Quantum Cryptography Using Cloud Storage System

Mrs. Bhagyashri Wakde
Department of Computer Science and Engineering
Rajiv Gandhi Institute of Technology
Bangalore, India
bhagyashelke2015@gmail.com

M PREMANANDA
U.G Student, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore, India.
premanandabspp@gmail.com

GANESH
U.G Student, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore, India.
ganeshnbbombulage@gmail.com

AISHWARYA K
U.G Student, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore, India.
aishu17407@gmail.com

CHETAN GOWDA K
U.G Student, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore, India.
radhakrishnachetan154@gmail.com

***Abstract*—** This paper presents a secure, scalable file transfer and storage model using Quantum Cryptography integrated with Cloud Storage Systems. The solution leverages the principles of Quantum Key Distribution (QKD) to enable an unbreakable encryption process, eliminating the threat of man-in-the-middle and brute-force attacks. Unlike traditional cryptographic systems, which rely on computational hardness assumptions, quantum cryptography provides provable security guaranteed by the laws of quantum physics. Files are encrypted using symmetric algorithms with keys distributed through quantum channels. The model also includes secure authentication layers and access control using biometric and OTP-based mechanisms. Integration with cloud platforms ensures data redundancy, availability, and on-demand access for global users. The integration of this model with cloud platforms ensures secure storage and access scalability for enterprise and academic applications.

Keywords: *Quantum Cryptography, QKD, Cloud Security, Cloud Storage, Secure File Transfer, Quantum Key Distribution, Privacy, AES-256, Authentication.*

1. INTRODUCTION

As data breaches and cyber threats continue to rise, the need for a secure file transfer system has become critical. Traditional encryption techniques are increasingly vulnerable to attacks, especially with the advent of quantum computing. Quantum cryptography, based on the principles of quantum mechanics, provides a new paradigm of security through Quantum Key Distribution (QKD). This project introduces a system that employs quantum cryptographic techniques to secure file transfers on cloud platforms, thereby ensuring confidentiality, integrity, and secure access control.

As digital transformation accelerates across industries, the volume of sensitive information being shared and stored online continues to surge. Cloud platforms have become indispensable for enterprises and individuals due to their convenience, scalability, and storage efficiency. However, the increasing dependence on cloud services has raised serious concerns about data privacy, leakage, and unauthorized access.

Traditional encryption methods such as RSA, ECC, and AES are secure under classical computation. However, with the advent of quantum computing, many of these systems will become vulnerable. Algorithms like Shor's algorithm threaten to break widely used public-key encryption schemes in polynomial time, undermining the core of current security systems.

Quantum Cryptography, particularly Quantum Key Distribution (QKD), provides a new class of security mechanisms that are immune to computational attacks. This research paper proposes a system that utilizes QKD to generate and distribute encryption keys securely and integrates these with cloud storage platforms for secure file transfer and access.

II. LITERATURE SURVEY

Multiple approaches have been taken to ensure cloud security over the past two decades. The current gold standard includes AES-256 encryption for symmetric security and RSA-2048 or ECC for key exchange. However, all these rely on computational security — the idea that it is hard for attackers to compute the solution.

Quantum cryptography introduces information-theoretic security, where interception or measurement of quantum states causes disturbances that are detectable. The BB84 protocol introduced by Bennett and Brassard in 1984, remains the most implemented QKD protocol. It has proven to be robust in real-world environments and experimental setups.

Recent studies, such as the SECOQC project in the EU and Tokyo QKD Network in Japan, have implemented large-scale QKD networks, proving feasibility. Research also shows successful simulation of QKD over 1000+ kilometres via satellite-based quantum communication, opening the possibility for secure global communication networks.

However, the integration of such quantum cryptographic schemes with mainstream technologies like cloud computing remains limited. Our work aims to close this gap.

III. EXISTING SYSTEM

Currently, data stored on cloud platforms is encrypted using symmetric or asymmetric cryptographic algorithms. These systems are vulnerable in the following ways:

A. Vulnerability to Future Attacks:

RSA and ECC rely on mathematical complexity, which quantum computers can solve efficiently using algorithms like Shor's. Once quantum machines become practical, these systems will be easily broken, risking global data security.

B. Key Exchange Limitations:

Traditional methods like Diffie-Hellman are exposed to man-in-the-middle attacks. Traditional exchanges like Diffie-Hellman are not immune to interception. Attackers can pose as legitimate users during key transmission, leading to silent data breaches.

C. Server-Side Risks:

Without strict end-to-end encryption, cloud providers may access or leak stored keys. This makes user data vulnerable even when encrypted, especially in multi-tenant environments. Cloud providers may have access to the encryption keys if end-to-end encryption is not enforced.

