A Quantum Key Distribution Based Secure Document Encryption and Retrieval System

Sivakami R, Sandeep Rishi J B, Rishikesh C, Risthish Anto A, Shivam Kumar M
Department of Computer Science and Engineering
Sona College of Technology, Salem, India

***Abstract*—The rapid growth of digital communication and cloud-based storage has significantly increased the risk of data breaches and unauthorized access to sensitive documents. Traditional cryptographic systems rely on computational complexity, which may be vulnerable to future quantum computing attacks. Quantum Key Distribution (QKD) offers information-theoretic security by leveraging the principles of quantum mechanics. This paper proposes a hybrid secure document encryption and retrieval system that integrates simulated QKD using the BB84 protocol with classical AES-256 encryption. The system is implemented using a Streamlit frontend and a FastAPI backend, enabling secure document upload, encryption, storage, and retrieval. Experimental evaluation demonstrates the feasibility, scalability, and enhanced security of integrating quantum-inspired key distribution into real-world document security applications.**

***Index Terms*—Quantum Key Distribution, BB84 Protocol, Secure Document Storage, AES-256, Quantum Cryptography, Hybrid Encryption**

I. Introduction

With the exponential growth of digital data exchange, ensuring secure storage and transmission of sensitive documents has become a critical challenge. Conventional cryptographic algorithms such as RSA and AES rely on computational hardness assumptions, which may be compromised by advances in quantum computing.

Quantum Key Distribution (QKD) provides a fundamentally secure approach to key exchange by exploiting quantum mechanical principles such as superposition and measurement disturbance. Any eavesdropping attempt introduces detectable anomalies in the communication channel. This work presents a practical implementation of a QKD-assisted secure document system that integrates quantum-inspired security into classical web architectures.

II. Motivation and Problem Statement

Existing secure document systems face several limitations:

- Vulnerability of classical key exchange mechanisms.
- Absence of real-time eavesdropping detection.
- Increasing threat from quantum-enabled adversaries.

Problem Statement: To design and implement a scalable secure document encryption and retrieval system that leverages Quantum Key Distribution for secure key generation and classical encryption for data protection.

III. Related Work

Bennett and Brassard introduced the BB84 protocol, laying the foundation for quantum cryptography. Subsequent studies explored QKD over optical fibers, free-space links, and satellite communication. Hybrid systems combining QKD with classical encryption have been proposed; however, many lack full-stack implementations. This work bridges that gap by providing an end-to-end system.

IV. System Architecture

The proposed system follows a layered architecture as shown in Fig. 1.

A. Frontend Layer

Implemented using Streamlit, the frontend enables users to upload documents securely and download decrypted files.

B. Backend Layer

The FastAPI backend handles QKD key generation, encryption/decryption, API logic, and communication with storage layers.

C. Storage and Database Layer

Encrypted documents are stored on the server filesystem, while metadata and quantum keys are stored in a relational database.

