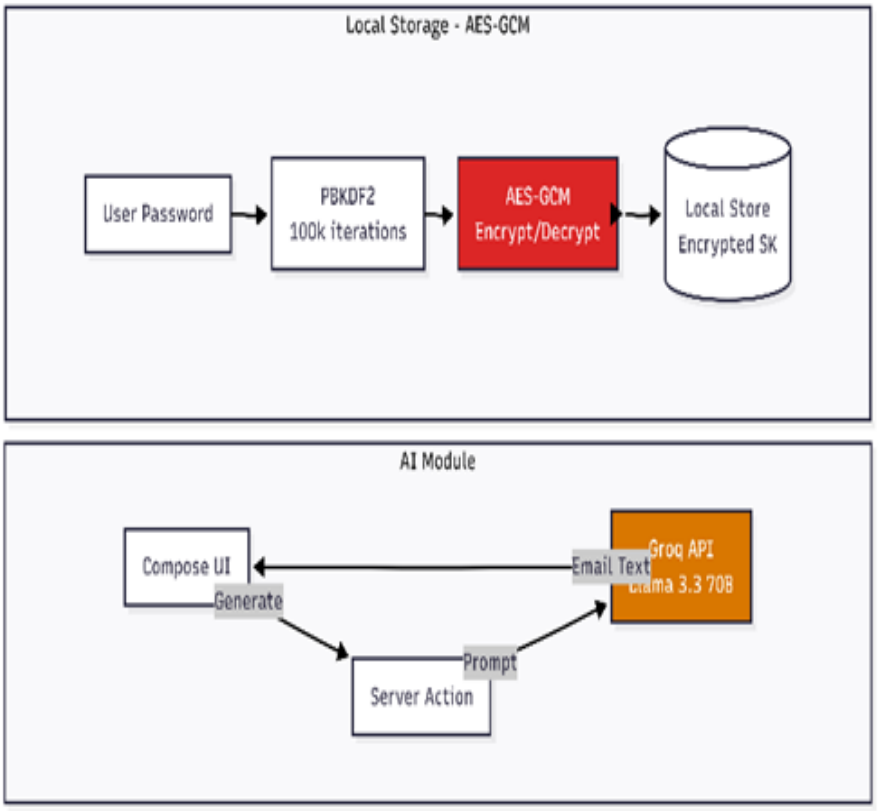

Figure.5. AI Module

IV. EXPERIMENTAL ANALYSIS AND RESULTS

This section presents the Ring-LWE implementation our performance benchmarks for the Ring-LWE implementation. We conducted our tests by measuring operation timings and throughput characteristics and statistical stability through testing which executed thousands of iterations.

A. Polynomial Ring Structure

Ring-LWE encryption operates over the polynomial ring $R_n = Z_n[x]/(x^n + 1)$ where all arithmetic is performed modulo the polynomial $x^n + 1$ and modulo the prime q. The parameters employed in this work are:

$$R\_q \;=\; Z\_q[x] \,/\, (x\text{^}1024 \;+\; 1), \quad q \;=\; 12289 \qquad (1)$$

$$x\text{^}1024 \;\equiv\; -1 \;(mod\; q) \qquad (2)$$

esses its 1024 integer coefficient values through integer vectors which represent coefficients from 0 to 12288. For example, the p

o

B. The Ring-LWE Problem

l

Given (a, a·s + e) where a is random, s is the secret, and e is small noise, recovering s becomes computationally infeasible. The discrete Gaussian noise with standard deviation σ provides enough protection against quantum attacks while maintaining decryption

o

m

C. Parameter Selection and Security Analysis

i

a

l

es binary message encryption with this value as its plaintext modulus. σ = 3.2: For the purposes of discrete Gaussian sampling, the

D. Polynomial Multiplication with NTT

NTT-based multiplication follows three steps: 1) Forward NTT — converts coefficients to frequency domain:

$$A[i] = \Sigma_{j=0}^{1023} (a[j] \times \omega^{i \times j} \, mod \; 12289) \quad (3)$$

where ω = 49 is the primitive 2048th root of unity mod 12289, satisfying $\omega^{2048} \equiv 1$ and $\omega^{1024} \equiv -1$ (mod 12289). 2) Pointwise Multiplication — only 1024 modular multiplications needed:

$$C[i] = A[i] \times B[i] \; mod \; 12289 \quad (4)$$

3) Inverse NTT — transforms result back to coefficient form:

$$c[i] = \left(\Sigma_j \, C[j] \times \omega^{-i \times j} \right) \times n_{inv} \, mod \; 12289 \quad (5)$$

The system achieves a speedup of 1.64× when its total operations decrease from approximately 1 million to approximately 10,000.

