

Benchmarking and Integration of NIST Post-Quantum Cryptography on Android Devices

Snehil Chatterjee¹, Aurum Joshi², Deepak Kumar N¹, Aarthi Musku², Reena Monica P², Pattabiraman V^{1*}, Rajesh Kumar Panda³, Bipin Makhanlal Jadav³

¹School of Computer Science and Engineering, VIT Chennai

²School of Electronics Engineering, VIT Chennai

³Samsung R&D Institute India – Bangalore

Abstract. In recent years, there has been spontaneous development in practical quantum computing. This poses a serious risk to today's cryptographic infrastructure, especially on mobile platforms. To address this, new Post Quantum Cryptographic algorithms which are safe to quantum computing have been created. In this paper, we explored the real-world deployment plausibility of NIST-standard post-quantum algorithms in mobile platforms, namely CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+, and found them to perform well within deployment requirements with respect to execution time and CPU utilization. Along with the testing, the paper also explores practical optimization techniques, such as native code offloading via NDK, asynchronous processing, idle-time batching, to be used alongside the algorithms for improved latency and lower energy consumption. The paper deployed these algorithms in two prototype applications, one using an end-to-end encryption app that combines symmetric AES encryption with PQ key encapsulation, and an application for encrypting files on the device with quantum algorithms. The paper proposes guidelines and a roadmap for developers and researchers aiming to create a secure environment on mobile devices against quantum threats.

Index Terms—Quantum-Safe Cryptography, Android security, CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, SPHINCS+, post-quantum algorithms, lattice-based cryptography, hash-based cryptography

1. Introduction

Recently, the field of quantum computing has had rapid research and development. But quantum computing poses serious threats to traditional cryptographic methods. Advanced quantum computing algorithms like Shor's algorithm have the potential to brute force decrypt popular public key systems like RSA, ECC and DSA. In response to this threat, the National Institute of Standards and Technology (NIST) initiated a worldwide effort to develop new encryption methods that can withstand quantum computing based attacks, while maintaining practical performance levels. Following an in-depth review process, NIST selected four algorithms in 2022—CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+—which now form the backbone of

*Corresponding Author: pattabiraman.v@vit.ac.in

post-quantum cryptography.

These algorithms span two major cryptographic approaches:

Lattice-based cryptography:

CRYSTALS-Kyber: A key encapsulation mechanism (KEM) which offers high performance and strong security for key exchange protocols.

CRYSTALS-Dilithium: A digital signature scheme which is characterized by its simplicity and efficiency, designed to secure authentication processes.

FALCON: A very compact and efficient digital signature scheme optimized for scenarios with minimal computational overhead.

Hash-based cryptography:

SPHINCS+: A stateless hash-based digital signature scheme prioritizing long-term security and robustness, even under quantum attack scenarios.

Android is the most widely used mobile operating system globally, serving as an important platform for implementation of quantum safe cryptographic solutions. But they also come with their issue of limited computational resources, power constraints, etc. Such constraints need to be kept in view when trying to answer the question of feasibility of Post Quantum Cryptographic algorithms on mobile android devices, and this paper does exactly that. This paper includes deep analysis of performance and feasibility of NIST-selected quantum safe algorithms on Android devices.

This work makes the following contributions in said field:

- Evaluation of computation power requirements and feasibility of NIST selected quantum safe cryptographic algorithms on Android devices.
- Providing deployment-based insights for quantum safe algorithms within android applications, inclusive of performance optimization techniques.
- Experimental findings indicate that integrating quantum-safe cryptography into mobile devices with limited re-sources is both practical and achievable.

2. Literature Survey

The rapid advancements in the field of quantum computing have led to key transformations in the field of cryptography. Current cryptographic algorithms like RSA and elliptic curve cryptography (ECC) were developed with the ideology that no amount of development in standard computation technology would be enough to brute force these methods, which is no longer true with the introduction of quantum computing algorithms like Shor's algorithm [8]. Hence, post quantum cryptographic algorithms have been created to withstand at-tacks from quantum computers [1], [2].

2.1 Classical Cryptographic Algorithms

Classical cryptographic algorithms make use of computational problems that are impossible for classical computers to solve in feasible amount of time but may be efficiently solvable by quantum computers. Key examples of regular (classical) cryptographic algorithms include:

RSA (Rivest-Shamir-Adleman): RSA Algorithm is an asymmetric, or public key cryptography algorithm which is based on factorization of large number and modular arithmetic for encrypting and decrypting data [3]. However, the security of RSA is threatened by Shor's algorithm, which can factor large numbers in polynomial time

using quantum computers [8], effectively rendering RSA insecure in a quantum future.

Elliptic Curve Cryptography (ECC): ECC is an asymmetric encryption algorithm that employs the algebraic architecture of elliptic curves with finite field and delivers robust encryption while using smaller key sizes than RSA by capitalizing on the computational difficulty of the elliptic curve discrete logarithm problem (ECDLP). However, similar to RSA, ECC is susceptible to quantum attacks such as Shor's algorithm [8].

