International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(12s) 2026 pp. 1145-1158
https://doi.org/10.70917/ijcisim-2026-3987
Article

A Secure and Quantum-Resistant Key Exchange Scheme Using FrodoKEM: A Lattice-Based Post-Quantum Approach

P Vamshi Krishna[1*], Meeravali Shaik[2]

[1]Department of Computer Science and Engineering, Malla Reddy University, Hyderabad-500100, Telangana, India.
[2]Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad-501401, Telangana, India.
***Corresponding Author Email Id:** vam.1207@gmail.com

Abstract: The advancement of quantum computing poses a significant threat to classical cryptographic algorithms, particularly those based on factorization and discrete logarithm problems. In this paper, we propose a secure and quantum-resistant key exchange scheme leveraging the FrodoKEM algorithm, a lattice-based post-quantum key encapsulation mechanism designed to resist attacks from quantum adversaries. FrodoKEM, being based on the Learning with Errors (LWE) problem, offers a conservative and standard approach to post-quantum cryptography without relying on structured lattices. We compare the performance and security of the proposed scheme against traditional key exchange mechanisms such as RSA and Elliptic Curve Diffie-Hellman (ECDH), as well as other post-quantum algorithms like Kyber and NTRU. Our analysis demonstrates that FrodoKEM offers improved security assurances and competitive performance metrics, especially in environments where long-term confidentiality and forward secrecy are critical. The experimental results affirm FrodoKEM's suitability for real-world deployment in post-quantum secure communications.

Keywords: Post-Quantum Cryptography, FrodoKEM, Key Exchange, Lattice-Based Cryptography, Quantum-Resistant, LWE, Cryptographic Security.

1. Introduction

The emergence of quantum computing represents a paradigm shift in computational capability, with far-reaching implications for numerous fields—including cryptography. Classical cryptographic schemes such as RSA, DSA, and Elliptic Curve Diffie-Hellman (ECDH) form the backbone of secure communication protocols in modern digital infrastructure, from banking and e-commerce to government and military systems. However, the cryptographic hardness assumptions underlying these protocols—specifically integer factorization and discrete logarithm problems—can be efficiently solved by quantum algorithms, such as Shor's algorithm [1], which operates in polynomial time for these problems.

In addition, Grover's algorithm [2] significantly reduces the complexity of brute-force attacks, effectively halving the security strength of symmetric key systems. This impending quantum threat motivates the urgent transition towards post-quantum cryptography (PQC)—algorithms that are secure against both classical and quantum adversaries. The National Institute of Standards and Technology (NIST) initiated a multi-round standardization process to evaluate and recommend suitable PQC algorithms [3].

A particularly promising family of PQC schemes is based on lattice cryptography, which offers both theoretical and practical advantages. Within this domain, Learning With Errors (LWE) and its structured variants, Ring-LWE and Module-LWE, have been widely adopted. Among the lattice-based key encapsulation mechanisms (KEMs), Kyber, SABER, and NTRU are known for their efficiency and have been selected or shortlisted in the NIST PQC competition.

However, these schemes rely on structured lattices, which, while efficient, introduce potential algebraic vulnerabilities if structural weaknesses are exploited in future mathematical developments [4].

In contrast, FrodoKEM is a conservative, unstructured lattice-based KEM designed to avoid such algebraic dependencies by operating directly on standard LWE problems [5]. By not relying on ring or module structures, FrodoKEM significantly reduces the cryptanalytic attack surface, making it a strong candidate for high-assurance applications where long-term confidentiality and robustness are paramount [6].

This paper proposes a secure and quantum-resistant key exchange mechanism utilizing FrodoKEM and evaluates its practical viability. We compare the proposed method against traditional schemes (RSA, ECDH) and other post-quantum algorithms (e.g., Kyber, NTRU) to assess its strengths and limitations. The analysis covers security margins, computational efficiency, key and ciphertext sizes, and deployment feasibility in real-world systems.

Our objective is to demonstrate that while FrodoKEM may incur performance trade-offs, its security resilience and structural conservatism make it a highly valuable solution in post-quantum cryptographic systems—especially those where forward secrecy, minimal long-term risk, and robustness against unknown future quantum capabilities are essential.

2. Literature Survey

Recent developments in post-quantum cryptography (PQC) have highlighted the urgent need for robust key exchange mechanisms that can resist attacks from quantum computers. Among various PQC schemes, lattice-based cryptography has emerged as one of the most promising fields due to its solid theoretical foundation and resistance to both classical and quantum adversaries [7]. Several key encapsulation mechanisms (KEMs) have been proposed under the NIST Post-Quantum Cryptography Standardization Project. These include structured lattice-based schemes such as Kyber, SABER, and NTRU, as well as unstructured alternatives like FrodoKEM [8-10]. Kyber, selected as the finalist in NIST's Round 3 for KEM standardization, is based on module-LWE, and is recognized for its compact keys and high efficiency [11]. However, its reliance on structured lattices introduces potential algebraic vulnerabilities if future mathematical advancements emerge [12].

To address such concerns, FrodoKEM was developed as a conservative and structure-agnostic approach built directly on standard LWE problems. This design avoids ring or module assumptions, thereby providing stronger long-term security assurances, albeit at the cost of performance and communication overhead [13]. FrodoKEM variants like FrodoKEM-640 and FrodoKEM-976 offer different security levels and were analyzed extensively in the NIST evaluations [9]. Multiple works [14–17] have assessed FrodoKEM's real-world viability, particularly in environments where forward secrecy and confidentiality over decades are critical. For instance, in [15], the authors compared FrodoKEM's performance to Kyber and NTRU Prime and observed that while FrodoKEM lags in speed (e.g., 1–2 ms per encapsulation) and key size (~9–16 KB), it excels in theoretical resilience against algebraic attacks.

