A PROJECT STAGE-I REPORT

on

"Quantum-Safe Communication using BB84"

submitted to the

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

Gayatri Bhosale (IT4005)

Shrikant Hundekar (IT4020)

Shriram Narkhede (IT4042)

Under the Guidance of: -

Prof. Pravin R. Kamble

DEPARTMENT OF INFORMATION TECHNOLOGY

KJ'S EDUCATIONAL INSTITUTE

TRINITY COLLEGE OF ENGINEERING AND RESEARCH, Pune

NEAR KHADI MACHINE CHOWK, KONDHWA ANNEX,PUNE-411048

Academic Year: 2025-26

CERTIFICATE

This is to certify that the project report entitled

"Quantum-Safe Communication using BB84"

submitted by

Gayatri Bhosale (IT4005)
Shrikant Hundekar (IT4020)
Shriram Narkhede (IT4042)

is a bonafide work carried out by them under the supervision and guidance of ***Prof. Pravin R. Kamble****, and it is approved for partial fulfillment of the requirement for the* ***BE (Information Technology Engineering)*** *course of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology Engineering).*

This project report has not been earlier submitted to any other institute or university for the award of any degree or diploma.

Prof. Pravin R. Kamble
Internal guide

Dr. Vilas Gaikwad
Head of Department (IT)

External Examiner

Dr. Abhijeet Auti
Principal (TCOER,Pune)

ACKNOWLEDGEMENT

We would like to take this opportunity to express our gratitude towards all the people who have in various ways helped in the successful completion of our project. We must convey our gratitude to **Prof. Pravin R. Kamble** for giving us the constant source of inspiration, personally correcting our work, and providing encouragement throughout the project.

We express deep gratitude towards **Dr. V. S. Gaikwad**, Head of the Information Technology Engineering Department, for providing support and giving his valuable time.

We also express our gratitude towards our **father, mother, other family members, and our friends** for encouraging us with their valuable suggestions and motivating us from time to time.

Group No. PR202526_08

Name of Students (Seat No) **Sign.**

Gayatri Bhosale (IT4005)

Shrikant Hundekar (IT4020)

Shriram Narkhede (IT4042)

DECLARATION

We, the undersigned, hereby declare that the Project entitled **"Quantum-Safe Communication using BB84"** submitted by us to **Trinity College of Engineering and Research, Pune**, for the award of the degree of **Bachelor of Engineering in Information Technology Engineering**, under the guidance of **Prof. Pravin R. Kamble** is our original work.

We further declare that to the best of our knowledge and belief, this work has not been previously submitted to this or any other university.

Place: Pune

Date: 14/10/2025

Group No. PR202526_08
Gayatri Bhosale (IT4005)
Shrikant Hundekar (IT4020)
Shriram Narkhede (IT4042)

Abstract

The rapid evolution of quantum computing introduces serious risks to existing cryptographic infrastructures, particularly those based on **RSA** and **Elliptic Curve Cryptography (ECC)**, which depend on computational complexity for security. Quantum algorithms such as Shor's and Grover's can efficiently compromise these systems, exposing modern communication networks to potential quantum-era vulnerabilities. To mitigate these threats, this project proposes a **Quantum-Safe Communication System** that integrates the **BB84 Quantum Key Distribution (QKD) protocol** with advanced **Post-Quantum Cryptography (PQC)** techniques. In this system, the BB84 protocol is emulated to establish a secure quantum key exchange between two users—Alice and Bob—via a simulated quantum channel. The raw quantum key undergoes **error correction** and **privacy amplification** to produce a refined secret key for secure communication. For message confidentiality and authentication, the system employs **One-Time Pad (OTP)** encryption alongside **HMAC-SHA3-256**, ensuring both secrecy and integrity. File-level protection is achieved through the **XChaCha20-Poly1305** algorithm, which offers high-speed authenticated encryption and resistance to nonce-reuse attacks. Additionally, a hybrid key management model is implemented using the lattice-based **Kyber Key Encapsulation Mechanism (KEM)**, combining quantum and classical cryptographic strengths. This hybrid framework guarantees end-to-end resilience against both traditional and quantum attacks, making it adaptable for future secure communication environments. The project effectively demonstrates the integration of **QKD**, modern symmetric encryption, and post-quantum algorithms into a unified framework—establishing a practical foundation for developing next-generation quantum-secure communication architectures.

Contents

Abbreviations

BB84 – Bennett–Brassard 1984 Quantum Key Distribution Protocol
QKD – Quantum Key Distribution
QBER – Quantum Bit Error Rate
OTP – One-Time Pad
HMAC – Hash-based Message Authentication Code
SHA-3 – Secure Hash Algorithm 3
SHA-256 – Secure Hash Algorithm 256-bit
HKDF – HMAC-based Key Derivation Function
AEAD – Authenticated Encryption with Associated Data
AES – Advanced Encryption Standard (for comparison/reference)
GUI – Graphical User Interface
API – Application Programming Interface
QPU – Quantum Processing Unit
Qubit – Quantum Bit
QKD-SIM – Quantum Key Distribution Simulation System
PQC – Post-Quantum Cryptography
QCS – Quantum Communication System

List of Figures

List of Tables

Synopsis

Name of the College	:	Trinity College of Engineering and Research
Name of the Branch	:	Information Technology
Name of the Students	:	Mr. Shrikant Hundekar
		Mr. Shriram Narkhade
		Miss. Gayatri Bhosale
Project Group Number	:	PR2025-26_08
Name of the Guide	:	Prof. Pravin R. Kamble
Broad Area	:	Cybersecurity and Cryptography
Project Title	:	Quantum-Inspired Cryptography Simulator Using Tensor Networks

1 Abstract

A high-fidelity simulator for quantum cryptographic systems is presented, leveraging the power of tensor networks to address the computational limitations of classical hardware. The simulator accurately models the evolution of quantum states and the effects of quantum measurements, enabling the emulation of protocols such as BB84. Through the use of advanced tensor contraction algorithms, the framework efficiently manages the entanglement and superposition characteristics intrinsic to quantum systems.

Key functionalities include the simulation of sender and receiver operations, as well as various eavesdropping attack scenarios, offering a comprehensive platform for evaluating the security of cryptographic keys. This tool serves as both an educational and research resource, bridging the gap between theoretical quantum cryptography and practical, classical-based experimentation. The framework maintains a detailed log of all simulated operations—from qubit generation to key distillation—facilitating in-depth post-simulation analysis. Its modular architecture allows for the easy integration of alternative components, such as diverse noise models or entanglement-based protocols, thereby supporting the exploration of a wide range of quantum cryptographic scenarios.

Keywords **:** Quantum Key Distribution (QKD),BB84 Protocol,Tensor Networks,Quantum Cryptography,Quantum State Simulation,Eavesdropping Detection,Tensor Contraction Algorithms, Quantum Entanglement,Cryptographic Key Security,Modular Simulation Framework

2 Introduction

In today's digital landscape, where communication underpins national defense, banking, and critical infrastructure, data security has become paramount. While classical encryption remains effective, it faces increasing risks from emerging quantum computing technologies, which could eventually break traditional cryptographic schemes [1]. Quantum cryptography—especially protocols like BB84—offers a theoretically secure alternative based on the laws of quantum mechanics [2]. The BB84 protocol, introduced by Bennett and Brassard, allows two parties to establish a shared secret key with guaranteed security against eavesdropping [3]. However, practical deployment of quantum hardware remains limited due to high costs and technical barriers [4].

To bridge this gap, this project simulates the BB84 Quantum Key Distribution (QKD) protocol using tensor networks on classical hardware. The simulation enables secure key generation between two parties while modeling quantum phenomena such as superposition and measurement-induced collapse, which expose eavesdropping attempts [5]. Tensor networks provide an efficient framework for simulating quantum systems by managing entanglement and reducing computational overhead [6]. Recent advances, including GPU acceleration and optimized contraction techniques, have significantly improved the performance of such simulations [8].

Existing tools such as NuQKD [1], QKDNetSim+ [2], and SimQN[4] contribute to modeling QKD in various contexts, though many lack full quantum-level fidelity or encryption capability. In parallel, research has demonstrated the power of tensor networks in simulating large-scale quantum circuits, optimizing performance on GPUs, and extending into applications like cryptanalysis. Community-based parallel tensor contraction has further enabled efficient simulation of complex quantum systems.

This work builds on these developments to offer a modular, high-fidelity simulation platform for exploring and demonstrating quantum cryptographic principles using classical computing resources.

3 Problem Statement

With the rise of quantum computing, traditional cryptographic methods are becoming less secure. While protocols like BB84 offer quantum-safe communication, real quantum hardware is expensive and not widely accessible. To address this, there is a need for a web-based simulation tool that can demonstrate BB84 Quantum Key Distribution (QKD) and secure communication on classical systems. This research aims to develop a user-friendly, web-based platform that uses tensor networks to simulate quantum behavior, enabling secure key generation, eavesdropping detection, and message encryption without requiring actual quantum devices.

4 Literature Review

4.1 Brief Review of Literature

Several simulation tools and models have been developed in recent years to study and emulate the BB84 Quantum Key Distribution (QKD) protocol. These systems aim to enhance understanding of quantum cryptographic principles in classical environments and to assess their resilience against noise and eavesdropping.

NuQKD is a modular, Python-based QKD simulation framework designed to replicate physical-layer conditions such as photon loss, dark counts, and detector inefficiency. It enables accurate calculation of the Quantum Bit Error Rate (QBER) under various scenarios. However, its functionality is limited to physical modeling and does not extend to message encryption or complex eavesdropping strategies, thereby limiting its practical applicability in full-stack cryptographic research [1].

QKDNetSim+ extends the NS-3 network simulator to incorporate QKD within classical network infrastructures. While it facilitates the study of QKD in networked environments, it lacks simulation of essential quantum protocol components like basis selection and quantum noise, making it more suitable for network-layer research than for protocol-level analysis [2].

An enhanced version of QKDNetSim+ has improved usability, modularity, and scalability. It supports the simulation of quantum channels across large-scale networks but still does not model low-level quantum operations such as qubit behavior or measurement collapse, which are critical for high-fidelity BB84 simulations [3].

SimQN, a platform introduced in IEEE Network, provides a network-focused environment for simulating QKD mechanisms including BB84. It features end-to-end modeling of quantum communication performance, accounting for quantum memory, entanglement distribution, and routing strategies. Although SimQN is geared toward infrastructure-level studies, it serves as an important bridge between quantum hardware simulation and network application design [4].

Recent developments in tensor network techniques have enabled efficient classical simulation of quantum circuits. One study proposed the use of Tree Tensor Networks (TTNs) to adapt simulation topologies to the expected entanglement patterns in quantum circuits. This approach reduced computational costs significantly, achieving speedups of up to two or-

ders of magnitude for circuits with up to 37 qubits [5]. Another study used tensor networks to simulate IBM's 127-qubit Eagle quantum processor. By optimizing network geometry and applying belief propagation for approximate contraction, the simulation achieved results more accurate than those from the physical device itself, highlighting the effectiveness of classical tensor network methods [6].

To further enhance performance, tensor network simulations have been optimized for modern GPUs using mixed-precision arithmetic and highly efficient GEMM operations. These techniques have demonstrated over 21 TFLOPS performance on a single A100 GPU, significantly outperforming CPU-based methods and previous GPU approaches [7]. A broader survey by Díez García and Márquez Romero outlines tensor network applications in areas such as machine learning, optimization, and quantum circuit simulation. The review clarifies how different tensor network architectures compare in accuracy, complexity, and domain suitability [8].

An innovative application of tensor networks is in cryptanalysis. Aizpurua et al. demonstrated the use of Matrix Product States (MPS) and Flexible-PEPS simulators to attack symmetric-key ciphers like S-AES and Blowfish. Their work suggests that different tensor architectures can offer varying benefits in terms of speed and scalability depending on key length and cipher structure [9]. Finally, a novel parallel algorithm was proposed using community detection to partition tensor networks into sub-networks. These are contracted in parallel, making the method particularly effective for highly entangled and random quantum circuits, and showcasing a new form of parallelism for computationally intensive quantum simulations [10].

Table 1: Literature Survey of BB84 Simulation Tools and Tensor Network Techniques

Tool / Study	**Focus Area**	**Limitations / Key Contributions**
NuQKD [1]	Physical-layer QKD simulation with photon loss, dark counts, detector inefficiency	High QBER analysis; lacks encryption and complex eavesdropping simulation
QKDNetSim+ [2]	NS-3 based simulation of QKD in classical networks	No quantum-level operations; more network-level analysis focus
Enhanced QKDNetSim+ [3]	Improved version with better scalability and modularity	Still lacks modeling of low-level quantum behaviors like qubit measurement
SimQN [4]	Network-focused simulation platform with BB84 support	End-to-end communication modeling; strong infrastructure focus, not protocol level
TTN-based simulation [5]	Use of Tree Tensor Networks for efficient quantum circuit emulation	Speedup in classical simulation up to 37 qubits; ideal for BB84-like systems
IBM Eagle processor simulation [6]	Simulation using tensor networks of 127-qubit system	More accurate than hardware using belief propagation and tensor optimization
GPU-optimized TN [7]	Tensor networks using GPU and mixed-precision for high performance	Achieved 21+ TFLOPS on A100; best suited for large-scale simulations
Diez García	Romero [8] Survey of tensor networks in ML and quantum simulation	Comparative study on architectures' suitability for cryptography and QKD
Aizpurua et al. [9]	Cryptanalysis using MPS and PEPS tensor networks	Showed varying success on S-AES and Blowfish; applicable to QKD security studies
Parallel TN algorithm [10]	Tensor networks partitioned and simulated in parallel using community detection	Efficient for highly entangled systems; accelerates complex QKD simulations

4.2 Significance/Relevance

The proposed project plays a pivotal role in democratizing access to quantum cryptography, specifically the BB84 Quantum Key Distribution (QKD) protocol, by eliminating the reliance on costly quantum hardware. As advancements in quantum computing increasingly threaten the robustness of classical encryption methods, there is a pressing need to understand and implement quantum-resistant cryptographic techniques. This project addresses that need by simulating the BB84 protocol using tensor network-based algorithms on classical computing systems.

