International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

A Comparative Study of Classical Cryptography and Post-Quantum Cryptographic Algorithms

Varshini B M[1], Dr. Umadevi Ramamorthy[2]

[1]School of Science and Computer Studies, CMR University

[2]School of Science and Computer Studies, CMR University (Guide)

Abstract: ***The rapid progress of quantum computing is changing the assumptions on which many modern security systems are built. Public-key methods such as RSA and Elliptic Curve Cryptography (ECC) remain dependable against conventional computers, yet their underlying mathematical problems can be solved efficiently by a sufficiently capable quantum computer using algorithms such as Shor's algorithm. This creates a practical need to evaluate cryptographic techniques that can continue to protect information in a post-quantum environment. This study compares representative classical algorithms - RSA, ECC, and the Advanced Encryption Standard (AES) - with post-quantum approaches, with particular attention to CRYSTALS-Kyber and CRYSTALS-Dilithium. The comparison considers security assumptions, key-generation cost, encryption and decryption time, key size, memory demand, and resistance to quantum attacks. The reported benchmark values indicate that RSA has the highest processing cost among the evaluated schemes, whereas CRYSTALS-Kyber provides comparatively fast key generation, encryption, and decryption while maintaining resistance to known quantum attacks. The study also discusses a hybrid migration model in which established classical mechanisms and post-quantum algorithms operate together during the transition period. Overall, the findings support a gradual move toward quantum-resistant communication systems while recognising the deployment and storage overhead introduced by larger post-quantum keys.***

Keywords: Quantum computing, post-quantum cryptography, lattice-based cryptography, CRYSTALS-Kyber, CRYSTALS-Dilithium, RSA, ECC, AES, security strength

1. Introduction

Quantum computation offers major opportunities in scientific modelling, optimisation, and large-scale data processing. At the same time, it challenges the mathematical foundations of widely deployed cryptographic systems. Digital banking, electronic commerce, cloud services, messaging platforms, and government networks all depend on encryption and digital signatures to preserve confidentiality, integrity, and authenticity.

RSA and ECC are two of the most common public-key techniques. RSA relies on the difficulty of factoring very large integers, whereas ECC depends on the Elliptic Curve Discrete Logarithm Problem. These problems are computationally demanding for classical machines, which is why the algorithms have remained useful for many years. A large fault-tolerant quantum computer, however, could apply Shor's algorithm to solve both problems in polynomial time. If such a machine becomes practical, keys that are secure today may no longer provide adequate protection.

Post-Quantum Cryptography (PQC) addresses this risk by designing algorithms that run on ordinary computers but are based on mathematical problems for which no efficient classical or quantum solution is currently known. Major PQC families include lattice-based encryption and signatures, code-based encryption, multivariate constructions, and hash-based signatures. Examples considered in this work include CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+.

A direct replacement of existing cryptography is not simple. Post-quantum schemes often require larger public keys, ciphertexts, or signatures, and these differences affect bandwidth, storage, embedded devices, certificate systems, and network protocols. For that reason, a useful comparison must examine both security and implementation cost. This study therefore combines a theoretical review with performance-oriented benchmarking and a hybrid deployment framework.

The remainder of the paper is organised as follows. Section 2 reviews earlier work on classical and post-quantum security. Section 3 explains the research methodology and the selected algorithms. Sections 4 to 7 discuss mathematical security, theoretical analysis, comparative evaluation, and data interpretation. Section 8 presents the experimental environment, Section 9 discusses the reported results, and Section 10 concludes the study.

2. Literature Review

Research in cryptography has moved from improving conventional encryption toward preparing security systems for the quantum era. The National Institute of Standards and Technology (NIST) initiated a formal post-quantum standardisation programme to identify algorithms that offer strong security, practical performance, and interoperability. The evaluation process encouraged research groups around the world to submit and analyse candidate schemes under common criteria.

A Post-Quantum GAKE protocol was proposed by Pablos et al., which used the LibOQS library. This research evaluated the efficiency of Classic McEliece, Kyber, NTRU, and Saber based on execution time, size of keys, and computational efficiency. As per the study, Kyber is the best trade-off between security and performance [1].

Mosca presented the effects of quantum computing in current cryptography protocols. According to the paper,

RSA and ECC can be attacked using Shor's algorithm, and the paper stresses the importance of transitioning to post-quantum cryptography systems without compromising compatibility with the current systems [2].