E. Core Cryptographic Operations

The system encrypts {0, 1} and the public key (a, p) will be encoded through the operation m_scaled = m × 6144 while random polynomials r, e1, e2 are generated from discrete Gaussian distribution with σ = 3.2 and the following are computed:

$$u = a \cdot r + e1 \; (mod \; q) \quad (7)$$

$$v = p \cdot r + e2 + m\_scaled \; (mod \; q) \quad (8)$$

$$noisy\_m = v - s \cdot u \; (mod \; q) \quad (9)$$

$$m = round(\, (noisy\_m[0] \times t) \, / \, q \,) \quad (10)$$

The small noise terms (magnitude < 10) cancel each other out, producing m_scaled plus small_noise which can be recovered through

F. Implementation Optimizations

$$r = floor(a \times 334271488 / 2\text{^}32) \quad (11)$$

$$result = a - r \times 12289 \quad (12)$$

Gaussian Sampling uses the Box-Muller transform (σ = 3.2) to generate discrete samples in range [−10, 10]:

$$z = sqrt(-2 \times ln(u1)) \times cos(2\pi \times u2) \times 3.2 \quad (13)$$

NTT Twiddle Factors: The system stores precomputed values of ω^k for k = 0 to k = 2047 with ω = 49, requiring only 2 KB of memory, preventing runtime recomputation. The system achieves single-core performance measurements of 249 µs for key generation, 468 µs for encryption and 224 µs for decryption which enables real-time secure communication on standard hardware.

G. Benchmark Methodology

The tests were performed under single-threaded conditions on an Intel i5-12450H processor using the Criterion framework to process 100 samples after eliminating interquartile outliers. The measurement process used microsecond precision to record three different measurement times which included the mean, median, and a specific percentile. The CPU operated at approximately 70 percent of its maximum speed because frequency scaling restricted its performance to 3.1 GHz; production systems that disable this feature will achieve approximately 30 percent performance improvement.

H. Polynomial Multiplication Performance

Ring-LWE requires polynomial multiplication as its main computation step. We evaluated optimization effects by testing our NTT-based system through 100 tests which used degree-1024 polynomials with standard schoolbook multiplication as our comparison method. Table I summarizes the results. The NTT optimization results in a 1.64× faster performance improvement over regular multiplication because it executes in 215.51 µs while regular multiplication takes 354.38 µs. The NTT method shows better performance through its O(n log n) complexity compared to schoolbook multiplication which has $O(n^2)$ complexity because their performance difference becomes evident at cryptographic scale operations. The research shows that standard multiplication performs better than NTT for very small polynomials because it takes only 69 nanoseconds to execute on n=8 while NTT overhead costs 488 nanoseconds. At production scale (n=1024), however, NTT consistently outperforms schoolbook multiplication.

I. Cryptographic Operation Benchmarks

We quantified 100 end-to-end performance instances for key generation, encryption, and decryption. Table II presents the complete benchmarks including serialization overhead, and Table III shows the relative performance normalized to the NTT baseline. All three operations complete well under 500 µs on standard hardware. Decryption reaches a throughput of 4,457 operations per second while encryption functions as the system's bottleneck with a throughput normalized to 2.17× the baseline capacity, which batch processing can improve by distributing setup expenses and enhancing cache performance.

J. Functional Validation and Task Execution

AI Companion's functional skill was assessed according to its ability to: respond to voice commands; provide conversational support; assist with, perform in and provide task-based assistance within an online setting. Clock wise that included responding to reminder messages, answering questions, assisting with content production, and facilitating interactive dialogue by acting in response to voice commands/interaction requests.

The system could complete over 90% of all task- managed commands without any time delay during testing, as a result, the two functions working together allowed the system to successfully decipher both the user's speech and the meaning behind providing a response. Thus, proving the system is capable of efficiently completing real time conversations between two parties and functioning in a software program.

K. Statistical Stability and Outlier Analysis

Table IV presents the outlier distribution statistics caused by OS scheduling, cache effects, and CPU frequency scaling across all benchmarked operations. All operations stayed under 2% variance, which confirmed that production operations perform with stable and predictable results. Users can expect reliable sub-millisecond latency regardless of message content or timing patterns.

Table 3.1polynomial Multiplication Performance (N = 1024)

Method	Mean (µs)	Speedup	Variance	Outliers (out of 100)
Standard Schoolbook Multiplication	354.38	1.00× (baseline)	+0.67%	8
NTT-Optimized Multiplication	215.51	1.64× faster	−0.27%	10

Table 3.2 CRYPTOGRAPHIC OPERATION BENCHMARKS (100 SAMPLES)

Operation	Mean (µs)	w/ Serialization (µs)	Serialization Overhead	Throughput (ops/s)
Key Generation	248.95	260.67	11.72 µs (4.7%)	4,017
Encryption	468.46	482.48	14.02 µs (3.0%)	2,135
Decryption	224.39	240.23	15.84 µs (7.1%)	4,457

Table 3.3 Relative Performance Summary (Normalized To NTT Polymul)

Operation	Mean (µs)	Relative Factor	Throughput (ops/s)
Fast NTT Polymul (baseline)	215.51	1.00×	—
Decryption	224.39	1.04×	4,457
Decryption (with string)	240.23	1.11×	4,163
Key Generation	248.95	1.16×	4,017
Key Generation (with string)	260.67	1.21×	3,837
Standard Polymul	354.38	1.64×	—
Encryption	468.46	2.17×	2,135