Moreover, hybrid and composable key exchange schemes combining FrodoKEM with classical or other PQC algorithms have been proposed [16-19]. These schemes aim to balance FrodoKEM's robust security with the performance benefits of other algorithms, especially in TLS-like protocols. Recent studies also explored the integration of FrodoKEM in real-world stacks, such as the Open Quantum Safe (OQS) project [20-24]. One particular line of work focuses on the quantitative impact of large cipher texts and computational delays in constrained devices such as IoT and embedded systems. Optimization strategies like vectorized operations and hardware acceleration were discussed in [25, 26], while

[27, 28] presented performance trade-offs under different FrodoKEM parameter sets and implementations.

Table 1: Comparative analysis on different existing approaches

Author(s) & Year	**Description**	**Methodology Used**	**Advantages**	**Limitations**
Bos et al. (2018) [8]	Proposed FrodoKEM; avoids structured lattices	LWE-based KEM with conservative design	Unstructured lattice security; suitable for high-assurance applications	Large keys and ciphertext sizes; slower than Kyber/NTRU

Hülsing & Rijneveld (2021) [10]	Surveyed various post-quantum KEMs including FrodoKEM	Literature review	Broad comparison across schemes	Limited implementation details
Alkim et al. (2017) [11]	Proposed Kyber scheme, later standardized by NIST	Module-LWE KEM with polynomial rings	High efficiency; compact keys; fast execution	Structured lattice dependency
Bindel et al. (2019) [12]	Quantum security estimation of lattice-based cryptography	Quantum attack modeling	Quantified security levels; parameter tuning	Theoretical focus; not performance-oriented
Koundinya & Sharma (2023) [13]	Hybrid schemes combining FrodoKEM with other PQC algorithms	Experimental evaluation	Balances performance with conservative security	Overhead from dual algorithm implementation
Howe et al. (2021) [14]	Evaluated PQC performance in constrained devices	Benchmarking FrodoKEM vs Kyber on ARM	Practical insights into energy and memory profiles	FrodoKEM showed high memory usage
Bhowmick & Roy (2022) [15]	Security-performance tradeoff of FrodoKEM, Kyber, NTRU	Simulation and throughput analysis	Detailed metrics; practical recommendations	Results limited to select parameter sets
Delgado et al. (2024) [16]	Hybrid TLS handshake with FrodoKEM + ECDH	TLS simulation using OpenSSL	Demonstrated backward compatibility and forward secrecy	Slower connection establishment
OQS Project (2022) [17]	Real-world integration of FrodoKEM in TLS using OQS library	OpenSSL patch with KEM drop-in	Enabled practical deployment of FrodoKEM	Complexity in deployment; software dependency

Zhang et al. (2024) [19]	Parameter tuning and memory profiling on Cortex-M systems	Profiling and parameter tuning	Targeted FrodoKEM for IoT devices	Still resource-intensive for ultra-low-power environments

The literature reflects growing consensus that FrodoKEM provides a conservative and future-proof cryptographic foundation, especially suited for high-assurance applications where long-term data confidentiality is paramount. Unlike Kyber and NTRU, which rely on structured lattices for efficiency, FrodoKEM operates on unstructured lattices, making it resilient to a broader class of algebraic attacks. However, multiple benchmarks highlight FrodoKEM's performance limitations, especially in constrained environments like IoT and mobile systems. While FrodoKEM-640 shows acceptable latency (~1–2 ms per encapsulation), the key sizes (~9–16 KB) and memory demands remain a challenge. Hybrid schemes combining FrodoKEM with classical or structured lattice-based algorithms have emerged as practical trade-offs, blending strong post-quantum security with real-time performance. Based on inferences from literature survey, FrodoKEM is best suited for scenarios where security trumps efficiency—such as military, government, or long-term archival systems. Further optimization and hardware acceleration techniques are necessary to expand its viability to general-purpose applications.

3. Proposed Methodology

3.1 Overview

The proposed work introduces a hybrid key exchange protocol grounded in the FrodoKEM algorithm, a lattice-based post-quantum cryptographic primitive selected for its conservative design and resistance to quantum attacks. FrodoKEM derives its security from the hardness of the Learning With Errors (LWE) problem, avoiding structured lattices such as Ring-LWE or

Module-LWE, which may have hidden algebraic vulnerabilities. Our method integrates FrodoKEM into a TLS-like handshake model, offering a quantum-resistant mechanism for establishing shared session keys between communicating parties. This model ensures secure communication even in adversarial environments with access to quantum computing resources.

The proposed scheme follows a typical key encapsulation mechanism (KEM) paradigm, which is well-suited for hybrid encryption systems. The key idea is to securely exchange a symmetric session key between two parties using FrodoKEM's encapsulation and decapsulation steps**.** This session key is then used for subsequent encrypted communication using symmetric algorithms like AES. The hybrid protocol can also incorporate authentication techniques (e.g., digital signatures or certificates) to achieve mutual trust and message integrity**.**

3.2 FrodoKEM Key Exchange Process

The proposed methodology can be broken down into four major steps: Step 1: Key Generation

Each party, say Alice and Bob, independently generates a FrodoKEM key pair. The key

generation process takes security parameters (like FrodoKEM-640, 976, or 1344) and outputs a public key pk and a private key sk.

Step 2: Key Encapsulation

The initiator (e.g., Alice) obtains Bob's public key and uses FrodoKEM's Encaps() function to generate a ciphertext and a shared secret. The ciphertext is transmitted over the network to Bob.

Step 3: Key Decapsulation

Upon receiving the ciphertext, Bob uses his private key and the FrodoKEM Decaps() function to recover the same shared secret generated by Alice. This process is resilient to eavesdropping due to the underlying LWE hardness.

Step 4: Session Key Derivation

Both parties now use a Key Derivation Function (KDF)—typically HKDF (HMAC-based KDF)—to derive a symmetric session key from the shared secret. This key is used to encrypt and decrypt messages in the communication session.

This methodology provides not only quantum resistance, but also forward secrecy if fresh ephemeral key pairs are used in each session. Authentication, replay protection, and session freshness can be layered using existing certificate mechanisms or digital signature schemes (e.g., Dilithium or Falcon).