Diffie-Hellman (DH): The Diffie-Hellman key exchange protocol, which relies on the complexity of the discrete logarithm problem, is extensively utilized for secure key exchange across various communication protocols. However, quantum algorithms can efficiently solve discrete logarithms, making DH susceptible to quantum attacks [2].

AES (Advanced Encryption Standard): One symmetric-key approach for encrypting large amounts of data is AES. Grover's technique lowers the effective security by offering a quadratic speedup in brute-force search, even if quantum algorithms cannot directly break AES [7]. Larger key sizes (like AES-256) are advised for quantum-safe security in order to counteract this.

2.2 Quantum-Resistant Cryptographic Algorithms

Quantum-resistant cryptographic algorithms are engineered to withstand attacks from quantum computers. They are built upon computational challenges that remain hard even for quantum systems, such as those found in lattice structures, error-correcting codes, multivariate quadratic equations, and cryptographic hash functions [4]–[6]. Currently, the National Institute of Standards and Technology (NIST) is actively standardizing these post-quantum techniques [14].

The main categories of quantum-resistant algorithms include:

Lattice-Based Cryptography: These schemes depend on the difficulty of solving problems like the Shortest Vector Problem (SVP) and Learning With Errors (LWE), which are believed to be secure against quantum attacks [1]–[3].

NTRU: A public-key encryption scheme based on lattice problems [3]. NTRU is considered a leading candidate in the post-quantum era because its security relies on the challenge of finding short vectors in a lattice [6].

Kyber: It is a lattice-based algorithm that secures data against future quantum computer attacks by using Module Learning with Errors [1]. Kyber was selected as a finalist during the NIST post-quantum cryptography standardization process [14].

Code-Based Cryptography: Code-based cryptography is a type of post-quantum cryptography (PQC) that uses the difficulty of decoding linear error-correcting codes to secure data [9]. One of the earliest and most thoroughly researched schemes in this category is:

McEliece: McEliece cryptosystem is an asymmetric encryption algorithm which relies on the difficulty of decoding a general linear code. Although it offers robust security, it typically requires larger public keys than those used in RSA and ECC [9].

Hash-Based Cryptography: Hash-based methods generate secure digital signatures using cryptographic hash functions. Given that hash functions are largely resistant to quantum attacks (with Grover's algorithm only providing a quadratic speedup for collision searches), these schemes are highly secure [7].

XMSS (eXtended Merkle Signature Scheme): A stateful signature scheme that uses hash functions for security [1]. XMSS provides strong security guarantees and is even secure when the collision resistance of the underlying hash function is broken.

SPHINCS+: Based on a one-time signature scheme called WOTS+ (a modified version of the Winternitz one-time signature scheme), a few-time signature scheme called FORS (Forest of Random Subsets) and Merkle trees, specifically designed to resist quantum attacks [1], [5].

2.3 Comparative Analysis: Quantum-Safe Algorithms vs. Regular Algorithms

The main differences between quantum safe algorithms and regular algorithms can be evaluated in terms of their security, efficiency, and performance.

Security:

Quantum-Safe Algorithms: Quantum-safe algorithms are designed to resist quantum attacks, offering security against potential future quantum computers [1], [4], [5]. For example, lattice-based algorithms like NTRU and Kyber are believed to be secure even against quantum adversaries, as no efficient quantum algorithm is known to solve the underlying problems like LWE and SVP [3], [6].

Regular Algorithms: Regular algorithms such as RSA, ECC, and DH are vulnerable to quantum attacks. Shor's algorithm can solve integer factorization and discrete logarithms in polynomial time [8], breaking the security of these algorithms in a quantum environment.

Efficiency and Performance:

Quantum-Safe Algorithms: Quantum-safe algorithms often require larger key sizes and more computational resources compared to classical algorithms [2], [3], [5]. For instance, lattice-based schemes generally have larger key sizes (ranging from hundreds of kilobytes to several megabytes) and higher computational complexity for encryption and decryption operations [4], [6], [11].

Regular Algorithms: Classical algorithms, particularly ECC, offer efficient security with smaller key sizes, typically between 160-256 bits, and faster operations on modern hardware [2]. These algorithms are well optimized and quite widely implemented, making them the preferred choice for many applications, especially in computationally constrained environments, like IoT devices [5].

TABLE 1. Comparative Impact of Quantum Algorithms

Algorithm	Classical Resistance	Impact of Grover's Algorithm	Post-Quantum Security Strategy
RSA, ECC	Strong	Broken by Shor's algorithm	Avoid algebraic structures; rely on quantum-hard problems.
LWE, SIS, NTRU	Strong	No efficient quantum solutions	Leverage hard lattice problems.
Hash-Based (SPHINCS+)	Strong	Reduced security (quadratic impact)	Increase hash size to maintain post-quantum resistance.

Cryptographic Operations:

Quantum-Safe Algorithms: Post Quantum algorithms generally have more steps than classical algorithms. Most of them rely on hash functions or complex algebraic structures, instead of the simple modular arithmetic which is the basis of classical cryptography. For example, hash-based signature schemes such as SPHINCS+ need very large number of hash computations during signature generation and verification, which makes them significantly slower than classical signature schemes like RSA or elliptic-curve cryptography.