By providing a user-friendly, web-based environment, the simulator enables users—including students, researchers, and developers—to visualize secure key exchange processes, simulate

eavesdropping attacks, and explore quantum-safe encryption mechanisms. The framework facilitates practical experimentation with core principles of QKD, thus bridging the gap between theoretical quantum security models and their practical, real-world applications. Ultimately, this contributes meaningfully to the field of post-quantum cryptography and enhances preparedness for future cybersecurity challenges.

5 Proposed Work

5.1 Objectives

1. Simulate the BB84 Protocol Implement the BB84 Quantum Key Distribution protocol on a classical computer using Tensor Networks to mimic quantum behavior efficiently.
2. Secure Key Generation Generate a secret key between two users (Alice and Bob) using quantum principles like superposition and measurement.
3. Eavesdropping Detection Simulate attacks by an eavesdropper (Eve) and show how quantum rules help detect such intrusions in the communication.
4. Interactive Visualization Provide a user-friendly web interface to visualize each step of the QKD process — from qubit transmission to key matching.
5. Quantum-Secure Messaging Use the generated key to encrypt/decrypt messages, demonstrating secure communication based on quantum cryptography

5.2 Methodology

The proposed project initiates with the design and development of a simulation framework that replicates the BB84 Quantum Key Distribution (QKD) protocol using Tensor Networks on classical computing systems. Tensor Networks are utilized to efficiently model quantum states and operations, allowing the simulation of quantum behavior without requiring access to quantum hardware.

Within this framework, two communicating parties—Alice and Bob—exchange qubits encoded in randomly selected bases. Following the BB84 protocol, the recipients perform measurements on the incoming qubits using randomly chosen bases as well. A classical public channel is then used to compare measurement bases, enabling the generation of a shared secret key after discarding mismatched bits.

To simulate real-world security challenges, the system incorporates a configurable eavesdropper module (Eve), which introduces quantum-level disturbances in the communication channel. This enables the demonstration of how such intrusions increase the Quantum Bit Error Rate (QBER), making eavesdropping detectable due to the fundamental principles of quantum mechanics.

An interactive web-based user interface has been developed to visualize the complete QKD process. This includes qubit generation, transmission, measurement, basis comparison, error checking, and key reconciliation. Additionally, the framework integrates an encryption module that leverages the generated quantum key to securely encrypt and decrypt messages between Alice and Bob. This practical extension illustrates the application of quantum-secure communication, reinforcing the system's value as an educational and experimental platform for students, researchers, and developers.

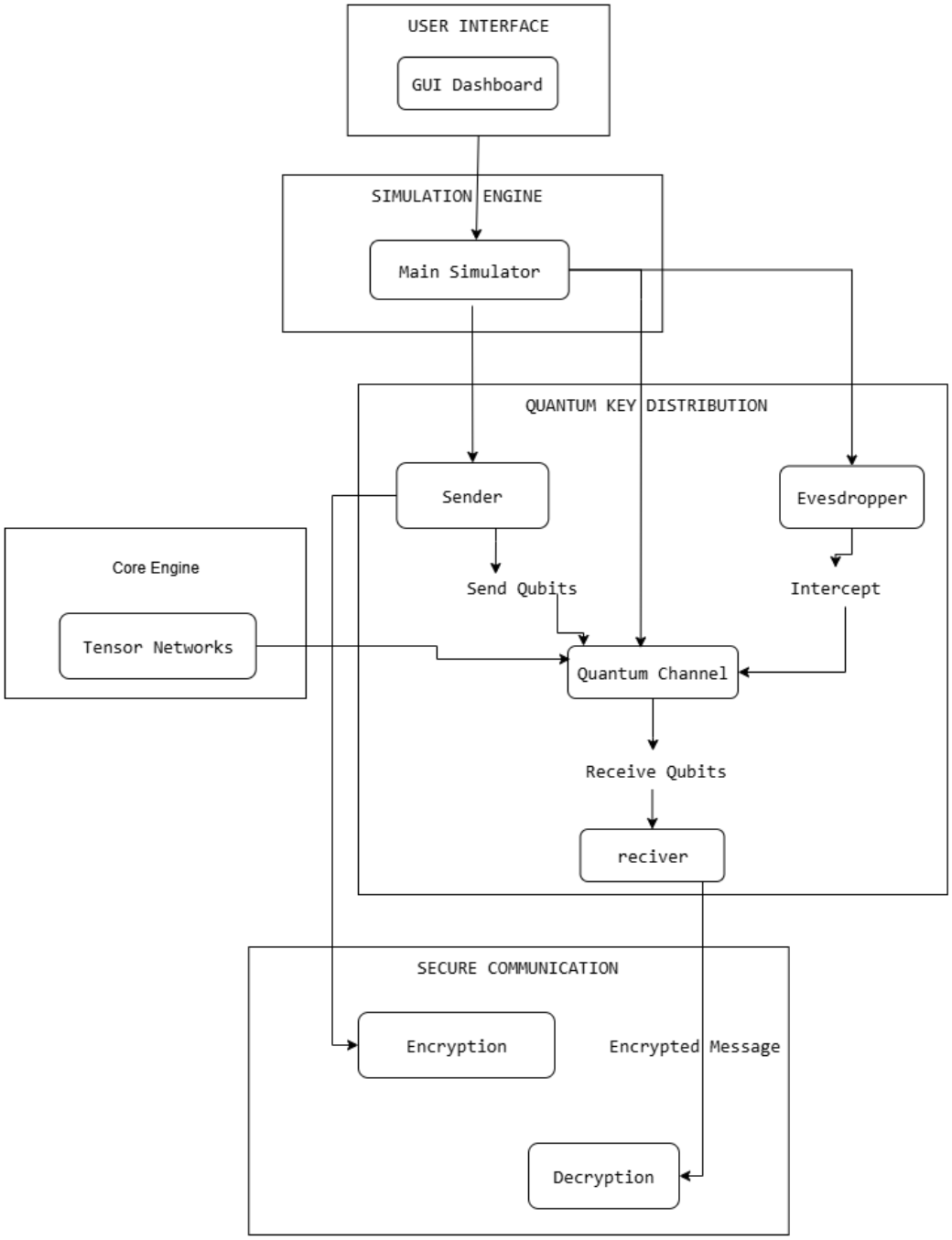


Figure 1: System Architecture

5.3 Schedule of Project Stage-I Work

Week No	Activity Planned
01	Finalization of Project Domain, Submission of 4–5 Proposed Topics, and Literature Survey (minimum 2–3 papers per topic).
02	Identification of Problem Statement/Title based on literature review, feasibility analysis, and gap identification. **Pre-Review Presentation** and Topic Finalization.
03	Documentation of Motivation, Objectives, and Defined Scope of the Finalized Project.
04	Comparative Study of Existing Methods/Tools, Feasibility Assessment, and Listing of Required Resources (tools, technologies, hardware/software).
05	Overview of Proposed System, Expected Output, and Basic Design. **Project Review – 1**.
06	System Architecture and Initial Design Elements (UML,Context-level DFD, High-Level Block Diagrams).
07	Documentation of System Requirements – Functional and Non-Functional.
08	Preparation of the SRS (Software Requirements Specification) Document as per the finalized problem statement.
09	Requirement Analysis using UML/ER Diagrams, Use Case Diagrams, Class/Activity Diagrams. **Project Review – 2**.
10	Detailed System Architecture, Design Specifications, and Algorithmic Analysis.
11	Module-level Testing Plan and Initial Test Case Design.
12	Detailed Design – Expanded DFD Levels, Module Specifications, and Data Dictionary
13	Development of Initial Working Modules (at least 3–4). Submit partial coding with documentation. Continue survey paper preparation.
14	Drafting of Research Paper, Finalization of Project Report Format, Timeline Planning. **Project Review – 3**.

Table 2: Schedule of Project Stage-I Work

Apart from above schedule, some additional weeks are required for below mentioned activities:

- Conference/Journal paper Preparation and Presentation.
- Semester-I project stage I Report Preparation.
- Semester-II project stage II Final project presentation and Report Preparation.

5.4 Software/Hardware Requirements:

Software Requirements:

- **Platforms:** Windows/Linux
- **Language:** Python
- **Libraries and Packages:** NumPy, Matplotlib, Seaborn, TensorNetworkX, cryptography
- **IDE/Editor:** VS code / Pycharm

Hardware Requirements:

- **4 GB+ RAM**
- **Storage :** 2 GB+ Free Space

References

[1] A. Gkouliaras, A. Nychis, and I. Pitas, "NuQKD: A modular quantum key distribution simulation framework for engineering applications," arXiv preprint arXiv:2310.12351, Oct. 2023.

[2] P. Soler, A. Chorti, and E. Papagiannaki, "QKDNetSim+: Extension of NS-3 for realistic quantum key distribution network simulation," arXiv preprint arXiv:2402.10822, Feb. 2024.

[3] D. Soler, I. Cillero, C. Dafonte, M. Fernández-Veiga, A. Fernández-Vilas, and F. J. Nóvoa, "QKDNetSim+: Improvement of the quantum network simulator for NS-3," SoftwareX, vol. 101685, Feb. 2024.

[4] L. Chen et al., "SimQN: A network-layer simulator for the quantum network investigation," IEEE Network, vol. 37, no. 5, pp. 182–189, Sep. 2023

[5] P. Seitz, I. Medina, E. Cruz, Q. Huang, and C. B. Mendl, "Simulating quantum circuits using tree tensor networks," Quantum, vol. 7, p. 964, Mar. 2023

[6] J. Tindall, M. Fishman, M. Stoudenmire, and D. Sels, "Efficient tensor network simulation of IBM's Eagle kicked Ising experiment," arXiv preprint, arXiv:2306.14887, Jun. 2023.

[7] F. Pan, X. Gao, L. Zhao, and L. Lin, "Efficient quantum circuit simulation by tensor network methods on modern GPUs," arXiv preprint, arXiv:2310.03978, Oct. 2023.

[8] M. Díez García and A. Márquez Romero, "Survey on computational applications of tensor network simulations," arXiv preprint, arXiv:2408.05011, Aug. 2024.

[9] B. Aizpurua, S. Patra, J. Etxezarreta Martinez, and R. Orús, "Hacking cryptographic protocols with tensor network attacks," arXiv preprint, arXiv:2409.04125, Sep. 2024.

[10] A. M. Pastor, J. M. Badia, and M. Castillo, "A community detection–based parallel algorithm for quantum circuit simulation using tensor networks," J. Supercomputing, Springer, vol. 80, pp. 5212–5233, Dec. 2023.

[11] A. Moiseevskiy, "Quantum-enhanced symmetric cryptanalysis for S-AES," arXiv preprint, arXiv:2304.05380, Apr. 2023

Miss. Gayatri Bhosale

Mr. Shrikant Hundekar

Mr. Shriram Narkhade — **Prof. Pravin R. Kamble**

Name of The Students — **Name of the Guide**

Dr. V. S. Gaikwad

Head of the Department (IT)

Chapter 1

Introduction

1.1 Overview

The developed quantum-safe communication framework combines the principles of **BB84 Quantum Key Distribution (QKD)** and **Post-Quantum Cryptography (PQC)** to establish a highly secure medium for exchanging both chat messages and files. In this system, Alice and Bob initiate the process by transmitting qubits through a quantum channel using the BB84 protocol to create an initial raw quantum key. Subsequent stages of basis matching and error correction refine this key into a final shared quantum secret that forms the core of the secure communication process.

Alongside the quantum-derived key, a complementary post-quantum key is produced using a PQC algorithm such as Kyber [4]. These two keys are then merged through a **Key Derivation Function (KDF)** to form a hybrid session key [8]. This hybridization improves overall security and fault tolerance—if one cryptographic layer is ever compromised, the remaining component continues to safeguard the session.

For instant messaging, encryption is implemented using a one-time pad derived from the hybrid key and verified with **HMAC-SHA3**, ensuring both data integrity and confidentiality [10]. File encryption employs **XChaCha20-Poly1305**, an authenticated encryption scheme known for its high performance and robustness against cryptographic attacks [9]. The system operates across both quantum and classical communication channels: the quantum channel facilitates secure key distribution, while the classical channel supports reconciliation and encrypted data transfer. Overall, this architecture offers a scalable, fault-tolerant, and quantum-resilient communication platform applicable to modern secure communication environments [5, 6].

1.2 Aims/Motivation

The primary objective of this project is to **design and develop a quantum-safe communication framework** capable of securely transmitting chat messages and files through the combined use of **BB84 Quantum Key Distribution (QKD)** and **Post-Quantum Cryptography (PQC)**. The system is intended to ensure **confidentiality, integrity, and authentication** of data against both conventional and quantum-based cyber threats, while maintaining practicality for real-world communication environments.

The emergence of **quantum computing** introduces serious risks to traditional encryption systems, as quantum algorithms may be capable of breaking widely adopted public-key schemes. This growing threat highlights the necessity for **quantum-resilient security architectures** that integrate quantum key distribution with post-quantum algorithms to safeguard sensitive data. Accordingly, this project focuses on the following objectives:

1. Safeguard confidential information from potential quantum computing attacks.

2. Investigate hybrid key generation models that merge QKD and PQC to reinforce communication security.

3. Enable authenticated and encrypted data exchange for both textual and file-based communication.

4. Translate theoretical advancements in quantum cryptography into a functional and scalable implementation suitable for modern communication infrastructures.

1.3 Objectives

1. To design and implement a **hybrid quantum-secure communication framework** that integrates **BB84 Quantum Key Distribution (QKD)** with **Post-Quantum Cryptography (PQC)** for enhanced protection.

2. To protect chat communications through **OTP and HMAC-SHA3**, ensuring data **confidentiality**, **integrity**, and **authentication** during transmission.

3. To safeguard file exchanges using the **XChaCha20-Poly1305** algorithm, enabling efficient and authenticated encryption for larger datasets.

4. To generate a **hybrid session key** by combining quantum-derived and post-quantum keys, thereby strengthening cryptographic robustness and resilience.

