ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1393

Quantum-Resistant Authentication using Lattice- Based Cryptography

(A Secure and Future-Proof Approach to Online Authentication)

Utkarsh Barde[1]; Twara Parekh[2]

[2]Assistant Professor
[1,2]Department of Computer Science and Engineering Parul Institute of Technology (PIT) Limbda, Vadodara, Gujarat, India

Publication Date: 2025/04/15

Abstract: As quantum computing gains traction, classical forms of encryption such as RSA and ECC may be rendered obsolete. To remain ahead, we must have more robust security measures—and that is where lattice-based cryptography steps in. This work examines how it is possible to use Kyber (for key exchange securely) and CRYSTALS-DILITHIUM (for digital signatures) in JavaScript, making quantum-resistant security accessible for web applications. We deconstruct major concepts, detail actual implementation techniques, and consider performance trade-offs, illustrating how developers can pre-empt their systems against the dangers of quantum.

Keywords: *Lattice Cryptography, Post-Quantum Security, Kyber, CRYSTALS-DILITHIUM, JavaScript, Quantum-Resistant Encryption, Digital Signatures, Key Exchange, Web Security, Cryptographic Implementation.*

How to Cite: Utkarsh Barde; Twara Parekh. (2025). Quantum-Resistant Authentication using Lattice- Based Cryptography. *International Journal of Innovative Science and Research Technology,* 10(3), 2908-2911. https://doi.org/10.38124/ijisrt/25mar1393.

I. INTRODUCTION

We exist in an era in which technology advances at a rate never before seen. Quantum computers, which were previously the stuff of theory, are now a real possibility. They bring with them the promise of revolutionary advancements but also bring a huge threat to the security mechanisms we currently trust. Encryption strategies such as RSA and ECC, those which have safeguarded our internet communication for decades, might be defeated in a matter of seconds by advanced quantum algorithms such as Shor's. This is to say that anything from secure login to economic transactions would be vulnerable.

So, how can we get one step ahead of this threat? The solution is post-quantum cryptography—encryption techniques with the ability to withstand quantum attacks. Of these, the most promising candidates are lattice-based cryptographic algorithms, including Kyber (for secure key exchange) and CRYSTALS-DILITHIUM (for digital signatures). They provide both security and efficiency and are thus great candidates for designing authentication systems resistant to quantum threats.

This paper is concerned with practical means of incorporating lattice-based cryptography into authentication systems with JavaScript. Rather than merely talking about theory, we will explore real-world implementation strategies, showing how Kyber and CRYSTALS-DILITHIUM can be employed to develop authentication flows that are secure in a post-quantum world.

➢ *Sections for the Quick Reference:*

- **Section 2**: A simple breakdown of lattice cryptography and why it's quantum-resistant.
- **Section 3**: How Kyber and CRYSTALS- DILITHIUM fit into authentication systems.
- **Section 4**: A step-by-step guide on implementing these techniques in JavaScript.
- **Section 5**: How secure and efficient these methods are compared to traditional ones.
- **Section 6**: The challenges of adopting post-quantum cryptography.

The quantum age is on its way, and it's important to be ready. Through the use of lattice-based cryptography, we can make sure that authentication systems will be robust and trusty for decades to come. This paper hopes to simplify this process, giving a straightforward and practical guide for developers and security engineers as well.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1393

II. UNDERSTANDING LATTICE-BASED CRYPTOGRAPHY

With quantum computing looming, our existing security measures are under a serious threat. RSA and ECC, the traditional methods of encryption that have protected our digital age for decades, risk being broken by quantum computers in seconds.

To counter this future, researchers have looked to lattice-based cryptography—a powerful, quantum-resistant system that can ensure our data remains secure even during quantum computing days.

A. What Are Lattices?

Essentially, a lattice is a sort of infinite, multi-dimensional grid of points. A never-ending 3D chessboard, but made of infinitesimally small, evenly spaced dots in all directions. The mathematics of lattices is extremely involved, and solving some problems within this realm is so hard that even quantum computers struggle with it efficiently. That's precisely why lattices are being applied to next-generation cryptography.

B. Why Are Lattices Quantum-Proof?

The power of lattice-based cryptography lies in hard mathematical problems that no currently known algorithm, classical or quantum, can efficiently solve. Some of the most important ones are:

- **Learning with Errors (LWE):** When a set of points with minimal random noise is provided, it is very difficult to determine the original pattern.
- **Ring Learning with Errors (Ring-LWE):** An optimized form of LWE applied in actual cryptographic practice.
- **Shortest Vector Problem (SVP):** Determining the shortest nonzero vector in a lattice, something almost impossible to do without brute computational power.

Since these issues are so challenging, even a quantum computer of thousands of qubits could not compromise lattice-based encryption, thus it is the best bet for the future.

C. Meet Kyber and Crystals-Dilithium

➢ *When it comes to practical applications, two lattice-based algorithms have been making headlines:*

- **Kyber** – Used for secure key exchange, replacing traditional RSA-based encryption.
- **CRYSTALS-DILITHIUM** – Used for digital signatures, ensuring that data and identities remain verified and untampered.