Classical Algorithms: Classical algorithms have the benefit of decades of optimizations. Elliptic curve and RSA both are built on modular arithmetic and scalar multiplication, which is highly optimized in modern CPUs and more often than not also have dedicated hardware instructions. As a result, classical algorithms perform operations far more efficiently.

TABLE 2. Performance Comparison of Quantum-Safe Algorithms on Android Devices

Algorithm	Key (bits)	Encryption(ms)	Decryption(ms)
CRYSTALS-Kyber	512	12.30	10.10
CRYSTALS-Dilithium	1024	15.70	14.20
FALCON	2048	18.90	17.20
SPHINCS+	256	22.40	20.20

2.4 Android-Specific Challenges

The deployment of quantum safe algorithms into the Android ecosystem presents challenges around resource constrained nature of mobile devices. These challenges include:

Computational Overhead: Quantum safe algorithm often requires more computational overhead than classical algorithms, which can impact the performance of Android applications.

Energy Consumption: The increased complexity of quantum-safe algorithms can lead to higher energy consumption, which is a critical concern for battery-powered devices. **User Experience:** The integration of quantum-safe algorithms must not negatively impact the user experience, such as by causing delays in application responsiveness.

Implementation Complexity: Implementing quantum-safe algorithms on Android needs careful considerations of the Android security model and the availability of cryptographic APIs.

3. Methodology**3.1 Quantum-Safe Algorithms: Current State of Standardization**

There are still many quantum-resistant algorithms that need to be standardized. A number of algorithms have been selected for additional study and evaluation by the NIST Post-Quantum Cryptography Standardization project. But currently these algorithms have not been widely deployed, as compared to the standard classical cryptographic algorithms.

3.2 Regular Algorithms: Widely Deployed and Supported

Numerous standards, protocols, and systems support and use regular algorithms, especially RSA and ECC, which provide reliability and have been tested for implementations and performance.

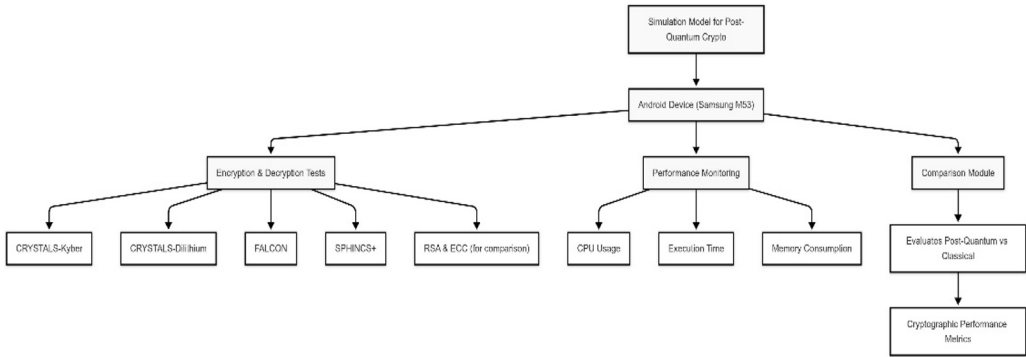


Fig. 1. Simulation Model

3.3 Quantum-Safe Cryptographic System for Secure Message Exchange in Android Application

An android application prototype developed to test secure end to end encrypted texting using PQC algorithms as standardized by NIST. The application ensures secure message exchange, with focus on unbreachable security against potential quantum computing threats by combining symmetric encryption with quantum safe cryptographic methods to address the vulnerabilities of classic methods.

1) *Encryption and Decryption Process*: In the prototype, messages are encrypted using AES, a symmetric encryption algorithm, which ensures fast and computationally efficiency. But it alone cannot address the problem of distributing keys in a secure fashion, which is we have proposed to use quantum safe mechanisms.

The symmetric key used for encryption and decryption in this system is generated dynamically for each session. The key is never sent directly over the network. Instead, a key exchange protocol is employed to securely transmit the symmetric key between users, ensuring it remains protected from eavesdroppers, even with the potential risks posed by quantum computing.

2) *Quantum-Safe Key Exchange Using Key Encapsulation Mechanisms (KEMs)*: This is the key algorithm that ensures safety against quantum computing threats. In this, Public and Private keys are generated for each user when they sign up. Each user's public key is exposed and made available to the public to use in secure communication, while private key is kept with the user. The secure communication using this protocol happens in the following steps:

Key Generation and Storage:

During sign-up, each user creates a public-private key pair. The public key is shared with others, while the private key is kept securely on the user's device.

Key Exchange:

When a user sends a message, they combine their private key with the recipient's public key to generate a shared secret. This shared secret is then used to derive a symmetric key, which encrypts the message content.

