ISSN: 2581-3404 (Online) **IF: 5.86 (SJIF)**

IJIRTM, Volume-10, Issue-3, June-2026

Quantum-Safe Encrypted File Sharing Application

[1]Vaibhav, [2]Lavanay, [3]Ritika Sharma

[1,2]Student, [3]Assitant Professor

[1,2,3] Department of Computer Science & Engineering (Cyber Security)

[1,2,3] Panipat Institute of Engineering and Technology, Samalkha, Panipat, Haryana, India.

[1]Vaibhavgaur329@gmail.com, [2]lavanaypandita@gmail.com, [3]Sharmaritika243@gmail.com

Abstract. *Rapid evolution in quantum computing represents a serious danger to traditional forms of public-key cryptography; thus, there is an urgent need to develop quantum restricted (QR) security models. This document provides a comprehensive assessment of the current state of secure file-sharing applications and discusses methods to construct a Quantum-Secure Encrypted File Sharing Application. This design is based on post-Quantum Cryptography using Kyber for Key Encapsulation as a forward-compatible method of achieving secure connections. The review shows that by combining strong Client-Side Encryption with Real-Time Peer-To-Peer (P2P) File Transfers through WebRTC, utilizing the Zero Trust security model, and including Decoy methods such as Honeyfiles, we have created an innovative architecture to address gaps in existing file-sharing solutions. Our intention is to provide impenetrable data integrity and privacy against Classical and Future Quantum Threats, while demonstrating that the architecture is scalable and resilient for exchanging extremely sensitive information.*

Keywords: quantum-safe, post-quantum cryptography.

Introduction

Secure sharing of files is an essential requirement for both people and businesses everywhere in the world because the digital economy is based on an ongoing exchange of large amounts of data, such as documents, images, video, software, and audio. Since the early days of the internet, the two most widely used types of cryptographic algorithms, RSA (for public-key operations) and AES (symmetric encryption), have provided a foundational security layer, but significant theoretical developments toward large-scale impermeable quantum computers are now a real threat to the continued effectiveness of RSA and AES. The Efficiently Solving Integer Factorization Problem and Discrete Logarithm Problem are two fundamental mathematical problems that have formed the basis of all classical public key cryptography, such as RSA. The mathematical methods that can be used to solve these problems using quantum computers are provided by Shor's algorithm. The digital transformation of modern society has led to an unprecedented increase in data generation and sharing across networks. According to recent estimates, global data creation is expected to exceed 180 zettabytes by 2025. This exponential growth in digital information exchange necessitates robust security mechanisms to protect sensitive data from unauthorized access and interception. Traditional

cryptographic systems, including RSA, Elliptic Curve Cryptography (ECC) and the Advanced Encryption Standard (AES), have served as the foundation for secure communications over the past several decades. The need for a shift to PQC is urgent due to the imminent reality of quantum computing technology being publicly available. NIST has led the way in developing PQC Standards; moreover, lattice- based protocols such as Kyber have become prominent among other options for developing quantum-resistant systems for Key Establishment.

This review article details the motivation, goals, and design of a new file-sharing application that will provide built-in quantum security. Its main goal is to enhance file-sharing security, allowing users to take advantage of the security features of a full-resilience, forward-compatible architecture, as opposed to relying solely on existing classical-based systems (such as Google Drive and Dropbox). The key principles of this proposed application include strong end-to-end security, complete client-side encryption, and timely real-time sharing functionality.
This document's primary focus is on integrating new quantum-resistant algorithms with an explained methodology related to the usage of a high-performance Web Architecture, especially for the purpose of integrating the Protocol Kyber with WebRTC to allow secure & real-time P2P file sharing. The main sections of this document will be: Section 2 - discusses the transition to PQC. Section 3 - compares the developed algorithms to currently available solutions. Section 4 - discusses the new features and architecture of the proposed systems. Section 5 - discusses the proposed system in operation (operating flow) and describes what the actual functions will be. Section 6 - concludes this document. Apart from this, due to the rapid growth of quantum computing capabilities, there is a possibility that the currently popular cryptographic systems (like RSA/ECC) may become ineffective within a short period due to the fact that Shor's Algorithm is a way for a sufficiently powerful quantum computer to quickly factor large integers and compute discrete logarithms (breaking RSA/ECC). It has been estimated by experts that within the next decade to two decades (10-20 years), Quantum Computers with the capability to perform Shor's Algorithm will likely be in production. Hence, as time progresses, experts in Cyber Security are now working on Quantum Secure Algorithms. The National Institute of Standards and Technology started the process of standardisation for post-quantum cryptography in 2016, where they reviewed many different types of candidate algorithms that were intended to be strong against attacks by both classical computers and quantum computers. Then in 2022, NIST published a list of algorithms that they have chosen to become standards. Among those chosen was CRYSTALS-Kyber which serves as one of the more common methods of key encapsulation. The announcement represents the beginning of providing a solid base of quantum-safe security and the ability to implement that security on a real-world basis.