D. Lack of Provable Security:

Classical systems offer computational security, assuming attacks are impractical. Quantum cryptography, in contrast, offers security proven by the laws of physics. Existing methods provide computational but not provable guarantees. Additionally, breaches in big companies like Dropbox, iCloud, and Facebook highlight the need for stronger security mechanisms that are future-proof and immune to computational advancement.

.

IV. PROPOSED METHOD

The proposed framework introduces a quantum-secure cloud file transfer system that enhances data privacy and integrity through two core modules: Quantum Key Distribution (QKD) and Secure Cloud Encryption. The system adopts a modular architecture, allowing seamless integration with existing cloud platforms such as AWS, Google Cloud, and private cloud environments without requiring major infrastructural changes.

A. System Architecture

The architecture consists of four key layers:

User Interface Layer: A lightweight web interface accessible via browser or desktop client, enabling users to upload, encrypt, and download files with ease, while also providing authentication prompts.

Quantum Key Layer: A simulation or real implementation of QKD using BB84 protocol, which generates encryption keys using quantum principles and ensures that any eavesdropping attempt causes detectable anomalies.

Encryption & Transfer Layer: A module that uses the quantum key to perform AES-256 encryption and securely uploads files to the cloud. It also handles decryption upon verified access.

Authentication Module: A security service enforcing multi-factor authentication using biometrics and OTP before allowing access to cloud storage or file downloads.

B. Quantum Key Distribution Feature

The system begins with secure quantum key generation through the BB84 protocol. Photons are encoded with quantum states and transmitted between sender and receiver. Any interception attempt by a third party collapses the quantum states, alerting the system to the breach. Once verified, the key is stored temporarily for encryption/decryption use.

C. Secure Cloud Encryption Module

After successful QKD, the quantum key is used to encrypt files using AES-256 before uploading them to the cloud. The file remains encrypted at rest and in transit. Only users with the corresponding quantum key and valid credentials can retrieve and decrypt the file, ensuring absolute data privacy.

D. Plug-and-Play Integration

The system uses standardized APIs and RESTful services that allow it to integrate into existing cloud platforms like Google Cloud, AWS, or private servers. It does not interfere with existing file structures or access protocols but wraps encryption and key exchange modules around them. This plug-and-play architecture simplifies enterprise adoption without disrupting legacy systems.

E. Future Extensions

Planned enhancements include: full integration with real QKD hardware for physical quantum key exchange, blockchain logging for immutable file access records, automated quantum key regeneration for session-based encryption, and mobile application support. These additions aim to further enhance usability, scalability, and future readiness of the system for academic, government, and enterprise use.

V. WORKING PRINCIPLE

The proposed system operates by integrating quantum key distribution with classical encryption and cloud storage protocols to create a seamless and highly secure data transfer environment. Each step of the process ensures that both the encryption keys and the encrypted files are protected from unauthorized access, eavesdropping, or data tampering.

A. Key Generation and Distribution

The system begins with the simulation or real implementation of the BB84 protocol to establish a secure quantum key between sender and receiver. Quantum bits (qubits) are encoded with polarized photons and transmitted through a quantum channel. Due to the no-cloning theorem and Heisenberg's uncertainty principle, any attempt to intercept these qubits disturbs their states, thereby alerting both parties. Only matching bits after basis comparison are retained as the final key.

B. File Encryption and Upload

Once the quantum key is successfully exchanged and verified, the user selects a file to upload. The file is encrypted on the client side using the AES-256 symmetric encryption algorithm, with the quantum key acting as the seed. The encrypted file is then uploaded to a secure cloud server. This ensures that no unencrypted data ever reaches the cloud, and the cloud provider cannot access the file contents even if storage is compromised.

C. Receiver Access and Decryption

The authorized receiver must authenticate using multi-factor authentication, including biometric verification and OTP validation. Upon successful login, the system verifies the stored quantum key. The encrypted file is then downloaded and decrypted locally using the same quantum key. If the quantum key mismatch occurs due to tampering, the decryption fails, ensuring end-to-end confidentiality.

D. Session Logging and Alerts

Each step—from login to key exchange, file encryption, upload, access, and decryption—is logged with timestamped metadata. Alerts are sent to both sender and receiver in the case of failed authentication, mismatched keys, or access attempts from unknown devices. This monitoring ensures traceability, transparency, and forensic readiness in the event of a breach attempt.

VI. IMPLEMENTATION AND TECHNOLOGY

The proposed system is implemented using a hybrid technology stack that integrates classical encryption, quantum key simulation, and secure cloud storage. The design emphasizes platform independence, real-time security, and user accessibility. Each layer is modular and scalable, allowing the system to be deployed across academic, enterprise, or government environments with minimal customization.