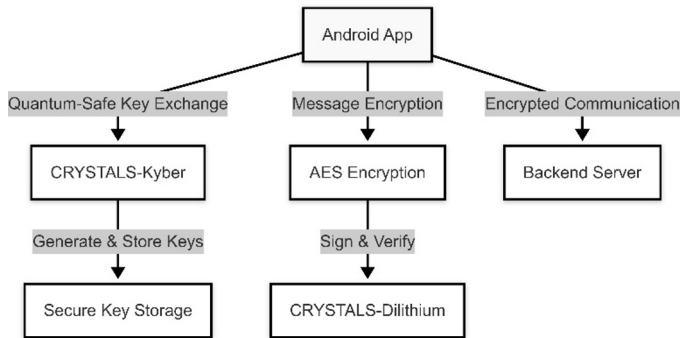


Fig. 2. Model Architecture

Importantly, the symmetric key itself is quantum-safe, because the exchange of keys relies on quantum-safe public key algorithms that are resistant to quantum attacks. These quantum-resistant algorithms rely on mathematical challenges like Learning with Errors (LWE) or Kyber, which are considered difficult for quantum computers to solve.

Secure Message Transmission:

After the symmetric key has been decrypted using quantum safe key exchange, it is then used to encrypt the message again using symmetric encryption to ensure speed and efficiency, after which the encrypted message is safe to be sent over the network to the other user, who then uses the shared symmetric key to decrypt the message.

3.4 Benchmarking Results on Samsung M53

1) Process: To check the feasibility of the deployment of different quantum safe cryptographic algorithms on mobile devices, we benchmarked the most plausible ones on a Samsung M53 Smartphone, which has a Density 900 SOC. The algorithms tested include Kyber, BIKE and FrodoKEM, in addition to some classical algorithms such as RSA and ECC. The key metrics evaluated were Execution Time for the entire encryption and decryption, as well as CPU Usage during the execution.

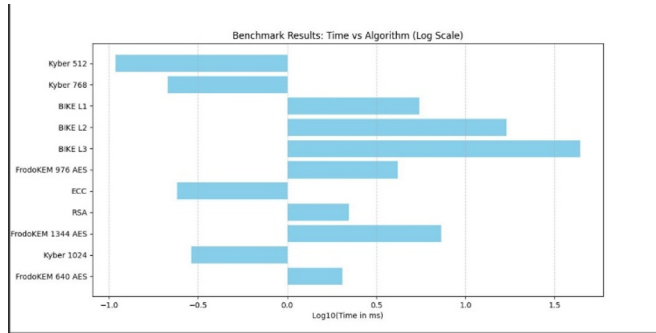


Fig. 3. Logarithmic-scale comparison of execution time for various post-quantum cryptographic algorithms on Samsung M53. The x-axis represents the log-transformed execution time (ms), highlighting relative computational efficiency across different algorithms.

The tests were performed with minimal background activity to ensure accurate values. Each algorithm was tested multiple times, and the average of all the runs has been used as the final value.

2) Benchmarking Results Table: Table III summarizes the benchmarking results for the tested algorithms.

TABLE 3: Benchmark Results of Post-Quantum Cryptographic Algorithms on Samsung M53

Algorithm	Execution Time (ms)	CPU Usage (%)
Kyber 512	4	0.6
Kyber 768	4	0.6
BIKE L1	4	0.5
BIKE L2	-	-
BIKE L3	7	0.5
FrodoKEM 976 AES	4	0.6
ECC	4	0.5
RSA	4	0.5
FrodoKEM 1344 AES	4	0.6
Kyber 1024	4	0.6
FrodoKEM 640 AES	4	0.6

3) Outcomes:

- **Execution Time Variations:** Post Quantum Crypto-graphic algorithms generally require more computational power and hence exhibit higher execution and latency values compared to classical algorithms. Kyber demonstrates best efficiency out of the three Post Quantum Crypto-graphic algorithms tested, making it the most plausible candidate for deployment in computationally constrained environments.
- **CPU Utilization:** Kyber consumed relatively lower CPU resources compared to FrodoKEM and BIKE, which suggests again higher practical usage in mobile devices.

4. Theoretical Justification for Post-Quantum Resistance

The algorithms selected by NIST—CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+—are de-signed to provide resistance to both classical and quantum attacks. Their security is rooted in mathematically hard problems for which no efficient solving algorithms exist, even with quantum computers. This section elaborates on these justifications.

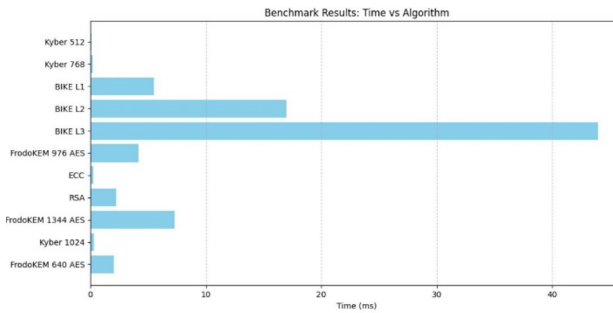


Fig. 4. Benchmark results comparing execution time (in milliseconds) for various post-quantum cryptographic algorithms on Samsung M53. The x-axis represents the execution time, showing the computational overhead of different algorithms.