5. To conduct **experimental validation** involving key generation, encryption and decryption processes, as well as secure message and file transfers with error recovery mechanisms.

6. To assess the **performance, reliability, and scalability** of the developed system for deployment in practical communication environments.

7. To identify and discuss **potential use cases and future enhancements** for integrating quantum-safe communication in real-world network infrastructures.

Chapter 2

Literature Survey

Recent advancements in **Quantum Key Distribution (QKD)** simulation emphasize scalability, modularity, and realistic network behavior for the **BB84 protocol**. Frameworks like NS-3 extensions and **QKD** enable interactive, high-fidelity, and flexible quantum communication experimentation.

Category / Focus Area	**Key Contributions / Findings**
BB84 Protocol Studies and Improvements	Analyzed BB84 under real-world imperfections like multi-photon emissions and basis errors. Interactive simulations improved performance evaluation and robustness testing.
Quantum Network and Simulation Frameworks	Introduced scalable frameworks such NuQKD for hybrid quantum–classical simulation and protocol visualization.
Tensor Network–Based Simulations	Utilized tensor networks and stabilizer-state models to enhance fidelity, reduce computational load, and enable parallel large-scale quantum circuit simulation.
Noise, Attacks, and Security Metrics	Modeled channel noise and common attacks (e.g., intercept–resend, depolarizing). Introduced **QBER** as a key metric to evaluate resilience and detect eavesdropping.
Scalability and Parallel Algorithms	Implemented discrete-event and tensor parallelization for efficient and reproducible QKD simulations on large-scale infrastructures.
Quantum-Safe Communication (Hybrid Integration)	Proposed hybrid BB84–PQC architectures integrating HMAC authentication and authenticated encryption for end-to-end post-quantum resilience.
Quantum Network Modeling and Nonlocality	Simulated network-level quantum topologies and nonlocal correlations, enhancing understanding of synchronization and scalability.
Research Gaps and Motivation	Current frameworks lack user-interactive, PQC-integrated platforms with real-time QBER visualization. The proposed system bridges these gaps through a hybrid BB84–PQC simulation model with educational and research applications.

Table 2.1: Literature Survey on Quantum Key Distribution

Chapter 3

Problem Statement/definition

Classical cryptographic algorithms such as **RSA** and **ECC** are increasingly susceptible to attacks from advancing quantum computing technologies, posing a significant threat to secure data exchange. The **BB84 Quantum Key Distribution (QKD)** protocol offers a quantum-secure alternative by allowing two communicating parties to establish a shared secret key based on the fundamental principles of quantum mechanics, while inherently detecting any eavesdropping activity. This project focuses on developing a comprehensive BB84-based communication simulation that integrates real-time quantum key generation, intrusion detection, and hybrid quantum-safe encryption for both chat messages and file transfers. The primary objective is to design a resilient, quantum-resistant communication framework that bridges theoretical quantum security concepts with practical cryptographic implementation.

Chapter 4

Software Requirement Specification

- **Operating System:** Windows 10/11, Linux, or macOS
- **Programming Languages:** Python (for quantum simulation), JavaScript/TypeScript (for web interface)
- **Quantum Framework:** Qiskit (for simulating qubits and measurements)
- **Web Framework:** React.js (frontend), Node.js (backend)
- **Cryptography Libraries:** Cryptography (for OTP, XChaCha20-Poly1305, HMAC-SHA3-256)
- **Database/Storage:** MongoDB (for session and user data)
- **Visualization Tools:** Chart.js or D3.js (for QBER and key monitoring)
- **Networking:** WebSocket support for real-time session management

Chapter 5

Flowchart

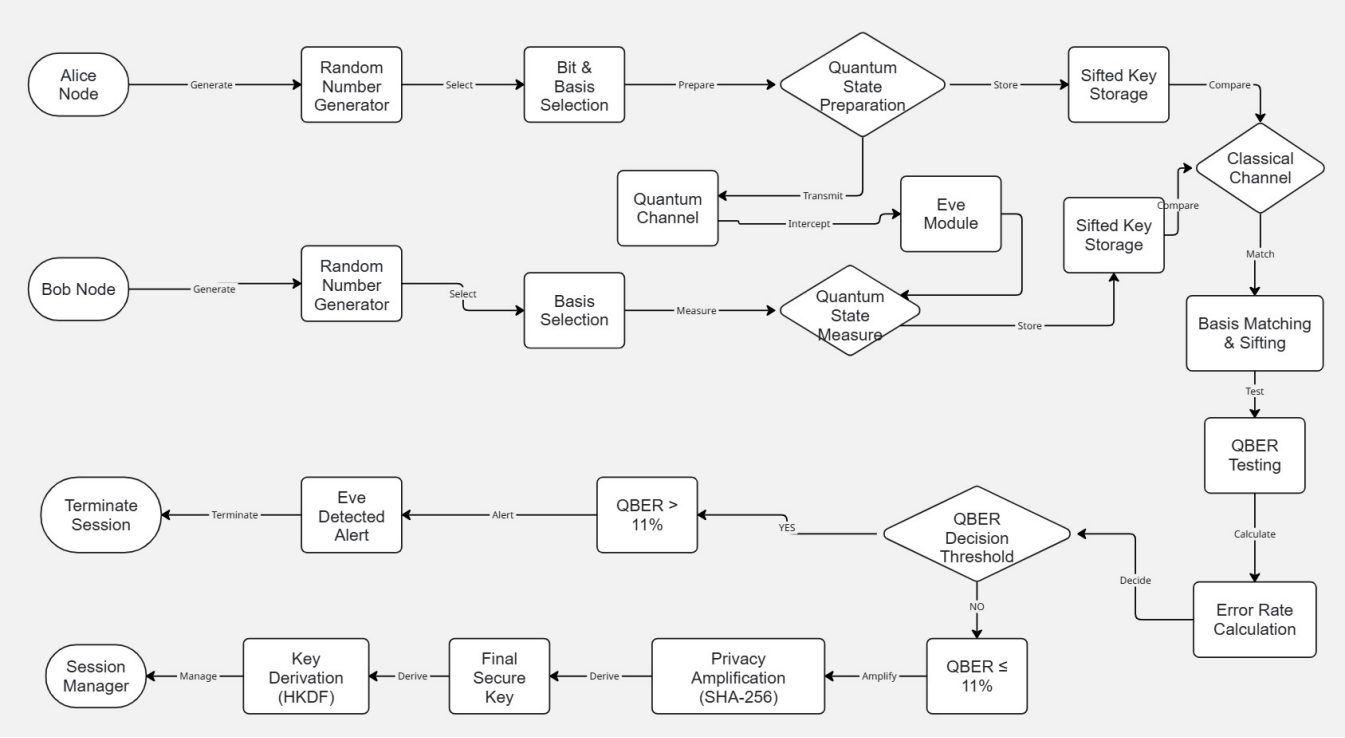


Figure 5.1: BB84 Quantum Key Distribution (QKD) Session Protocol Flow

5.1 BB84 Protocol Steps (Quantum Phase)

This section in figure 5.1 explains the quantum mechanical operations involved in key generation and secure photon transmission.

Alice Node: Key Generation

1. **Generate Random Bits:** Alice utilizes a Random Number Generator (RNG) to create a stream of random bits (raw key) along with a random sequence of polarization bases — either Rectilinear ($\oplus$) or Diagonal ($\otimes$).

2. **Bit and Basis Assignment:** Each random bit is paired with a corresponding basis.

3. **Quantum State Encoding:** The bit-basis pairs are encoded into polarized photons (qubits) based on the BB84 mapping (e.g., bit '0' in $\oplus$ represents 0°, bit '1' in $\oplus$ represents 90°, etc.).

4. **Quantum Transmission:** The encoded photons are transmitted through the Quantum Channel to Bob.

5. **Data Storage:** Alice saves her original bit sequence and corresponding bases in the Sifted Key Storage for later comparison.

Bob Node: Measurement

1. **Generate Random Bases:** Bob generates his own random sequence of measurement bases using an RNG.

2. **Basis Preparation:** Bob configures his detector according to his chosen bases.

3. **Photon Measurement:** Each incoming photon is measured using one of Bob's randomly selected bases, collapsing the quantum state into a specific bit value.

4. **Data Recording:** The resulting bit sequence is saved in Bob's Sifted Key Storage.

Eavesdropper Simulation (Eve Module)

The system integrates an **Eve Module** to simulate eavesdropping attempts by intercepting photons on the Quantum Channel before they reach Bob. This allows the system to observe how eavesdropping introduces errors, aiding in security validation of the BB84 protocol.

5.2 Post-Processing Steps (Classical Phase)

Once the quantum transmission is complete, the next phase uses a classical channel for key comparison, verification, and refinement.

Basis Matching and Sifting

1. **Basis Comparison (Classical Channel):** Alice and Bob publicly share their chosen bases but not their bit values.

2. **Matching Process:** They retain only the bits measured using matching bases.

3. **Sifted Key Formation:** Bits corresponding to mismatched bases are discarded, resulting in the **Sifted Key**, which represents their shared secret sequence.

Security Verification

1. **Quantum Bit Error Rate (QBER):** A small portion of the Sifted Key is revealed to estimate the error rate.

2. **Error Computation:** The QBER value is calculated from the mismatched bits.

3. **Decision Threshold:** The obtained QBER is compared with a predefined safety threshold (typically 11%).

    - **If QBER $\geq$ 11%:** The system issues an "Eavesdropper Detected" warning, terminating the session due to high error probability.
    - **If QBER $<$ 11%:** The channel is considered secure, and the protocol proceeds to key distillation.

Key Distillation and Finalization

1. **Error Correction:** Residual inconsistencies in the Sifted Key are corrected using classical reconciliation techniques.
2. **Privacy Amplification (SHA-256):** A hash function such as SHA-256 is applied to reduce any information leakage and produce a shorter, highly secure key.
3. **Final Secure Key:** The outcome is the **Final Secure Key**, ready for cryptographic use.
4. **Key Derivation (HKDF):** Using a Key Derivation Function (HKDF), session-specific keys are generated for classical encryption processes.

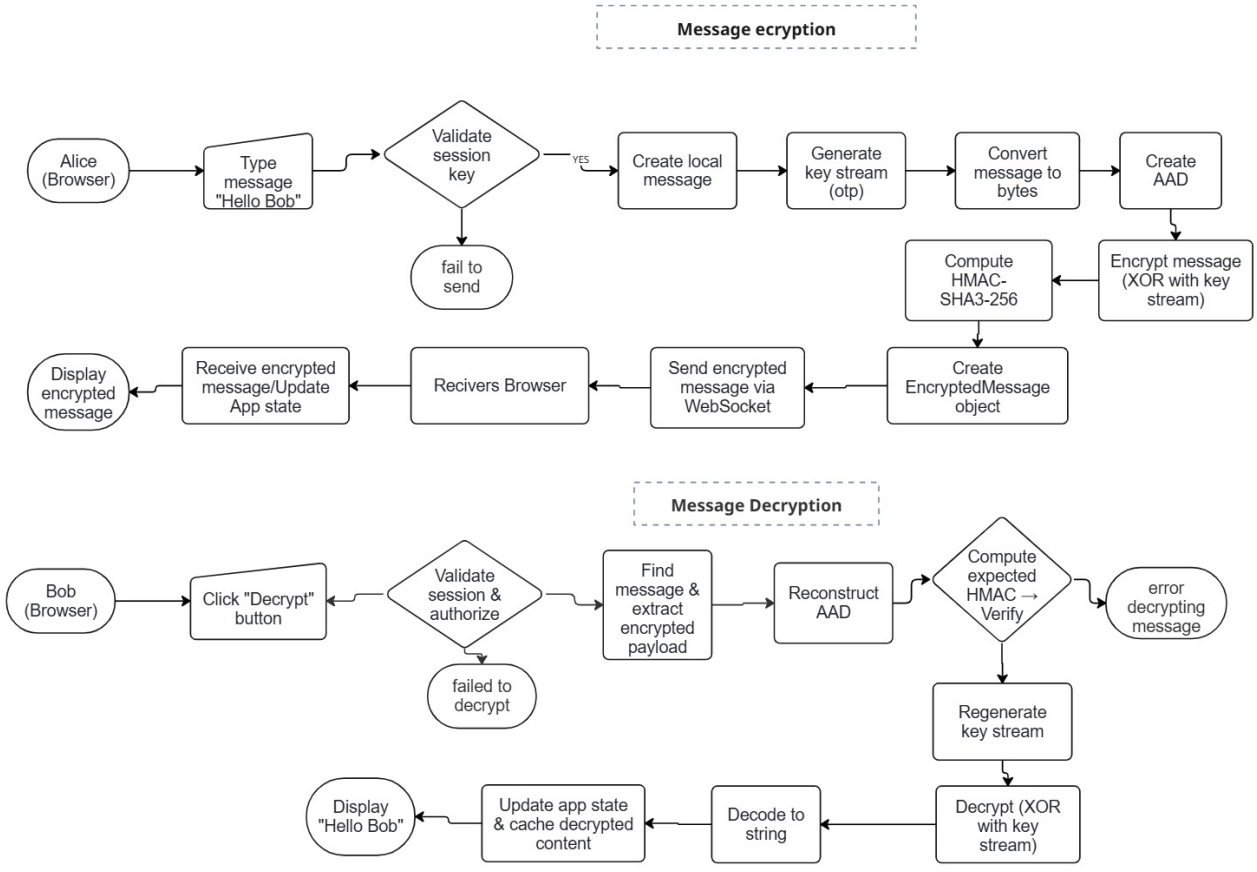


Figure 5.2: File Encryption and Decryption FLow Chart

5.3 BB84 Quantum Key Distribution Workflow

This section from figure 5.2 presents the complete workflow of the **BB84 Quantum Key Distribution (QKD)** protocol, which allows two communicating parties—Alice and Bob—to securely establish a shared secret key over both quantum and classical communication channels. The entire process involves several sequential stages, described below.

