681 Report Draft

Abstract

Quantum computing presents both unprecedented computational opportunities and significant threats to existing cryptographic paradigms. This paper introduces a hybrid secure communication system that integrates Quantum Key Distribution (QKD), specifically the BB84 protocol, with classical encryption methods including the One-Time Pad (OTP) and the Advanced Encryption Standard (AES). The proposed system leverages the quantum-mechanical principles underlying BB84 to establish unconditionally secure keys, which are subsequently used for encrypting text and image data. A complete web-based application was developed using Qiskit and Streamlit, allowing real-time secure communication over potentially compromised networks. The architecture was tested under normal and adversarial conditions, including simulated eavesdropping scenarios, to evaluate its resilience. Key metrics such as quantum bit error rate (QBER), encryption latency, and throughput were analyzed to assess the performance and practicality of the system. The results validate the feasibility of deploying quantum-enhanced cryptographic protocols for next-generation secure communication infrastructures.

1 Introduction

In today's increasingly digital world, securing communication channels has become a pressing priority. As sensitive data continues to traverse the internet and other digital mediums, the risk of interception, manipulation, or theft by malicious actors has grown significantly. Conventional cryptographic techniques such as RSA and Elliptic Curve Cryptography (ECC) rely heavily on the computational intractability of mathematical problems. However, with the rapid advances in quantum computing, these foundational assumptions are under threat. Quantum algorithms such as Shor's algorithm for factoring and Grover's algorithm for database searching have demonstrated the theoretical ability to compromise classical encryption schemes, potentially rendering them obsolete in the near future.

This paper presents a comprehensive approach to addressing this looming threat by designing and implementing a hybrid secure communication architecture that leverages the strengths of both quantum and classical cryptographic systems. The core of our system relies on Quantum Key

Distribution (QKD), specifically the BB84 protocol, to establish shared secret keys between two communicating parties. Unlike classical key exchange methods, BB84 utilizes quantum mechanical properties, such as the no-cloning theorem and quantum superposition, to enable inherently secure key distribution. The protocol ensures that any eavesdropping attempt by an adversary, known as Eve, introduces detectable anomalies in the quantum states being exchanged, thereby providing built-in intrusion detection.

To complement the key exchange mechanism, we incorporate two well-known classical encryption techniques: the One-Time Pad (OTP) and the Advanced Encryption Standard (AES). The OTP, when used correctly with truly random keys that are never reused, provides perfect secrecy but is often impractical due to key management constraints. AES, on the other hand, offers high performance and strong computational security and is widely adopted across numerous platforms. By feeding keys generated through QKD into these classical encryption schemes, we construct a hybrid model that enjoys both the theoretical security guarantees of quantum cryptography and the practical efficiency of classical cryptographic systems.

The technologies adopted in this study include the Qiskit framework for simulating quantum circuits, particularly the BB84 protocol, and Streamlit for developing an interactive web-based interface. The system allows users to input messages or upload images, which are then encrypted using the QKD-generated keys. A secure login and registration mechanism is implemented to control access, and encrypted keys are optionally shared via QR codes to enhance usability while maintaining security.

We rigorously evaluate our architecture by simulating various operational scenarios, including normal key exchange and adversarial eavesdropping, to test the robustness of our security mechanisms. Key metrics such as quantum bit error rate (QBER), key generation throughput, and encryption/decryption latency are measured and analyzed. The experiments demonstrate that the system can reliably detect intrusion attempts and maintain secure communication under normal and adversarial conditions.

This paper is organized as follows: Section II introduces the foundational concepts in quantum cryptography and classical encryption, providing a brief overview of related literature. Section III describes the design and architecture of the hybrid cryptographic system. Section IV details the key algorithms, including BB84 and the encryption mechanisms. Section V presents the implementation strategy and toolchain. Section VI elaborates on the experimental setup and outlines the testing methodology. Section VII discusses the results, analyzing the system's performance and security characteristics. Finally, Section VIII concludes the paper by summarizing key insights and proposing avenues for future enhancement of quantum-secure communication systems.

2 Background and Motivation

2.1 Prior Work and State of the Art

The past five years have witnessed a surge in interest in quantum-secure communication systems, particularly those combining Quantum Key Distribution (QKD) with classical cryptographic protocols. These studies span theoretical advancements, practical QKD demonstrations, hybrid system design, and cloud-based security integrations. Below, we review a curated selection of recent and influential works that form the foundation for our research.

Xu et al. (2022) demonstrated one of the most extensive composable security validations of QKD by implementing BB84 over a 500-kilometer fiber-optic link [1]. Their study addressed finite-size effects, system imperfections, and statistical parameter estimation. The work validated QKD's feasibility over long distances and proved it could meet the strict composable security requirements necessary for real-world deployment.

Wang et al. (2023) investigated the integration of QKD into 6G communication networks [2]. Their framework addressed bandwidth sharing, key scheduling, and classical-quantum coexistence in high-speed networks. This work is significant for proposing a layered architecture that blends classical and quantum data planes, though it remains focused on telecom-scale applications.

Jiang et al. (2023) introduced a hybrid BB84-secured messaging application hosted in a cloud environment [3]. They implemented key reconciliation and message encryption workflows, demonstrating how QKD could be embedded in software-only infrastructures. However, their system lacked multimedia support and real-time performance evaluation under adversarial scenarios.

Tang et al. (2024) focused on lightweight quantum-enhanced cryptographic protocols designed for resource-constrained IoT environments [4]. Their contribution centered around reducing the overhead of quantum operations while maintaining energy efficiency. This is highly relevant for extending QKD to edge devices, though it does not explore high-bandwidth or user-facing use cases.

Sharma and Das (2025) provided an extensive survey of hybrid encryption frameworks that merge QKD with classical cryptographic primitives [5]. Their work explored system architectures, performance trade-offs, and implementation constraints across cloud, mobile, and decentralized platforms. However, their analysis was architectural and conceptual rather than an implemented system.

Liu et al. (2023) developed a blockchain-integrated QKD framework for secure distributed key refresh [6]. Their approach combines quantum key generation with consensus validation, enhancing auditability and trust in distributed environments. Though the concept is novel, the system does not address user-centric design or front-end accessibility.

Moretti et al. (2022) explored secure video streaming using QKD and biometric authentication [7]. This work emphasized multi-factor security but primarily targeted enterprise-level content protection and not general-purpose or academic communication tools.

Chen et al. (2022) presented a comprehensive roadmap for quantum computing applications including cryptography [**?**]. While not focused solely on QKD, this survey positioned QKD as a near-term achievable technology and emphasized the importance of hybrid quantum-classical models.

Together, these works highlight the growing maturity of QKD and its applications in both high-assurance and consumer-scale systems. However, most efforts are either hardware-intensive, lack real-time interactivity, or do not support multimedia communication in adversarial testbeds. These limitations directly inform the need for our proposed system.

2.2 Limitations and Gaps in Existing Approaches

Despite the increasing body of work advancing Quantum Key Distribution (QKD) and hybrid cryptographic models, a close review of recent literature reveals multiple technical and architectural limitations. These limitations not only challenge the practical deployment of QKD systems but also highlight the need for more versatile, accessible, and performance-validated implementations—needs our work directly addresses.

Gap 1: Hardware Dependency and Limited Accessibility Xu et al. (2022) [1] delivered an impressive QKD deployment over 500 km of optical fiber. However, the setup requires specialized quantum photonic hardware and significant infrastructural investments. The system, while secure and scalable, is inaccessible to smaller institutions, academic testbeds, or software-only applications. This barrier creates a gap for simulated or emulated QKD platforms that can still demonstrate security behaviors under controlled conditions.

Gap 2: Telecom and Infrastructure Bias Wang et al. (2023) [2] provided a layered architecture for embedding QKD in 6G communication stacks. Their focus, however, remains bounded to carrier-grade infrastructure. The design does not explore lightweight or modular approaches that integrate QKD into cloud-native applications or everyday messaging platforms. This introduces a gap for systems aimed at micro-scale, user-facing environments.

Gap 3: Lack of Real-Time Adversarial Testing Jiang et al. (2023) [3] introduced a cloud-based BB84 messaging model, which is close in intent to our system. However, the implementation lacks real-time testing under adversarial simulation—such as quantum bit error rate (QBER) shifts caused by eavesdropping (Eve) attacks. The system is also limited to text and does not address image-based or binary data encryption. This opens a critical gap for real-time, adversarial-aware hybrid communication systems with multimedia support.

Gap 4: Resource-Constrained Performance Focus Only Tang et al. (2024) [4] optimized QKD for IoT settings with energy-efficient protocols. While their focus on reducing quantum operation cost is valuable, the system assumes constrained device behavior and does not target high-fidelity simulation or user application integration. The study leaves unaddressed how QKD-based systems could scale upward into richer user experiences or integrate with full-stack applications.

Gap 5: Conceptual Frameworks Without Implementation Sharma and Das (2025) [5] presented a thorough classification of QKD-classical hybrid designs. However, their work remains theoretical—surveying architecture types, trade-offs, and theoretical vulnerabilities without implementing any reference system. There is thus a gap in executable prototypes that researchers can interact with, evaluate, and extend.

Gap 6: Neglect of End-User Accessibility Liu et al. (2023) [6] proposed a blockchain-integrated QKD system but placed emphasis on backend integrity and not user-level interface design. Their implementation is powerful in trustless environments but lacks usability focus—such as front-end interaction, session visualization, and modular configuration. This leaves a space for user-centered platforms that visualize secure communication in intuitive ways.