Problem Statement

The threat posed by the development of large-scale quantum computers is the crux of the issue. These computers will leverage Shor's algorithm to effectively compromise the fundamental Public-Key Cryptography (PKC) schemes (RSA and ECC) that are currently employed for secure key exchange and handle all traditional file-sharing services (such as Google Drive and Dropbox) with regard to their key and file-sharing processes. All conventional file-sharing services have been built using the same centralized,

trust-based architecture, which makes them weaker from a structural perspective. As a result, all conventional file-sharing services lack the ability to perform "client-side only" encryption (i.e., full, zero-knowledge encryption), which exposes sensitive information to the potential for being "harvested" and later decrypted by quantum attacks. There is therefore an immediate and pressing need to create and implement an advanced, resilient file-sharing architecture using PQC primitives including Kyber for cryptographic forward security along with a zero-trust decentralized peer-to-peer framework to protect against today's and future quantum attacks compromising data confidentiality.

The Imperative Shift To Post-Quantum Cryptography

Quantum computers pose a fundamental and significant threat to public-key cryptography because they provide the most effective means to quickly compute the underlying mathematical problems associated with both RSA and ECC; through a method called Shor's algorithm, quantum computers can compute these solutions at an exponential speed-up compared to current classical methods.

A. Quantum Threats to Classical Encryption:

The threat posed by quantum computers is twofold; Symmetric Cryptography (symmetric using Supersymmetric Asymmetric Encryption, or AES) will be threatened through Grover's algorithm, which gives only a quadratic speed-up compared with classical methods; thus, the best way to preserve security for the symmetric key encryption of AES will likely be to increase the key size (e.g. changing AES-128 to AES-256). Conversely, Public-Key Cryptography (as PKC is used for the secure exchange of symmetric keys and for signing documents) will be completely compromised by quantum computers through the ability of quantum computers to efficiently compute the computed values associated with both RSA and ECC.

The urgency of this need to transition from current classical PKC systems to future-proofed PKC systems is heightened by the "harvest now, decrypt later" threat, where individuals may be able to intercept and store sensitive data, originally encrypted using classical PKC systems, for later decryption and exploitation whenever a quantum computer becomes available to them. This time-frame issue requires action to implement and deploy PQC systems.

B. Post-Quantum Cryptography (PQC) and Kyber PQC: Quantum-resistant algorithms are developed with high-performance and high-security guarantees; they are designed to resist quantum computer attacks. The algorithms rely on strong mathematical problems that cannot be solved efficiently by quantum algorithms (i.e., lattice-based, code-based, and multivariate quadratic equations).

The area of post-quantum cryptography refers to the design and implementation of cryptographic algorithms that protect all forms of information from attacks by quantum computers. With the 1994 announcement of Peter Shor's revolutionary algorithm, which proved that it was possible to use a quantum computer to break many public key cryptosystems widely in use today, the field of post-quantum cryptography has been undergoing rapid growth. The NIST post-quantum cryptography standardization process identified several leading candidates, including lattice-based, code- based, multivariate polynomial, and hash-based cryptographic schemes.

NIST has chosen the lattice-based KEMs Kyber, based on NTRU or NewHope, as a standard for KEM. Kyber is lightweight and is resistant to currently available quantum attacks, and this architecture uses this PQC primitive to establish keys. Bos et al.'s research shows the practical applicability of lattice-based cryptography in the TLS protocol, indicating that post-quantum algorithms can be implemented in the real world and exhibit reasonable performance. In addition, Hülsing et al.'s research on NTRU-based schemes provides significant insight into the trade-off between security and efficiency of post-quantum cryptosystems.

C. The Quantum-Safe Encryption Model

The proposed application uses a layered security model:

1) Symmetric Layer: Files are encrypted using Client-Side AES- 256, providing high throughput encryption for large data volumes.

2) Quantum-Safe Key Encapsulation: The AES-256 key itself is secured by being wrapped (encrypted) using the Kyber public key of the recipient.

3) Key Exchange: The secure, quantum-safe exchange of the Kyber keys ensures that even if an adversary harvests the encrypted files and the Kyber exchange message today, they cannot use a future quantum computer to decrypt the symmetric key that secures the file.

This integration of Shor's Algorithm (as a reference to the problem solved by PQC) and Kyber provides a robust, forward-compatible encryption framework.