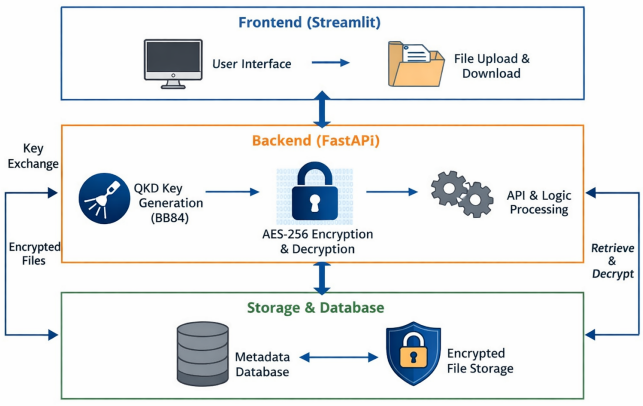

Fig. 1. Layered Architecture of the QKD-Based Secure Document System

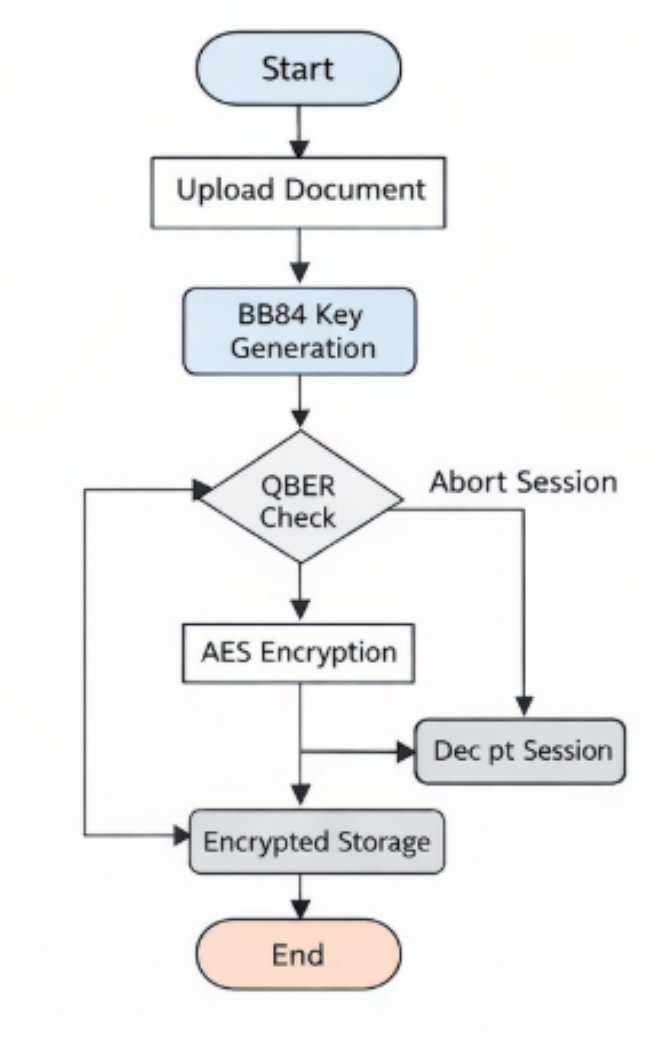

Fig. 2. Secure Document Upload and Retrieval Workflow

V. Quantum Key Distribution Methodology

The BB84 protocol is simulated to generate shared secret keys. Alice prepares random bit sequences using randomly chosen bases, while Bob measures them using independent random bases. After basis reconciliation, matching bits form the shared key.

The Quantum Bit Error Rate (QBER) is calculated as:

$$QBER = \frac{N_{error}}{N_{total}} \times 100 \quad (1)$$

If the QBER exceeds a threshold, the session is terminated, indicating possible eavesdropping.

VI. Encryption and Decryption Process

The generated quantum key is mapped to a 256-bit key and used in AES encryption.

A. Encryption

- Document padding to meet block size requirements.
- AES-256 encryption of document data.
- Storage of encrypted files.

B. Decryption

- Retrieval of encrypted document.
- Fetching corresponding quantum key.
- AES decryption to restore original document.

VII. Implementation Details

The system is implemented in Python using modern frameworks. Table I summarizes the technology stack.

TABLE I
Technology Stack

Component	Technology
Frontend	Streamlit
Backend	FastAPI, Uvicorn
QKD Simulation	BB84 Protocol
Encryption	AES-256 (PyCryptodome)
Database	SQLite (SQLAlchemy)