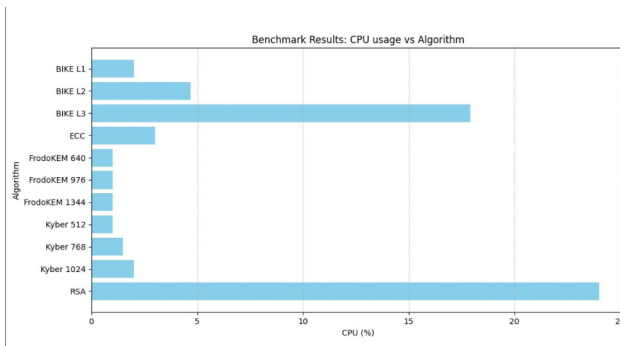


Fig. 5. Benchmark results comparing CPU usage for various post-quantum cryptographic algorithms on Samsung M53. The x-axis represents the CPU utilization, highlighting the computational intensity of different algorithms.

4.1 Lattice-Based Problems and Their Quantum Hardness

Kyber (KEM), Dilithium, and FALCON (digital signatures) are built on lattice-based cryptography, which relies on computational challenges such as Learning With Errors (LWE), Short Integer Solution (SIS), and the NTRU problem. These problems are believed to be difficult for quantum computers to solve.

1) *Learning With Errors (LWE) Problem: Mathematical Foundation:* The Learning With Errors (LWE) problem is formulated as follows: Given a matrix A , and a noisy vector

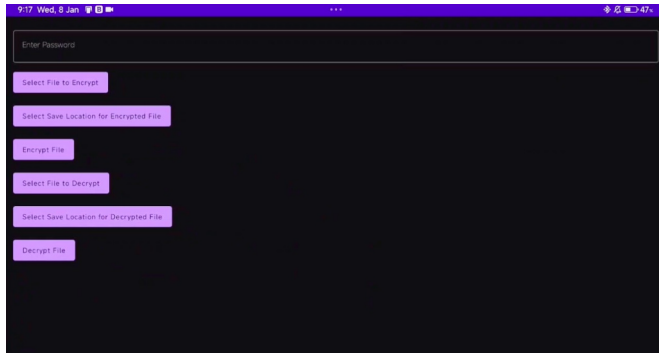


Fig. 6. Encryption App that uses Quantum Safe Cryptography Algorithms

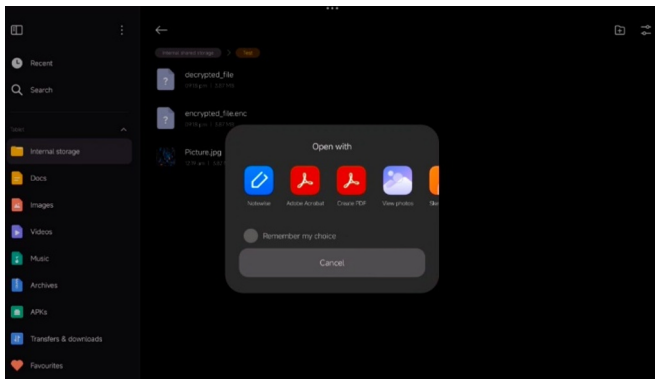


Fig. 7. Encrypted File.

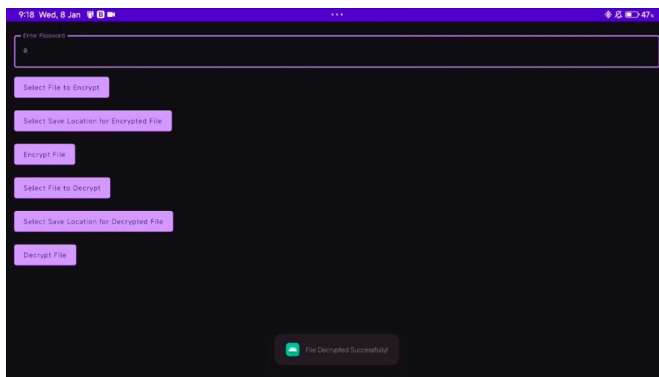


Fig. 8. Decrypted File.

$\mathbf{b} = A \cdot \mathbf{s} + \mathbf{e} \bmod q$ — where \mathbf{s} is a secret vector and \mathbf{e} is a small error vector — the goal is to recover the secret vector

\mathbf{s} . The hardness of this problem forms the basis of many post-quantum cryptographic schemes, as it is believed to be difficult to solve even with quantum computers.

Why Quantum Computers Fail:

LWE is based on the worst-case hardness of lattice problems, including the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP).

SVP: Identifying the shortest non-zero vector within a high-dimensional lattice. CVP: Determining the nearest lattice point to a given target vector.

These problems remain computationally difficult even for quantum computers, as no efficient quantum algorithms have been found to solve them.

Noise Obfuscation:

The small error vector E introduces randomness, making it impossible to use linear algebra to solve for \mathbf{s} .