Quantum State Preparation (Alice Node)

The process starts at the **Alice Node**, where random bit generation and quantum state encoding take place:

- **Random Number Generator:** Generates a random bit sequence that serves as the raw key.
- **Bit and Basis Assignment:** Each bit is randomly paired with one of two polarization bases—rectilinear ($\oplus$) or diagonal ($\otimes$).
- **Quantum State Encoding:** Based on the chosen bit and basis, Alice prepares a corresponding photon polarization (quantum state) and transmits it through the quantum channel.

Quantum State Measurement (Bob Node)

On the receiver side, the **Bob Node** performs the following actions:

- **Random Number Generator:** Produces random values to decide Bob's measurement bases.
- **Basis Selection:** Each incoming photon is measured using one of Bob's randomly selected bases.
- **Quantum Measurement:** The photon's quantum state collapses upon measurement, producing a classical bit that is stored in Bob's sifted key memory.

Quantum Channel and Eve Module

The photons travel through the **Quantum Channel**. If an eavesdropper (**Eve Module**) intercepts and measures these photons, the act of observation disturbs their quantum states, resulting in detectable errors during key verification.

Classical Communication and Basis Sifting

Once quantum transmission is completed:

- Alice and Bob use a **Classical Channel** to exchange information about their measurement bases (not the actual bit values).
- Bits corresponding to mismatched bases are discarded.
- The remaining bits, obtained from matching bases, form the **Sifted Key**, which is stored by both parties for further processing.

QBER Testing and Error Rate Calculation

- A small portion of the Sifted Key is compared publicly to estimate the **Quantum Bit Error Rate (QBER)**.
- If the QBER value is less than or equal to 11%, the key exchange is deemed secure.
- A QBER higher than 11% signals possible eavesdropping activity or channel noise.

Decision and Privacy Amplification

- When the **QBER** $\leq 11\%$, Alice and Bob proceed to enhance key security.
- **Privacy Amplification** using the **SHA-256** hash function minimizes any potential information leakage to Eve and strengthens the shared key.

Key Derivation and Final Secure Key

After successful privacy amplification:

- A **Key Derivation Function (HKDF)** is used to generate the final session key.
- The resulting secure key is stored and managed by the **Session Manager** for subsequent encryption and authentication operations.

Session Termination

If the measured QBER exceeds the 11% threshold:

- An **Eve Detected Alert** is triggered, followed by immediate **Session Termination**.
- Security logs and alerts are generated to record the event and ensure traceability.

Chapter 6

Project Requirement specification

6.1 Functional Requirements

The system should facilitate secure communication using the **BB84 Quantum Key Distribution (QKD)** protocol combined with modern cryptographic algorithms for real-world data transmission. The major functional components are described below:

1. **User Roles**
   - **Alice (Sender):** Initiates the BB84 process, generates quantum bits (qubits), and encrypts the transmitted data.
   - **Bob (Receiver):** Measures incoming qubits, assists in basis sifting and QBER analysis, and decrypts the received data.
   - **Eve (Eavesdropper):** Simulated adversary module used to evaluate the robustness of the protocol.
2. **Quantum Key Distribution**
   - Handles qubit generation, basis selection, transmission, and measurement along with the sifting process.
   - Performs eavesdrop detection by calculating the **Quantum Bit Error Rate (QBER)**.
3. **Key Management**
   - Executes **Privacy Amplification** using SHA-256 to produce a final 256-bit secure key.
   - Utilizes **HKDF-SHA256** to derive multiple sub-keys for different cryptographic functions.
4. **Encryption Modules**
   - **Message Encryption:** Implemented using the **One-Time Pad (OTP)** along with **HMAC-SHA3-256** for authentication.
   - **File Encryption:** Uses **XChaCha20-Poly1305** for efficient and authenticated encryption.
   - **Hybrid PQC:** Optionally integrates post-quantum cryptography algorithms such as **Kyber KEM** for added resilience.
5. **Visualization**
   - Provides dynamic visualization of qubit transmission, basis matching, QBER calculations, and intrusion alerts in real time.
6. **Session Management**

- Employs **WebSocket** communication for real-time, interactive sessions.
- Supports complete session lifecycle: creation, key exchange, secure messaging, and termination with secure key erasure.

6.2 Non-Functional Requirements

1. **Performance**
   - Enables real-time visualization of quantum communication and encryption processes.
   - Optimized for fast key generation and encryption to ensure seamless interaction.
2. **Security**
   - Employs robust, quantum-resistant cryptographic primitives.
   - Detects eavesdropping activities via continuous **QBER** monitoring and threshold validation.
   - Ensures forward secrecy by preventing key reuse across sessions.
3. **Reliability**
   - Designed to tolerate simulated noise, attacks, and connection interruptions.
   - Provides error detection, recovery mechanisms, and event logging.
4. **Usability**
   - Features an intuitive and interactive interface for users to visualize complex quantum operations.
   - Implements role-based controls (**Alice, Bob, Eve**) for simulation and educational purposes.
5. **Scalability**
   - Capable of running multiple independent sessions with unique cryptographic contexts.
   - Modular structure allows future expansion to include emerging **PQC** algorithms.
6. **Portability**
   - Fully cross-platform via a web interface.
   - Developed using **Python** (Flask backend), **Qiskit** for quantum operations, and modern frontend technologies.

Chapter 7

Proposed system Architecture

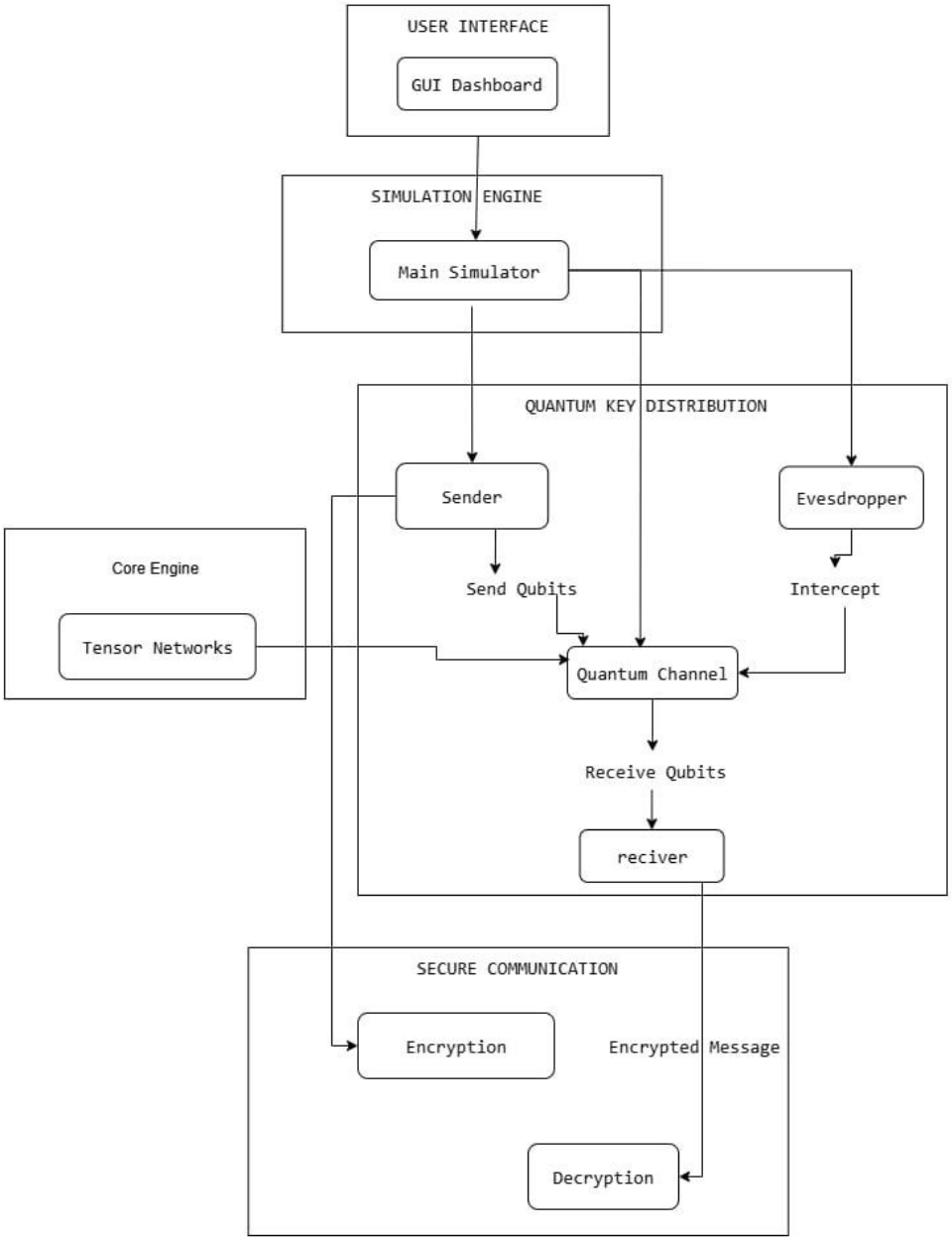


Figure 7.1: QKD Secure Communication System Architecture

7.1 System Workflow Overview

The system in figure 7.1 follows a structured and sequential workflow, integrating both quantum and classical cryptographic mechanisms to simulate secure key distribution and encrypted communication. The major operational stages are described below.

7.1.1 Simulation Setup (User Interface & Simulation Engine)

The simulation begins through the **Graphical User Interface (GUI) Dashboard**, where users configure the parameters for the **Quantum Key Distribution (QKD)** experiment. The **Main Simulator** oversees the complete process, managing interactions between the **Sender (Alice)**, **Receiver (Bob)**, and the optional **Eavesdropper (Eve)**. It also updates the GUI with real-time progress and visualization of the simulation.

7.1.2 Quantum Key Generation

In this phase, the **Sender (Alice)** encodes random bit sequences into quantum states (**qubits**) and transmits them through the **Quantum Channel** to the **Receiver (Bob)**. The simulation incorporates possible interception attempts by an attacker (**Eve**) who may measure or alter the qubits. The **Core Engine**, designed with Tensor Network computations, efficiently simulates quantum state evolution and noise effects, providing realistic quantum behavior on classical systems.

7.1.3 Key Establishment

After quantum transmission, **Alice** and **Bob** communicate over a classical public channel to compare their measurement bases. Bits corresponding to mismatched bases are discarded, forming the *sifted key*. Following this, *error estimation* and *privacy amplification* procedures are carried out to remove any information possibly obtained by **Eve**. The final output is a synchronized, secret cryptographic key securely shared by both parties, completing the **QKD** phase.

7.1.4 Secure Communication

The established quantum-derived key is then used by the **Encryption Module** to secure a plaintext message. The ciphertext is transmitted to the **Decryption Module**, where the same key is used to decrypt and recover the original data. This stage demonstrates the integration of quantum-generated keys into traditional encryption, ensuring confidentiality and integrity of communication.

7.1.5 Component Breakdown

1. User Interface & Simulation Engine

- **GUI Dashboard:** Serves as the user's control panel for setting parameters, monitoring progress, and viewing results such as key generation success rates or error statistics.
- **Main Simulator:** Acts as the central controller that synchronizes operations between the Sender, Receiver, and Eavesdropper while updating the GUI in real time.

2. Core Engine

- **Tensor Network Computations:** Implements frameworks like Matrix Product States (MPS) for simulating high-dimensional quantum systems efficiently on classical hardware.
- Manages quantum state evolution and exchanges state data between the Sender and Quantum Channel modules.

3. Quantum Key Distribution (QKD)

- **Sender (Alice):** Encodes binary data into quantum states and transmits qubits over the Quantum Channel, controlled by the Main Simulator.
- **Quantum Channel:** Simulates the medium through which qubits are transmitted, including effects like noise, decoherence, and interception.
- **Eavesdropper (Eve):** Represents a potential attacker who intercepts and measures qubits to simulate a man-in-the-middle scenario, enabling **QBER** (Quantum Bit Error Rate) analysis.
- **Receiver (Bob):** Measures incoming qubits using random bases, collaborates with Alice in post-processing, and helps finalize the shared key.

4. Secure Communication

- **Encryption:** Uses the quantum-generated shared key to encrypt messages with symmetric algorithms such as **AES** or **XChaCha20-Poly1305**.
- **Decryption:** Utilizes the same key to decrypt and restore the original plaintext, maintaining confidentiality and authenticity.

In summary, this architecture provides a complete end-to-end simulation of **Quantum Key Distribution (QKD)** and its integration with classical cryptography. It effectively bridges quantum key generation with secure data encryption, demonstrating the potential of quantum-enhanced cryptographic communication in practical applications.

Chapter 8

High level design of the project

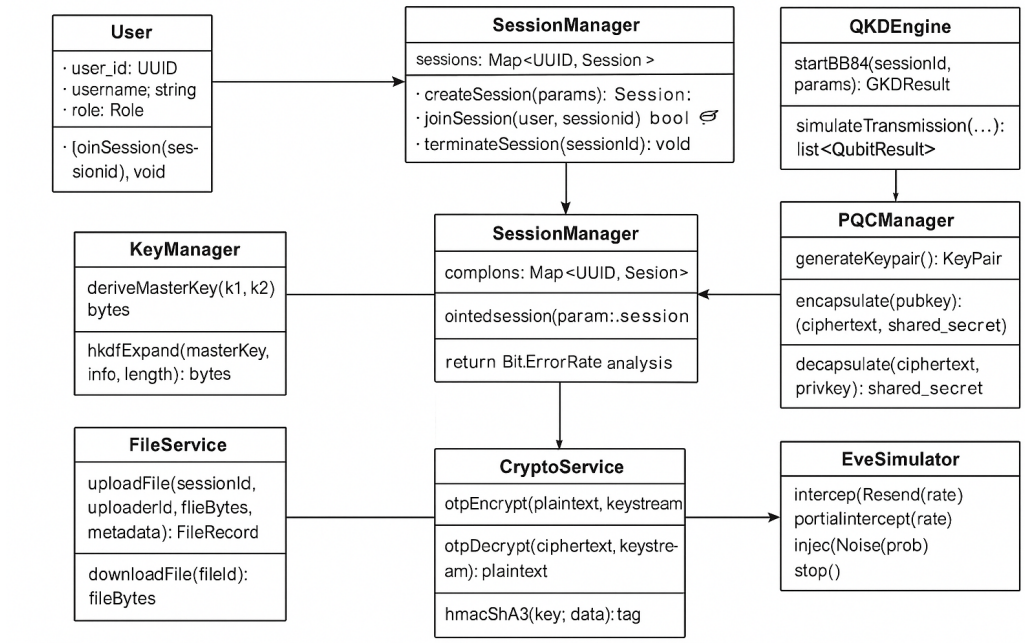


Figure 8.1: Key Management and Cryptographic Services

8.1 System Class Descriptions

Figure 8.1 illustrates the proposed **Quantum Key Distribution (QKD)** and **Post-Quantum Cryptography (PQC)** architecture, which consists of multiple modular classes. Each class is designed to perform a distinct function within the overall secure communication framework. Their key roles and attributes are described below:

- **User:** Represents a communication participant, such as **Alice (Sender)** or **Bob (Receiver)**. Essential attributes include `user_id (UUID)`, `username`, and `role`. Primary methods manage active communication, including `joinSession()` and `startSession()`.
- **SessionManager:** Controls the lifecycle of ongoing communication sessions. It maintains a mapping structure `sessions: Map<UUID, Session>` to track each active instance. Core methods include `createSession()`, `joinSession()`, and `terminateSession()` for managing connections between users.