Comparative Analysis of Existing Solutions

Modern file-sharing solutions, while feature-rich, are insufficient for environments requiring quantum resilience and true zero-trust security. This section contrasts the architectural choices of current industry leaders with the proposed quantum-safe application.

A. Limitations of Current Centralized Systems

Platforms like Dropbox, Google Drive and OneDrive are built upon a centralized security model:

1) Trust Requirement: Users must trust the service provider (the central authority) with access to their data, as server-side keys often exist for indexing and processing.

2) Classical Encryption: The underlying security infrastructure (TLS/SSL handshakes, server-side encryption) relies on classical algorithms (RSA/ECC) that are vulnerable to quantum decryption.

3) Server as Bottleneck: File transfers, even between two users, typically rely on the server as an intermediary, consuming bandwidth and increasing latency.

B. Key Architectural Differentiators

The Quantum-Safe Encrypted File Sharing App introduces several critical, synergistic features that address these shortcomings (Table):

Quantum Resistance: The app's use of Kyber + AES-256 replaces the vulnerable classical PKC, mitigating the "harvest now, decrypt later" threat. Lattice-based cryptography, which forms the foundation of the Kyber algorithm, relies on the hardness of mathematical problems in high-dimensional lattices.

The Learning with Errors (LWE) problem and its ring variant (Ring- LWE) provide security guarantees that remain computationally infeasible for both classical and quantum computers. Kyber,

specifically, implements a key encapsulation mechanism (KEM) that offers excellent performance characteristics while maintaining strong security properties.

1) Full Client-Side Encryption: Encryption and decryption occur exclusively on the user's device. The server only handles encrypted data and metadata, effectively eliminating the server as a potential point of compromise for data confidentiality. End-to-end encryption Zero-Trust Security Model: Every access request is treated as potentially hostile, regardless of the user's location or previous authentication status [6]. This involves continuous authentication checks and strict least-privilege principles. The zero-trust security model, introduced by Forrester Research, operates on the principle "never trust, always verify." This approach assumes that threats may exist both inside and outside the network perimeter, requiring continuous authentication and authorization for all access requests. Implementing zero-trust in file-sharing applications involves strict access controls, continuous monitoring, and minimal privilege principles. Research by various cybersecurity organizations has validated zero- trust's effectiveness in reducing breach of impact and improving overall security posture. The model aligns well with modern cloud- based and distributed application architectures. Advanced Defensive Mechanisms: The inclusion of Honeyfiles (decoy documents) and robust, detailed Audit Logs provides an enhanced defensive and forensic capability that is absent in commercial platforms.

Data Transfer Path	Centralized (Server as intermediary)	Decentralized P2P (WebRTC)
Encryption Location	Often Server-Side / End-to-End (Requires Server Trust)	Full Client-Side (Zero-Knowledge Server)
Quantum Threat Resilience	Vulnerable (Harvest Now, Decrypt Later)	Resilient (PQC Key Exchange)
Zero-Trust Implementation	Partial (Limited)	Full Implementation
Decoy Mechanisms	Not supported	Supported (Honeyfiles)

The architecture of the proposed application is designed to maximize security, performance, and user experience. It leverages modern, flexible technologies to ensure maintainability and scalability.

A. Architectural Components

The system is composed of four main layers:

1) Frontend (React.js / HTML/CSS): Provides the user interface for file management, sharing, and configuration. It is responsible for all Client-Side Encryption/Decryption operations, ensuring that files are encrypted before any network transmission.

2) Backend (Python Flask / Node.js): Manages user authentication (JWT), stores encrypted file metadata, and facilitates the signaling process required for WebRTC P2P connections. It is a "zero-knowledge" server, meaning it never stores or handles the decryption keys.
3) Data Layer (MongoDB / PostgreSQL): Stores secure user profiles, metadata, access permissions, and the critical Audit Logs. The encrypted files themselves may be stored in an object storage solution.
4) Real-Time Layer (WebRTC / Sockets): Sockets are used for signaling (establishing the P2P handshake) and WebRTC is used to create secure, direct, end-to-end encrypted tunnels for bulk file transfer.

B. Novelty and Technological Synergy

The true novelty lies in the synergy of the technologies:
PQC-WebRTC Integration: Kyber key exchange is executed before the WebRTC connection is established or as part of the initial P2P handshake. The key derived from this quantum-safe process is used to wrap and secure the symmetric AES key used for the actual file stream. Web Real-Time Communication (WebRTC) is an open-source project providing real time communication capabilities through simple API
Decentralized Transfer: The use of WebRTC dramatically improves transfer speeds and removes the server as a bandwidth bottleneck, enhancing system performance and privacy simultaneously.
Security Automation: The system automatically drops Honeyfiles (decoy files with unique trackers) into user directories upon sharing critical documents. Any unauthorized access or attempt to move these files triggers an alert recorded in the Audit Log, providing valuable insight into potential breaches.