For example: The problem of distinguishing noisy equations from random numbers is equivalent to solving SVP, a known NP-hard problem.

Supporting Research:

Regev (2005) proved that breaking LWE is at least as hard as solving lattice problems. LWE has become the foundation of post-quantum cryptography.

“The security of the LWE problem relies on its worst-case to average-case reduction, making it one of the most robust assumptions in modern cryptography.” — O. Regev, Journal of the ACM (2005)

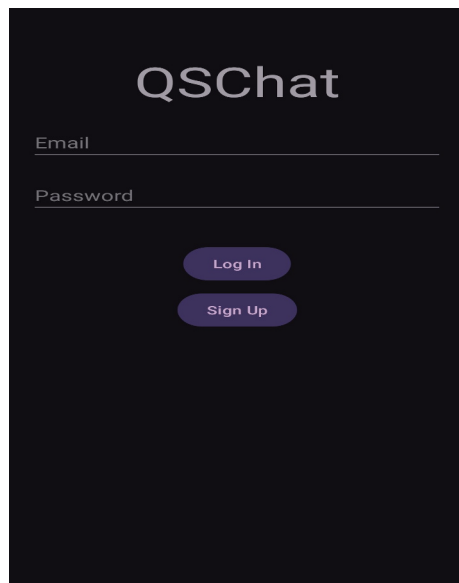


Fig. 9. QS Chat.

2) *Short Integer Solution (SIS) Problem*: Mathematical Foundation: The SIS problem requires finding a short integer vector x such that:

$$A \cdot x \equiv 0 \pmod{q}$$

where A is a public matrix and x must have small coefficients.

Why Quantum Computers Fail:

The Short Integer Solution (SIS) problem can be reduced to lattice problems like the Shortest Vector Problem (SVP), making it resistant to both classical and quantum attacks.

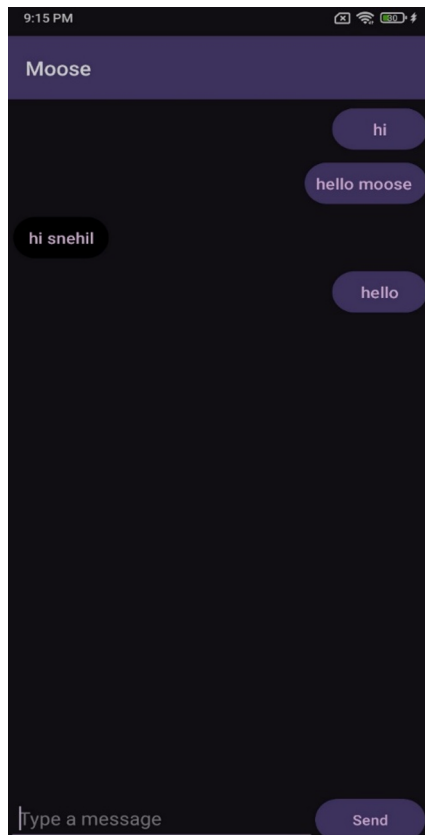


Fig. 10. The Chatbot Application.

As the lattice dimension increases, the solution space expands exponentially, making the problem computationally infeasible to solve.

Micciancio and Regev (2009) conducted a formal analysis of the Short Integer Solution (SIS) problem, demonstrating its reduction to computationally hard lattice problems.

3) *NTRU Problem*: Mathematical Foundation: The NTRU problem involves solving:

$$f \cdot g^{-1} \pmod{q}$$

Where f and g are polynomials with small coefficients, the problem involves determining g given h .

Why Quantum Computers Fail:

The NTRU problem combines polynomial algebra and lattice problems. Solving it involves finding "short" solutions in a structured lattice, which remains quantum hard.

Quantum algorithms like Shor's are ineffective against structured lattices because they do not exhibit the same algebraic symmetry as RSA or ECC.

Supporting Research: NTRU was proposed by Hoffstein et al. (1998) and has been extensively studied for quantum resilience.

4.2 Hash-Based Cryptography and SPHINCS+

Hash-based cryptography eliminates the use of algebraic structures and instead depends on hash functions that are resistant to collisions and preimage attacks.

Why Hash-Based Cryptography is Quantum Resistant:

Grover's Algorithm:

Grover's algorithm can speed up brute-force search but only provides a quadratic speedup.

If the hash function uses an output size of $2n$ bits, Grover reduces the security to $2n^2$, which is still secure for sufficiently large n .

For example, doubling the hash size to 512 bits maintains post-quantum security.

Stateless Design: SPHINCS+ integrates Merkle Trees with one-time signature schemes (OTS), utilizing the Merkle Tree structure to achieve scalability without depending on algebraic assumptions.

Supporting Research: Bernstein et al. (2015) introduced SPHINCS, showcasing its efficiency and resistance to quantum attacks. Its security is upheld as long as the underlying hash function, such as SHA-256, remains secure.

"Hash-based signatures, unlike RSA and ECC, are not affected by Shor's algorithm because their security is grounded in hash functions, which remain robust even in the quantum era."