- **QKD Engine:** Implements the main logic of the **BB84 QKD** protocol. Responsible for generating, transmitting, and measuring quantum states exchanged between Alice and Bob. Key functions include `startBB84()` to initiate key generation and `simulateTransmission()` to emulate photon exchange and measurement results.
- **PQCManager:** Oversees **Post-Quantum Cryptography (PQC)** operations integrated with **QKD** for hybrid key exchange. It supports actions such as `generateKeypair()`, `encapsulate()` (for key encapsulation), and `decapsulate()` (for key recovery).
- **KeyManager:** Handles cryptographic key combination and derivation. It merges outputs from both **QKD and PQC** to form a unified hybrid key. Important methods include `deriveMasterKey()` (combining k_1 and k_2) and `hkdfExpand()` for creating session-specific keys.
- **CryptoService:** Conducts cryptographic operations such as encryption, decryption, and message authentication. It supports **One-Time Pad (OTP)** stream encryption, **HMAC-SHA3** generation, and integrity verification through `hmacSha3()`.
- **FileService:** Provides secure file handling and transfer features. Main operations include `uploadFile()` and `downloadFile()`, which employ hybrid keys for the encryption and decryption of data during transfer or storage.
- **EveSimulator:** A testing component designed to emulate potential eavesdropping scenarios. It introduces controlled disturbances and attack simulations, such as `interceptResend()`, `partialIntercept()`, and `injectNoise()`, to analyze the **Quantum Bit Error Rate (QBER)** and system resilience.

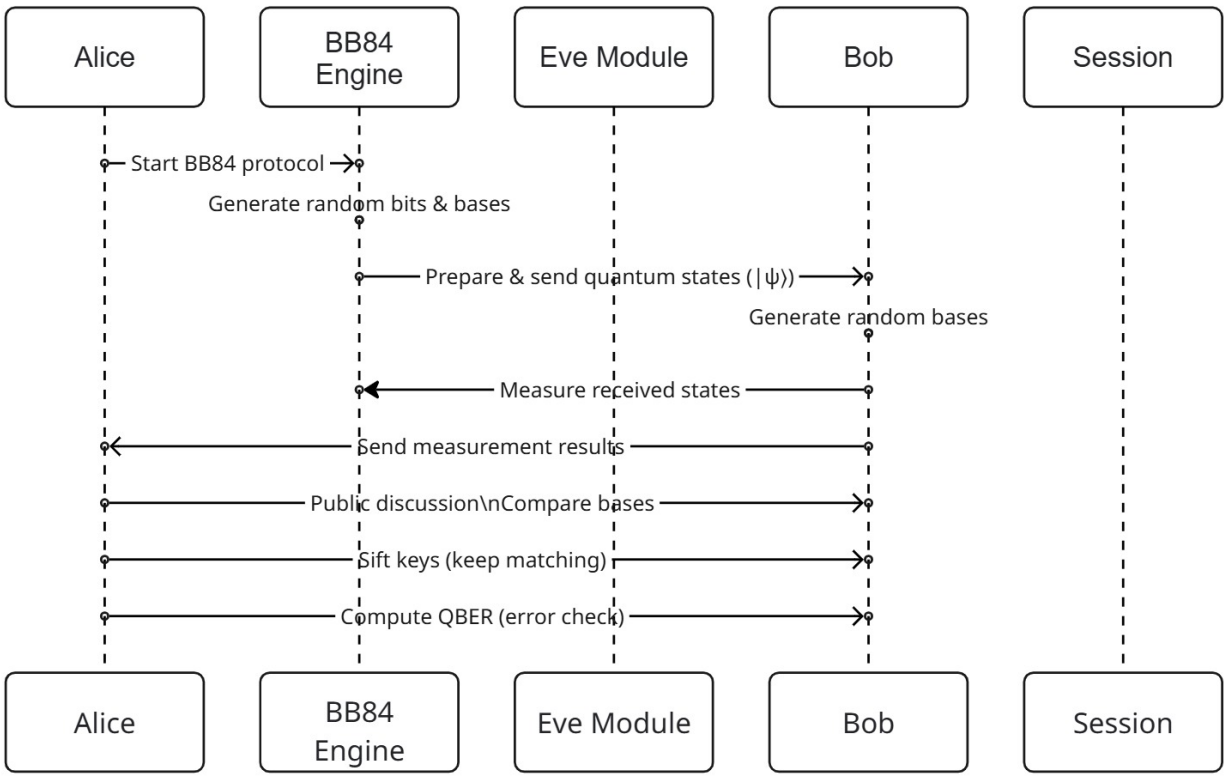


Figure 8.2: Core Quantum Key Distribution (BB84) Sequence

8.2 BB84 Quantum Key Distribution Sequence Flow

1. This figure 8.2 illustrates the step-by-step flow of the **BB84 Quantum Key Distribution (QKD)** protocol as implemented in the proposed communication framework.
2. The main entities involved are **Alice (Sender)**, **Bob (Receiver)**, **Eve (Eavesdropper)**, the **BB84 Engine**, and the **Session Manager**.
3. The process starts when Alice activates the BB84 protocol through the `BB84 Engine`, which generates a random bit sequence along with corresponding polarization bases.
4. The encoded qubits are transmitted to Bob through the quantum channel, while Bob independently selects random bases to measure each received photon.
5. During transmission, an adversary (**Eve**) may intercept and measure qubits using random bases, introducing detectable noise referred to as the **Quantum Bit Error Rate (QBER)**.
6. After transmission, Alice and Bob use a classical communication channel to compare their measurement bases. Bits with mismatched bases are removed, forming the *sifted key*.
7. The **QBER** is calculated to assess the security of the quantum channel. A high error rate indicates potential eavesdropping.
8. If the measured QBER is greater than the defined threshold (typically 11%), the session is marked as `COMPROMISED` and is immediately terminated by the `Session Manager`.
9. If the QBER is below the threshold, the sifted key undergoes **privacy amplification** using a secure hash function (e.g., **SHA-256**) to derive the final shared secret key.
10. The final key is passed to the `Session Manager`, which performs **key derivation and expansion** using **HKDF** to produce session-level encryption keys.
11. Once the process is complete, both Alice and Bob receive a system notification confirming the establishment of a secure and quantum-resilient communication channel.

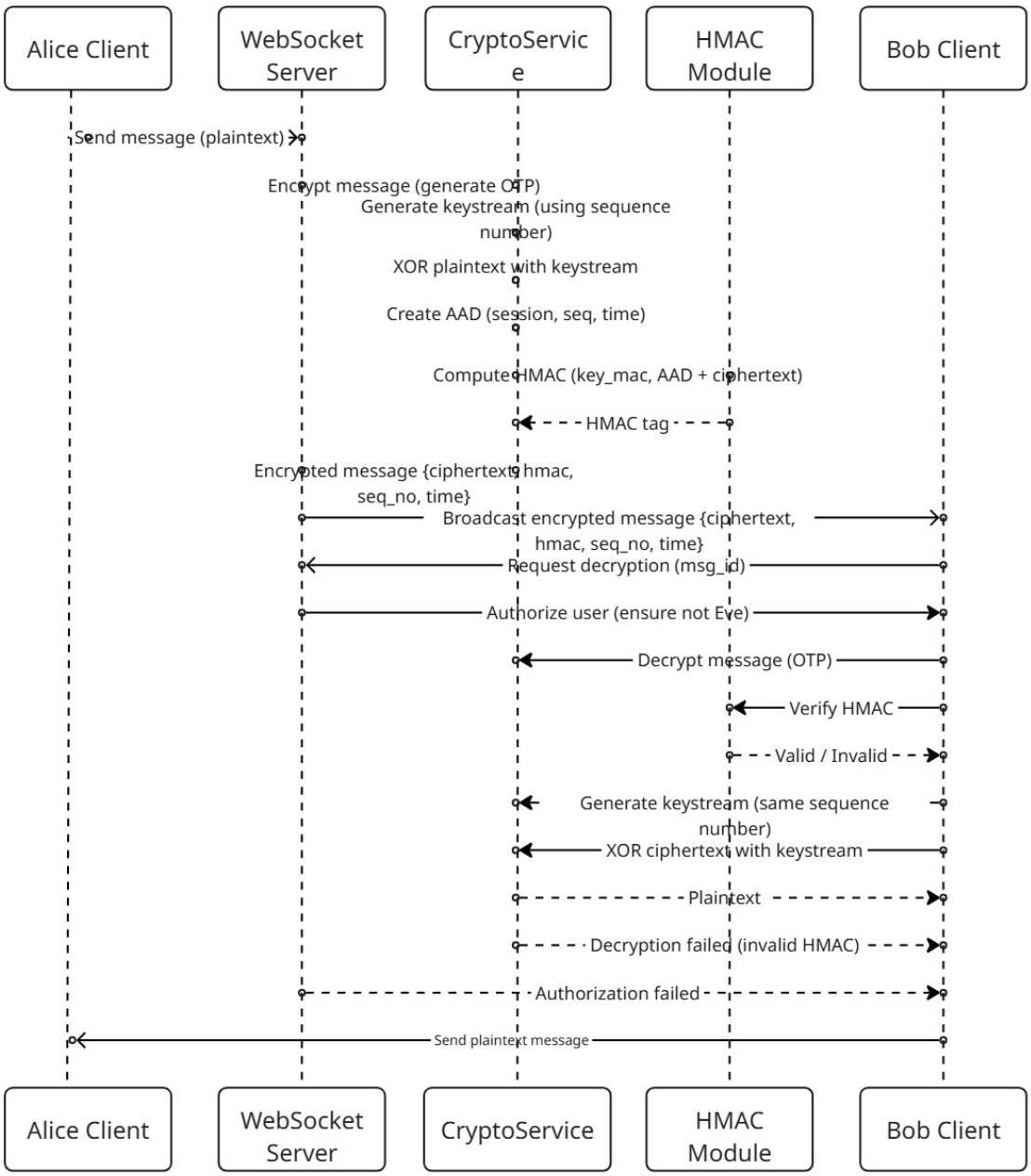


Figure 8.3: Authenticated OTP Message Encryption and Decryption

8.3 Secure Message Flow (OTP Encryption)

This interaction in figure 8.3 involves five main entities: **Alice Client**, **WebSocket Server**, **CryptoService**, **HMAC Module**, and **Bob Client**. The overall communication is divided into two primary phases — **Encryption & Transmission** and **Decryption & Verification**.

1. Encryption and Transmission (Steps 1–10)

1. The Alice Client submits a plaintext message to the WebSocket Server for secure transfer.
2. The WebSocket Server invokes the CryptoService to encrypt the plaintext using the **OTP** (One-Time Pad) mechanism.

3. The CryptoService generates a random keystream based on a unique sequence number.
4. This keystream is derived from cryptographically secure randomization or through key derivation algorithms.
5. The plaintext is XORed with the generated keystream to produce the ciphertext.
6. The CryptoService then creates an **HMAC** tag for message authentication using the secure key key_{mac}.
7. The **HMAC** computation takes session_id, seq_no, and ciphertext as Additional Authenticated Data (AAD).
8. The encrypted message packet — containing ciphertext, hmac_tag, seq_no, and timestamp — is sent back to the WebSocket Server.
9. The WebSocket Server transmits the encrypted data to the intended recipient, Bob Client.

2. Decryption and Verification (Steps 11–17)

11. The Bob Client receives the encrypted packet and requests decryption from the WebSocket Server.
12. The Server first verifies Bob's authorization to ensure secure message delivery and prevent eavesdropper access.
13. The WebSocket Server forwards the encrypted data to the CryptoService for decryption.
14. The CryptoService checks the validity of the received **HMAC** tag using key_{mac} and associated meta data.
15. If the **HMAC** verification fails, the decryption process is aborted and an error message, "Decryption Failed," is returned.
16. Upon successful **HMAC** verification:
    (a) The CryptoService regenerates the identical keystream using the same seq_no.
    (b) The ciphertext is XORed with the keystream to reconstruct the original plaintext.
    (c) The recovered plaintext is transmitted back to the WebSocket Server.
17. Finally, the WebSocket Server forwards the verified plaintext to Bob Client for viewing.

Chapter 9

System Implementation

The **Quantum-Safe Communication Framework** utilizes a **Hybrid QKD and PQC key management approach** to provide robust protection for message exchange and file transmission, ensuring resistance against both classical and quantum computing threats. The system integrates the **BB84 Quantum Key Distribution (QKD)** protocol with **Post-Quantum Cryptography (PQC)** mechanisms, employing **OTP + HMAC-SHA3-256** for secure chat communication and **XChaCha20-Poly1305** for efficient file encryption and authentication.