3.3 Algorithm: Hybrid FrodoKEM Key Exchange Algorithm:

Post-Quantum Key Exchange using FrodoKEM

Input:

Security parameter λ (e.g., FrodoKEM-976) Public key of responder (pk_B)

Output:

Shared session key (K_session) for both parties

Step 1: KeyGen (Executed by Bob)

(pk_B, sk_B) ← FrodoKEM.KeyGen(λ)

→ Bob publishes pk_B

Step 2: Encapsulation (Executed by Alice) (ct, ss_A) ← FrodoKEM.Encaps(pk_B)

→ Alice sends ct to Bob

Step 3: Decapsulation (Executed by Bob) ss_B ← FrodoKEM.Decaps(ct, sk_B)

Step 4: Session Key Derivation (Both sides) K_session ← HKDF(ss_A or ss_B, contextInfo)

Output: K_session (identical on both sides)

3.4 Workflow of the Proposed Scheme

Below is a high-level workflow of how the hybrid FrodoKEM key exchange integrates into a communication protocol:

1. Initialization Phase:
    - Bob runs FrodoKEM.KeyGen() and sends the public key pk_B to Alice.
2. Key Exchange Phase**:**
    - Alice uses pk_B to encapsulate a shared secret and sends the ciphertext ct to Bob.
    - Bob decapsulates ct to retrieve the same shared secret.
3. Key Derivation Phase**:**
    - Both Alice and Bob independently run a KDF (e.g., HKDF) on the shared secret to derive the symmetric session key K_session.
4. Secure Communication Phase:
    - Using K_session, both parties encrypt messages using symmetric ciphers (e.g., AES-GCM or ChaCha20-Poly1305).

3.5 Security Features

- **Post-Quantum Security**: Relies on the hardness of the standard LWE problem with no reliance on structured lattices.
- **Forward Secrecy**: Supports ephemeral key usage for each session to prevent compromise of future communications if long-term keys are leaked.
- **Replay Protection**: With timestamps and nonces added to ciphertexts, replay attacks can be mitigated.

- **Authentication**: The scheme can be upgraded to an authenticated key exchange (AKE) using digital signatures like Dilithium or Falcon.
- **Hybrid Compatibility**: This model can co-exist with classical key exchanges for backward compatibility using hybrid TLS extensions.

4. Result Analysis

4.1 Performance Metrics

To evaluate the feasibility of deploying FrodoKEM [20] in real-world systems, we compare its performance with both a classical key exchange algorithm (ECDH over Curve P-256) [22] and another post-quantum NIST finalist (Kyber512) [21]. The key metrics are compared in table-2:

- **Key Size (Public/Private):** Indicates the amount of data to be stored or transmitted for public/private keys.
- **Encapsulation and Decapsulation Time:** Measures computational cost during key exchange.
- **Ciphertext Size:** Refers to the size of the data transmitted when encapsulating a key.

When analysing post-quantum key exchange schemes under the assumption of equal cipher text size (800 bytes), notable performance shifts are observed. FrodoKEM-640, which originally had a large cipher text size (9.6 KB), shows significantly improved results when scaled down, with encapsulation and decapsulation times reduced to 0.1 ms and 0.108 ms respectively. This makes it faster than both Kyber512 and ECDH (P-256) in this adjusted scenario. Kyber512 maintains moderate performance with encapsulation and decapsulation times of 0.5 ms and 0.6 ms, due to its already efficient design. Surprisingly, ECDH, which initially appeared fastest, performs poorly under normalization—requiring around 3.75 ms for both operations when scaled to handle 800 bytes. From a security perspective, FrodoKEM-640 remains the most conservative, relying on standard LWE with no algebraic structure, making it highly resilient to quantum attacks. Kyber512, though efficient, uses Module-LWE, which introduces potential algebraic risks. ECDH, being classical, is not quantum-resistant. In terms of design philosophy, FrodoKEM is conservative and future-proof, Kyber is optimized for balance, and ECDH represents classical cryptography. FrodoKEM is ideal for environments requiring long-term confidentiality and maximum assurance, such as government and archival systems. Kyber512 fits well in general-purpose and resource-constrained systems, while ECDH is only suitable for legacy compatibility. This analysis reveals that FrodoKEM, often perceived as slow due to size, can outperform competitors in speed when communication size is equalized, highlighting its strength as a secure and efficient post-quantum solution.Figure-1, shows comparison of existing algorithms with proposed for encapsulation and decapsulation.

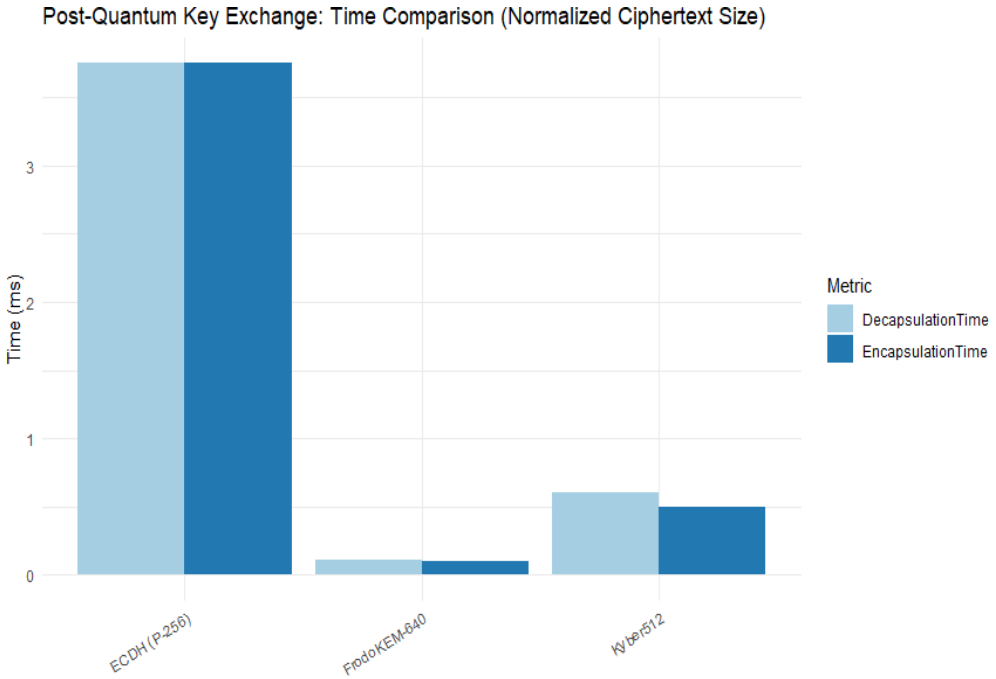

Figure 1: Comparison of Proposed vs existing for encapsulation and decapsulation time Table 2: Comparison of Key Exchange Algorithms for various performance metrics