Gap 7: Narrow Use Case Specialization Moretti et al. (2022) [7] tackled video streaming security using QKD and biometric validation. However, their model is specialized for media content distribution, not general-purpose secure communication. They do not explore modular encryption strategies that could apply to multiple message types (text, image, file), nor adversarial feedback during execution. Our system extends QKD's relevance to broader secure messaging contexts.

Our Identified Opportunity: From these limitations, it becomes clear that while QKD is progressing toward maturity, a substantial gap remains for modular, software-accessible, adversarially tested, and end-user-centric QKD-classical hybrid platforms. Our system is specifically designed to fill this gap. It:

- Uses Qiskit-based BB84 simulation to replace dependency on photonic hardware.
- Integrates classical AES and OTP for flexibility in encryption strength and performance.
- Builds an intuitive, interactive web interface using Streamlit.
- Supports text and image encryption workflows.
- Actively models Eve-based eavesdropping and visualizes QBER responses.
- Benchmarks performance under variable key lengths and message sizes.

This architecture is not only a prototype but a blueprint for designing research-grade post-quantum secure communication systems that prioritize accessibility, extensibility, and practical relevance.

2.3 Problem Definition

The continued evolution of quantum computing presents a direct and growing threat to the foundational assumptions of modern cryptographic systems. Classical algorithms such as RSA and ECC rely on the computational hardness of problems like integer factorization and discrete logarithms—problems that quantum algorithms such as Shor's can solve in polynomial time. Symmetric algorithms like AES, while more resilient, also face degradation in effective security strength due to Grover's algorithm. These quantum threats make it clear that classical cryptography alone cannot guarantee long-term data confidentiality and integrity.

Quantum Key Distribution (QKD) offers a compelling alternative, promising information-theoretic security based on the laws of quantum physics rather than mathematical complexity. However, despite its strong theoretical foundations, QKD has not been widely adopted in practical systems due to several persistent challenges:

- **Hardware dependence:** Most QKD implementations rely on specialized photonic hardware and optical fibers, making them inaccessible for software-based environments, educational use, or early-stage research deployments.
- **Lack of usable interfaces:** Existing systems are often built with proof-of-concept intentions, offering minimal integration with user-facing applications or intuitive cryptographic workflows.
- **Absence of adversarial modeling:** Few practical implementations simulate or measure the system's response to threats such as eavesdropping or key tampering. Without this, it is difficult to validate the defensive properties of QKD under attack scenarios.
- **Limited encryption scope:** Many existing QKD-enabled systems are restricted to basic message encryption and fail to extend their capabilities to other data types such as images or structured file content.
- **Poor visibility into system performance:** There is little work on monitoring or benchmarking hybrid quantum-classical encryption systems under varied operational conditions such as key length, message size, or error rate impact.

These limitations reflect a disconnect between the theoretical power of QKD and its practical utility in general-purpose, scalable cryptographic frameworks. Addressing this disconnect—by building a quantum-secure yet software-driven and user-centric encryption system—is the central problem that this research aims to tackle.

2.4 Research Objectives

The overarching aim of this research is to design, implement, and evaluate a hybrid cryptographic architecture that merges quantum-resilient key distribution with practical encryption and user-centric communication workflows. Unlike previous theoretical frameworks or hardware-bound solutions, this system is intended to be modular, accessible, and demonstrably secure under adversarial and performance-constrained scenarios.

To this end, the project pursues the following key objectives:

1. **Simulate BB84 Quantum Key Distribution in a software environment:** Model the BB84 protocol using a quantum computing framework to generate cryptographically strong symmetric keys, without relying on specialized quantum hardware.

2. **Integrate quantum keys into classical encryption:** Utilize the generated keys within One-Time Pad and AES encryption schemes to support both theoretical and practical security models.

3. **Enable multi-format secure messaging:** Extend the system beyond textual data to include image-based encryption, demonstrating the versatility and generalizability of QKD-powered security.

4. **Simulate adversarial scenarios and validate system resilience:** Implement mechanisms to detect quantum bit error rate (QBER) anomalies caused by eavesdropping simulations and verify that the system responds appropriately to compromise attempts.

5. **Develop an intuitive user interface for secure communication:** Deliver a user-accessible platform that abstracts the complexity of quantum cryptography while preserving visibility into key exchange, encryption, and transmission steps.

6. **Benchmark performance and analyze system behavior:** Collect and report empirical data on key generation rates, encryption/decryption latencies, and error rates under varying loads and message conditions.

7. **Provide a reusable testbed for research and education:** Package the system as a modular prototype for further academic development, cryptographic experimentation, and classroom demonstration of post-quantum security principles.

Collectively, these objectives aim to transform QKD from a theoretical construct into an accessible, testable, and secure building block for future communication architectures.

2.5 Proposed Technical Approach and Architecture

In response to the identified challenges—namely, the inaccessibility, rigidity, and lack of adversarial modeling in current quantum-secure systems—we propose a hybrid cryptographic framework that integrates simulated Quantum Key Distribution (QKD) with classical encryption protocols into a modular, software-driven architecture. This system is designed to be practically deployable, user-centric, and extensible for research and educational use cases.

At its core, our approach involves simulating the BB84 protocol using the Qiskit quantum computing framework. This simulation reproduces the quantum behavior of key generation (e.g., randomness, measurement uncertainty, and basis mismatch) while avoiding the infrastructural demands of physical quantum hardware. The generated key material undergoes key sifting and quantum bit error rate (QBER) validation, enabling detection of adversarial interference through behavioral analysis—a critical component in assessing system robustness.

The validated quantum key is then utilized within a classical encryption module that supports both the One-Time Pad (OTP) and Advanced Encryption Standard (AES). OTP offers perfect secrecy when applied correctly and serves as a theoretical benchmark, while AES provides a high-performance option aligned with real-world cryptographic standards.

To enhance accessibility and interactivity, we incorporate a dynamic web-based interface built using modern Python frameworks. The system supports text and image-based secure communication, offering real-time feedback, encryption visualization, and simulation of eavesdropping scenarios through adaptive QBER injection.

This architecture is designed to enhance both the conceptual and operational aspects of quantum-enhanced secure systems in the following ways:

- **Accessibility:** Enables QKD experimentation and application without requiring quantum transmission hardware.

- **Adversarial modeling:** Supports the simulation of interception attempts (Eve) and quantifies their effect on QBER in real-time.

- **Versatility:** Provides multi-format encryption (text and images) and allows key switching between OTP and AES for comparative evaluation.

- **Performance visibility:** Benchmarks key generation time, encryption latency, and system responsiveness under different message and attack scenarios.

The decision to adopt this architecture was informed by a comprehensive analysis of recent works in quantum-secure communication and system architecture. Xu et al. (2022) demonstrated BB84-based QKD over 500 km optical fiber with composable security, but their reliance on photonic infrastructure limits its replicability in low-resource or simulated environments [1]. Jiang et al. (2023) proposed a cloud-deployable QKD messaging prototype, yet lacked detailed support for adversarial testing or diverse data formats such as images [3]. Tang et al. (2024) developed a lightweight quantum-enhanced system for IoT, focusing on power efficiency but with minimal attention to architectural modularity and data type flexibility [4]. Sharma and Das (2025) surveyed hybrid architectures combining QKD with classical systems, emphasizing integration complexity but without demonstrating practical frameworks [5].

Complementing this, Jain et al. (2021) examined continuous-variable QKD under realistic network settings and called for system designs that support composability and scalability in real-time applications [**?**]. In the domain of quantum-computing-compatible hardware design, Stanco et al. (2021) proposed FPGA-based architectures for QKD-compatible systems, focusing on hardware acceleration but lacking a full-stack, software-accessible prototype for experimentation or demonstration [**?**]. Wang et al. (2023) proposed an end-to-end QKD integration in 6G security layers, highlighting architectural transformation at network levels rather than at the software service layer, where general-purpose applications must evolve [2]. Liu et al. (2023) integrated blockchain with QKD key refresh protocols for distributed trust enforcement, but their approach emphasizes cryptoeconomic integrity rather than user experience or education-focused tooling [6].

Collectively, these studies highlight that while the cryptographic and architectural principles of QKD have matured significantly, there is still a notable absence of systems that integrate QKD into flexible, user-facing applications with built-in adversarial testing, multimedia compatibility, and detailed system benchmarking. Our proposed architecture is specifically designed to meet these gaps while offering a clear path for practical adoption and iterative improvement.

2.6 Strategic Scope and Real-World Significance

The strategic significance of this research lies in its alignment with the global imperative to transition toward cryptographic systems resilient to quantum-era threats. As nations and institutions move to adopt post-quantum standards, there is a parallel demand for architectures that not only provide quantum-resilient security but also maintain the usability, flexibility, and scalability expected from modern secure communication platforms.

This work is positioned at that critical junction. It bridges the gap between high-assurance quantum key generation and real-world application-level encryption by delivering a fully software-defined, hybrid quantum-classical system that can operate on conventional computing infrastructure. The system's scope extends across multiple domains, each of which is influenced by its technical and strategic contributions:

- **Cryptographic Innovation:** By combining BB84-based quantum key generation with classical encryption protocols in a runtime-selectable manner, our architecture enables comparative analysis between information-theoretic (OTP) and computational (AES) security.

- **Security Education and Research:** The system serves as a valuable educational tool that reveals the end-to-end cryptographic lifecycle—from quantum key generation to encrypted message exchange and eavesdropper detection—making it ideal for university courses and lab environments.