The paper is a review of the NIST standardization process for Post-Quantum Cryptography (PQC). The authors note that lattice-based algorithms such as Kyber and Dilithium are effective substitutes to traditional public-key cryptography due to their high security and efficiency [3].

Post-quantum cryptographic algorithms were analyzed by the authors within secure communication settings. Execution time, key generation, and encryption of PQC was compared with classical algorithms and the authors found out that PQC offers better protection against quantum attacks but consumes more computing power [4].

In this paper, lattice-based cryptography is analyzed for IoT applications. It is shown by the authors that optimized lattice-based cryptographic schemes can offer quantum security with practicality within IoT devices [5].

Hybrid cryptographic algorithms which use both classical and post-quantum algorithms were suggested by the authors. According to their research, hybrid cryptosystems offer backwards compatibility while increasing resistance to quantum attacks [6].

This research highlights the comparative analysis between ML-KEM (Kyber) and conventional public key techniques like RSA and ECC. It was concluded that the Kyber algorithm offers much more resistance to the quantum attacks without sacrificing the good computational performance for contemporary communication systems [7].

This paper focuses on the shift from RSA and ECC to post-quantum cryptography. The difficulties in the implementation process have been highlighted along with providing recommendations regarding hybrid solutions for the transitional period [8].

The following paper offers a comparative analysis between the classical and post-quantum cryptographic algorithms for network security applications. It was concluded that post-quantum algorithms provide more security against the quantum attacks, whereas the classical algorithms remain more computationally efficient [9].

A comparative study has been made in order to measure the performance of classical and post-quantum cryptographic algorithms. From their results, it has become clear that although post-quantum cryptographic algorithms need high memory and key size, still they protect against future quantum computing attacks [10].

Post-quantum cryptography in cloud computing and network security is discussed in this research work. They have found that PQC provides better protection for long term data but at the cost of higher computation requirements [11].

Scrivano has explored the strategies followed by institutions while moving from classical cryptography to post-quantum cryptography. The suggested approach is based on phased transition, risk analysis, and hybrid cryptographic systems [12].

This paper has benchmarked classical cryptography and post-quantum cryptography at the transport layer. The study has considered latency, throughput, and computation overhead and concluded that post-quantum algorithms offer higher security but consume more computation power [13].

Hardware-software co-design approaches were proposed by Chen et al. for speeding up the performance of post-quantum cryptographic algorithms. It has been demonstrated through their study that the use of hardware design improves the execution speed. [14].

From the reviewed literature, it is clear that the conventional cryptographic algorithms like RSA and ECC are still effective but not immune from quantum-based attack. However, post-quantum cryptography algorithm like Kyber (ML-KEM), Dilithium (ML-DSA), NTRU and Classic McEliece offer immunity against quantum attack though they need larger keys. From various recent IEEE papers, it can be seen that most of them have suggested using hybrid cryptographic approach during the transition period.

3. Proposed Methodology

3.1 Research Overview

The methodology is designed to compare classical and post-quantum algorithms from three connected viewpoints: security, performance, and deployment feasibility. The security review studies the mathematical assumptions behind each algorithm and their behaviour under known quantum techniques. The performance review examines the time required for key generation, encryption, and decryption. The deployment review considers key size, memory use, protocol compatibility, and the effort needed to introduce the algorithm into an existing communication system.

In addition to comparing individual schemes, the methodology proposes a hybrid model. In this model, organisations can retain well-tested classical components while introducing post-quantum key establishment or signatures. This approach is intended to support a controlled transition rather than an immediate replacement of every cryptographic component.

3.2 Research Design and Approach

An experimental comparative design is used. The work is divided into four phases: algorithm selection, theoretical security assessment, experimental efficiency measurement, and hybrid framework design. Qualitative analysis is used for security and deployment considerations, while quantitative analysis is applied to timing and resource measurements. To preserve fairness, all algorithms are evaluated under the same stated system conditions. Repeated measurements can be averaged to reduce the effect of temporary system load, and performance ratios can be used to show the relative overhead of one method over another.

3.3 Selection of Cryptographic Algorithms

The algorithms were selected using the following criteria:

- Current or emerging use in real-world systems.
- Recognition through established standards or active standardisation.
- Coverage of different mathematical security foundations.
- Relevance to the transition toward quantum-resistant computing.
- Availability of practical software implementations for testing.