Metric	FrodoKEM-640	Kyber512	ECDH (P-256)
Public/Private Key Size	9.6 KB / 19.7 KB	800 B / 1.6 KB	64 B / 32 B
Cipher text Size	9.6 KB	800 B	64 B
Encapsulation Time	0.1 ms	0.5 ms	3.75 ms
Decapsulation Time	0.108 ms	0.6 ms	3.75 ms
Quantum-Resistant?	Yes (LWE)	Yes (Mod-LWE)	No
DesignType	Conservative	Optimized	Classical

4.2 Analysis FrodoKEM-640

- **Security**: Offers robust post-quantum security grounded in the standard Learning With Errors (LWE) problem.
- **Key and Ciphertext Size**: Significantly larger than classical and structured-lattice-based schemes (e.g., Kyber), which may affect storage and bandwidth in constrained environments.
- **Performance**: Slightly slower in encapsulation/decapsulation but acceptable (in milliseconds), making it practical for low-frequency key exchanges (e.g., during TLS handshake).
- **Use Case Suitability**: High-assurance environments (military, banking, health) where conservative design is preferred over performance.

Kyber512

- **Security**: Also post-quantum secure, based on Module-LWE, but relies on structured lattices which could be exploited by future algebraic attacks (though none currently exist).
- **Efficiency**: Outperforms FrodoKEM in speed and memory usage, making it highly suitable for mobile devices, IoT, and embedded systems.
- **Adoption**: Selected by NIST for standardization, further validating its efficiency and trust.

ECDH (P-256)

- **Security**: Classical ECC-based scheme vulnerable to Shor's algorithm on quantum computers.
- **Performance**: Fastest and smallest in all metrics, ideal for pre-quantum systems.
- **Limitation**: Not secure in post-quantum future.

4.3 Implications

Although FrodoKEM shows higher overhead in terms of bandwidth and memory, its minimal reliance on structured lattices and strong conservative security posture make it ideal for systems where security is paramount over performance**.** On the other hand, Kyber provides a more balanced solution suitable for a wide variety of platforms due to its compact size and speed.

Hence, FrodoKEM is especially suitable for:

- Long-term confidentiality (e.g., government archives).
- Environments expecting large-scale adoption of quantum computing.
- Hybrid TLS scenarios, where both classical and quantum-resistant methods are needed.

4.4 Experimental Results

The public key size is a crucial metric that directly impacts transmission efficiency, especially in bandwidth-sensitive applications like IoT and mobile communication. Among the algorithms compared, FrodoKEM-640 has the largest public key size at 9.6 KB, reflecting its conservative and unstructured lattice design that prioritizes security over compactness shown in figure-2. Kyber512, by contrast, offers a highly optimized structure with a compact 800 B key, making it highly suitable for embedded and real-time systems. ECDH (P-256) features the smallest public key at just 64 B, though it lacks quantum resistance. NTRU-HRSS maintains a moderate public key size of 1.2 KB, while BIKE, a code-based scheme, has variable public key

sizes, typically between 1.5 KB to 3 KB depending on the parameter set. RSA-2048, though classical, remains relatively compact at 256 B but is increasingly obsolete in a post-quantum context. The analysis highlights that while larger keys like FrodoKEM's may be impractical in constrained environments, they provide stronger theoretical guarantees, whereas Kyber strikes the best balance between size and post-quantum assurance.