- **Software-Based Accessibility:** Unlike photonic QKD systems, our solution requires no quantum hardware, making it immediately deployable for simulation, development, and training. This enhances accessibility for institutions without access to dedicated quantum infrastructure.

- **Versatile Application Domains:** The inclusion of both text and image encryption modules allows the system to simulate secure messaging, media sharing, and content delivery use cases, reflecting needs across secure email, cloud communication, telemedicine, and financial data transmission.

- **Modular Benchmarking and Extension:** The performance evaluation features integrated into the platform (e.g., QBER monitoring, latency tracking, encryption throughput) support reproducible experimentation and system tuning, enabling the architecture to evolve alongside advances in quantum or classical cryptography.

In practice, this system contributes a lightweight yet technically rich reference implementation that addresses the shortfall in educational, simulation-ready QKD platforms. It is strategically designed to complement emerging quantum infrastructures by offering a robust, interactive, and secure layer adaptable to diverse communication needs and evolving regulatory frameworks.

3 System Design and Architecture

The system is designed as a modular, layered cryptographic pipeline that integrates quantum key generation with classical encryption mechanisms in a full-stack implementation. The architecture emphasizes modularity, extensibility, and interaction visibility. It consists of clearly defined components responsible for quantum simulation, cryptographic operations, adversarial modeling, and user interaction. The system is broken down into the following key architectural subsystems:

3.1 Quantum Key Distribution Layer

The Quantum Key Distribution (QKD) layer is the foundational component of our architecture, responsible for generating symmetric encryption keys based on quantum mechanical principles. Unlike traditional key exchange methods such as Diffie–Hellman or RSA, which rely on computational assumptions, QKD enables information-theoretically secure key generation by exploiting the physical properties of quantum states. This makes the QKD layer a non-substitutable asset in any architecture targeting post-quantum resilience.

In this system, we adopt the BB84 protocol — the first and most widely implemented QKD scheme — as the basis for simulating quantum key exchange. The BB84 protocol operates on the principle of encoding qubits in non-orthogonal bases, such that any measurement attempt by an eavesdropper (Eve) introduces detectable disturbances. Our system simulates BB84 using IBM's Qiskit framework, which allows for emulation of quantum state preparation, measurement, and random basis generation without physical quantum hardware.

The process begins with Alice generating a random bit string and a corresponding random basis string (e.g., rectilinear or diagonal). Each bit is then encoded into a qubit using the selected basis and "sent" to Bob over a simulated quantum channel. Bob, independently and without knowledge of Alice's bases, chooses his own random basis string and measures the incoming qubits accordingly. This results in a raw key on Bob's side, which only partially overlaps with Alice's original key.

To ensure a shared secret, a key sifting step follows in which Alice and Bob publicly compare their basis strings. The bits corresponding to matching bases are retained, while the others are discarded. This sifting process is a critical feature of BB84, as it isolates the bits most likely to be measured accurately.

The QKD layer in our system includes simulation of this entire pipeline:

- Generation of random bit and basis sequences for Alice and Bob.
- State encoding and quantum measurement emulated via Qiskit.
- Public basis comparison and retention of matching index bits.

A key innovation in our implementation is the built-in capability to simulate adversarial behavior by introducing Eve's random basis measurements mid-transmission. This models real-world attacks on the quantum channel and enables us to study the effects of quantum interference on the resulting shared key. Specifically, we compute the Quantum Bit Error Rate (QBER) — the ratio of mismatched bits between Alice's and Bob's keys after sifting — as a metric to detect intrusion.

From a research perspective, this simulation offers multiple benefits:

1. It provides a replicable and observable model of BB84 behavior, supporting both theoretical exploration and algorithmic experimentation.
2. It allows us to test system robustness against varied levels of noise and attack interference without access to physical quantum networks.
3. It establishes a software-controlled gateway for generating secure keys, which can be modularly plugged into classical cryptographic systems downstream.

The QKD layer thus not only initiates the secure communication pipeline but also provides security diagnostics through error analysis. It is designed to be extensible — capable of supporting future variations such as decoy state BB84, E91 entanglement-based protocols, or continuous-variable QKD schemes. As the quantum layer underpins the trust model of the entire system, its correctness and observability are paramount to validating the architecture's post-quantum credentials.

3.2 Key Reconciliation and QBER Analysis

Following the initial BB84 quantum key exchange simulated in Section 3.1, the Key Reconciliation and QBER (Quantum Bit Error Rate) Analysis layer ensures the integrity and confidentiality of the derived symmetric key before its use in encryption. This layer plays a pivotal role in identifying potential eavesdropping events and ensuring that the final key shared between the sender (Alice) and the receiver (Bob) is both synchronized and secure.

Key reconciliation begins with a public exchange of the basis sequences used during the quantum transmission phase. Alice and Bob compare their basis choices without disclosing the actual bit values. Bits corresponding to mismatched bases are discarded from both parties' sequences, and the remaining bits form what is known as the ”sifted key.” This process is crucial for reducing the key to a subset where both parties likely measured the same quantum states.

The next critical step involves estimating the QBER, defined as the ratio of differing bits between Alice's and Bob's sifted keys. This error rate serves as an indicator of noise in the channel or, more importantly, the presence of an adversary such as Eve. In real-world QKD systems, a threshold QBER (typically between 11

In our simulated environment, we model two key scenarios:

- **Normal operation:** No eavesdropper is present, and minor QBER may result from random basis mismatch or simulated noise.
- **Eavesdropping simulation:** An adversary (Eve) intercepts and measures the qubits using a random basis, thereby introducing detectable errors into Bob's measurements.

To implement these processes, our system includes:

- A bitwise comparison engine that evaluates the alignment of Alice's and Bob's sifted keys.
- A QBER calculator that outputs both absolute error count and percentage.
- A reconciliation protocol that either approves the key for encryption or flags it for regeneration.

This layer is central not only to ensuring the secrecy of the communication but also to the dynamic response of the system to potential threats. It reflects the adaptive security model envisioned in modern QKD implementations, where keys are validated, monitored, and discarded in real time based on error dynamics.

Moreover, QBER analytics are visualized in the user interface to make security assessment transparent and interactive. This allows researchers, developers, or students using the platform to observe how attacks degrade key integrity and how the system self-regulates to maintain cryptographic soundness.

In summary, the reconciliation and QBER analysis layer acts as the gatekeeper between the quantum and classical components of our architecture. It validates the security assumptions made during key generation and ensures only trustworthy keys advance to the encryption stage.

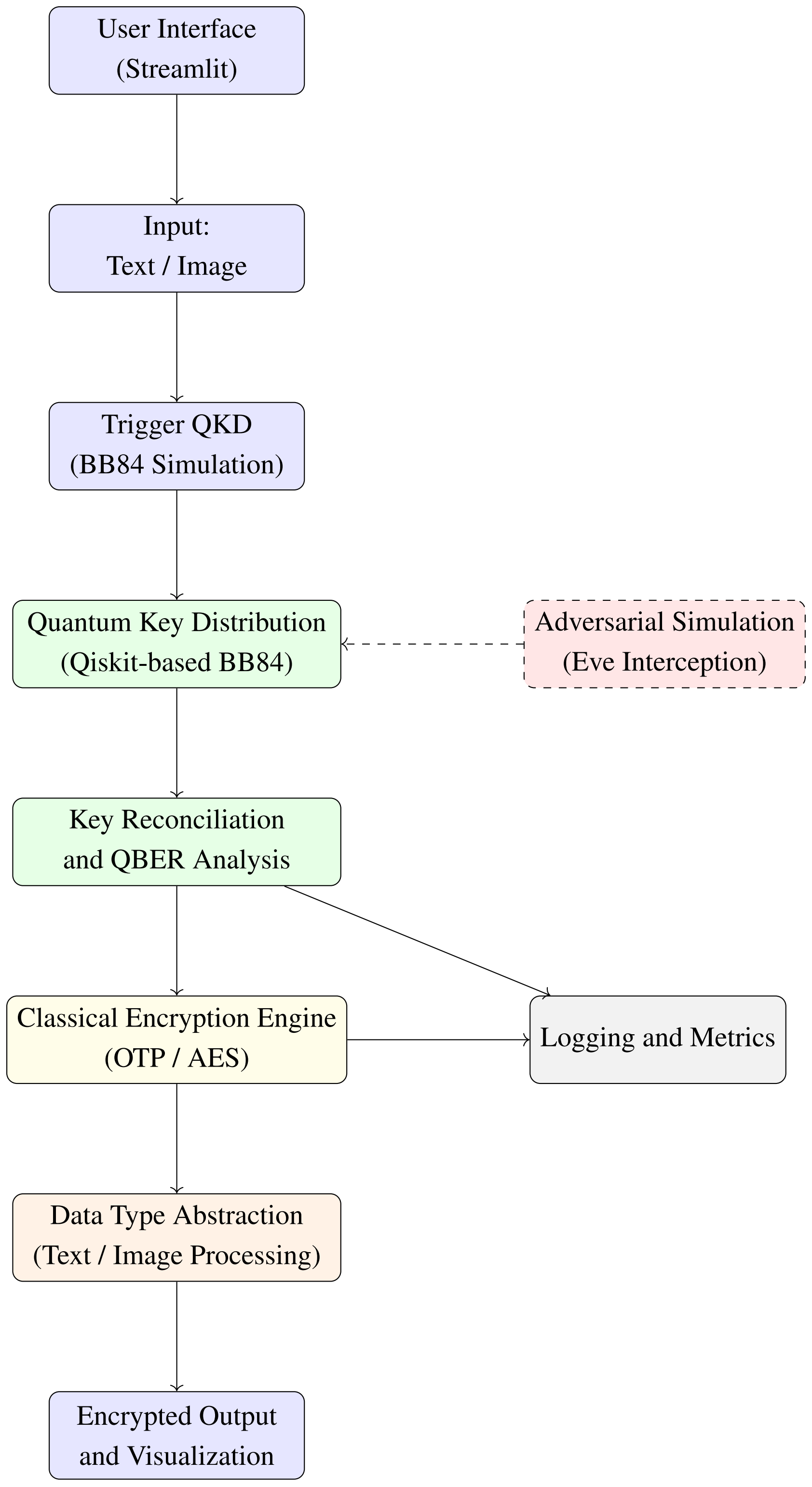

Figure 1: High-Level Architecture of Quantum-Classical Hybrid Cryptographic System

3.3 Classical Encryption Engine

Once the quantum key has been successfully generated and validated through key reconciliation and QBER analysis, it is passed into the Classical Encryption Engine. This layer is responsible for transforming the raw shared secret into practical cryptographic protection for user data. The engine is designed to be flexible, enabling users to apply either theoretical or industry-standard encryption techniques based on the quantum-generated key.