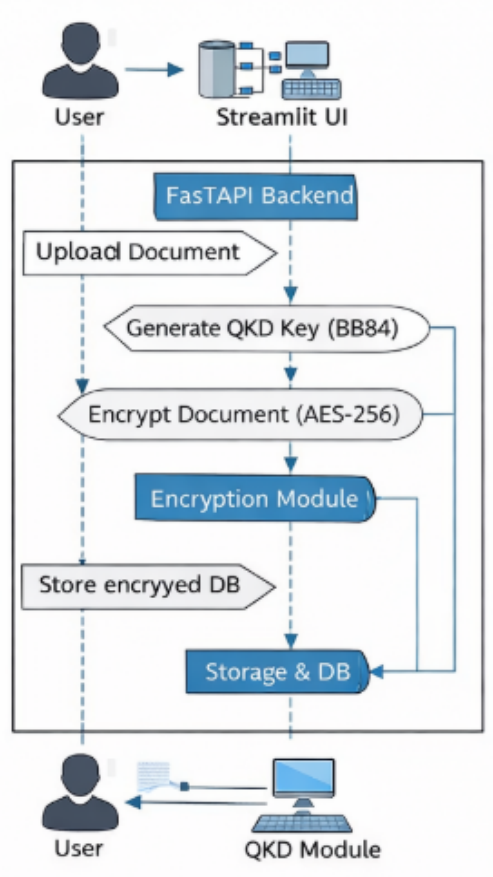

Fig. 3. Sequence Diagram Showing Frontend–Backend Interaction

VIII. Threat Model and Security Analysis

The system considers man-in-the-middle, eavesdropping, and brute-force attacks. QKD detects interception through elevated QBER values, while AES-256 ensures resistance to brute-force attacks.

IX. Performance Evaluation

Experiments were conducted on different file types. Table II summarizes performance results.

TABLE II
PERFORMANCE EVALUATION

File Type	Size (KB)	Encrypt (ms)	Decrypt (ms)
Text	50	12	10
PDF	500	28	25
Image	1024	45	42

X. Comparison with Classical Systems

TABLE III
COMPARISON WITH CLASSICAL SECURE SYSTEMS

Feature	Classical Systems	Proposed System
Key Exchange	RSA/DH	QKD (BB84)
Eavesdrop Detection	No	Yes
Quantum Resistance	Limited	High
Security Basis	Computational	Information-Theoretic

XI. Mathematical Model of the Proposed System

Let $U = \{u_1, u_2, ..., u_n\}$ represent the set of authorized users and $D = \{d_1, d_2, ..., d_m\}$ represent the set of documents.

The system can be formally defined as a tuple:

$$S = (U, D, K_q, E, De)$$

where:

- K_q is the quantum-generated secret key obtained using BB84,
- E represents the encryption function,
- De represents the decryption function.

For a document $d_i \in D$, encryption is performed as:

$$C_i = E(d_i, K_q)$$

Decryption is defined as:

$$d_i = De(C_i, K_q)$$

Security is guaranteed if:

$$Pr(K_q = K_e) < \epsilon$$

where K_e is the key obtained by an eavesdropper and ϵ is a negligible probability.

XII. Detailed BB84 Algorithm Description

The BB84 protocol operates through the following steps:

A. Initialization

Alice generates a random binary sequence:

$$B_A = \{b_1, b_2, ..., b_n\}$$

and a random basis sequence:

$$\Theta_A = \{\theta_1, \theta_2, ..., \theta_n\}$$

where $\theta_i \in \{+, \times\}$.

B. Transmission

Each bit is encoded into a quantum state and transmitted over the quantum channel.

C. Measurement

Bob independently selects a random basis sequence:

$$\Theta_B = \{\phi_1, \phi_2, ..., \phi_n\}$$

and measures the received states.

D. Sifting

Alice and Bob publicly compare their bases and retain only matching positions:

$$K_{raw} = \{b_i \mid \theta_i = \phi_i\}$$

E. Error Estimation

A subset of K_{raw} is compared to compute the Quantum Bit Error Rate (QBER).

XIII. Secure Workflow and Sequence of Operations

The secure document lifecycle follows a well-defined sequence:

1) User uploads a document via the frontend.
2) Backend initiates BB84-based quantum key generation.
3) Generated key is validated using QBER estimation.
4) Document is encrypted using AES-256.
5) Encrypted document is stored securely.
6) Upon request, encrypted data is retrieved.
7) Decryption is performed using the shared quantum key.

This workflow ensures confidentiality at every stage of document handling.

XIV. Security Proof Sketch

The security of the proposed system is derived from the combination of QKD and AES.

A. Key Security

BB84 guarantees that any eavesdropping attempt introduces detectable disturbances. Let E be an adversary attempting to intercept the channel.

$$Pr(E \text{ remains undetected}) \leq 2^{-n}$$

B. Data Security

AES-256 provides computational security such that:

$$Pr(\text{break AES}) \approx 2^{-256}$$

Thus, the joint system achieves strong security against both classical and quantum adversaries.