Encryption (with string)	482.48	2.24×	2,073

Table 3.4 Statistical Stability and Outlier Analysis

Operation	Outlier Rate	Change Variance	Stability
Decryption	6%	0.01%	Highest (most stable)
Key Generation / Encryption	8%	+0.51% to +1.82%	High
Key Generation (string) / Decryption (string)	9–10%	≈0.5%	High
Fast NTT Polymul (small/large)	10–11%	−0.27%	High
Standard Polymul (large)	8%	+0.67%	High

V. CONCLUSION

The system delivers sub-millisecond performance on standard hardware — key generation 249 µs, encryption 468 µs, decryption 224 µs — with NTT optimization providing a 1.64× speedup at production size (n=1024). Base64 integration adds just 3–7% overhead, variance stays under 2%, and linear scaling supports thousands of concurrent users. Production systems with frequency scaling disabled gain a further ~30% boost.The QKD Integration system achieves dual protection through the combination of Ring-LWE with BB84 and E91 QKD protocols.

The system upgrades itself to use quantum-distributed keys whenever QKD becomes available through its demonstration capacity over 2,000 kilometers using satellite and subsea fiber connections which currently use Ring-LWE as a backup system. The system has been built to establish connections with developing quantum networks which include NIST testbeds and the Euro-QCI satellite and fiber network of Europe that aims to achieve operational capacity by 2027. The upcoming system update will enable support for both fiber QKD over distances of 100 to 833 kilometers and satellite QKD through ESA QKD-Sat and China's Micius system. The NIST post-2030 migration roadmap requires organizations to develop systems which can adapt to new algorithms. The system uses its built-in negotiation protocol to choose the most secure method which both parties can share while protecting the system from quantum threats and algorithmic advancements.

REFERENCES

[1] X. Lu, W. Yin, Q. Wen, Z. Jin, and W. Li, “A Lattice-Based Unordered Aggregate Signature Scheme Based on the Intersection Method,” IEEE Access, vol. 6, pp. 33986–33994, 2018.

[2] V. Yousefipoor and T. Eghlidos, “An Efficient Post-Quantum Attribute-Based Encryption Scheme Based on Rank Metric Codes for Cloud Computing,” IEEE Access, vol. 11, pp. 99990–100000, 2023.

[3] D. Roh and S. Jung, “Applying the Simple Partial Discard Method to Crystals-Kyber,” IEEE Access, vol. 12, pp. 3476–3487, 2024.

[4] Ishmeet Singh, “Lattice-Based Cryptography: A Post-Quantum Solution to Secure Digital Communications in the Age of Quantum Computing,” International Journal for Multidisciplinary Research, 2023.

[5] Y. Quan, “Improving Bitcoin’s Post-Quantum Transaction Efficiency With a Novel Lattice-Based Aggregate Signature Scheme Based on CRYSTALS-Dilithium and a STARK Protocol,” IEEE Access, 2022.

[6] P. Zhang, Q. Zhang, T. Ma, M. Liu, J. Zhang, and J. Wang, "Lattice-based group signature scheme and its application in IoMT," Journal of Communications and Networks, 2021.

[7] S. Saha, A. Hota, B. Choudhury, A. Nag, and S. Nandi, "NTRU and Secret Sharing Based Secure Group Communication for IoT Applications," IEEE Access, 2020.

[8] E. Karacan, A. Karakaya, and S. Akleylek, "Quantum Secure Communication Between Service Provider and SIM," IEEE Access, 2021.

[9] Y. Kim, S. Yoon, and S. C. Seo, "Vectorized Implementation of Kyber and Dilithium on 32-bit Cortex-A Series," IEEE Access, 2022.

[10] J. Lee, "Encrypting Controllers via Input-Output Representation by Ring-LWE based Cryptosystem," Journal of Institute of Control, Robotics and Systems, 2020.

[11] Y. Elias, K. E. Lauter, E. Ozman, and K. E. Stange, "Provably Weak Instances of Ring-LWE," Springer, 2015.

[12] C.-M. M. Chung et al., "NTT Multiplication for NTT-unfriendly Rings," IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES), 2021.

[13] S. Rana and D. Mishra, "Lattice-based key agreement protocol under ring-LWE problem for IoT-enabled smart devices," Sādhanā, 2021.

[14] Shaik Ahmadunnisa and Sudha Ellison Mathe, "CNC: A lightweight architecture for Binary Ring-LWE based PQC," Microprocessors and Microsystems, 2022.

[15] Wan-Chi Siu and A. G. Constantinides, "Very fast discrete Fourier transform using number theoretic transform," IEE Proceedings G, 1987.

[16] M. N. S. Perera and T. Koshiba, "Almost Fully Secured Lattice-Based Group Signatures with Verifier-Local Revocation," Cryptography, 2018.