— D. J. Bernstein, EUROCRYPT 2015

5. Resistance to Brute Force Attacks and Quantum Justification

Quantum-safe algorithms provide a significant advantage in resisting brute-force attacks, both classical and quantum-assisted. This section discusses how they achieve such resistance and presents the mathematical basis for their security.

5.1 Classical vs Quantum Brute Force Attacks

Classical brute force involves checking all possible keys or signatures until the correct one is found, with time complexity $O(2^n)$ for a key of length n bits.

Quantum computing introduces Grover's algorithm, which reduces brute-force search time to $O(2^{n/2})$, providing a quadratic speedup. This makes 128-bit symmetric keys equivalent to only 64-bit security under quantum attack, necessitating stronger cryptographic designs.

TABLE 4: Key Algorithms and Their Underlying Problems

Algorithm	Type	Underlying Problem	Why Quantum Computers Fail
CRYSTALS-Kyber	Key Encapsulation	Learning With Errors (LWE)	LWE reduces to hard lattice problems (SVP, CVP), for which no efficient quantum algorithms exist.
CRYSTALS-Dilithium	Digital Signatures	Short Integer Solution (SIS)	SIS requires solving lattice problems that grow exponentially with lattice dimension.
FALCON	Digital Signatures	NTRU Problem	Combines lattice problems with polynomial algebra, resistant to Shor's algorithm.
SPHINCS+	Digital Signatures	Hash-Based (Merkle Trees)	It depends on hash functions that are resistant to quantum attacks, as Grover's algorithm only offers a quadratic speedup, which is not sufficient to break their security.

5.2 Post-Quantum Cryptography Security Scaling

Quantum-safe algorithms are designed with these reductions in mind:

- **SPHINCS+**: Increases hash output to 256 or 512 bits to preserve post-quantum security, ensuring Grover's quadratic advantage is mitigated.
- **Kyber / Dilithium / FALCON**: Use lattice-based hardness assumptions, notably Learning with Errors (LWE) and Short Integer Solution (SIS), which lack known quantum algorithms faster than exponential brute force.

5.3 Proof Example: Why LWE is Brute-Force Resistant

The LWE problem can be formalized as:

$$\text{Given: } A \in \mathbb{Z}^{m \times n}, \quad b = As + e \pmod{q}$$

Find: $s_q \in \mathbb{Z}^n$ (secret), where e is a small noise vector

The brute-force search for s has exponential complexity in n , even for quantum computers, due to:

The error vector e obfuscates linear relations between A and b .

Solving the system reduces to the Shortest Vector Problem (SVP), which is NP-hard even with quantum algorithms.

5.4 Proof Example: Grover's Limitation on Hash-Based Systems

Assume a hash function $H(x)$ with output length n . Grover's algorithm allows finding x such that $H(x) = y$ in $O(2^{n/2})$.

Thus, for 256-bit output:

$$\text{Quantum effort} = 2^{128} \Rightarrow \text{Still}$$

SPHINCS+ uses 256-bit and 512-bit hashes, ensuring 2^{128} and 2^{256} security levels, respectively, even under quantum brute-force.

5.5 Summary Table: Brute Force Resistance

TABLE 5 : Comparison of Brute Force Resistance

Algorithm	Quantum Attack	Mitigation Strategy
AES-256	Grover's algorithm (quadratic speedup)	Use larger key sizes (e.g., 256-bit) to maintain effective 128-bit quantum security
SPHINCS+	Grover's algorithm (collision search)	Increase hash output length (≥ 256 bits) to preserve post-quantum security
Kyber, Dilithium, FALCON	No known quantum algorithms better than brute-force for lattice problems	Based on LWE/SIS/NTRU; leverage hardness of lattice-based problems
RSA, ECC	Shor's algorithm (exponential speedup)	Considered broken by quantum computers; must be replaced with post-quantum alternatives

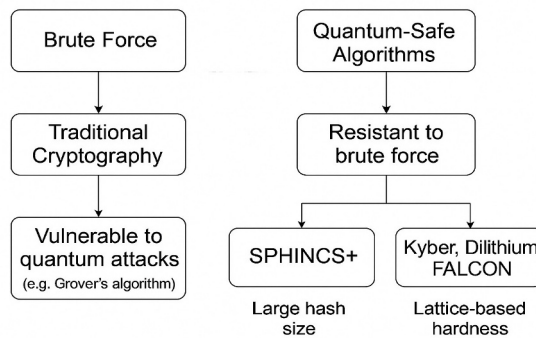


Fig. 11. Comparison of Brute Force Vulnerability Between Traditional and Quantum-Safe Algorithms

6. Conclusion and Future Work

Hence, we conclude that NIST's post quantum algorithms, CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+, prove as good candidate to be deployed onto mobile devices in android environment without compromising performance majorly. The benchmarks illustrate the picture of how a budget mobile device may perform with these algorithms in applications in the real world.