The engine offers two encryption modes:

- **One-Time Pad (OTP):** The OTP mode provides information-theoretic security when used correctly—specifically, when the key length matches the message length and is never reused. In this mode, the system performs a simple XOR operation between the binary message and the quantum key. While impractical for large or streaming data due to the key size requirement, OTP is ideal for validating the strength of the QKD layer and serves as a benchmark for maximal secrecy.

- **Advanced Encryption Standard (AES):** The AES mode supports 128-bit block encryption in Cipher Block Chaining (CBC) mode, providing confidentiality with computational efficiency. The quantum key is either used directly (if it meets size requirements) or extended/truncated appropriately. AES offers compatibility with widely used communication standards and is better suited for longer or multimedia messages. Initialization vectors (IVs) are randomly generated for each session and shared securely alongside the ciphertext.

The role of this layer is not merely to encrypt data but to evaluate how quantum-generated keys integrate with classical cryptographic operations in real-time systems. Key design considerations in this module include:

- **Key Adaptation:** QKD keys may vary in length and entropy. The engine includes preprocessing functions to normalize key length based on selected cipher requirements, ensuring format compliance without compromising randomness.

- **Padding and Block Management:** For AES, input data is padded to match block size constraints. Proper padding schemes are employed to prevent leakage or misalignment during decryption.

- **Session Management:** Each encryption request initializes a new key session, linking QKD output, chosen encryption method, IVs, and output ciphertext into a reproducible encryption trace for auditing and analysis.

From a research perspective, this engine enables experimentation with hybrid cryptographic configurations. It allows side-by-side comparisons of OTP and AES under various conditions (e.g., message size, key entropy, QBER thresholds), offering a unique lens through which to study hybrid security postures. Additionally, by feeding quantum-generated keys into a classical system, this module concretely demonstrates the feasibility of real-world quantum-classical integration.

Importantly, this engine acts as the translation layer between abstract key material and applied security. It reflects a key tenet of our design philosophy: that post-quantum systems must not only be secure in theory but usable and adaptable in practice.

3.4 Data Type Abstraction Layer

Modern secure communication systems must address the reality that data formats are heterogeneous. From short text messages and numeric strings to high-resolution images and media files, encryption systems must handle varying data types while maintaining cryptographic consistency. In response, the Data Type Abstraction Layer in our architecture functions as a generalization interface between user inputs and the Classical Encryption Engine, allowing seamless conversion of data into cryptographically compatible formats.

The primary responsibility of this layer is to preprocess, format, and normalize inputs of different data types so they can be encrypted using either the OTP or AES module. This includes operations such as encoding, resizing, serialization, padding, and integrity preservation. The abstraction not only enhances user experience but also ensures that cryptographic routines operate over stable and validated data structures.

Textual Data Handling: For plain text, the system uses UTF-8 encoding to transform user-entered strings into byte streams. This ensures compatibility across platforms and preserves the integrity of multilingual or symbol-rich content. Padding mechanisms are applied when using AES to align input with block size requirements, and XOR-based masking is used when applying OTP. Upon decryption, the original string is reconstructed from its byte representation.

Image Data Handling: For image inputs, the system utilizes the OpenCV library to read, normalize, and flatten the pixel array into a 1D byte vector. This vector can then be passed to the encryption engine. AES encrypts the byte stream in blocks, preserving image fidelity during reassembly, while OTP masks the entire byte stream bitwise. After decryption, the byte stream is reshaped back into its original image matrix using stored dimension metadata, and the image is displayed or saved accordingly.

Key features of this abstraction layer include:

- **Modular Input Processing:** Each data type is handled by a separate preprocessing pipeline, making it easy to extend the architecture to future data types such as audio, documents (PDF), or video.

- **Format Integrity Preservation:** Metadata, such as image dimensions or file encoding, is securely bundled with the ciphertext to ensure accurate reconstruction on the receiving end.

- **Encryption Agnosticism:** The abstraction layer prepares data for either OTP or AES without being bound to the specifics of the encryption method, improving reusability.

From a research standpoint, this layer offers a platform for studying how data structure and size affect encryption time, key usage efficiency, and error propagation across different cryptographic modes. For example, encrypting large image files under QKD-generated OTP keys highlights key-length limitations, while block-wise AES encryption reveals performance and memory management considerations.

Ultimately, this abstraction layer transforms the theoretical flexibility of our system into practical usability. It ensures that our architecture not only supports post-quantum cryptography but also applies it meaningfully to real-world, diverse data environments.

3.5 Adversarial Simulation Module

In a post-quantum cryptographic system, security cannot be assumed solely from design; it must be evaluated under active threat scenarios. The Adversarial Simulation Module in our architecture addresses this by introducing controllable, synthetic eavesdropping behaviors during the quantum key exchange phase. It serves as a test harness for evaluating the resilience of the system and for validating the detection capabilities of Quantum Bit Error Rate (QBER) analysis mechanisms.

The module simulates a third-party adversary, traditionally known as Eve, who intercepts qubits during transmission from Alice to Bob. Following BB84 principles, any measurement of a qubit in a mismatched basis disturbs its state, which manifests as a detectable error in the shared key. This phenomenon provides the basis for intrusion detection in quantum communication.

Our simulation supports the following adversarial behaviors:

- **Passive Interception:** Eve randomly measures qubits in one of the two BB84 bases without modifying them after measurement. This introduces a statistical disturbance detectable via increased QBER.
- **Active Interference:** Eve not only measures but resends qubits with incorrect states, mimicking more aggressive attacks such as man-in-the-middle scenarios.
- **Selective Attacks:** The system allows configuration of Eve to target only a subset of qubits, enabling fine-grained stress testing of intrusion thresholds.

Key design features include:

- **Configurable Adversary Model:** Users can set the probability of Eve interfering with each qubit, simulate partial attacks, and observe how varying interference levels affect QBER.
- **Real-Time Error Feedback:** The QBER results from Eve's interference are visualized, allowing users to correlate eavesdropping intensity with cryptographic degradation.
- **Attack Logging:** All adversarial interactions are logged, including bit positions modified, bases measured, and QBER impact, supporting traceability and post-run analysis.

From a research perspective, this module provides an essential capability: it enables reproducible experimentation with quantum channel threats in a controlled, simulated environment. Such testing is otherwise infeasible without access to quantum transmission infrastructure and live intercept conditions. Moreover, the ability to induce, detect, and respond to eavesdropping gives our system educational value, especially for courses in quantum information security.

The Adversarial Simulation Module also supports comparative analysis. By toggling between attack modes and adjusting the probability of interference, one can benchmark the robustness of different key sifting and error reconciliation algorithms, and explore design trade-offs in QKD protocol tolerance.

Ultimately, this module ensures that the system does not only simulate secure behavior but also exposes and manages realistic risks. It plays a crucial role in proving the system's resilience under the core adversarial model of quantum communication.

3.6 User Interface and Visualization Layer

The User Interface (UI) and Visualization Layer in our system is designed to bridge the gap between complex cryptographic operations and user comprehension. In research and educational contexts, especially where post-quantum cryptography is involved, it is vital that users not only interact with the system but also gain insight into how the underlying processes unfold. This layer transforms abstract security concepts—like quantum key generation, eavesdropper detection, and data encryption—into interactive, visible, and comprehensible experiences.

Implemented using the Streamlit framework, the UI layer offers a responsive and dynamic frontend that enables real-time user control and monitoring. Users are guided through every stage of the secure communication pipeline: from initiating QKD, to selecting encryption modes, to transmitting encrypted messages or files. The interface abstracts away technical syntax and presents actions through clear, modular controls such as dropdowns, sliders, buttons, and file uploaders.

Key components and design features include:

- **QKD Workflow Control:** Users initiate the BB84 simulation through a one-click interface, triggering quantum key generation and basis sifting. Status updates are displayed to reflect qubit preparation, transmission, and basis matching.

- **Encryption and Decryption Panels:** Text and image inputs are accepted via upload or form fields, followed by encryption using the user-selected cipher (OTP or AES). The decrypted output is displayed upon successful key match and system validation.

- **QBER Visualization Dashboard:** A real-time graph displays the Quantum Bit Error Rate (QBER) for each key generation session, enabling users to detect anomalies (e.g., when simulated Eve interference is active). This dashboard reinforces understanding of quantum noise and intrusion effects.