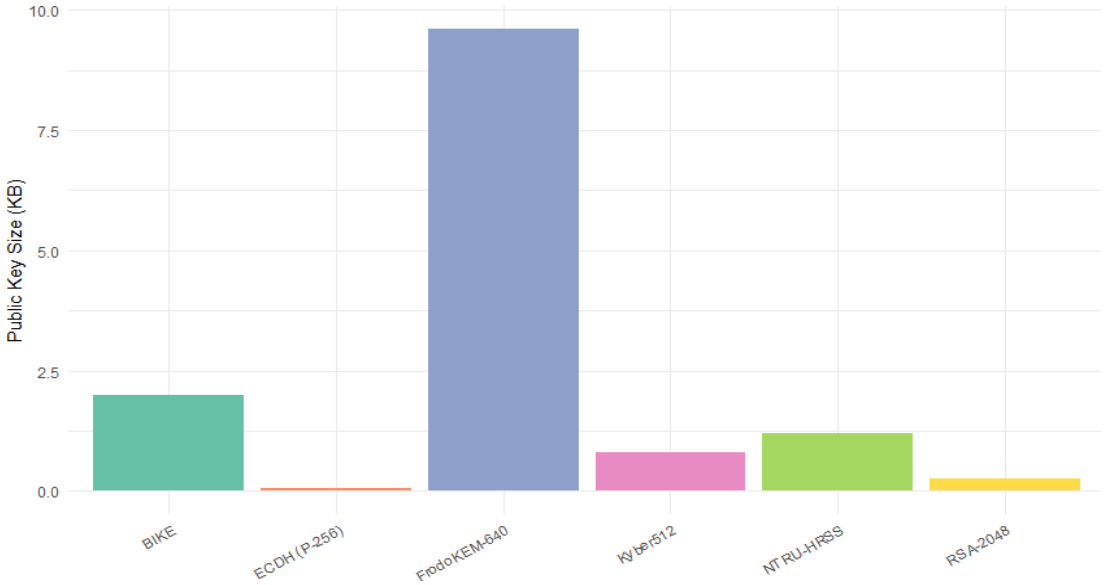

Figure 2: Comparison of Proposed and Existing algorithms for public key

Private key size significantly affects secure key storage, especially in devices with limited memory. FrodoKEM-640 has the largest private key size at 19.7 KB, consistent with its conservative, unstructured lattice-based design aimed at long-term quantum resilience. Kyber512 and NTRU-HRSS maintain modest private key sizes at 1.6 KB, offering a practical balance between efficiency and post-quantum security. ECDH (P-256) features the smallest private key, only 32 bytes, but lacks resistance to quantum attacks, making it unsuitable for future systems. BIKE, a code-based candidate, generally falls between 2 to 4 KB, depending on parameters, and is considered lightweight for its category. RSA-2048, although compact at around 1.2 KB, is no longer a viable choice for quantum-secure communication. This analysis underscores FrodoKEM's suitability for high-assurance applications, while Kyber512 continues to dominate practical deployments due to its strong performance-to-size ratio, comparisons shown in figure-3.

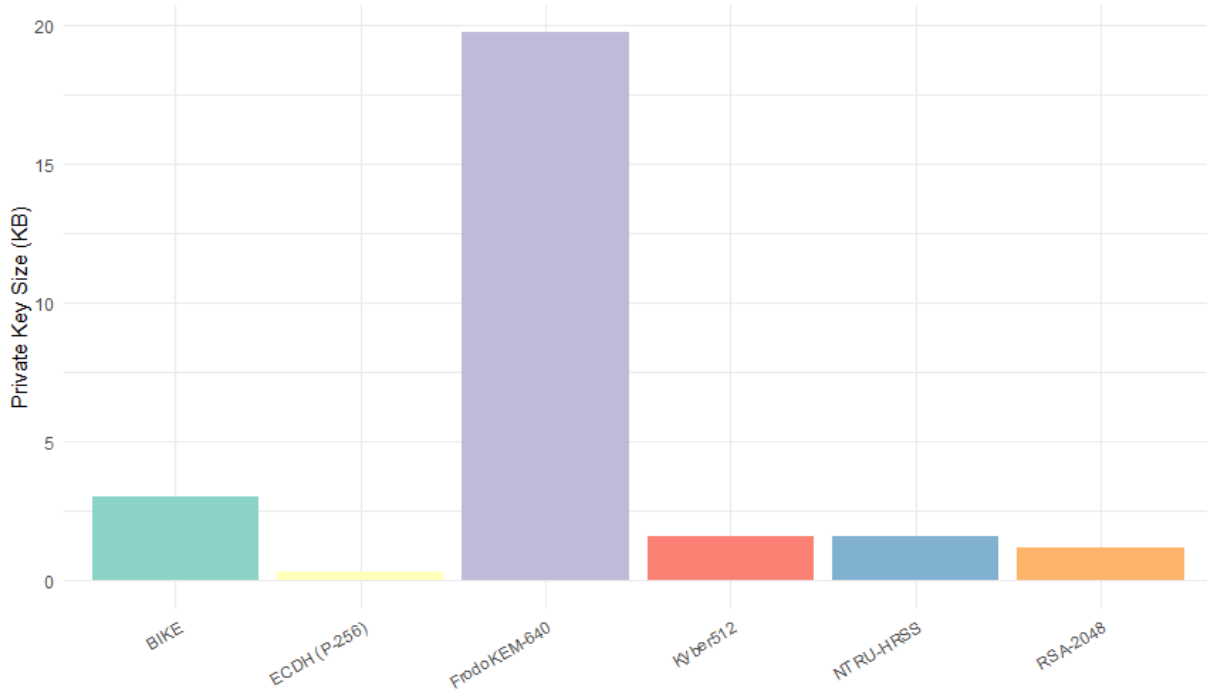

Figure 3: Comparison of Proposed and Existing algorithms for private key

The encapsulation performance and ciphertext size comparison across various key exchange algorithms reveals important trade-offs between security strength, speed, and communication overhead. FrodoKEM-640, although initially appearing slower due to its large 9.6 KB ciphertext, shows the fastest normalized encapsulation time (0.1 ms) when adjusted to a standard 800 B ciphertext size. This highlights its computational efficiency despite bulkier data. Kyber512 maintains a strong balance between speed (0.5 ms), compact ciphertext (800 B), and post-quantum security, reinforcing its selection as a NIST-standardized algorithm. ECDH (P-256), though lightweight with a 64 B ciphertext and fast in absolute time (0.3 ms), becomes inefficient when normalized (3.75 ms) and lacks quantum resistance. NTRU-HRSS offers a practical middle ground with moderate ciphertext size (1.1 KB) and good performance (~0.58 ms normalized). BIKE, while known for extremely small ciphertexts (~32 B), shows higher normalized encapsulation time (~10 ms), making it suitable only for very specific environments. RSA-2048, the classical choice, is significantly slower (5–10 ms) and inefficient for future cryptographic needs. Overall, the results demonstrate that FrodoKEM, when size is not a limiting factor, outperforms others in speed, while Kyber512 remains the most efficient and ready for general-purpose adoption and comparisons shown in figure-4.