XV. Reliability and Fault Tolerance Analysis

The system ensures reliability through persistent storage and database-backed key management. Even in the event of system restart or network failure, encrypted documents remain intact.

Failure scenarios such as partial uploads, key mismatch, or corrupted data are handled through validation checks and controlled session termination.

XVI. Extended Performance Evaluation

To further evaluate scalability, encryption and decryption times were measured for increasing file sizes. Results indicate a linear relationship between file size and processing time.

$$T_{enc}(n) = O(n), \quad T_{dec}(n) = O(n)$$

This confirms that the system can scale efficiently for large document repositories.

XVII. Comparative Cryptographic Strength Analysis

Compared to RSA-based systems, the proposed approach eliminates dependency on factorization or discrete logarithm assumptions. Table IV highlights the comparison.

TABLE IV
Cryptographic Strength Comparison

Scheme	Security Basis	Quantum Safe
RSA-2048	Integer Factorization	No
ECC	Discrete Logarithm	No
AES-256	Symmetric Complexity	Partially
QKD + AES	Physical + Computational	Yes

XVIII. Practical Deployment Case Study

A simulated deployment was conducted for secure academic document handling. Examination papers were encrypted using QKD-generated keys and stored securely. Access was restricted to authorized personnel only.

The system successfully prevented unauthorized access while maintaining usability and low latency.

XIX. Discussion on Quantum Readiness

The proposed system prepares existing infrastructure for a post-quantum era. By decoupling key distribution from computational assumptions, the system ensures long-term confidentiality even with the advent of large-scale quantum computers.

XX. Ethical Implications and Responsible Use

While strong encryption enhances privacy, it must be deployed responsibly. Access control, audit logging, and compliance with legal frameworks are essential to prevent misuse.

XXI. Formal Algorithmic Representation

This section presents a structured algorithmic description of the proposed secure document encryption and retrieval system.

A. Algorithm 1: Secure Document Upload

1) User selects a document d_i via frontend interface.
2) Backend initializes BB84 quantum key generation.
3) Alice generates random bit sequence and bases.
4) Bob measures received qubits using random bases.
5) Perform basis reconciliation and key sifting.
6) Compute Quantum Bit Error Rate (QBER).
7) If QBER $>$ threshold, abort session.
8) Else, derive quantum key K_q.
9) Encrypt document using AES-256: $C_i = E(d_i, K_q)$.
10) Store C_i and metadata in secure storage.

B. Algorithm 2: Secure Document Retrieval

1) User requests document d_i.
2) Backend retrieves encrypted file C_i.
3) Fetch associated quantum key K_q.
4) Decrypt document: $d_i = D(C_i, K_q)$.
5) Transmit decrypted document securely.

XXII. Communication Protocol and Data Flow Analysis

The system employs a hybrid communication protocol combining classical HTTP-based REST APIs with quantum-inspired key distribution mechanisms.

A. Frontend to Backend Communication

The frontend communicates with the backend using RESTful APIs over HTTP. File upload requests encapsulate document data, while responses return status codes and metadata.

B. Quantum Key Exchange Phase

Although simulated, the QKD phase follows the same logical separation as real-world quantum channels:

- Quantum Channel (simulated): Used for qubit transmission.
- Classical Channel: Used for basis comparison and error correction.

C. Backend to Storage Communication

Encrypted files are written to persistent storage, while keys and metadata are stored in the database layer. This separation ensures minimal exposure of sensitive information.

XXIII. Key Management and Lifecycle Model

Effective key management is critical for cryptographic security. The lifecycle of quantum keys in the proposed system consists of the following stages:

1) **Generation**: Keys are generated per document upload session.
2) **Validation**: Keys are validated using QBER estimation.
3) **Usage**: Keys are used exactly once for encryption and decryption.
4) **Storage**: Keys are stored securely in encrypted form.
5) **Expiration**: Keys may be invalidated after predefined usage or time limits.

This lifecycle ensures forward secrecy and limits key reuse vulnerabilities.