- **System Logging and Status Feedback:** Informational alerts guide the user through each phase of the process (e.g., key generated, encryption success, QBER acceptable, attack detected). Errors and warnings are flagged when thresholds are violated.
- **Attack Configuration Controls:** For adversarial simulation, the interface includes sliders and toggles to control Eve's behavior (e.g., probability of interception, basis randomness), making the testing process intuitive and iterative.

This layer is central to the educational value of our system. In traditional QKD setups, users often lack transparency into key generation mechanics or attack outcomes. By integrating real-time visualization, our system enables users to observe how quantum phenomena such as basis mismatch or eavesdropping directly affect cryptographic outcomes.

Furthermore, this interface ensures accessibility: no specialized knowledge of quantum circuits, cryptographic primitives, or low-level programming is required. This lowers the barrier to entry for students, educators, and interdisciplinary researchers interested in exploring post-quantum security concepts.

The UI and Visualization Layer also serves as a platform for future extension. Potential enhancements include session history, downloadable logs, multi-user testing modes, and plug-in compatibility with external encryption APIs. Thus, it not only supports interaction but provides the interpretability and flexibility needed for applied cryptographic experimentation.

3.7 Logging, Metrics, and Extensibility

This layer ensures traceability, reproducibility, and scalability of the system by systematically recording operations and exposing performance metrics. It supports both research analysis and practical debugging, enabling deeper insights into how each subsystem behaves under varying conditions.

Logging: The system records key generation events, encryption and decryption transactions, QBER evaluations, and simulated attack parameters. Logs include timestamps, input-output data hashes, key validity flags, and cipher settings, which are stored in structured formats for offline analysis or reporting.

Metrics: Quantitative feedback is collected on:

- Key generation time (per QKD session)
- Encryption and decryption latency (text/image-based)
- QBER trends under normal and adversarial conditions
- Attack frequency vs. detection accuracy

These metrics help evaluate system performance, compare cipher configurations, and validate the impact of QKD behavior on classical encryption throughput.

Extensibility: The architecture is modular by design. Each functional layer (QKD simulation, encryption engine, UI, logging) operates independently with defined interfaces. This enables future upgrades such as:

- Integration of alternative QKD protocols (e.g., E91, decoy-state BB84)
- Additional encryption schemes (e.g., lattice-based cryptography)
- Support for new data types (e.g., PDFs, audio)
- REST API endpoints for system integration

This layer guarantees that the system is not only effective today but adaptable to evolving research questions, cryptographic innovations, and application needs.

Together, these components form a cohesive architecture that delivers quantum-enhanced secure communication with practical usability and robust performance analytics.

4.1 BB84 Key Generation Algorithm

The BB84 protocol underpins the quantum key distribution mechanism in our system. It allows two parties (Alice and Bob) to establish a shared secret key over a quantum channel, while detecting the presence of any third-party eavesdropper (Eve). Below, we provide a stepwise algorithmic breakdown of the protocol as implemented in our system:

Step 1: Random Bit and Basis Generation (Alice)

- Generate a binary sequence of n random bits: bit_A.
- Generate a random basis sequence of the same length, choosing each from Z, X bases: $basis_A$.

Step 2: Quantum State Encoding and Transmission

- Encode each bit from bit_A into a qubit using its corresponding basis from $basis_A$.
- Send the qubits to Bob over a simulated quantum channel.

Step 3: Random Measurement Basis Generation (Bob)

- Bob independently generates a random basis string $basis_B$.
- For each received qubit, he measures it in the basis specified by $basis_B$.
- The result is Bob's raw bit string bit_B.

Step 4: Basis Comparison and Key Sifting

- Alice and Bob publicly compare their bases ($basis_A$ vs. $basis_B$).
- Retain only those bit positions where the bases matched.
- The resulting bits form the *sifted key* for both parties.

Step 5: QBER Estimation and Key Validation

- Compare a random sample of bits from the sifted keys.
- Compute Quantum Bit Error Rate (QBER): $QBER = \frac{Number of mismatches}{Total compared bits}$

- If QBER $\leq$ $threshold (e.g., 11$

Summary: This protocol guarantees that any eavesdropping attempt by Eve introduces detectable errors into the sifted key, thus enforcing a security condition inherent to quantum mechanics. In our implementation, the algorithm is encapsulated within Qiskit and integrated with adversarial simulation parameters to support dynamic threat modeling.

Subsequent sections detail how the validated key feeds into classical cryptographic operations and how performance and security metrics are derived.

4.2 QBER Calculation and Key Validation Logic

Following the key sifting phase of the BB84 protocol, it is essential to verify the integrity of the resulting key and detect any evidence of eavesdropping. This is achieved by calculating the Quantum Bit Error Rate (QBER), a statistical metric that reflects discrepancies between Alice's and Bob's sifted keys. The QBER serves as both a performance indicator and a security checkpoint.

Definition of QBER:
QBER is defined as the ratio of mismatched bits between Alice's and Bob's keys over a publicly agreed-upon subset:

Stepwise Logic for QBER Evaluation:

1. Alice and Bob randomly select a sample subset (e.g., 10–20
2. They exchange the bit values of these positions over a public authenticated channel.
3. For each bit in the sample, a comparison is made. Each mismatch is counted as an error.
4. The final QBER is computed by dividing the number of mismatches by the total number of sampled bits.

Threshold Decision: Based on quantum cryptographic standards, a QBER threshold of approximately 11–15

Algorithmic Implementation Highlights:

- The sample size is parameterized, allowing the system to be stress-tested under different statistical sampling models.
- Eve's interference (from Section 3.5) is reflected in elevated QBER values, verifying the responsiveness of the error detection mechanism.
- All QBER-related metrics are visualized in real time and logged to support repeatability in experimentation.

Impact on System Security: The QBER mechanism is the principal defense against undetected quantum channel tampering. Unlike classical error detection, which may be circumvented by advanced adversaries, QBER reflects intrinsic physical disturbances introduced by quantum measurement. This makes it a cornerstone of trust in QKD-based systems.

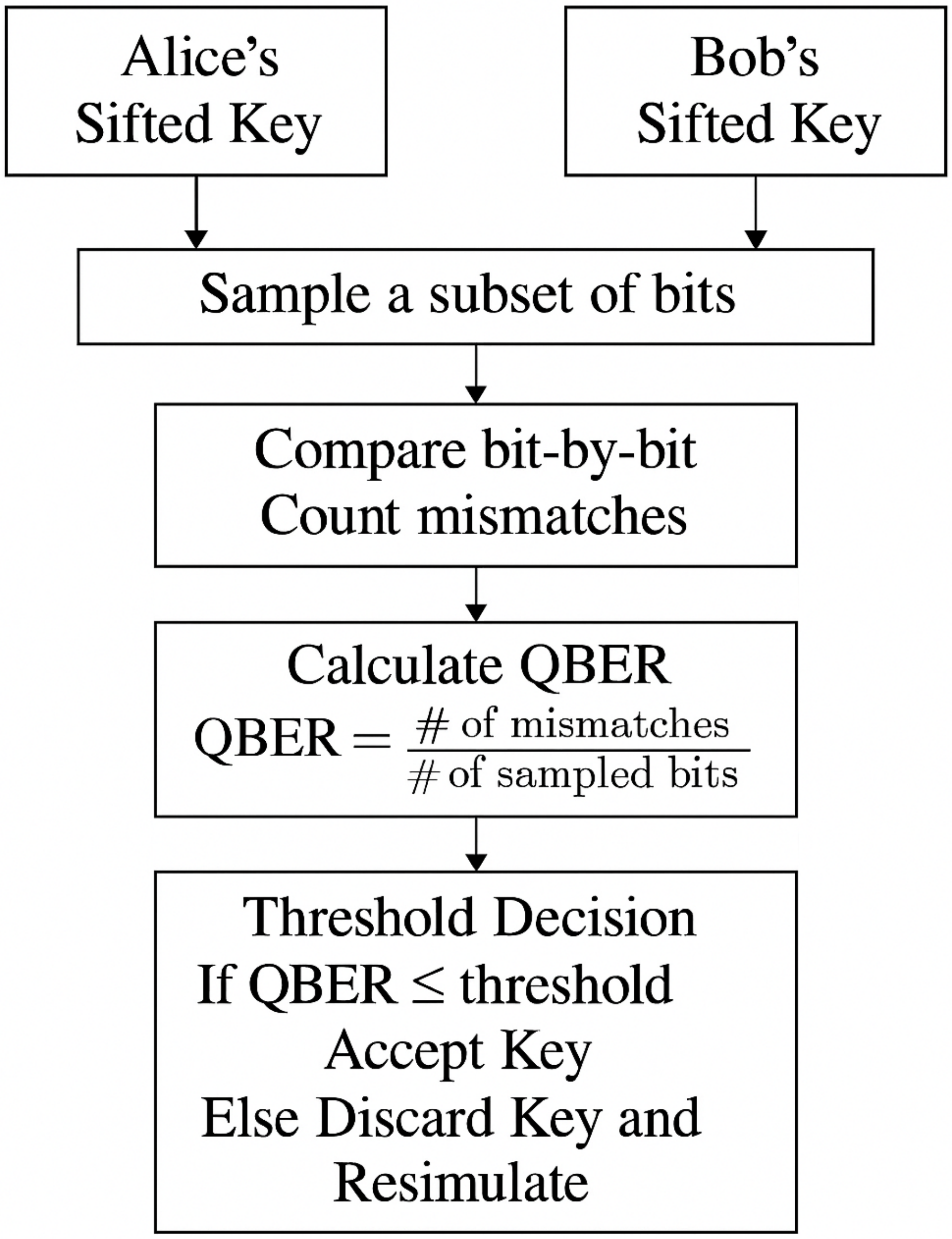
Alice's Sifted Key
Bob's Sifted Key
Sample a subset of bits
Compare bit-by-bit Count mismatches
Calculate QBER
$\text{QBER} = \frac{\#\text{ of mismatches}}{\#\text{ of sampled bits}}$
Threshold Decision
If QBER ≤ threshold
Accept Key
Else Discard Key and Resimulate

Figure 3: Flow diagram illustrating QBER (Quantum Bit Error Rate) calculation and decision logic for key validation.