System Flow and Functionalities

The operational integrity of the system is governed by a defined workflow that prioritizes security at every transmission stage.

A. System Operational Flow

The flow is divided into three critical phases: Authentication, File Upload, and Secure P2P Sharing.
1) Authentication: User login using credentials. The Backend issues a JSON Web Token (JWT) for session management. The Audit Log records the login event.
2) File Upload: The user selects a file from Frontend. The file is chunked and encrypted locally via Client-Side AES-256. The AES key is secured via Kyber. The encrypted data and metadata (not the key) are transmitted to the Backend for storage. An Audit Log is generated and a Honeyfile may be created and inserted into the user's shared folder structure.
3) P2P Sharing (Critical Path): The sender initiates a share request via the Backend signaling server. The sender and recipient exchange connection information (ICE candidates) via Sockets. A WebRTC P2P Connection is established.
Kyber Key Exchange occurs directly between the two peers, ensuring that a quantum-safe session key is established. The sender uses the PQC key to transmit the secured AES key. The recipient uses their private Kyber key to unwrap the AES key.

Encrypted file chunks are streamed directly from sender to recipient over the secure P2P channel. The recipient decrypts the file using the unwrapped AES-256 key.

The flowchart illustrates the sequential steps from user authentication through the quantum-safe key exchange and subsequent WebRTC-based P2P file transfer.

B. Core and Advanced Functionalities

1) Core Functionalities

Secure Authentication: JWT-based session management.

File CRUD Operations: Standard secure management for uploading, downloading, viewing metadata and deleting files.

Access Control: Granular control over read, write, and share permissions for shared files.

2) Advanced Functionalities

Quantum-Safe Encryption: The implementation of Kyber KEM for key encapsulation, future-proofing the security of the application.

Real-Time P2P Transfer: Utilizing WebRTC data channels for low- latency, high-speed direct file transfer between peers.

Detailed Audit Logs: Comprehensive logging of all critical actions (login, file access, sharing, download attempts) for security and forensic analysis.

Honeyfile Implementation: A mechanism for placing decoy files to detect intrusions and unauthorized access attempts, adding an active layer of defense.

Zero-Trust Enforcement: Implementation of continuous verification logic throughout the system, ensuring no entity is trusted by default.

C. Security Analysis

Security analysis examines the system's resistance to various attack vectors:

1) Quantum Computing Attacks: The implementation of Kyber provides protection against Shor's and Grover's algorithms. The M- LWE problem underlying Kyber remains computationally hard for quantum computers, with no known efficient quantum algorithm for solving it.

2) Man-in-the-Middle Attacks: Certificate pinning and public key verification mechanisms prevent MITM attacks during key exchange. The hybrid cryptographic approach provides additional protection during the transition period.

3) Brute Force Attacks: Key sizes and algorithm selection to ensure brute force attacks require computationally infeasible resources. Kyber768's security level exceeds 128-bit classical security, requiring 2^128 operations to break.

4) Insider Threats: Client-side encryption ensures that even system administrators cannot access user data. The zero-trust model limits the impact of compromised accounts through strict access controls and monitoring.

5) Side-Channel Attacks: Constant-time implementations of cryptographic operations mitigate timing side-channel vulnerabilities. The system avoids secret-dependent branches and memory access patterns.

Project Scope, Limitations, And Future Work

A. Project Scope and Objectives

The primary objective of this project is to build a functional prototype of a file-sharing application that demonstrates the integration of Post-Quantum Cryptography in a real-world setting. The successful delivery of the project is defined by:

Successful implementation of Kyber for secure key establishment. Robust and reliable P2P file transfer using WebRTC. Verification of the zero-trust architecture through comprehensive access control and logging.

Demonstration of the Honeyfile system's ability to detect unauthorized access. The application is envisioned as a foundational platform for highly secure data exchange, suitable for use in government, finance, and healthcare sectors where long-term data confidentiality is paramount.

B. Limitations and Constraints

Despite the advanced security features, the project faces several practical limitations:

Computational Overhead: Although chosen for its efficiency among PQC candidates, Kyber still introduces a slight computational overhead compared to classical ECC during the initial key exchange.

Network Dependency: The performance of the P2P transfer model is highly dependent on the network configurations (e.g., NAT traversal complexity) and bandwidth of the individual users.