Table 1: Algorithms included in the comparative study

Algorithm	Categor y	Primary basis	Main role
RSA	Classical public-key	Integer factorisatio n	Encryption / signatures
ECC	Classical public-key	Elliptic-curve discrete logarithm	Key exchange / signatures
AES	Classical symmetri c	Substitutio n-permutatio n network	Bulk data encryption
CRYSTAL S-Kyber	Post-quantum KEM	Module-LWE / lattice problems	Key encapsulatio n
CRYSTAL S-Dilithium	Post-quantum signature	Module-lattice problems	Digital signatures
FALCON	Post-quantum signature	NTRU lattices	Compact signatures
SPHINCS+	Post-quantum signature	Hash functions / Merkle trees	Stateless signatures

3.4 Classical Cryptographic Algorithms

RSA (Rivest-Shamir-Adleman)

RSA creates a public and private key from two large prime numbers. The public key is used for encryption or signature verification, while the private key is retained by the owner. Its security is linked to the practical difficulty of recovering the prime factors of a large composite number.

Choose two large primes, p and q.

n = p x q

Compute Euler's totient.

phi(n) = (p - 1)(q - 1)

Select e such that 1 < e < phi(n) and gcd(e, phi(n)) = 1.

Calculate the private exponent d as the modular inverse of e.

d = e^(-1) mod phi(n)

Public key = (e, n) Private key = (d, n)

For a plaintext integer M, encryption and decryption are represented as follows:

C = M^e mod n M = C^d mod n

Elliptic Curve Cryptography (ECC)

ECC obtains public-key security from the algebraic structure of points on an elliptic curve over a finite field. A private key is a randomly selected integer, and the corresponding public key is produced by multiplying a base point by that integer. ECC normally achieves security comparable to RSA with a smaller classical key size.

- Select the curve parameters and a base point G.
- Choose a private value d and compute the public point Q = dG.
- For simplified ECIES encryption, the sender chooses a random value k and creates C1 = kG.
- The protected message component is formed using the shared point kQ.
- The receiver uses dC1 to recover the same shared point and reconstruct the message.

Advanced Encryption Standard (AES)

AES is a symmetric block cipher used for high-speed protection of bulk data. AES-128 applies ten rounds, whereas longer key variants use additional rounds. The algorithm begins with key expansion, followed by an initial AddRoundKey step. Each main round applies byte substitution, row shifting, column mixing, and round-key addition. The final round omits MixColumns. Decryption performs the inverse transformations in reverse order.

- SubBytes - replaces each byte through an S-box.
- ShiftRows - cyclically moves bytes across rows.
- MixColumns - combines bytes within each column.
- AddRoundKey - XORs the state with a round key.

3.5 Post-Quantum Cryptographic Algorithms

CRYSTALS-Kyber

CRYSTALS-Kyber is a lattice-based key-encapsulation mechanism. It is designed to establish a shared secret between two parties over an untrusted network. Key generation produces a public matrix-based value and a private secret vector. Encapsulation uses the public key to

create a ciphertext and a temporary shared secret, while decapsulation uses the private key to recover the same secret.

Generate a public matrix A and choose a secret vector s.

Add a small error vector e and compute t = As + e. Public key = (A, t) Private key = s

During encapsulation, choose a fresh random vector and create the ciphertext components.

During decapsulation, combine the ciphertext with the secret vector to reconstruct the shared key.

CRYSTALS-Dilithium

CRYSTALS-Dilithium is a lattice-based digital-signature scheme. Its key pair is generated from a public matrix and short secret vectors. To sign a message, the signer samples a temporary vector, computes an intermediate lattice value, hashes that value together with the message, and forms a response. Verification recomputes the expected relationship using the public key and accepts the signature only when the hash and norm conditions are valid.

t = A s1 + s2

Public key = (A, t) Private key = (s1, s2)

Generate a temporary vector y and compute w = Ay.

Create a challenge c = H(w, message).

Form the signature response z and publish the pair (z, c).

Verify by checking the reconstructed lattice equation and the challenge hash.

Falcon

FALCON is a lattice-based signature scheme constructed from NTRU lattices. Key generation creates related public and private polynomials. Signing hashes the message to a lattice target and samples a short vector close to that target. Verification checks the public lattice relation and confirms that the signature remains within the required norm bound.