XXIV. Extended Experimental Methodology

The evaluation of the proposed system was conducted under controlled experimental conditions.

A. Experimental Setup

Experiments were performed on a system with the following configuration:

- Processor: Intel Core i7
- RAM: 16 GB
- Operating System: Windows 11
- Python Version: 3.11

B. Dataset Description

A diverse dataset consisting of text files, PDF documents, and image files of varying sizes was used to evaluate system performance.

C. Evaluation Metrics

The following metrics were used:

- Encryption Time
- Decryption Time
- Key Generation Time
- Storage Overhead

XXV. Statistical Performance Analysis

To ensure consistency, each experiment was repeated multiple times and average execution times were recorded.

Let T_{enc} and T_{dec} represent encryption and decryption times respectively.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} T_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(T_i - \mu)^2}$$

The low standard deviation observed across experiments indicates stable and predictable system performance.

XXVI. Storage Overhead and Optimization Analysis

The encrypted files incur minimal storage overhead compared to original files due to symmetric encryption. Metadata storage overhead is negligible relative to file sizes.

Future optimizations include compression-before-encryption and deduplication techniques for large-scale deployments.

XXVII. Long-Term Vision and Research Directions

In the long term, the proposed architecture can serve as a foundation for fully quantum-secure cloud services. Integration with quantum networks, satellite-based QKD, and post-quantum authentication schemes represents promising research directions.

XXVIII. Limitations

The system relies on simulated QKD rather than physical quantum channels. Additionally, AES-ECB mode can be replaced with authenticated encryption schemes for improved security.

XXIX. Applications

- Secure examination paper transmission
- Medical record storage
- Legal and government document security
- Research data protection

XXX. Future Scope

Future enhancements include real QKD hardware integration, AES-GCM encryption, blockchain-based audit logs, and multi-user authentication.

XXXI. Conclusion

This paper presented a QKD-based secure document encryption and retrieval system that integrates quantum-inspired key distribution with classical cryptography. The proposed approach demonstrates enhanced security, scalability, and feasibility for real-world deployment.

References

[1] Quantum Cryptography and Key Distribution for Secure Communication in the Post Quantum World, published 2 Jan 2026, DOI:10.1109/ICSCN67106.2025.11308316. Describes hybrid integration of QKD and post-quantum cryptography for future secure systems.

[2] Secure method of communication using Quantum Key Distribution, published 3 Jul 2025. This work develops secure encrypted communications using the BB84 protocol and web applications.

[3] Quantum cryptography for secure cloud data storage and transmission, published 15 Oct 2025. Proposes a hybrid encryption system integrating QKD with AES-256 for secure cloud environments.

[4] Hybrid QKD-based framework for secure enterprise communication, 2024. Discusses integration of QKD with classical/post-quantum cryptography for enterprise security.

[5] Advancements in secure quantum communication and cryptographic protocols, 2025. Reviews BB84, E91, and other QKD protocols with security and error correction techniques.

[6] Secure and scalable file encryption for cloud systems via PQC and QKD integrating AES-CTR, 2025. Proposes distributed file encryption combining QKD and post-quantum key encapsulation.

[7] Eliminating single points of trust: a hybrid quantum and post-quantum blockchain approach, 2025. Combines QKD with blockchain to remove centralized trust issues.

[8] Quantum key distribution through quantum machine learning, 2025. Reviews QKD security, eavesdropping detection, and future QKD research directions with quantum ML.

[9] A hybrid multi-node QKD-ECC architecture for securing IoT, 2025. Studies security for IoT using QKD and elliptic curve cryptography for resilient multi-node networks.

[10] H. E. Mozo, "Quantum-Classical Hybrid Encryption Framework Based on Simulated BB84 and AES-256: Design and Experimental Evaluation," arXiv:2511.02836, 2025. Experimental evaluation of simulated BB84 and AES-256 hybrid encryption.

[11] A. Raj and V. Balachandran, "A Hybrid Encryption Framework Combining Classical, Post-Quantum, and QKD Methods," arXiv:2509.10551, 2025. Framework combining classical, PQC, and QKD for secure communication.