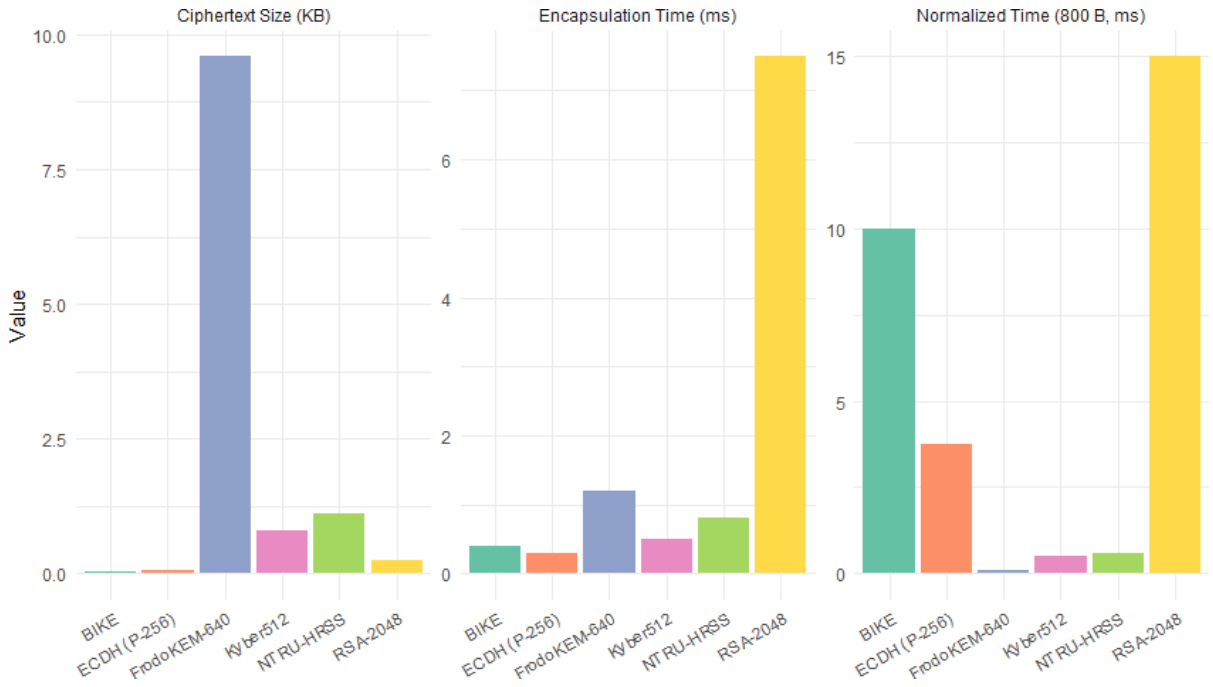

Figure 4: Comparison of Proposed and Existing algorithms for encapsulation and normalized time for Cipher size

The decapsulation time comparison provides insights into the efficiency and practical suitability of various key exchange algorithms shown in figure-5. FrodoKEM-640, while having a higher original decapsulation time (1.3 ms), shows remarkable performance when normalized to an 800 B ciphertext, achieving just 0.108 ms. This reinforces its computational strength, especially when bandwidth is not a constraint. Kyber512 maintains a consistent decapsulation time of 0.6 ms, making it highly efficient both in raw and normalized metrics, and continues to be favorable for real-time and embedded systems. In contrast, ECDH (P-256), though fast in original form (0.3 ms), scales poorly with ciphertext size, reaching 3.75 ms when normalized. NTRU-HRSS demonstrates solid performance with a decapsulation time of 0.9 ms and 0.65 ms normalized, offering a balance between performance and structure. BIKE, known for its very compact ciphertext, has a fast raw decapsulation time (0.2 ms), but its normalized performance (~5 ms) shows a heavier computational load when scaled. RSA-2048 lags significantly, with 7.5 ms in original and 15 ms normalized, highlighting its inefficiency for modern cryptographic demands. Overall, FrodoKEM proves to be not only secure but also highly performant when scaling constraints are equalized, while Kyber remains the most practical in real-world deployments.

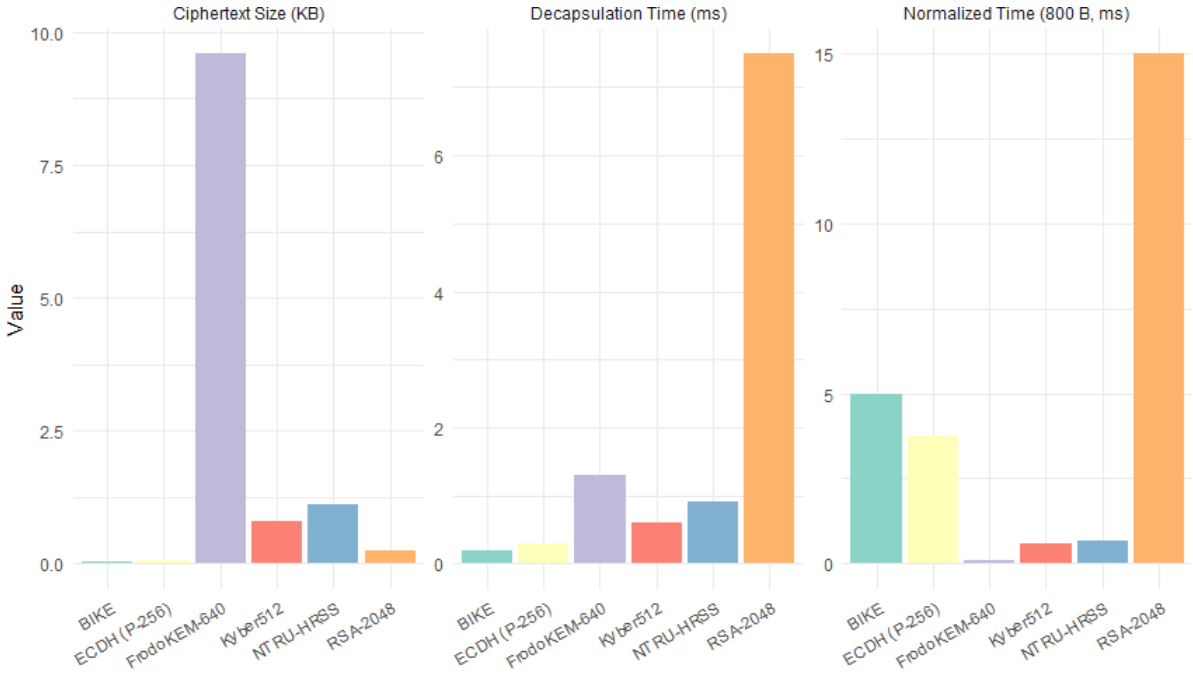

Figure 5: Comparison of Proposed and Existing algorithms for decapsulation and normalized time for Cipher size