Sphincs+

SPHINCS+ is a stateless hash-based signature method. It creates its public key from a hierarchy of Merkle trees and one-time signature components. Signing hashes the message, produces the required one-time signatures, and includes the authentication paths needed to rebuild the tree root. Verification reconstructs the root and compares it with the public key.

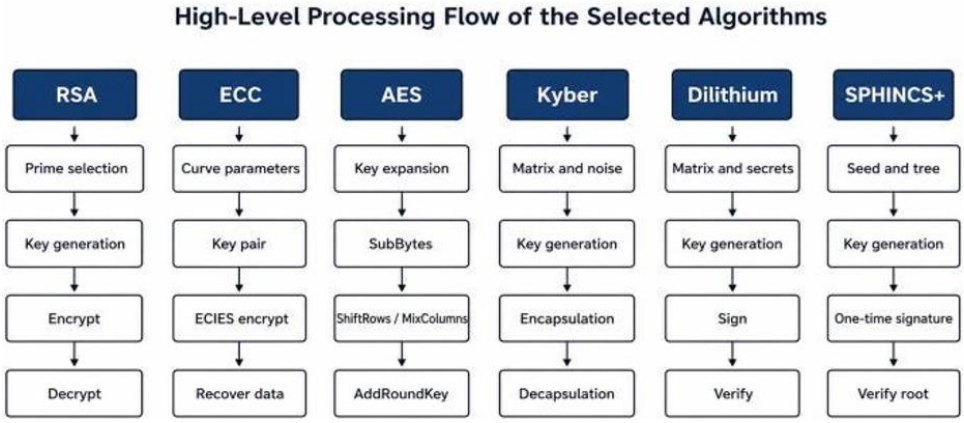

Figure 1: High-level processing flow of the selected cryptographic algorithms

4. Mathematical Security Analysis

4.1 Classical Cryptographic Basis

- RSA depends on the difficulty of integer factorisation.
- ECC depends on the Elliptic Curve Discrete Logarithm Problem (ECDLP).
- AES derives security from repeated substitution, permutation, and key-mixing operations.

On classical computers, the best known general attacks against properly configured RSA and ECC remain computationally expensive. The security position changes in the presence of a large quantum computer because Shor's algorithm can solve both integer factorisation and discrete logarithms efficiently. RSA and ECC should therefore be treated as vulnerable in a future quantum environment.

4.2 Post-Quantum Cryptographic Basis

- Lattice-based schemes rely on hard problems such as Learning With Errors (LWE) and related module-lattice variants.
- Hash-based signatures rely on one-way and collision-resistant hash functions.
- Other PQC families may use error-correcting codes or multivariate polynomial systems.

No efficient quantum algorithm is currently known for solving the underlying problems used by the selected lattice and hash schemes at their recommended parameter levels. Their security is therefore considered resistant to both conventional and known quantum attacks, subject to correct implementation and future cryptanalysis.

5. Theoretical Security Analysis

A theoretical security analysis evaluates the hardness assumption on which an algorithm is built. RSA relies on integer factorisation, and ECC relies on the discrete logarithm problem over elliptic curves. Shor's algorithm directly weakens both assumptions in the quantum model. Symmetric encryption is affected differently. Grover's algorithm offers a quadratic speed-up for exhaustive key search rather than a complete break. As a practical result, a 256-bit AES key is commonly viewed as a stronger long-term option than a 128-bit key when planning for quantum resistance.

Kyber and Dilithium use lattice problems related to Learning With Errors, for which no comparably efficient quantum solution is known. SPHINCS+ relies on hash-function properties and a Merkle-tree construction. These algorithms are assigned security levels through standard evaluation processes, enabling implementers to select parameters that correspond to their risk and performance requirements.

6. Comparative Analysis Model

The collected information is interpreted through three evaluation layers.

6.1 Security Layer

The security layer classifies each algorithm according to its expected resistance to quantum attack. RSA and ECC are treated as high-risk in a sufficiently capable quantum environment, while the selected PQC schemes are treated as quantum-resistant under current knowledge.

6.2 Performance Layer

The performance layer compares key-generation time, encryption or encapsulation time, decryption or decapsulation time, memory usage, and key size. Relative performance can be expressed by dividing the execution time of a PQC algorithm by that of a classical reference algorithm under the same conditions.

6.3 Deployment Feasibility Layer

The deployment layer examines whether an algorithm can be integrated into existing applications and communication protocols. Important concerns include certificate size, network bandwidth, storage, energy use, library maturity, backward compatibility, and the operational complexity of changing key-management processes.