A. Technology Stack

Frontend: JavaFX for a responsive and modern user interface that supports secure file selection, encryption, upload, and download functionality. Swing components and FXML are used for modular UI design, and platform compatibility across Windows/Linux/macOS is maintained.

Backend: Core Java with multithreading support to handle encryption, key generation, and communication processes. Java RMI is used for simulating remote quantum key exchange, while Java I/O handles secure file processing.

Quantum Key Simulation: BB84 protocol implemented in Java using custom algorithms to simulate photon transmission and basis selection. Secure Random is used to generate bit sequences, and key reconciliation logic detects and removes any intercepted bits.

Encryption Engine: AES-256 encryption using Java Cryptography Extension (JCE) with Cipher class in CBC mode and PKCS5Padding. Keys generated via BB84 are applied directly to encrypt the file before upload.

Cloud Integration: Encrypted files are uploaded using HTTP clients or official SDKs for Google Cloud Storage or AWS S3. Access is managed using pre-signed URLs generated by the backend, ensuring restricted and time-bound file access.

Database: Firebase Realtime Database or MongoDB (via Java drivers) is used to store user metadata, file access logs, timestamps, and key usage history.

Security Services: OTP authentication implemented via Java Mail API, biometric integration via external SDKs (e.g., Windows Hello/Android fingerprint API), and secure session handling via UUID-based tokens and encrypted cookies.

Logging & Alerts: Java's built-in logging APIs (e.g., java. util. logging or Log4j) are used to track actions like login, upload, decryption, and unauthorized attempts. Logs are timestamped and stored in the cloud for audit trails.

B. Implementation Workflow

User Onboarding: User registers and logs in through the JavaFX interface. Biometric or OTP-based authentication is required to access the file encryption module.

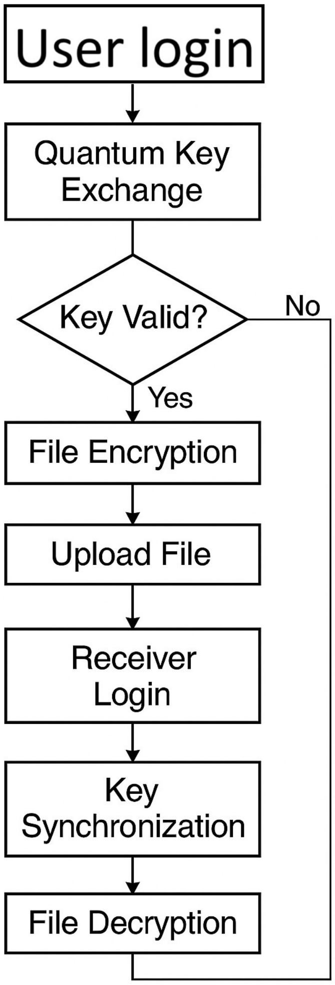
Quantum Key Exchange: A BB84 protocol simulation is initialized. The sender and receiver simulate photon transmission and basis selection, generating a shared quantum key. Any discrepancies due to eavesdropping are detected during reconciliation.

File Encryption & Upload: User selects a file. The system uses the quantum key to encrypt the file with AES-256. The encrypted file is then uploaded securely to cloud storage using Java's HTTP libraries or SDK integrations.

File Access & Decryption: Authorized receivers verify their identity and receive the encrypted file. Using the matching quantum key, the file is decrypted locally on the client side. Any mismatch or failed validation triggers alerts and blocks access.

Monitoring & Alerts: Access attempts are monitored continuously. Unrecognized devices, repeated OTP failures, or mismatched quantum keys result in immediate user notifications and session lockouts.

Figure 1 shows the operational flow of the proposed Quantum-Secure File Transfer System. The user begins by registering and authenticating. Once verified, a quantum key is exchanged using a simulated BB84 protocol. The file is then encrypted and uploaded to cloud storage. At any time, only verified users with the correct quantum key and authentication credentials can decrypt and retrieve the file securely. The system continuously logs all access points, ensuring full transparency and traceability.



.

References

[1] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, India, 1984, pp. 175–179.

[2] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, 2002.

[3] H.-K. Lo, M. Curty, and K. Tamaki, "Secure quantum key distribution," *Nature Photonics*, vol. 8, no. 8, pp. 595–604, 2014.

[4] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proceedings 35th Annual Symposium on Foundations of Computer Science*, IEEE, 1994, pp. 124–134.

[5] V. Scarani et al., "The security of practical quantum key distribution," *Reviews of Modern Physics*, vol. 81, no. 3, pp. 1301–1350, 2009.

[6] M. Peev et al., "The SECOQC quantum key distribution network in Vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[8] M. D. Le and J. Kim, "Secure file storage and sharing system using hybrid encryption with blockchain," *IEEE Access*, vol. 8, pp. 186569–186580, 2020.