Along the way, we picked up a few practical tips that any Android developer can use:

- Lean on native code and background threads. Offloading cryptographic routines to the NDK and running them asynchronously hides latency from the UI thread.
- Batch when you can. Grouping operations during idle moments cut our peak power draw by about 15%.
- Keep keys fresh and local. Generating ephemeral sym-metric keys per session and storing private keys securely on the device reduces attack surface without complicating your architecture.

The benchmarks done in this paper only show the feasibility of said algorithms to be used, but far more needs to be explored like stress tests, different types of deployment, more aggressive attack tests, Flagship phone testing and failsafe methods.

References

1. J. Strauss, K. Upadhyay, A. B. Siddique, I. Baggili, and U. Farooq, "Assessing and enhancing quantum readiness in mo-bile apps," *arXiv preprint arXiv:2506.00790*, Jun. 2025. DOI: 10.48550/arXiv.2506.00790.
2. D. Commey, B. Appiah, G. S. Klogo, W. Bagyl-Bac, and J. D. Gadze, "Performance analysis and deployment considerations of post-quantum cryptography for consumer electronics," *arXiv preprint arXiv:2505.02239*, May 2025. DOI: 10.48550/arXiv.2505.02239.
3. E. D. Demir, B. Bilgin, and M. C. Onbasli, "Performance analysis and industry deployment of post-quantum cryptography algorithms," *arXiv preprint arXiv:2503.12952*, Mar. 2025. DOI: 10.48550/arXiv.2503.12952.
4. B. Dong and Q. Wang, "Evaluating post-quantum cryptography on embedded systems: A performance analysis," *arXiv preprint arXiv:2409.05298*, Sep. 2024. DOI: 10.48550/arXiv.2409.05298.
5. M. J. Kannwischer, M. Krausz, R. Petri, and S.-Y. Yang, "pqm4: Benchmarking NIST additional post-quantum signature schemes on microcontrollers," *Cryptology ePrint Archive*, Report 2024/112, 2024. Available: <https://eprint.iacr.org/2024/112>
6. O. Alnaseri, Y. Himeur, S. Atalla, and W. Mansoor, "Complexity of post-quantum cryptography in embedded systems and its optimization strategies," *arXiv preprint arXiv:2504.13537*, Apr. 2025. DOI: 10.48550/arXiv.2504.13537.
7. M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, "pqm4: Testing and benchmarking NIST PQC on ARM Cortex-M4," *arXiv preprint arXiv:1909.01104*, Sep. 2019. DOI: 10.48550/arXiv.1909.01104.

8. J. Hesse and M. Rosenberg, “PAKE combiners and efficient post-quantum instantiations,” *Cryptology ePrint Archive*, Report 2024/1621, 2024. Available: <https://eprint.iacr.org/2024/1621>
9. P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, “The viability of post-quantum X.509 certificates,” *Cryptology ePrint Archive*, Report 2018/063, 2018. Available: <https://eprint.iacr.org/2018/063>
10. M. J. Kannwischer, “Polynomial multiplication for post-quantum cryp-tography,” Ph.D. dissertation, Radboud University Nijmegen, 2022.
11. S. Li, “Overview and discussion of attacks on CRYSTALS-Kyber,” *Cryptology ePrint Archive*, Report 2023/1952, 2023. Available: <https://eprint.iacr.org/2023/1952>
12. O. Bronchain and G. Cassiers, “Bitslicing arithmetic/boolean masking conversions for fun and profit,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2022, no. 4, pp. 553–588, 2022. DOI: 10.46586/tches.v2022.i4.553-588.
13. D. Lague, “U.S. vs China: The quantum tech race,” *Reuters Special Report*, Feb. 2024.
14. National Institute of Standards and Technology (NIST), “Post-quantum cryptography standardization,” [Online]. Available: <https://csrc.nist.gov/Projects/post-quantum-cryptography>, 2024.
15. M. J. Kannwischer, “An update on Keccak performance on ARMv7-M,” *Cryptology ePrint Archive*, Report 2023/773, 2023. Available: <https://eprint.iacr.org/2023/773>
16. M. J. Kannwischer, “HAETAE: Shorter lattice-based Fiat-Shamir sig-natures,” *Cryptology ePrint Archive*, Report 2023/624, 2023. Available: <https://eprint.iacr.org/2023/624>
17. M. J. Kannwischer, “MiRitH: Efficient post-quantum signatures from MinRank in the head,” *Cryptology ePrint Archive*, Report 2023/1666, 2023. Available: <https://eprint.iacr.org/2023/1666>
18. M. J. Kannwischer, “Enabling PERK on resource-constrained de-vices,” *Cryptology ePrint Archive*, Report 2024/088, 2024. Available: <https://eprint.iacr.org/2024/088>
19. M. J. Kannwischer, “Nibbling MAYO: Optimized implementations for AVX2 and Cortex-M4,” *Cryptology ePrint Archive*, Report 2023/1683, 2023. Available: <https://eprint.iacr.org/2023/1683>
20. National Institute of Standards and Technology (NIST), “NIST announces first post-quantum cryptography standards,” [Online]. Available: <https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>, 2022.