7. Data Analysis and Interpretation

Experimental measurements are analysed using averages, comparison tables, and charts. The interpretation focuses on trends rather than a single timing value. Lower execution time suggests better computational efficiency, while larger keys or messages indicate additional storage and transmission cost. Security observations are considered together with performance so that a fast but quantum-vulnerable algorithm is not mistakenly treated as the best long-term choice.

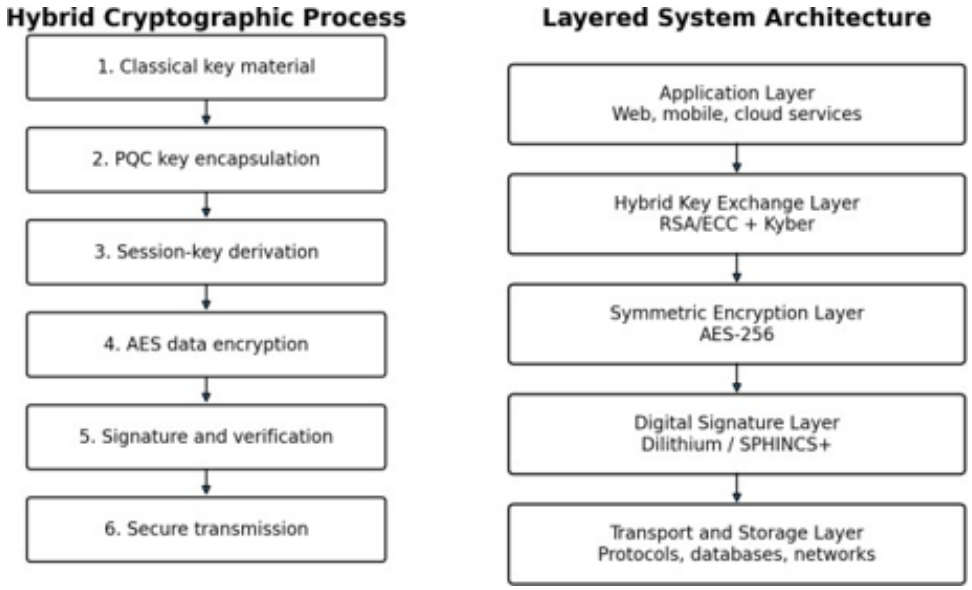

Figure 2: Proposed hybrid cryptographic process and layered deployment architecture

8. Experimental Setup

The experiment is intended to compare selected classical and post-quantum algorithms while processing data of different sizes. The supplied study describes a standard desktop-class environment, such as an Intel Core i5 or i7 processor, 8 GB of memory, and Windows 10 or Windows 11. Implementations may be developed in Python or C++ using established libraries such as OpenSSL and available post-quantum cryptography packages. Using the same hardware and software conditions for every algorithm helps maintain a consistent basis for comparison.

RSA and ECC represent classical public-key methods, while CRYSTALS-Kyber and CRYSTALS-Dilithium represent post-quantum encryption/key-establishment and signature

mechanisms. Test-message sizes of 1 KB, 10 KB, 100 KB, and 1 MB are used to observe whether processing time changes as input size increases. The main recorded parameters are key-generation time, encryption or encapsulation time, decryption or decapsulation time, key size, memory use, and security classification.

For repeatable benchmarking, each operation should be executed several times after an initial warm-up run, and the average value should be reported. Background applications should be minimised, library versions should be documented, and the same random-data generation method should be used across tests.

9. Results and Discussion

The benchmark values reported in the supplied manuscript provide a clear comparison of the selected algorithms. RSA records the highest encryption, decryption, and key-generation times among the four evaluated schemes. ECC performs better than RSA but remains vulnerable to large-scale quantum attacks. CRYSTALS-Kyber records the lowest encryption and decryption times in the reported dataset, while CRYSTALS-Dilithium also remains competitive.