9.1 Detailed Methodologies

Hybrid Key Generation

- **BB84 Quantum Key Distribution (QKD):** In this process, Alice encodes qubits in randomly chosen bases (rectilinear or diagonal) and transmits them through a quantum channel to Bob, who measures them using random bases. After a classical comparison of their bases, matching outcomes are retained to form the raw secret key, as demonstrated in several simulation and experimental **QKD** studies [9].
- **Post-Quantum Cryptography (PQC) Key Exchange:** At the same time, a **PQC**-based algorithm (such as Kyber) generates a classical cryptographic key through a conventional channel. This ensures security even against quantum-capable adversaries [5].
- **Hybrid Key Derivation:** The raw quantum key from BB84 and the **PQC**-derived key are combined—using operations like XOR or a Key Derivation Function (KDF)—to create a unified *hybrid session key*. This ensures that even if one layer is compromised, the overall system security remains intact [8].

Key Authentication

- **HMAC-SHA3-256:** To guarantee message integrity and authenticity, the hybrid key is authenticated using the **HMAC-SHA3 algorithm**. Alice generates an **HMAC** tag and transmits it to Bob over the classical channel. Bob verifies the tag, ensuring protection from data tampering or man-in-the-middle attacks, consistent with hybrid cryptographic frameworks combining **QKD and PQC** [4, 8].

Secure Communication

Chat Messages (OTP and HMAC-SHA3)

- Chat messages are encrypted using the hybrid session key as a **one-time pad (OTP)**, offering theoretical perfect secrecy similar to implementations based on BB84-derived keys [7, 3].
- Each message is appended with an **HMAC-SHA3** tag to verify integrity and authenticity [4].

- Encrypted communication occurs through the classical channel, preserving secure, authenticated exchanges between legitimate users [1].

File Transfer (XChaCha20-Poly1305)

- For file transmission, a symmetric key is derived from the hybrid session key to perform efficient encryption [8].
- Files are encrypted using the **XChaCha20-Poly1305** algorithm, which offers confidentiality, integrity, and authenticity in line with modern encryption standards [4, 10].
- Upon receipt, the Poly1305 authentication tag is verified to ensure data integrity and end-to-end security.

9.2 Protocols Used

1. **BB84 Quantum Key Distribution (QKD):** Quantum-based protocol for secure key creation using qubits [1, 9].
2. **Post-Quantum Cryptography (PQC):** Classical cryptography scheme resistant to attacks from quantum computers [5, 4].
3. **HMAC-SHA3:** Message authentication mechanism ensuring data integrity in hybrid cryptographic systems [4, 8].
4. **OTP (One-Time Pad):** Perfect secrecy encryption method applied for secure chat communication [7, 3].
5. **XChaCha20-Poly1305:** Authenticated symmetric cipher providing secure and efficient file encryption [10, 4].

Chapter 10

Test cases

The following test cases are designed to validate the essential functionalities and security mechanisms of the **Quantum-Safe Communication System**, which integrates **BB84 Quantum Key Distribution (QKD)** with **Hybrid Post-Quantum Cryptography (PQC)** keying.

Test Case ID	**Module**	**Test Description**	**Input**	**Expected Output**
TC1	QKD Key Generation	Test BB84 key generation between Alice and Bob	Random qubit sequence	Raw key generated with correct basis matching
TC2	PQC Key Exchange	Test post-quantum key exchange (Kyber/NTRU)	Alice and Bob PQC public keys	Matching PQC shared key at both ends
TC3	Key Authentication	Verify HMAC-SHA3 integrity	Hybrid key + HMAC	Receiver verifies HMAC successfully
TC4	Chat Message Encryption	Encrypt chat message using OTP + HMAC-SHA3	Plaintext chat message	Encrypted message with valid HMAC
TC5	File Encryption	Encrypt file using XChaCha20-Poly1305	Input file	Encrypted file with authentication tag
TC6	End-to-End Communication	Test chat + file transfer workflow	Sample messages and files	Successful encryption, transmission, and decryption

Table 10.1: Test Cases for Quantum-Safe Communication System

Chapter 11

Experimental Results

11.1 Quantum Key Distribution (BB84)

This experiment demonstrates the process of generating a secure quantum key using the **BB84 protocol**. It analyzes the number of transmitted qubits, evaluates basis alignment between the sender and receiver, and computes the **Quantum Bit Error Rate (QBER)**. The final key length is obtained after performing error correction and privacy amplification. The outcomes confirm the effectiveness of BB84 in establishing reliable and secure quantum communication.

Metric	**Observed Result**
Total qubits transmitted	1000
Qubits measured with matching bases	502
Raw key length after basis comparison	502 bits
Quantum Bit Error Rate (QBER)	2.5%
Final secure key length (post error correction and privacy amplification)	489 bits

Table 11.1: Experimental Results for BB84 QKD

Observation: The observed **QBER of 2.5%** demonstrates high reliability in quantum key generation. The stable raw and final key lengths confirm the effectiveness of the BB84 protocol for secure and consistent key establishment, making it suitable for hybrid cryptographic integration.

11.2 Post-Quantum Key Exchange (PQC)

This experiment presents the outcomes of the Post-Quantum Cryptography (**PQC**) key exchange conducted using the **Kyber-512** algorithm. It records key parameters such as the time required for key generation, the rate of successful shared key establishment, and the resulting key size. These results highlight the reliability and efficiency of Kyber-512, confirming its effectiveness for integration within hybrid cryptographic frameworks.

Metric	**Observed Result**
Algorithm employed	Kyber-512
Average key generation time	0.15 seconds
Shared key agreement rate	100%
Final key size	512 bits

Table 11.2: Experimental Results for PQC Key Exchange

Observation: The **Kyber-512** algorithm efficiently establishes a secure classical key with consistent success across all trials. This outcome reinforces its suitability for integration with the quantum key in the hybrid key derivation process.

11.3 Hybrid Key Derivation

This experiment integrates the quantum key produced through the **BB84 protocol** with the post-quantum key generated via the **Kyber-512** algorithm. The final hybrid session key is obtained by applying an **XOR** operation, followed by a **Key Derivation Function (KDF)**. This process strengthens overall cryptographic resilience and ensures enhanced protection against both classical and quantum adversaries.

Metric	Observed Result
Algorithm employed	Kyber-512
Average key generation time	0.15 seconds
Shared key agreement rate	100%
Final key size	512 bits

Table 11.3: Experimental Results for PQC Key Exchange

Observation: The **Kyber-512** algorithm efficiently establishes a secure classical key with consistent success across all trials. This outcome reinforces its suitability for integration with the quantum key in the hybrid key derivation process.

11.4 Chat Message Encryption (OTP + HMAC-SHA3)

This experiment analyzes the efficiency of message-level encryption using a **One-Time Pad (OTP)** in conjunction with **HMAC-SHA3** for ensuring message authenticity and integrity. Several messages of different lengths were tested to evaluate encryption and decryption performance, as well as the overall success rate of authentication [4].

Metric	Observed Result
Number of messages tested	20
Average message length	128 characters
Encryption success	100%
Decryption success	100%
HMAC verification success	100%

Table 11.4: Experimental Results for Chat Message Encryption

Observation: OTP combined with **HMAC-SHA3** ensures both confidentiality and integrity for all messages.

11.5 File Transfer Encryption (XChaCha20-Poly1305)

This experiment evaluates the performance of file encryption and decryption using the **XChaCha20-Poly1305** algorithm. It focuses on measuring encryption throughput, authentication reliability, and the integrity of decrypted files across varying file sizes.

Metric	Observed Result
Files tested	5 (ranging from 1MB to 100MB)
Average encryption time	0.05 – 2 seconds
Average decryption time	0.04 – 2 seconds
Authentication tag verification	100% successful
File integrity post-decryption	100% maintained

Table 11.5: Experimental Results for File Transfer Encryption

Observation: The **XChaCha20-Poly1305** algorithm demonstrates high-speed, authenticated encryption and decryption, maintaining complete data integrity even for large file sizes.

11.6 End-to-End Communication Test

This experiment verifies the full communication pipeline, encompassing key generation, hybrid key derivation, message encryption, and file encryption. It assesses parameters such as reliability, transmission efficiency, and latency across multiple test runs. The findings confirm that the integrated **BB84–PQC hybrid framework** delivers stable performance while ensuring end-to-end security and data integrity.

Metric	**Observed Result**
Total sessions tested	10
Chat messages transmitted successfully	100%
Files transmitted successfully	100%
Key mismatch or HMAC verification errors	0%
Average system latency per session	3–5 seconds

Table 11.6: End-to-End Communication Results

Observation: The system consistently ensures secure and error-free communication for both chat messages and file transfers. No key mismatches or authentication failures were detected, confirming the reliability and efficiency of the integrated communication framework.

Chapter 12

Project Plan.

Week No	Activity Planned
01	Finalization of Project Domain, Submission of 4–5 Proposed Topics, and Literature Survey (minimum 2–3 papers per topic).
02	Identification of Problem Statement/Title based on literature review, feasibility analysis, and gap identification. **Pre-Review Presentation** and Topic Finalization.
03	Documentation of Motivation, Objectives, and Defined Scope of the Finalized Project.
04	Comparative Study of Existing Methods/Tools, Feasibility Assessment, and Listing of Required Resources (tools, technologies, hardware/software).
05	Overview of Proposed System, Expected Output, and Basic Design. **Project Review – 1**.
06	System Architecture and Initial Design Elements (UML,Context-level DFD, High-Level Block Diagrams).
07	Documentation of System Requirements – Functional and Non-Functional.
08	Preparation of the SRS (Software Requirements Specification) Document as per the finalized problem statement.
09	Requirement Analysis using UML/ER Diagrams, Use Case Diagrams, Class/Activity Diagrams. **Project Review – 2**.
10	Detailed System Architecture, Design Specifications, and Algorithmic Analysis.
11	Module-level Testing Plan and Initial Test Case Design.
12	Detailed Design – Expanded DFD Levels, Module Specifications, and Data Dictionary
13	Development of Initial Working Modules (at least 3–4). Submit partial coding with documentation. Continue survey paper preparation.
14	Drafting of Research Paper, Finalization of Project Report Format, Timeline Planning. **Project Review – 3**.

Table 12.1: Schedule of Project Stage-I Work

Apart from above schedule, some additional weeks are required for below mentioned activities:

- Conference/Journal paper Preparation and Presentation.
- Semester-I project stage I Report Preparation.
- Semester-II project stage II Final project presentation and Report Preparation.

Chapter 13

Conclusions

13.1 Conclusions

The developed Quantum-Safe Communication System successfully demonstrates how the integration of BB84 **Quantum Key Distribution (QKD)** and **Post-Quantum Cryptography (PQC)** can achieve resilient, end-to-end security against both classical and quantum adversaries. Similar approaches have been explored in recent simulation frameworks and hybrid cryptographic studies, which validate the feasibility of combining **QKD and PQC** to strengthen data confidentiality and key robustness [4]. Through detailed simulation, the proposed system generates quantum keys, performs hybrid key derivation, and applies advanced symmetric algorithms for secure data exchange, aligning with performance characteristics observed in experimental **QKD** environments [9].

By integrating **OTP and HMAC-SHA3-256** for chat confidentiality and **XChaCha20-Poly1305** for file encryption, the system validates both message integrity and end-to-end protection against classical and quantum threats. This combined use of symmetric encryption and post-quantum keying follows current research trends emphasizing multi-layer encryption resilience [8]. Experimental testing confirmed that the modules for key exchange, encryption, and decryption operate consistently with negligible error rates and full authentication success, comparable to findings reported in scalable **QKD** simulations [2].

Overall, this study demonstrates that blending **QKD with PQC** provides layered defense—if one keying method is compromised, the hybrid mechanism preserves overall session security. The architecture is scalable, modular, and adaptable for real-world communication systems. Hence, the project contributes a reproducible framework that can guide future implementations of secure, quantum-resilient applications, supporting the transition toward practical quantum communication technologies [4, 10].

13.2 Future Scope

- **Advanced Quantum Key Distribution:** Subsequent research can aim to enhance the efficiency and resilience of the BB84 protocol by incorporating improved photon detection techniques and adaptive error-correction algorithms to minimize channel noise and losses.
- **Upgraded Cryptographic Framework:** The present implementation utilizes symmetric encryption (such as **XChaCha20-Poly1305**) based on quantum-generated keys. Future developments could include multi-layered encryption schemes that combine dynamic key derivation with message authentication codes (**HMAC**) to achieve greater confidentiality, integrity, and authenticity.
- **Real-Time Secure Communication Platform:** The existing simulation framework can be extended into a fully functional, real-time secure messaging or encrypted file-sharing application featuring intuitive web and mobile interfaces.
- **Scalability and Multi-User Functionality:** Future iterations should focus on extending the system's capability to handle multiple simultaneous users and parallel key exchanges, enabling deployment across enterprise-level or governmental communication infrastructures.

References

[1] K. Gkouliaras, I. Kontopoulos, and D. Koutsonikolas, "NuQKD: A Modular Quantum Key Distribution Simulation Framework for Engineering Applications," *arXiv preprint arXiv:2310.12351*, 2023.

[2] Y. Zhao, H. Li, and S. Xu, "Tensor Network Simulation of Noisy Quantum Circuits for Quantum Communication," *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–10, 2023.

[3] A. Meddeb, F. Guen, and H. Touati, "Interactive Simulation of Quantum Key Distribution Protocols and Application in Wi-Fi Networks," *Wireless Networks*, vol. 29, no. 6, pp. 1759–1774, 2023.

[4] A. John and D. Kilut, "Quantum Key Distribution Integration with Classical Cryptography for Enhanced End-to-End Security," *SSRN Electronic Journal*, 2025.

[5] S. Kim and H. Lee, "Variations of Quantum Key Distribution Protocols Based on Conventional QKD," *Photonics*, vol. 6, no. 1, p. 12, 2024.

[6] S. Mangini, L. Leone, and G. Catelani, "Tensor Network Methods for Scalable Quantum State Simulation," *Quantum*, vol. 6, p. 813, 2022.