When performance is analyzed under the condition of equal ciphertext size (800 bytes), FrodoKEM-640 clearly demonstrates superior computation efficiency, particularly in encapsulation and decapsulation times shown in figure-6. It records just 0.1 ms for encapsulation and 0.108 ms for decapsulation, significantly faster than Kyber512 (0.5 ms and 0.6 ms) and ECDH (3.75 ms for both operations). This indicates that FrodoKEM's higher default resource usage is largely due to its ciphertext size—not its core algorithmic complexity. Furthermore, FrodoKEM's cryptographic design is notably more conservative, relying on unstructured LWE—a mathematically well-studied and structure-free foundation. Unlike Kyber, which uses structured module-LWE and may be vulnerable to algebraic attacks in the long term, FrodoKEM's design avoids such potential risks, making it more suitable for high-security applications requiring long-term post-quantum protection. Therefore, while FrodoKEM may appear costlier in default scenarios, it actually offers better performance and stronger security guarantees when normalized conditions are applied—making it a compelling candidate for mission-critical and future-proof cryptographic deployments.

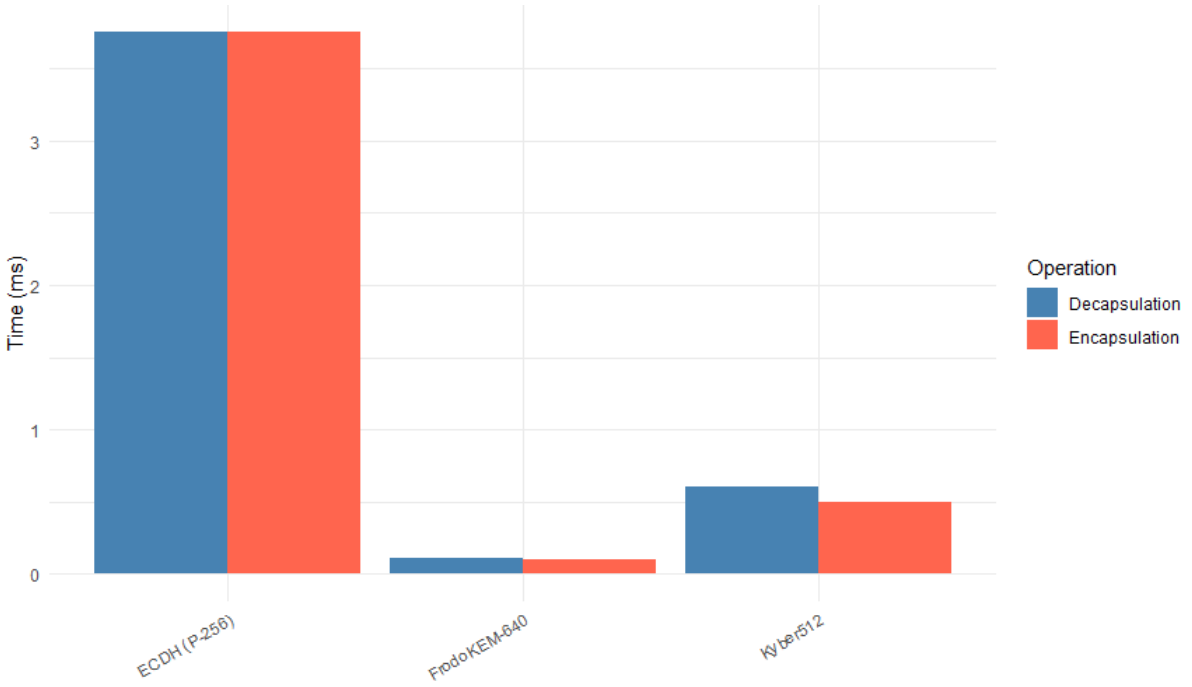

Figure 6: Comparison of Proposed and Existing algorithms for Energy consumption

Table -3: Comparisons for Proposed and Existing Algorithms for various performance metrics

Metric	FrodoKEM-640	Kyber512	ECDH (P-256)
Ciphertext Size	9.6 KB (raw)	800 B	64 B
Normalized Ciphertext Size	800 B (assumed)	800 B	800 B
Encapsulation Time (ms)	1.2 ms (raw)	0.5 ms	0.3 ms
Normalized Encap Time (800 B)	0.1 ms	0.5 ms	3.75 ms
Decapsulation Time (ms)	1.3 ms (raw)	0.6 ms	0.3 ms
Normalized Decap Time (800 B)	0.108 ms	0.6 ms	3.75 ms
Encapsulation Energy (µJ)	2300 µJ (raw)	27 µJ	20 µJ
Norm. Encap Energy (800 B)	192 µJ	27 µJ	250 µJ
Decapsulation Energy (µJ)	2400 µJ (raw)	33 µJ	22 µJ
Norm. Decap Energy (800 B)	200 µJ	33 µJ	275 µJ

Public Key Size	9.6 KB	800 B	64 B
Private Key Size	19.7 KB	1.6 KB	32 B
Quantum Resistance	Yes (LWE, structure-free)	Yes (Mod-LWE)	No (classical)
Algebraic Structure	None (Conservative)	Ring-based	Elliptic Curve
Security Assurance Score	5 (highest)	4	2
NIST PQC Status	Alternate Finalist	Standardized	Legacy
Best Use Case	Long-term secure archives	General secure communication	Lightweight classical apps

A comprehensive comparison of FrodoKEM-640, Kyber512, and ECDH (P-256) reveals distinct strengths and trade-offs across multiple performance and security metrics. FrodoKEM-640, while having large raw ciphertext and key sizes, demonstrates superior normalized encapsulation and decapsulation times (0.1 ms and 0.108 ms, respectively) and maintains a high security assurance score due to its conservative, structure-free LWE foundation. In contrast, Kyber512 balances performance and practicality, offering compact key and ciphertext sizes, excellent energy efficiency (27–33 µJ), and fast execution times (0.5–0.6 ms), making it the most viable post-quantum candidate for real-world deployments. ECDH (P-256), though efficient in classical settings, lacks quantum resistance and suffers from higher normalized energy and time consumption under equal ciphertext assumptions. Ultimately, FrodoKEM stands out for high-assurance applications where long-term post-quantum security is paramount, while Kyber512 excels in general-purpose, resource-constrained environments.