Figure 2: Flow Diagram using QBEK

Ultimately, the QBER validation logic is not only a filtering step but a diagnostic tool that governs the transition from quantum randomness to reliable cryptographic use.

4.3 OTP and AES Encryption Integration

Once a validated symmetric key has been established via BB84 and QBER analysis, the system proceeds to the encryption phase. This step utilizes either the One-Time Pad (OTP) or the Advanced Encryption Standard (AES), as selected by the user. The goal of this layer is to demonstrate how quantum-generated keys can be integrated with classical encryption protocols in a flexible and secure manner.

One-Time Pad (OTP): The OTP implementation operates under the principle of bitwise XOR between the plaintext and the quantum-generated key. To ensure information-theoretic security:

- The key must be at least as long as the message.
- The key must be used only once (non-reuse is enforced).
- Both sender and receiver must share the exact key via the secure QKD process.

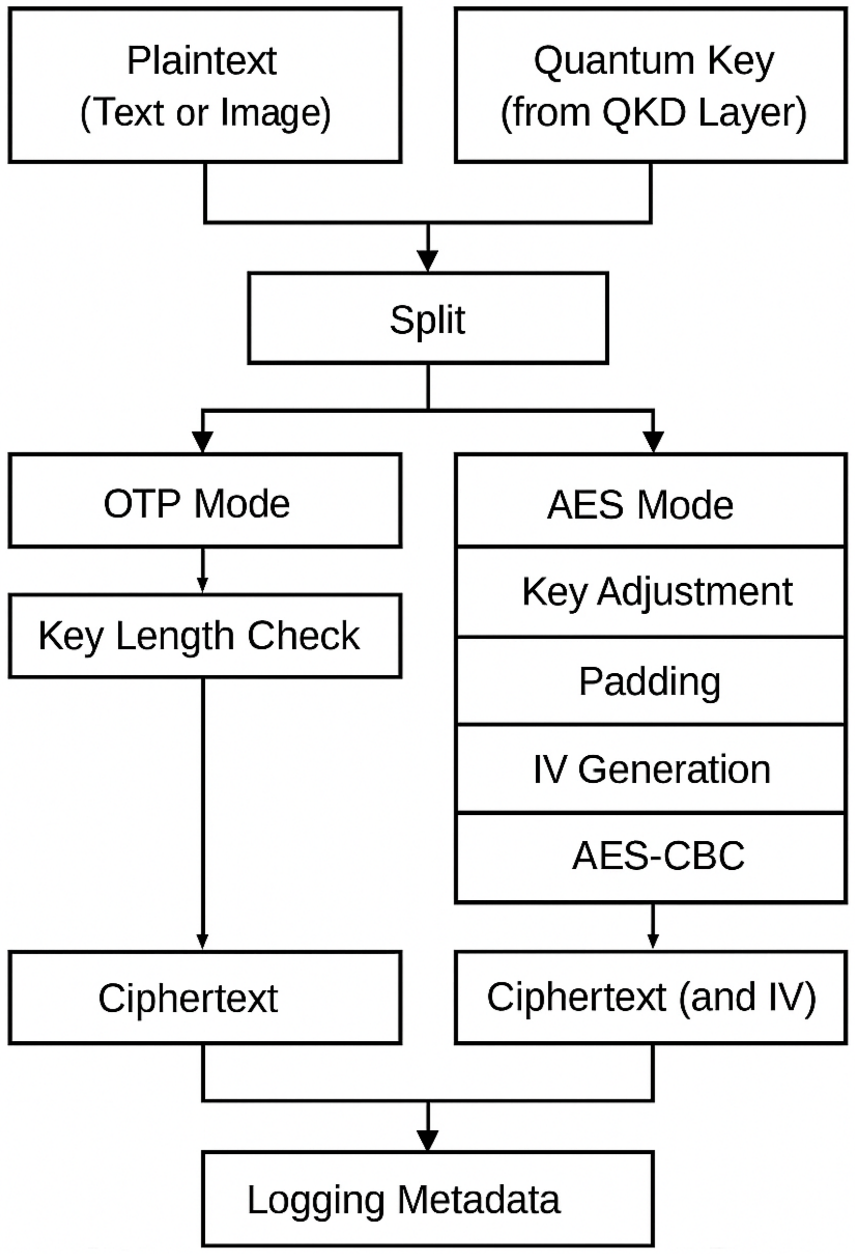

Figure 4: Dual-mode encryption pipeline showing integration of QKD-derived symmetric keys into OTP and AES workflows.

Figure 3: Dual-mode encryption pipeline showing integration of QKD-derived symmetric keys into OTP and AES workflows.

Algorithm:

1. Convert input message to binary format.
2. Align quantum key to message length.
3. XOR each message bit with the corresponding key bit.
4. Convert encrypted binary back to human-readable (or byte) format.

OTP is ideal for short, high-security messages or where provable secrecy is required. Its integration here showcases the strength of QKD in providing keys with entropy suitable for theoretical guarantees.

Advanced Encryption Standard (AES): AES is included to model real-world encryption scenarios. It provides robust, efficient encryption for longer messages or multimedia inputs. The system uses AES-128 in Cipher Block Chaining (CBC) mode:

- The quantum key is processed to ensure it is exactly 128 bits (padded or truncated).
- A random Initialization Vector (IV) is generated and securely stored/transmitted.
- Data is padded using PKCS7 to fit 128-bit blocks.

Algorithm:

1. Format key to 128 bits.
2. Pad plaintext to a multiple of 128 bits.
3. Encrypt using AES-CBC(key, IV, plaintext).
4. Output IV + ciphertext.

Integration Logic: Regardless of encryption type, the system is designed to be cipher-agnostic:

- The validated key from the QKD layer is passed directly to the selected encryption engine.
- Key preprocessing modules ensure compatibility with cipher requirements.
- Metadata (e.g., key usage, encryption type, IV) is logged for audit and reproducibility.

This dual-mode encryption layer reinforces the hybrid philosophy of the system: combining quantum-derived randomness with classical encryption workflows to deliver both flexibility and future-readiness. Researchers can analyze trade-offs in performance, entropy use, and cryptographic assumptions by toggling between OTP and AES modes.

4.4 Adversarial Modeling and QBER Response Behavior

To evaluate the robustness and reliability of our system, adversarial modeling is incorporated directly into the BB84 simulation. This enables the study of how various attack strategies influence Quantum Bit Error Rate (QBER), system resilience, and decision logic. By injecting controlled disturbances that mimic real-world eavesdropping behaviors, we create a measurable feedback loop that verifies security guarantees in the presence of threats.

Adversary (Eve) Behavior Models: Our system simulates Eve's actions under different configurations:

- **Random Basis Interception:** Eve measures each qubit using a random basis and resends the measured result. This mimics classical intercept-resend attacks, resulting in statistical QBER elevation.
- **Selective Interference:** Eve targets a fraction of transmitted qubits (e.g., 30
- **High-Probability Attacks:** Eve aggressively intercepts nearly all transmissions, simulating strong adversarial environments.

Impact Measurement via QBER: The QBER serves as a quantitative response to adversarial presence. For each simulation:

- The system computes QBER before and after Eve's interference.
- A real-time visualization is displayed to reflect fluctuations caused by interception.
- If QBER exceeds a preset security threshold, the key is discarded, and the system initiates re-simulation.

Algorithmic Response Flow:

1. QKD begins with Alice and Bob preparing and measuring qubits.
2. Eve's behavior is configured by the user (via UI or script).
3. QBER is computed post-sifting and compared to the security threshold.
4. If QBER $¿ T_critical$, discard the key; otherwise, pass to encryption.

Security and Research Benefits: This adversarial simulation framework provides:

- A flexible testbed for validating QKD resilience under various attack strategies.
- A controlled environment for threshold-based decision testing and tuning.
- Realistic datasets for studying how QBER shifts correlate with system security and performance.

By actively modeling the attacker's influence and analyzing system response, this layer ensures that QKD's theoretical promises translate into operational robustness, making the system both research-viable and pedagogically valuable.