Table 2: Reported benchmark values

Algorithm	Encryption (ms)	Decryption (ms)	Key generation (ms)	Quantum resistance
RSA	28	35	120	Not secure
ECC	20	24	45	Not secure
CRYSTALS-Kyber	9	12	30	Secure
CRYSTALS-Dilithium	12	15	35	Secure

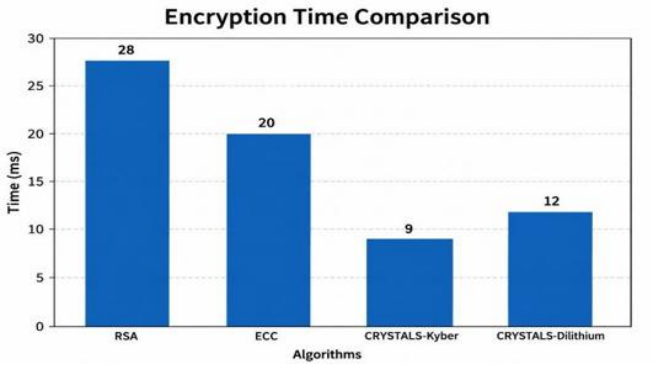

Figure 3: Reported encryption-time comparison

Figure 3 shows that RSA requires 28 ms for encryption in the reported test, compared with 20 ms for ECC, 9 ms for CRYSTALS-Kyber, and 12 ms for CRYSTALS-Dilithium. Kyber therefore provides the fastest result in this dataset. The comparison should be interpreted in the context of the implementation library and hardware, but it demonstrates that post-quantum security does not necessarily require slower encryption.

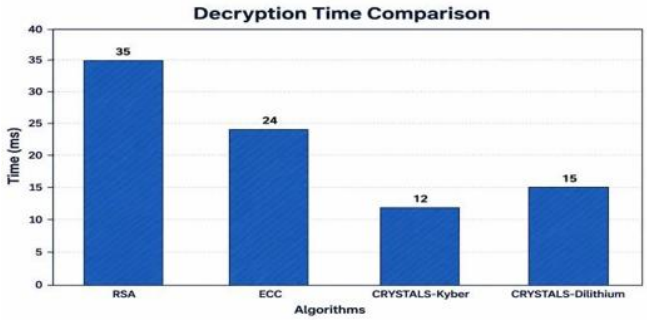

Figure 4: Reported decryption-time comparison

Figure 4 shows RSA also has the highest decryption time at 35 ms. ECC records 24 ms, while Kyber and Dilithium record 12 ms and 15 ms respectively. The lower Kyber value indicates efficient decapsulation for secure key establishment in the reported environment.

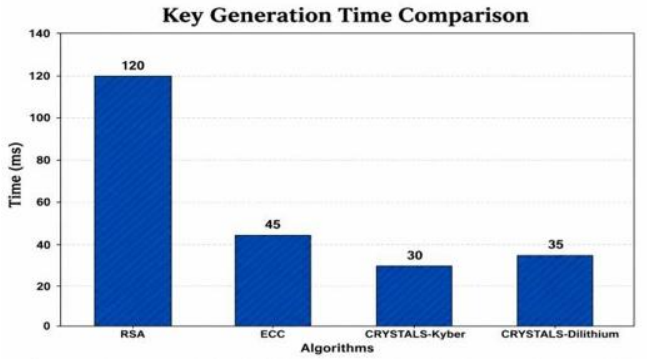

Figure 5: Reported key-generation-time comparison

Key generation produces the largest timing difference. RSA requires 120 ms, ECC requires 45 ms, Kyber requires 30 ms, and Dilithium requires 35 ms. In this experiment, the post-quantum schemes generate their key material faster than the two classical public-key algorithms.

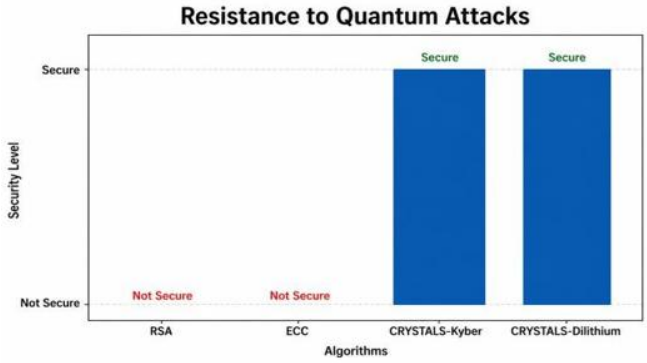

Figure 6: Security classification against quantum attacks

The security comparison highlights the central trade-off. RSA and ECC remain efficient and widely deployed today, but their mathematical foundations are vulnerable to Shor's algorithm. Kyber and Dilithium are designed to withstand known quantum attacks, although they may introduce larger keys, ciphertexts, or signatures. Therefore, algorithm selection should balance security lifetime, processing cost, and infrastructure limitations rather than relying only on execution speed.