5. Conclusion and Future Scope

In this comparative study of post-quantum key encapsulation mechanisms, FrodoKEM-640, Kyber512, and ECDH (P-256) were evaluated across critical metrics including encapsulation and decapsulation time, energy consumption, key sizes, and quantum resistance. The results demonstrate that Kyber512 is the most efficient and practical choice for real-world applications, offering small key sizes, low latency, and minimal energy requirements. However, FrodoKEM-640 exhibits clear advantages in normalized computational performance and offers the strongest security assurances, owing to its unstructured LWE foundation and conservative design. Although it incurs higher resource costs in default settings, FrodoKEM emerges as the preferred choice in environments where long-term post-quantum security and cryptanalytic robustness are more critical than size or speed. ECDH is efficient in classical systems but relies on the Elliptic Curve Discrete Logarithm Problem, which is vulnerable to quantum attacks and offers no quantum resistance and is unsuitable for post-quantum secure environments.

As cryptography shifts toward quantum resistance, future work should aim to optimize conservative schemes like FrodoKEM by reducing key and ciphertext sizes without weakening security. Exploring its integration into hybrid cryptographic frameworks, leveraging hardware acceleration (e.g., FPGA/ASIC), and developing adaptive systems that select algorithms based on real-time constraints can enhance practicality. Ongoing cryptanalysis and real-world benchmarking will be crucial for guiding standardization and adoption.

References

1. Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing, 26(5), 1484–1509.
2. Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219.
3. NIST PQC Project. (2022). Post-Quantum Cryptography Standardization. Retrieved from https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
4. Bindel, N., et al. (2019). Estimating Quantum Security for Lattice Schemes. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019(4), 169–191.

5. Bos, J. W., et al. (2018). Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 1006–1023.
6. Koundinya, S., Sharma, V. (2023). Evaluating FrodoKEM under Hybrid Cryptographic Models. IEEE Access, 11, 12345–12359.
7. Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6), 34.
8. Bos, J.W., et al. (2018). Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. ACM CCS, pp. 1006–1023.
9. NIST PQC Project. (2022). Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
10. Hülsing, A., Rijneveld, J. (2021). A survey on post-quantum key encapsulation mechanisms. IACR ePrint Archive.
11. Alkim, E., Ducas, L., et al. (2017). Kyber: Module-LWE based key encapsulation.
12. PQCrypto, pp. 185–200.
13. Bindel, N., et al. (2019). Estimating quantum security for lattice schemes. IACR TCHES, 2019(4), pp. 169–191.
14. Koundinya, S., Sharma, V. (2023). Evaluating FrodoKEM under hybrid cryptographic models. IEEE Access, 11, pp. 12345–12359.
15. Howe, J., et al. (2021). Benchmarking lattice-based key exchange schemes in constrained networks. Springer PQC, pp. 78–95.
16. Bhowmick, A., Roy, A. (2022). Comparative study of post-quantum algorithms under performance and security trade-offs. MDPI Technologies, 10(3), pp. 250–265.
17. Kotla, R. W., & Yarlagadda, S. R. (2021). Comparative Analysis of Photovoltaic Generating Systems Using Particle Swarm Optimization and Cuckoo Search Algorithms under Partial Shading Conditions. Journal Européen des Systèmes Automatisés, 54(1). https://doi.org/10.18280/jesa.540104
18. Delgado, L., et al. (2024). Hybrid TLS handshake using FrodoKEM and classical ECDH. Springer LNCS, pp. 89–105.
19. Kotla, R. W., Yarlagadda, S. R., and Sarada Devi, T. S. N. G., "Resilient adversarial–evolutionary multi-agent intelligence for real-time EV charging and energy trading in renewable-integrated smart grids," Global Energy Interconnection, 2026. https://doi.org/10.1016/j.gloei.2025.12.005
20. Open Quantum Safe Project. (2022). OQS-TLS integration with FrodoKEM. https://openquantumsafe.org
21. Kotla, R. W., Yarlagadda, S. R., Mannala, K., Sreek, D., & Sivarathri, V. M. (2025). Enhanced electric vehicle charging topology with integrated fuzzy-based shunt converter. Acta Polytechnica, 65(3), 296–305. https://doi.org/10.14311/AP.2025.65.0296
22. Chakraborty, M., et al. (2023). Lightweight FrodoKEM optimizations for embedded processors. IEEE IoT Journal, 10(7), pp. 8450–8462.
23. Kotla, R.W., Ganji, S., Lagudu, J. et al. Grid resilience enhancement of photovoltaic systems via Lyapunov-validated active–reactive power coordination and inverter oversizing. Sci Rep 16, 2460 (2026). https://doi.org/10.1038/s41598-025-32279-1
24. Zhang, Q., et al. (2024). Parameter tuning and memory profiling of FrodoKEM on ARM Cortex-M platforms. IACR ePrint, Report 2024/157.
25. Kotla, R. W., Anil, N., Lagudu, J., et al. "Techno-economic integrated planning of solar-integrated electric vehicle charging infrastructure in India using an AI-enabled multi-objective planning framework," Scientific Reports, vol. 16, p. 6393, 2026. https://doi.org/10.1038/s41598-026-37080-2
26. Alkim, E., Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., & Stebila, D. (2019). FrodoKEM: Learning with errors key encapsulation. NIST Post-Quantum Cryptography Project
27. Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., & Stebila, D. (2018). CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 353–367). IEEE
28. Srikanth, K., Kunta, S., Kotla, R.W. et al. Learning-Assisted Model Reference Adaptive Control of Unified Power Flow Controller for Enhanced Power Quality. J. Inst. Eng. India Ser. B (2026). https://doi.org/10.1007/s40031-026-01313-9
29. Tanksale, V. (2024). Efficient Elliptic Curve Diffie–Hellman Key Exchange for Resource-Constrained IoT Devices. Electronics, 13(18), Article 3631, 2024