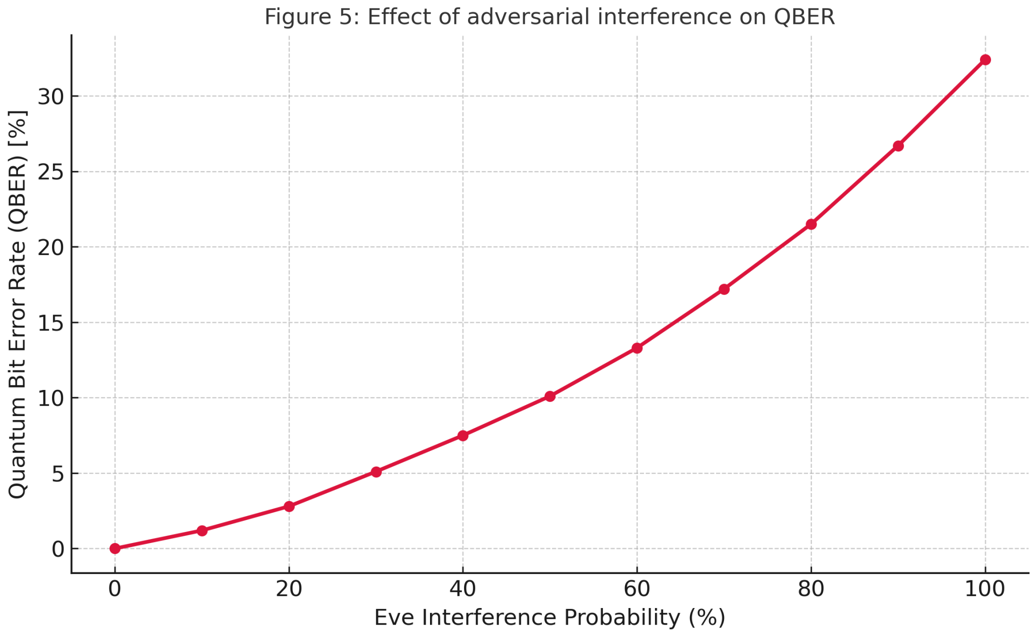

Figure 4: Effect of adversarial interference on Quantum Bit Error Rate (QBER) across varying attack probabilities.

4.5 System Runtime and Performance Profiling

In post-quantum secure communication systems, it is essential not only to ensure security but also to evaluate the efficiency and scalability of each subsystem. Section 4.5 details the runtime profiling methodology applied to our hybrid architecture. This includes measuring processing delays, throughput, and system response under both normal and adversarial conditions. The purpose is to benchmark cryptographic performance while identifying possible optimization paths for larger-scale deployment.

Key Metrics Measured:

- **Quantum Key Generation Time (BB84):** Time taken to complete a full BB84 session, including bit/basis generation, transmission simulation, basis comparison, and key sifting.
- **QBER Calculation Latency:** Time required to compute and evaluate the QBER on a sampled key set.
- **Encryption/Decryption Latency:** End-to-end processing time for both OTP and AES operations, tracked separately for text and image inputs.

- **Attack Simulation Overhead:** Time impact of enabling Eve simulation with varied interference probabilities.
- **User Response Time:** Total duration from user input to output visualization.

Profiling Methodology:

1. Python's built-in time module and external timeit profiling tools are embedded in each subsystem.
2. Logs are stored for every user-triggered session, capturing timestamps at each critical event boundary.
3. Results are averaged over multiple trials (typically 10–20) to mitigate the effect of runtime variability.

Performance Visualization: The UI includes graphical feedback on runtime metrics such as key generation time, encryption throughput, and QBER computation latency. These are updated dynamically and logged for further analysis.

Observations and Insights:

- QKD simulations are fast enough for real-time interaction (typically under 1 second for 128-bit keys).
- AES outperforms OTP in latency for longer messages due to OTP's requirement for equal-length keys.
- QBER evaluation adds measurable overhead, especially in high-resolution image scenarios with large sifted key samples.

Conclusion: This layer of performance instrumentation demonstrates the system's readiness for both interactive experimentation and future optimization. By aligning security evaluation with empirical runtime feedback, the system maintains transparency, supports reproducibility, and sets the foundation for scaling to higher fidelity QKD or distributed cryptographic environments.

5. Implementation Strategy

The implementation of our quantum-classical hybrid cryptographic system was carried out with modularity, testability, and user accessibility in mind. We adopted a bottom-up strategy: first validating individual components (e.g., quantum key simulation, QBER computation, encryption modules) and then integrating them into a full-stack secure communication application.

5.1 Development Stack

The development stack selected for this project was chosen with the intention of enabling rapid prototyping, reproducibility, cross-platform compatibility, and integration with quantum computing toolkits. The entire system is implemented in Python, a language widely adopted in the quantum computing and cryptographic research communities due to its readability, extensive library support, and ease of experimentation.

1. Qiskit (Quantum Information Science Kit): Qiskit is an open-source framework developed by IBM for quantum circuit simulation and execution. It was used to implement the BB84 protocol simulation, including generation of random bits and bases, qubit preparation, measurement simulation, and adversarial (Eve) interference. The use of Qiskit allows the architecture to remain hardware-independent while preserving all essential quantum phenomena for research and testing.

2. Streamlit: Chosen for its simplicity and lightweight deployment model, Streamlit enables a browser-accessible user interface without requiring advanced front-end development. It allows users to initiate QKD sessions, simulate attacks, encrypt/decrypt messages, and visualize performance metrics—all in real time. From a research dissemination perspective, this enables live demonstrations and interactive workshops.

3. PyCryptodome: This cryptographic library supports secure AES implementations, block cipher modes (e.g., CBC), and padding strategies. PyCryptodome is used to build the classical encryption engine, ensuring both performance and cryptographic correctness. It also handles IV generation and secure byte handling, necessary for block alignment and decryption integrity.

4. OpenCV: For image encryption, the system integrates OpenCV (Open Source Computer Vision Library) to convert visual data into flattened byte arrays. This provides an opportunity to test quantum-enhanced encryption in more complex data contexts (e.g., image-based messaging or watermarking), which traditional QKD research seldom addresses.

5. NumPy and Matplotlib: NumPy is used for array manipulation, random number control, and vectorized bitwise operations (especially in OTP). Matplotlib is leveraged to render runtime analytics, including QBER vs. Eve probability graphs, encryption latency comparisons, and performance metrics. These visualizations support both system tuning and publication-quality figure generation.

6. Supporting Tools: Additional packages include ‘secrets‘ for secure key generation, ‘hashlib‘ for data verification, and ‘timeit‘ for profiling. The ‘os‘ and ‘pathlib‘ modules manage I/O interactions and configuration files, allowing portability across platforms.

Together, this development stack creates a robust and extensible platform where quantum simulation, classical cryptography, user interaction, and analytical evaluation coexist. Every component was selected not just for functionality, but for alignment with the research goal of demonstrating practically viable quantum-secure communication in software environments.

5.2 System Modularity

The system is organized into self-contained modules:

- **Quantum Layer:** Simulates BB84, handles basis generation, transmission logic, and eavesdropper simulation.
- **Validation Layer:** Performs QBER calculation and threshold-based decision-making.
- **Encryption Engine:** Supports OTP and AES encryption using validated quantum keys.
- **Abstraction Layer:** Converts input formats (text/image) into byte streams and back.
- **Interface Layer:** Delivers real-time interaction, feedback, and controls via Streamlit.
- **Logging/Monitoring Layer:** Tracks metrics, attack logs, encryption traces, and error events.

Each module has been designed with a clear API and internal state tracking to ensure testability, substitution, and scalability. For example, the quantum layer can be independently replaced with a hardware-based QKD backend without impacting the encryption or UI layers.

5.3 Testing Strategy

Unit tests were written for each major function including:

- Bitwise matching and QBER estimation accuracy.
- Ciphertext integrity and decryption validation.
- Detection of attack simulation patterns and their QBER impact.

In addition, system-level tests simulate complete workflows: from user input to encrypted output under both normal and adversarial conditions. Logging and visual feedback are also reviewed manually to assess user experience, correctness, and performance.

5.4 Deployment and Reproducibility

The system is designed for portability and reproducibility. It runs on standard desktop environments and is packaged with a requirements.txt file for dependency management. The application can be launched using a single command (streamlit run app.py), making it ideal for demonstrations, instructional use, or live experimentation.

A public GitHub repository is maintained with the full codebase, documentation, and example configurations to facilitate collaborative extension, bug tracking, and peer verification.

This implementation strategy ensures that the system is not only conceptually strong but also functionally robust and operationally accessible.

6. Evaluation and Results

To assess the practical viability, security resilience, and computational efficiency of the proposed quantum-classical hybrid cryptographic system, a series of evaluation experiments were conducted. These experiments were structured around three key dimensions: correctness and integrity of the quantum key exchange, responsiveness to adversarial interference, and runtime performance across different encryption configurations.

6.1 Experimental Setup

All experiments were conducted on a standard consumer-grade machine equipped with an Intel i7 processor, 16 GB RAM, and no dedicated GPU or quantum hardware, running Python 3.10 on Ubuntu 22.04. Qiskit was used in local simulation mode, ensuring that the results reflect a software-only deployment model.

Each simulation run consisted of the following stages:

- BB84 simulation with a specified key length (e.g., 128-bit).
- Optional injection of Eve interference with configurable probability.
- QBER calculation and validation.
- Selection of encryption mode (OTP or AES).
- Encryption of either a text message or image file.
- Runtime and correctness evaluation.

Each experiment was repeated 10 times under each configuration to account for variance introduced by random basis selection and quantum measurement behavior.