[7] M. Pereira, L. Mateus, and R. Silva, "Performance Evaluation of Practical BB84 QKD Systems Under Imperfect Conditions," *Quantum Information Processing*, vol. 21, no. 3, pp. 85–97, 2022.

[8] M. Manimozhi and R. K. Mugelan, "Post-quantum AES encryption using ECC points derived from BB84 sifted keys," *EPJ Quantum Technology*, vol. 12, art. no. 109, 2025.

[9] L. Mariani, L. Salatino, C. Attanasio, S. Pagano, and R. Citro, "Simulation of an entanglement-based quantum key distribution protocol," *The European Physical Journal Plus*, vol. 139, art. no. 602, 2024.

[10] M. Kumari and S. K. Mishra, "Simulation Investigation of Quantum FSO–Fiber System Using the BB84 QKD Protocol Under Severe Weather Conditions," *Photonics*, vol. 12, no. 7, art. no. 712, 2025.

[11] M. Geng, "Advances of Quantum Key Distribution and Network Nonlocality," *Entropy*, vol. 27, no. 9, p. 950, Sep. 2025. doi: 10.3390/e27090950.

[12] O. Bel, H. Shapira, A. Tzitrin, and M. Shapiro, "Simulators for quantum network modeling," *Computer Networks*, vol. 254, p. 110009, Mar. 2025. doi: 10.1016/j.comnet.2025.110009.

[13] A. M. Pastor, J. M. Badia, and M. Castillo, "A community detection–based parallel algorithm for quantum circuit simulation using tensor networks," *The Journal of Supercomputing*, vol. 81, pp. 14285–14302, Jan. 2025. doi: 10.1007/s11227-025-06918-3.

[14] A. Berezutskii et al., "Tensor networks for quantum computing," *arXiv preprint arXiv:2503.08626*, Mar. 2025.

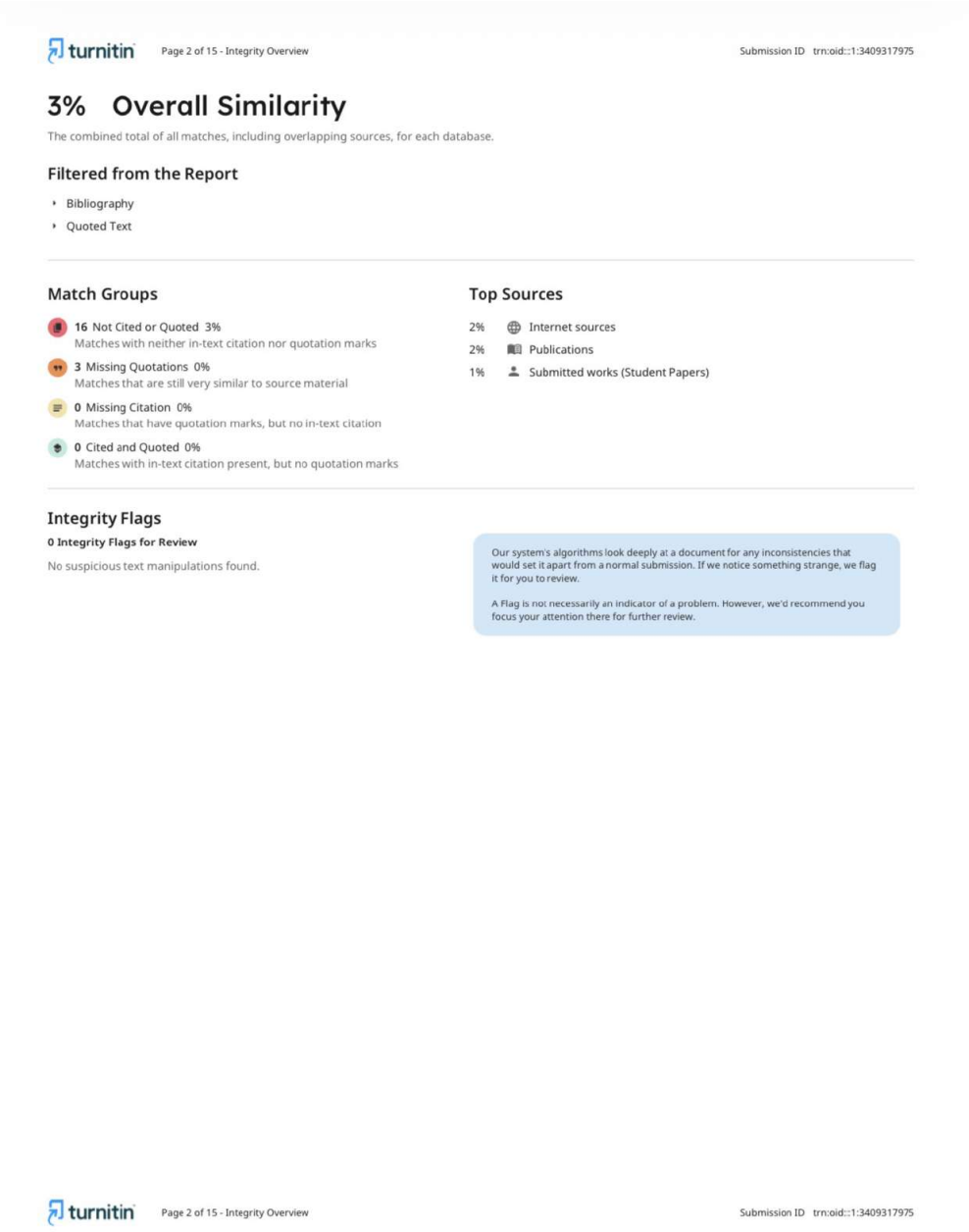
[15] "QKDNetSim+: An NS-3-Integrated Simulator for Quantum Key Distribution Networks", "Soler, J. and Fernandez, E. and Perianes, S. and Sanchez, A. and Martin, R.,2024.

[16] "NuQKD: A Modular Quantum Key Distribution Simulation Framework", "Gkouliaras, Georgios and Koukoumidis, Evangelos and Christidis, Spyridon and Vlachos, George,2024.

[17] S. Masot-Llima and A. Garcia-Saez, "Stabilizer Tensor Networks: Universal Quantum Simulator on a Basis of Stabilizer States," *Physical Review Letters*, vol. 133, no. 23, p. 230601, Dec. 2024. doi: 10.1103/PhysRevLett.133.230601.

Appendix

A. Plagiarism Report:



turnitin Page 2 of 15 - Integrity Overview Submission ID trn:oid:::1:3409317975

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

16 Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
2% Publications
1% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 15 - Integrity Overview Submission ID trn:oid:::1:3409317975

B. Base Paper:

1. Quantum Key Distribution (QKD) Integration with Classical Cryptography for Enhanced End-to-End Security.

C. Tools used / Hardware Components specifications:

This section lists the software tools, libraries, and hardware components used for implementing and testing the **Quantum-Safe Communication System**, which integrates BB84 Quantum Key Distribution (QKD) and Post-Quantum Cryptography (PQC).

Software Tools and Libraries

1. **Python 3.10:** The main programming language used for system development, simulation, and cryptographic integration.

2. **Qiskit:** Used for simulating quantum circuits, qubit transmission, and BB84 key generation.

   Syntax Example:
   ```
   from qiskit import QuantumCircuit, execute, Aer
   qc = QuantumCircuit(1,1)
   qc.h(0)  % Apply Hadamard gate
   qc.measure(0,0)
   ```

3. **PyCryptodome:** Provides cryptographic primitives such as HMAC-SHA3, XChaCha20-Poly1305, and KDF operations.

   Syntax Example:
   ```
   from Crypto.Hash import SHA3_256, HMAC
   h = HMAC.new(key, msg, SHA3_256)
   h.hexverify(tag)
   ```

4. **LibPQCrypto:** Used for implementing the Post-Quantum Cryptography (Kyber-512) algorithm for classical key exchange.

D. Published Papers and Certificates:

Publication status

1. International Journals

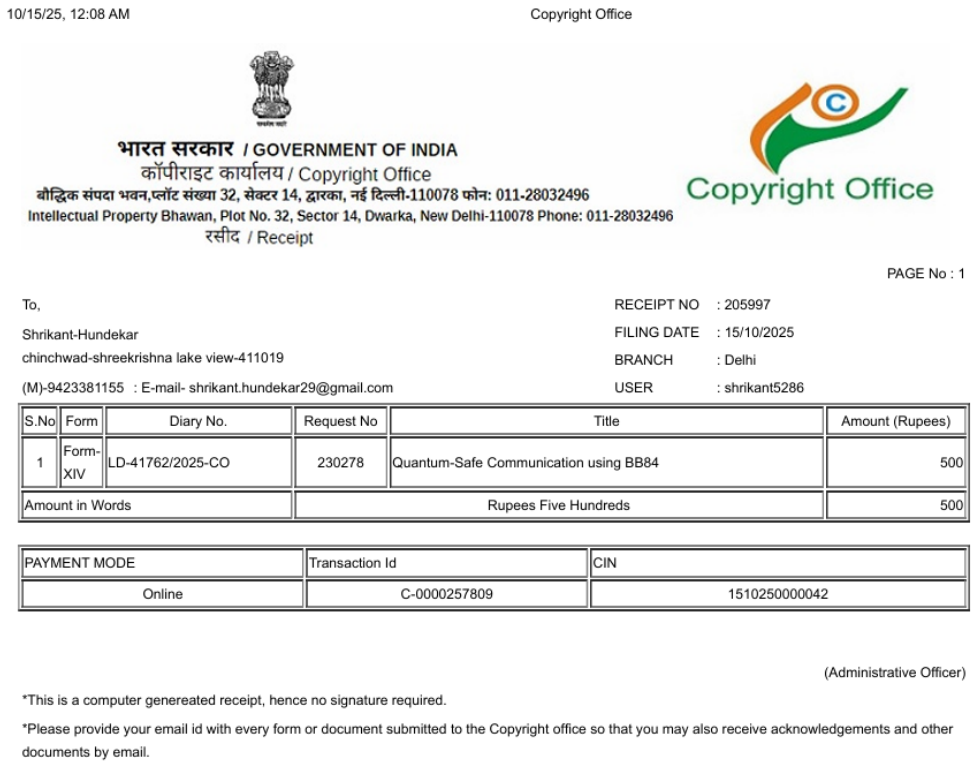
1. International Journal of Research and Analytical Reviews(IJRAR). **Status: Paper got Accepted .**
2. International Journal of Computer Science Education (IJCSED). **Status: Selected but not registered.**

2. International Conferences

1. International Conference on Intelligent Computing and Vision Technologies(ICICVT), In association with Punyashlok Ahilyadevi Holkar Solapur University, Solapur **Status: Under Review Process**
2. All accepted and presented papers of ICICVT 2025 will be published by SciTePress.
3. SciTePress is an international publisher indexed in SCOPUS, EI, Google Scholar, DBLP, Semantic Scholar, and Microsoft Academic..

E. Copyright Details:

Diary Number of Copyright File:- LD-41762/2025-CO



10/15/25, 12:08 AM Copyright Office

भारत सरकार / GOVERNMENT OF INDIA
कॉपीराइट कार्यालय / Copyright Office
बौद्धिक संपदा भवन,प्लॉट संख्या 32, सेक्टर 14, द्वारका, नई दिल्ली-110078 फोन: 011-28032496
Intellectual Property Bhawan, Plot No. 32, Sector 14, Dwarka, New Delhi-110078 Phone: 011-28032496
रसीद / Receipt

Copyright Office

PAGE No : 1

To,
Shrikant-Hundekar
chinchwad-shreekrishna lake view-411019
(M)-9423381155 : E-mail- shrikant.hundekar29@gmail.com

RECEIPT NO : 205997
FILING DATE : 15/10/2025
BRANCH : Delhi
USER : shrikant5286

S.No	Form	Diary No.	Request No	Title	Amount (Rupees)
1	Form-XIV	LD-41762/2025-CO	230278	Quantum-Safe Communication using BB84	500
Amount in Words			Rupees Five Hundreds		500

PAYMENT MODE	Transaction Id	CIN
Online	C-0000257809	1510250000042

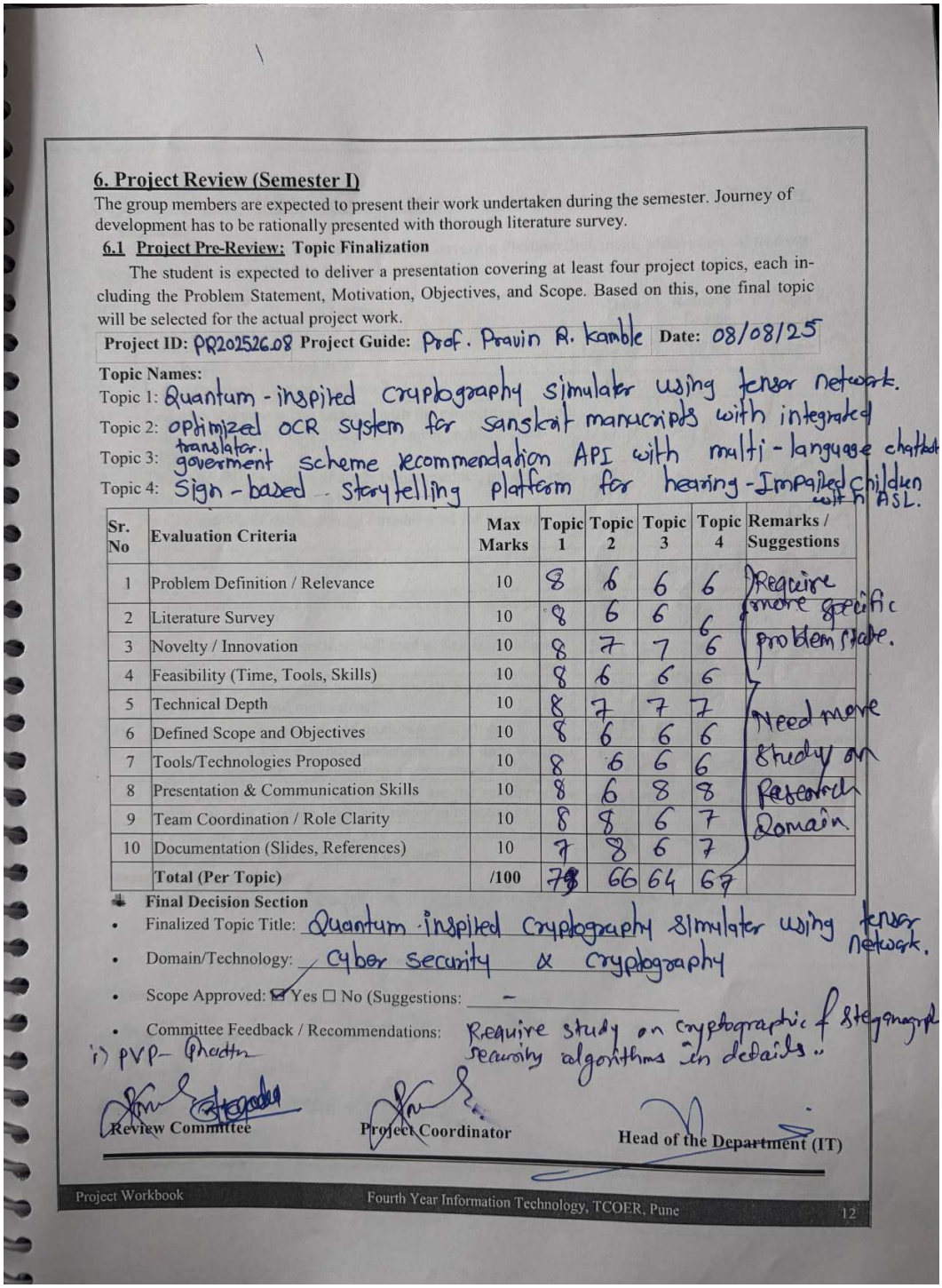
(Administrative Officer)

*This is a computer genereated receipt, hence no signature required.