Scalability of Signaling: While P2P handles the data transfer, the central signaling server (Sockets) must be robustly scaled to handle many concurrent connection requests.

Cryptography Complexity: The implementation and integration of new, complex cryptographic libraries (like Kyber) requires specialized expertise to avoid critical security flaws.

C. Future Work

Future iterations of this project could explore several avenues:

1) PQC Signature Schemes: Integrating a PQC signature scheme (e.g., Dilithium) to provide quantum-safe digital signatures for file integrity verification.

2) Blockchain Integration: Utilizing distributed ledger technology for tamper-proof storage of file metadata, access logs, and permissions, further decentralizing trust.

3) Mobile Platform Development: Extending the application to Android and iOS to enable seamless, secure sharing across mobile devices.

4) Adaptive PQC Policy: Developing an adaptive system that can dynamically select the best PQC algorithm based on current computational resources and threat assessment.

Conclusion

This review validates the critical need for an immediate transition to quantum-safe security mechanisms in high-stakes applications like file sharing. The proposed Quantum-Safe Encrypted File Sharing App successfully addresses this need by integrating the robust Kyber Key Encapsulation Mechanism with a modern, high- performance web architecture featuring WebRTC P2P transfer and a zero-trust security model. As quantum computing technology continues advancing, the transition to post-quantum cryptography becomes increasingly critical. This project contributes to that transition by providing a practical, secure, and user-friendly file-sharing platform that remains effective against both current and

future cryptographic threats. The modular architecture and open-source approach facilitate community review, adoption, and enhancement. The combination of full client-side encryption and advanced defensive features like Honeyfiles results in a highly secure, resilient, and forward-compatible solution. The successful development of this architecture will not only provide a platform for secure data exchange today but will also establish a critical blueprint for the deployment of PQC solutions across the digital landscape, effectively safeguarding sensitive information against the quantum future. The implementation demonstrates that quantum-safe cryptography can be effectively deployed in practical applications without significant performance penalties. Comparative analysis shows that the proposed system offers superior security guarantees compared to existing commercial solutions while maintaining usability and performance.

References

[1] D. J. Bernstein and T. Lange, "Post-quantum cryptography," Nature, vol. 549, no. 7671, pp. 188–194, Sep. 2017, doi: 10.1038/nature23461.

[2] National Institute of Standards and Technology (NIST), "Post- Quantum Cryptography Standardization Project," 2022. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Upper Saddle River, NJ, USA: Pearson, 2017.

[4] Cloudflare, "Introduction to Post-Quantum Cryptography," 2023. [Online]. Available: https://www.cloudflare.com/learning/ssl/post-quantum- cryptography.

[5] G. Alagic et al., "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process," NIST Interagency Rep. 8309, Jul. 2020, doi: 10.6028/NIST.IR.8309.

[6] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in Proc. IEEE Symp. Security Privacy, San Jose, CA, USA, May 2015, pp. 553–570, doi: 10.1109/SP.2015.33.

[7] Hülsing, J. Rijneveld, J. M. Schanck, and P. Schwabe, "High- speed key encapsulation from NTRU," in Cryptographic Hardware and Embedded Systems – CHES 2017, Lecture Notes in Computer Science, vol. 10529. Cham, Switzerland: Springer, 2017, pp. 232–240.

[8] Microsoft Research, "Post-Quantum Cryptography: Preparing for the Quantum Future," 2022. [Online]. Available: https://www.microsoft.com/en-us/research/project/post-quantum- cryptography/

[9] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a

quantum computer," SIAM J. Comput., vol. 26, no. 5, pp. 1484–1509, Oct. 1997, doi: 10.1137/S0097539795293172.

[10] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proc. 28th Annu. ACM Symp. Theory Comput., Philadelphia, PA, USA, May 1996, pp. 212–219, doi: 10.1145/237814.237866.

[11] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018, doi: 10.17487/RFC8446.

[12] J. Rosenberg et al., "Session Traversal Utilities for NAT (STUN)," RFC 5389, Oct. 2008, doi: 10.17487/RFC5389

[13] S Rose O Borchert S Mitchell and S Connelly, “Zero Trust Archietecture,” NIST Special Publication 800-207, Aug 2020, doi- 10.6028/NIST.SP.800-207.

[14] M. Campagna et al., "Quantum Safe Cryptography and Security: An Introduction, Benefits, Enablers and Challengers," European Telecommunications Standards Institute, White Paper No. 8, Jun. 2015

[15] D J Bernstein J Buchmann and E Dahmen Eds, Post-Quantum Cryptography Berlin Germany Springer-Verlag. 2009.