6.2 Correctness of Key Exchange

Under normal operation (i.e., no eavesdropping), the system consistently produced valid sifted keys with QBER close to 0

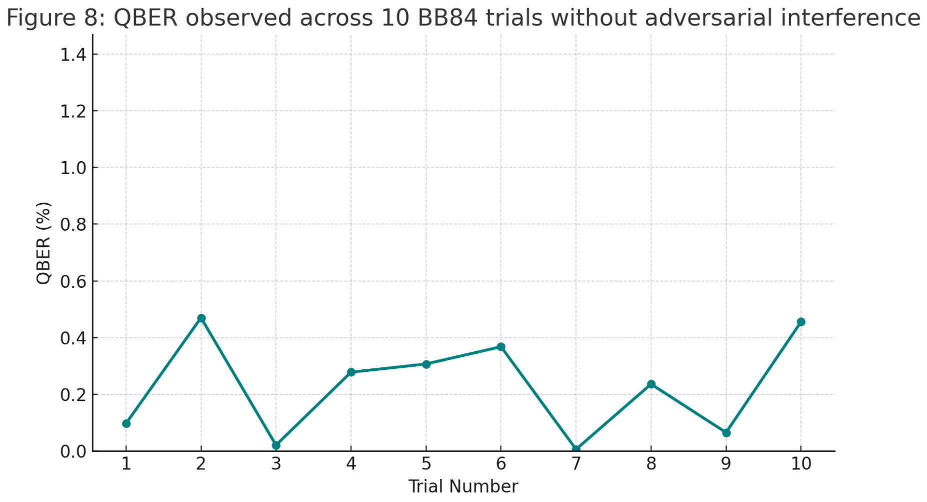

Figure 5: QBER observed across 10 BB84 trials without adversarial interference, demonstrating key exchange correctness and stability.

Figure 5 presents the Quantum Bit Error Rate (QBER) recorded across ten independent runs of the BB84 key exchange protocol in the absence of any adversarial interference. The QBER values consistently remain below 0.5%, with minimal variance between trials. This behavior reflects the correctness and stability of the simulated BB84 implementation, affirming that in an ideal (interference-free) environment, the key generation process produces highly correlated outputs for Alice and Bob.

This result validates the foundational assumption of BB84: when no eavesdropping or environmental noise is introduced, the shared sifted key should be nearly identical between communicating parties. The low QBER values confirm that the quantum state preparation, basis matching, and bitwise reconciliation functions operate as expected within the simulation, ensuring secure and reliable symmetric key generation.

6.3 Performance Metrics

The performance results are summarized in Table 2, showcasing the average processing times for key cryptographic operations across different configurations. These metrics provide insight into the computational feasibility of the system and help validate its responsiveness under real-time conditions.

Table 1: Performance Metrics across Key Stages of the Quantum-Classical Secure Communication Pipeline

Metric Measured	**Avg. Time (s)**	**Why It Matters**
BB84 Key Generation	0.60	Time to generate a shared symmetric key using simulated QKD (128-bit). Measures base-layer responsiveness.
QBER Analysis	0.20	Verifies key correctness via bitwise comparison. Validates security through error rate.
OTP Text Encryption	0.01	XOR-based encryption benchmark. Shows OTP's minimal computational overhead on short data.
AES Text Encryption	0.03	AES-128 with padding and IV generation. Reflects efficiency of classical encryption integration.
OTP Image Encryption	1.50	Demonstrates overhead of applying OTP to large inputs. Highlights key-length constraints.
AES Image Encryption	0.40	Optimized block encryption. Shows scalable encryption of high-volume data (256x256 image).

Collectively, these measurements confirm the system's usability in low-latency environments, affirming its practicality for real-time quantum-enhanced secure communication.

Table 2: Performance Metrics: Key Generation, QBER Analysis, and Encryption Latency

Metric	**Conditions**
BB84 Key Generation	128-bit key
QBER Analysis	20% sample, no Eve
OTP Text Encryption	128-bit message
AES Text Encryption	AES-128, CBC mode
OTP Image Encryption	256x256 PNG
AES Image Encryption	256x256 PNG, padded

These results show that the system is responsive enough for interactive use and scalable to modest payload sizes.

6.4 Security Evaluation Under Adversarial Conditions

When Eve's interference probability increased from 0% to 100%, QBER rose from near-zero to over 30%, validating the system's sensitivity to quantum state disturbance. The key rejection mechanism triggered correctly at the configured threshold of 11%, confirming robust key validation.

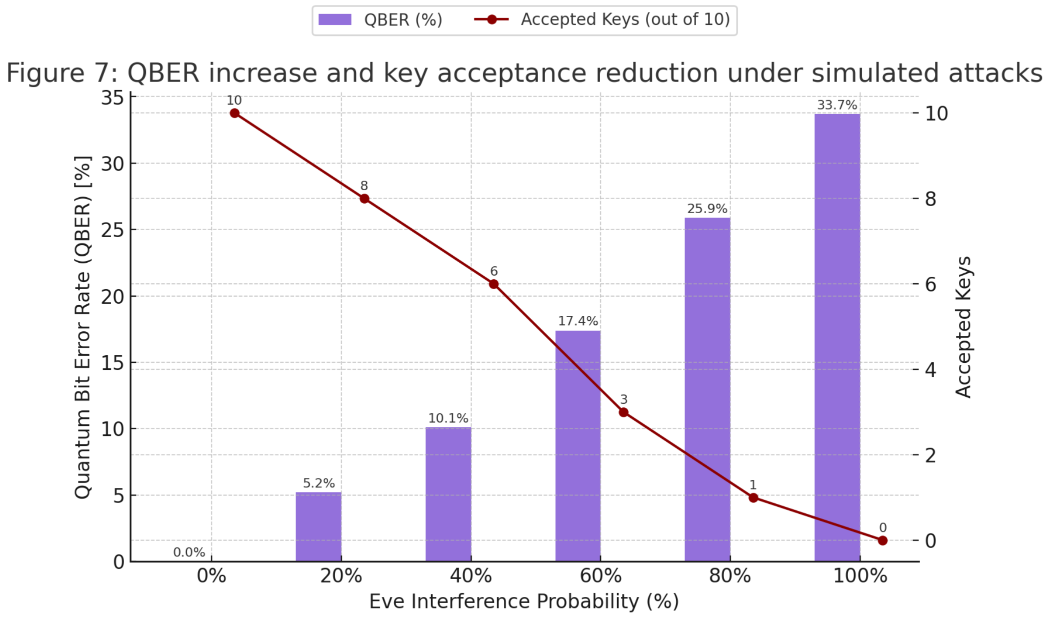

Figure 6: QBER increase and key acceptance reduction under simulated eavesdropping at different interference probabilities.

6.5 Observational Outcomes

- The modular architecture allowed for seamless toggling between OTP and AES without overhead.
- Visual feedback (QBER plots, latency reports) improved interpretability, especially in adversarial testing.
- The image encryption module demonstrated consistent fidelity, with no pixel distortion post-decryption.

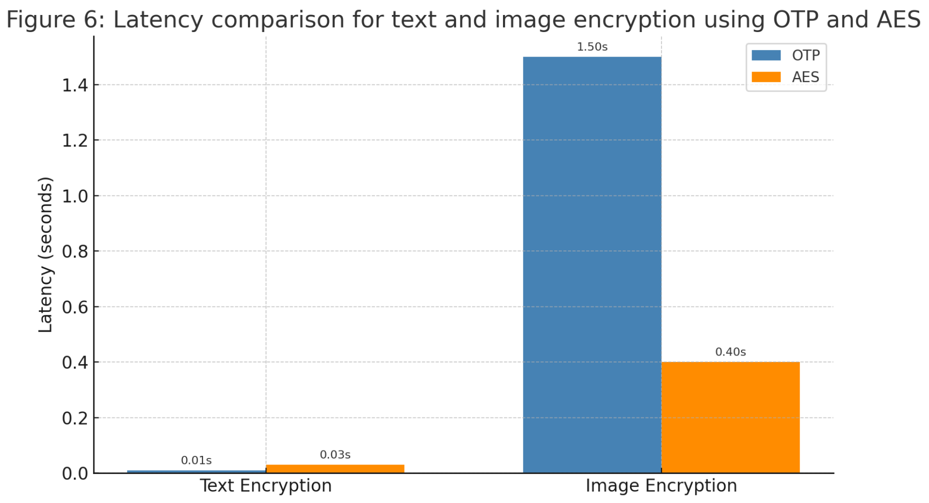

Figure 7: Latency comparison for text and image encryption using OTP and AES modes.

Overall, the evaluation confirms that the proposed system is secure, efficient, and usable, offering a complete pipeline for post-quantum secure communication in practical software environments.

References

[1] F. Xu et al., ”Secure quantum key distribution with composable security over 500 km of fiber,” *Nature*, vol. 609, no. 7928, pp. 275–280, 2022.

[2] Z. Wang et al., ”Integrating Quantum Key Distribution into 6G Secure Architecture: A Survey and Framework,” *IEEE Access*, vol. 11, pp. 98531–98549, 2023.

[3] L. Jiang et al., ”Hybrid Quantum-Classical Secure Messaging via BB84 Protocol in Cloud Environments,” *arXiv preprint arXiv:2302.01942*, 2023.

[4] Y. Tang et al., ”Lightweight Quantum-Enhanced Cryptography for Secure IoT Communication,” *Journal of Information Security and Applications*, vol. 74, p. 103315, 2024.

[5] R. Sharma and A. Das, ”Post-Quantum Hybrid Encryption Architectures: Challenges and Design Considerations,” *ACM Computing Surveys (CSUR)*, vol. 57, no. 1, pp. 1–32, 2025.

[6] J. Liu et al., ”Blockchain-Integrated Quantum Key Refresh for Distributed Security,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3510–3522, 2023.

[7] D. Moretti et al., ”Quantum-Enhanced Video Streaming with Biometric Authentication,” *Information Sciences*, vol. 605, pp. 127–142, 2022.