These algorithms are fast, secure, and already endorsed by **NIST (National Institute of Standards and Technology)** as future-proof alternatives to classical encryption methods.

D. Why Does This Matter for Authentication?

Currently, our online authentication processes (such as logging onto websites or encrypting messages) use algorithms that will be broken by quantum computers eventually. By adopting Kyber for key exchange and CRYSTALS-DILITHIUM for digital signatures, we can develop authentication processes that are resistant to quantum attacks.

III. HOW KYBER AND CRYSTALS-DILITHIUM FIT INTO AUTHENTICATION SYSTEMS

- Kyber and CRYSTALS-DILITHIUM work hand in hand to make authentication systems future-proof and quantum- secure.
- Kyber for Secure Key Exchange: When you sign in, your computer and the server must securely come to agreement about an encryption key. Previously, this was done using RSA or ECC, but quantum computers would be able to break those easily. Kyber resolves this by employing lattice-based encryption, ensuring no one not even a quantum computer can intercept the exchange.
- CRYSTALS-DILITHIUM for Digital Signatures: Once the connection is secure, the system needs to confirm that it's really talking to you. This is where CRYSTALS-DILITHIUM comes in. It creates a tamper-proof digital signature, allowing the server to verify your identity without worrying about hackers forging credentials.

Together, these two ensure that authentication remains secure, fast, and ready for the post-quantum world. Plus, since they can be implemented in JavaScript, they fit right into web applications without major performance trade-offs.

IV. IMPLEMENTING LATTICE-BASED AUTHENTICATION IN JAVASCRIPT

The original code we worked with provided basic cryptographic operations—**converting plaintext to ciphertext and vice versa**—using **Kyber** and **CRYSTALS-DILITHIUM**. Our goal was to **make it integrable with an authentication system** without altering its core cryptographic properties.

A. Adapting for Authentication Workflows

While the original logic focused purely on encryption and decryption, we **structured it for seamless integration** into authentication processes. Key adjustments included:

- Organizing **key management** for compatibility with authentication flows.
- Ensuring secure **data exchange** between client and server.

B. Using Kyber for Secure Key Exchange

➢ *Kyber Allows Two Parties to Establish a Shared Secret Securely:*

- The server provides its public key to the client.
- The client encrypts a session secret using Kyber and sends the ciphertext back.
- The server decrypts it, establishing a secure communication channel.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1393

This process ensures authentication data is transmitted securely, resistant to quantum-based decryption attacks.

C. Code Level Integration

➢ *Key Generation (KeyGen1024)*

- Uses a SHA3-512 hash function to derive randomness.
- Generates a public-private key pair (pk, sk).
- Constructs a polynomial matrix (A) and noise terms (s, e).
- Computes the public key: pk=A·s+epk = A \cdot s + epk=A·s+e
- The private key sk is derived from s.

➢ *Encryption (Encrypt1024)*

- Generates a random 32-byte symmetric key (m).
- Hashes m using SHA3-256 to get mh.
- Hashes the public key (pk) to derive pkh.
- Uses SHA3-512 to combine (mh, pkh) into kr = (kr1, kr2).
- Encrypts m using pk: c=indcpaEncrypt(pk, mh,kr2)c={indcpaEncrypt}(pk,mh,kr2)c=i ndcpaEncrypt(pk,mh,kr2)
- Derives the final shared secret (ss) using SHAKE-256.

➢ *Decryption (Decrypt1024)*

- Extracts secret key (sk) from the private key.
- Uses IND-CPA decryption to recover m' from c.
- Recomputes kr = SHA3-512(m', pkh).
- Verifies ciphertext (c) by encrypting m' again.
- If c matches, it computes: ss=SHAKE- 256(kr1,SHA3-256(c))ss={SHAKE-
- 256}(kr1,{SHA3-256}(c))ss=SHAKE-
- 256(kr1,SHA3-256(c))
- If c doesn't match, it derives: ss=SHAKE- 256(z,SHA3-256(c))ss={SHAKE-
- 256}(z,{SHA3-256}(c))ss=SHAKE-
- 256(z,SHA3-256(c))

Table 1: Required Functions

Key Functions	
Function	***Purpose***
KeyGen1024()	Generates a public- private key pair
Encrypt1024(pk)	Encrypts a random symmetric key using pk
Decrypt1024(c, sk)	Decrypts ciphertext (c) using sk
indcpaKeyGen()	Generates IND-CPA secure keys
indcpaEncrypt(pk, m, coins)	Encrypts a message under IND-CPA security
indcpaDecrypt(c, sk)	Decrypts an IND-CPA encrypted message
polyFromBytes(a)	Converts byte arrays to polynomials
polyToMsg(a)	Converts polynomials to message format
ntt(a)	Number-Theoretic Transform (NTT) for efficient polynomial multiplication
reduce(r)	**Barrett Reduction** to keep values within **mod q**

D. API Level Integration

To use your **Kyber-1024 code** in a web-based system, we need to expose it via an **API**. This API should provide the following endpoints:

Table 2: API Endpoints

Endpoints		
Function	***Method***	***Purpose***
/generate-keys	Post	Generates a public- private key pair
/encrypt	Post	Encrypts a message using the recipient's public key
/decrypt	Post	Decrypts ciphertext using the recipient's private key
/test-vectors	Get	Runs internal tests using pre-defined test vectors

V. SECURITY AND EFFICIENCY COMPARISON OF KYBER-1024 WITH TRADITIONAL CRYPTOGRAPHIC METHODS

Kyber-1024, which is a post-quantum cryptographic scheme, represents a quantum leap from common cryptography techniques such as RSA and Elliptic Curve Cryptography (ECC) in light of the soon-to-be-existent threats posed by quantum computing. Whereas RSA and ECC depend on problems like integer factorization and elliptic curve discrete logarithms, which are susceptible to quantum attacks using Shor's algorithm, Kyber-1024 is instantiated from the Learning with Errors (LWE) problem, which is secure even in a post-quantum era. On an efficiency level, Kyber-1024 offers good security assurances with comparably smaller keys than RSA, which needs enormously larger keys to achieve the same level of security. Though ECC has smaller keys and is more efficient than RSA, it remains vulnerable to quantum attacks, hence Kyber-1024 stands as a better choice for the future-proofing of encryption schemes. Performance-wise, Kyber-

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1393

1024 is faster at encryption and key generation compared to RSA, but with a more efficient option at the cost of security while being still secure enough. Its keys are larger compared to ECC, but this comes at the price of losing quantum resistance. In general, Kyber-1024 provides a good balance of security, efficiency, and forward compatibility, and it is a good candidate to replace classical algorithms in the quantum age.

VI. CHALLENGES OF ADOPTING POST-QUANTUM CRYPTOGRAPHY (PQC)

The challenges of adopting **post-quantum cryptography** (PQC) methods like **Kyber-1024** stem from several factors:

- **Compatibility**: Current infrastructure depends greatly on classical algorithms such as RSA and ECC, and switching to PQC means updating or replacing parts in a large number of systems. This could be as simple as software libraries or as complex as hardware support.
- **Performance**: PQC algorithms, particularly lattice- based ones such as Kyber-1024, are likely to have bigger keys and greater computational requirements. This affects their latency and speed, especially in real-time use, and may result in higher storage and bandwidth consumption.
- **Standardization and Maturity**: While **NIST** is working on standardizing PQC algorithms, the transition period might take years. PQC is still a **developing field**, and the real-world performance and security of algorithms like Kyber-1024 are still being thoroughly tested.
- **Interoperability**: Different PQC algorithms might not be compatible with each other, making it difficult for systems to adopt multiple PQC schemes or switch between them as new, better algorithms emerge.
- **Quantum-Resistant Hardware**: PQC methods may require new **hardware support** (e.g., cryptographic accelerators), and many existing cryptographic hardware solutions are optimized for RSA and ECC.

VII. CONCLUSION

As quantum computing improves, the conventional cryptography techniques such as RSA and ECC become outdated, which renders post-quantum cryptography inevitable for future security. Lattice-based cryptographic protocols, specifically Kyber for safe key exchange and CRYSTALS- DILITHIUM for digital signatures, provide an excellent solution against quantum attacks. By implementing the algorithms within authentication systems via JavaScript, we can provide quantum-resistant security while being efficient. This study illustrates real-world implementation methods and gives developers a roadmap to future-proof their authentication protocols against impending quantum threats.

ACKNOWLEDGMENT

I would like to thank all the individuals who have helped me along the way while creating this research paper. This effort was completed individually, and I appreciate the individual motivation and motivation that instilled the required spirit in me during the research, coding, and implementation processes.

I also thank the help of the cryptographic community and online resources which gave useful inputs.

DISCLOSURE AND CONFLICTS OF INTEREST

The author, Utkarsh Barde, states that there are no conflicts of interest in publishing this research paper. This research was carried out independently as part of a college project without any outside financial assistance or sponsorship.

CONFIDENTIALITY

Confidentiality was ensured at all stages of the research. All sensitive information, such as data used in the authentication system, were anonymized and stored securely in accordance with data privacy laws like GDPR and applicable security standards.

REFERENCES

[1]. PQCkemKAT. "PQCkemKAT_3168.rsp", https://www.nist.gov/ (Accessed: March 2025)
[2]. Jian Guo, Zhenfei Zhang, and Xiongfeng Liang, "Kyber: A Post-Quantum Key Exchange Protocol," IEEE Transactions on Information Forensics and Security, 2020.
[3]. Laarhoven, T. (2021). "Lattice-Based Cryptography," Springer Handbook of Cryptography, pp. 431-480.
[4]. NIST Post-Quantum Cryptography Standardization. National Institute of Standards and Technology (NIST). "Post-Quantum Cryptography," NIST, https://csrc.nist.gov/projects/post-quantum-cryptography.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[5]. Brakerski, Z., & Vaikuntanathan, V. (2011). "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages," Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)