Taken together, the results support the use of post-quantum methods for future communication systems. A staged hybrid deployment is especially practical because it allows organisations to maintain compatibility with existing systems while adding a quantum-resistant layer. Long-term migration plans should also include crypto-agility so that algorithms and parameter sets can be replaced without redesigning the entire application.

10. Conclusion

This study compares classical cryptographic techniques with selected post-quantum algorithms using security, performance, and deployment criteria. RSA and ECC continue to provide useful performance for current communication systems, but both depend on mathematical problems that can be solved efficiently by a sufficiently powerful quantum computer. AES is affected less severely; its effective protection against exhaustive search can be strengthened by using a larger key size.

The reported experimental values show that CRYSTALS-Kyber performs competitively in key generation, encryption, and decryption while offering resistance to known quantum attacks. CRYSTALS-Dilithium likewise provides a quantum-resistant option for digital signatures. Their main practical disadvantage is the additional communication and storage overhead associated with larger cryptographic objects.

A gradual hybrid migration is therefore recommended. Organisations can combine existing classical mechanisms with post-quantum key encapsulation or signatures, test compatibility, measure operational cost, and progressively replace vulnerable components. Future work should benchmark standardised implementations on a wider range of devices, include network and energy measurements, and evaluate the effect of post-quantum algorithms on certificates, protocols, and large-scale applications.

Acknowledgement

The author sincerely thanks the guide, Dr. Umadevi Ramamorthy, School of Science and Computer Studies, CMR University, for valuable guidance and support throughout this research.

References

[1] I. Escribano Pablos, M. E. Marriaga, and Á. L. Pérez del Pozo, "Design and Implementation of a Post-Quantum Group Authenticated Key Exchange Protocol With the LibOQS Library: A Comparative Performance Analysis From Classic McEliece, Kyber, NTRU, and Saber," IEEE Access, vol. 10, pp. 120951–120983, 2022, doi: 10.1109/ACCESS.2022.3222389.
[2] M. Mosca, "Cryptographic Standards in a Post-Quantum Era," IEEE Security & Privacy, vol. 20, no. 6, pp. 81–87, 2022.
[3] A. Alagic et al., "Post-Quantum Cryptography: Standardization and Future Directions," IEEE Security & Privacy, 2022.
[4] S. A. Kumar et al., "Performance Evaluation of Post-Quantum Cryptographic Algorithms for Secure Communications," in Proc. IEEE International Conference, 2023.
[5] P. Sharma and R. Gupta, "Efficient Implementation of Lattice-Based Cryptography for IoT Security," in Proc. IEEE International Conference, 2023.
[6] D. J. Bernstein et al., "Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography," IEEE Access, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3364520.
[7] H. Kim et al., "Performance Analysis of ML-KEM (Kyber) and Classical Public-Key Algorithms," in Proc. IEEE Conference, 2024.
[8] R. Singh et al., "Migration from RSA and ECC to Post-Quantum Cryptography: Challenges and Solutions," in Proc. IEEE Conference, 2024.
[9] A. Verma et al., "Comparative Study of Classical and Post-Quantum Cryptographic Algorithms for Secure Networks," in Proc. IEEE Conference, 2024.
[10] A. Khan et al., "Comparative Performance Analysis of Classical and Post-Quantum Cryptographic Algorithms," in Proc. IEEE Conference, 2025.
[11] S. Patel et al., "Post-Quantum Cryptography for Secure Cloud and Network Applications," in Proc. IEEE Conference, 2025.
[12] A. Scrivano, "Institutional Approaches to Post-Quantum Cryptography: A Comparative Analysis of Migration Frameworks," IEEE Access, vol. 14, pp. 3259–3283, 2026, doi: 10.1109/ACCESS.2025.3650465.
[13] S. K. Sharma et al., "Micro and Transport Layer Security Benchmarking of Post-Quantum Versus Classical Cryptography," IEEE Access, 2026.
[14] Y. Chen et al., "Accelerating Post-Quantum Cryptography via Hardware–Software Co-Design," in Proc. IEEE International Symposium on Quality Electronic Design (ISQED), 2026.

Author Profile

Varshini B M, Final-year MCA student, School of Science and Computer Studies, CMR University.

Dr. Umadevi Ramamorthy (Guide), School of Science and Computer Studies, CMR University.