*Please provide your email id with every form or document submitted to the Copyright office so that you may also receive acknowledgements and other documents by email.

Figure 13.1: Copyright status

F. Reviews Evaluation Sheet:



6. Project Review (Semester I)

The group members are expected to present their work undertaken during the semester. Journey of development has to be rationally presented with thorough literature survey.

6.1 Project Pre-Review: Topic Finalization

The student is expected to deliver a presentation covering at least four project topics, each including the Problem Statement, Motivation, Objectives, and Scope. Based on this, one final topic will be selected for the actual project work.

Project ID: PR2025208 **Project Guide:** Prof. Pravin A. kamble **Date:** 08/08/25

Topic Names:

Topic 1: Quantum-inspired cryptography simulator using tensor network.

Topic 2: optimized OCR system for sanskrit manuscripts with integrated translator.

Topic 3: goverment scheme recommendation API with multi-language chatbot

Topic 4: Sign-based storytelling platform for hearing-Impaired children with ASL.

Sr. No	Evaluation Criteria	Max Marks	Topic 1	Topic 2	Topic 3	Topic 4	Remarks / Suggestions
1	Problem Definition / Relevance	10	8	6	6	6	Require more specific problem state.
2	Literature Survey	10	8	6	6	6	
3	Novelty / Innovation	10	8	7	7	6	
4	Feasibility (Time, Tools, Skills)	10	8	6	6	6	
5	Technical Depth	10	8	7	7	7	Need more study on Research Domain
6	Defined Scope and Objectives	10	8	6	6	6	
7	Tools/Technologies Proposed	10	8	6	6	6	
8	Presentation & Communication Skills	10	8	6	8	8	
9	Team Coordination / Role Clarity	10	8	8	6	7	
10	Documentation (Slides, References)	10	7	8	6	7	
	Total (Per Topic)	**/100**	79	66	64	67	

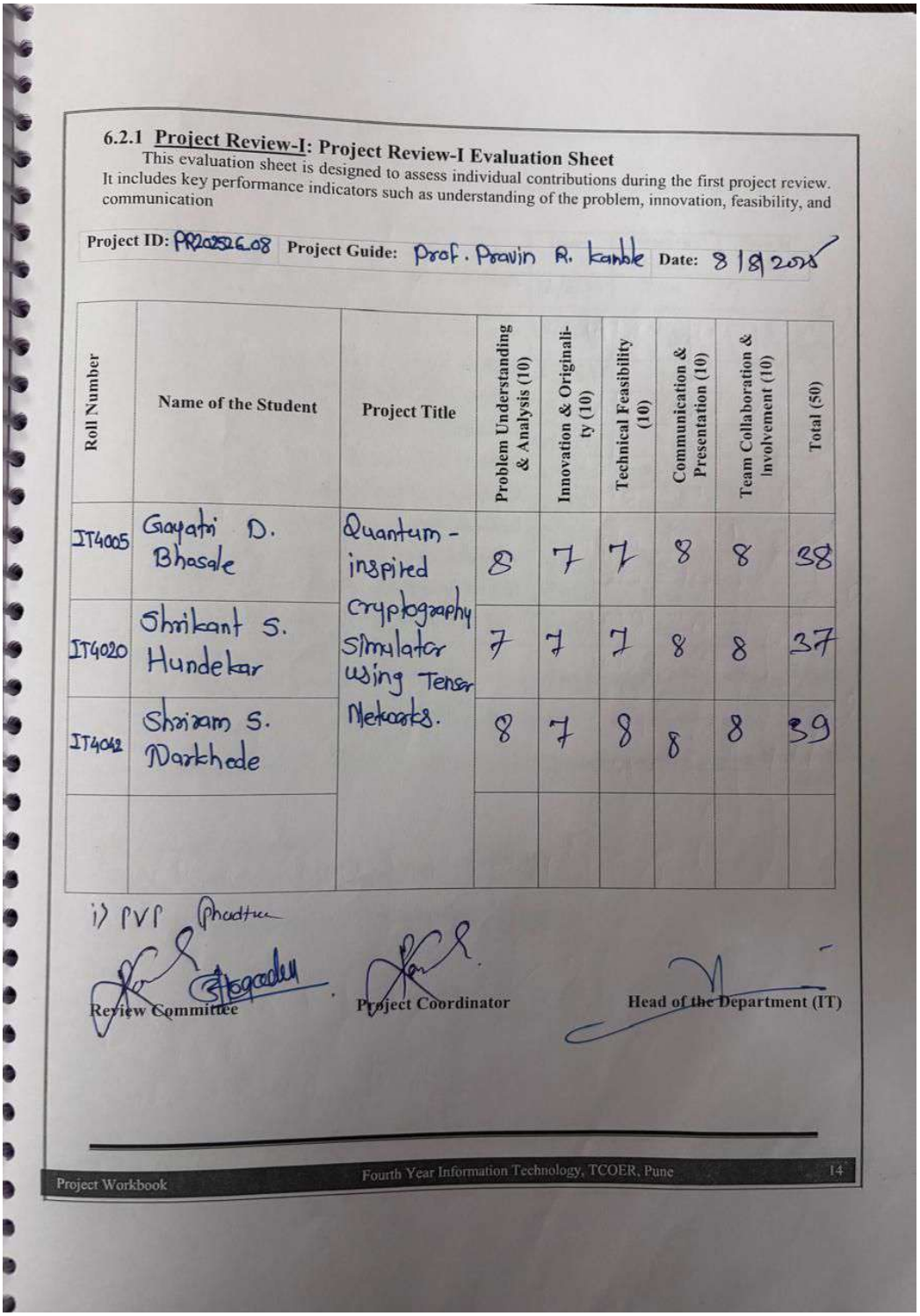
Final Decision Section

- Finalized Topic Title: Quantum-inspired cryptography simulator using tensor network.
- Domain/Technology: Cyber security & cryptography
- Scope Approved: ☑ Yes ☐ No (Suggestions: –
- Committee Feedback / Recommendations: Require study on cryptographic & steganography security algorithms in details..

i) PVP- Pune

Review Committee | Project Coordinator | Head of the Department (IT)

Project Workbook | Fourth Year Information Technology, TCOER, Pune | 12



6.2.1 Project Review-I: Project Review-I Evaluation Sheet

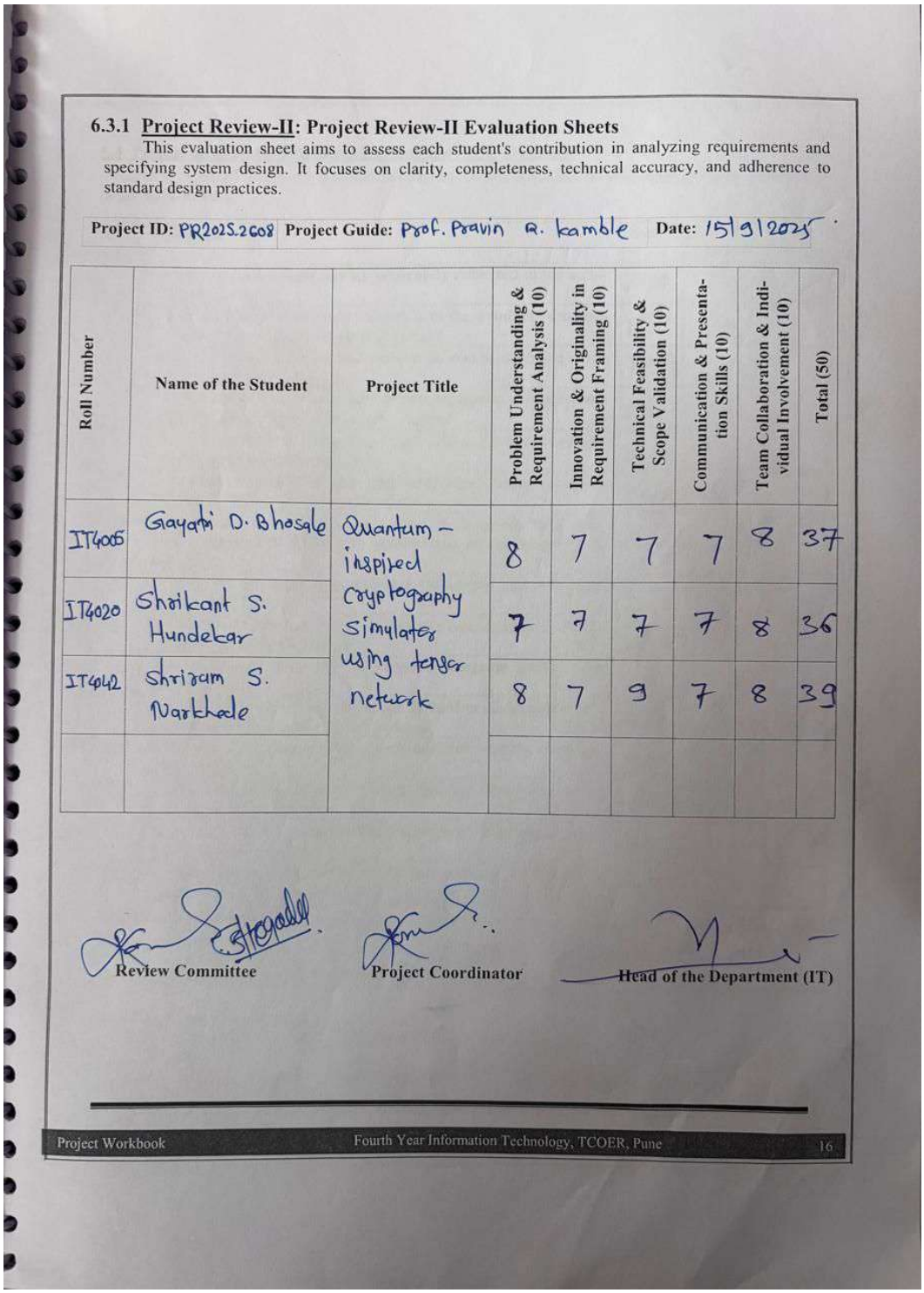
This evaluation sheet is designed to assess individual contributions during the first project review. It includes key performance indicators such as understanding of the problem, innovation, feasibility, and communication

Project ID: PR2025608 **Project Guide:** Prof. Pravin R. Kanke **Date:** 8/8/2025

Roll Number	Name of the Student	Project Title	Problem Understanding & Analysis (10)	Innovation & Originality (10)	Technical Feasibility (10)	Communication & Presentation (10)	Team Collaboration & Involvement (10)	Total (50)
IT4005	Gayatri D. Bhasale	Quantum-inspired cryptography simulator using Tensor Networks.	8	7	7	8	8	38
IT4020	Shrikant S. Hundekar		7	7	7	8	8	37
IT4042	Shriram S. Narkhede		8	7	8	8	8	39

1) PVP Phadtare

Review Committee Project Coordinator Head of the Department (IT)



6.3.1 Project Review-II: Project Review-II Evaluation Sheets

This evaluation sheet aims to assess each student's contribution in analyzing requirements and specifying system design. It focuses on clarity, completeness, technical accuracy, and adherence to standard design practices.

Project ID: PR2025.2608 **Project Guide:** Prof. Pravin R. kamble **Date:** 15/9/2025

Roll Number	Name of the Student	Project Title	Problem Understanding & Requirement Analysis (10)	Innovation & Originality in Requirement Framing (10)	Technical Feasibility & Scope Validation (10)	Communication & Presentation Skills (10)	Team Collaboration & Individual Involvement (10)	Total (50)
IT4005	Gayatri D. Bhosale	Quantum-inspired cryptography simulator using tensor network	8	7	7	7	8	37
IT4020	Shrikant S. Hundekar		7	7	7	7	8	36
IT4042	Shriram S. Narkhede		8	7	9	7	8	39

Review Committee Project Coordinator Head of the Department (IT)

Project Workbook Fourth Year Information Technology, TCOER, Pune 16

6.4.1 Project Review-III : Internal Evaluation Sheet (Semester I)

This evaluation sheet is designed to assess the student's overall performance in Project Stage-I. It covers understanding of the problem, analytical and design capabilities, planning, documentation quality, and individual contribution through presentations and discussions.

Project ID: PR2025.26.08 **Project Guide:** Prof. Pravin R. kamble **Date:** 16/10/25

Project Topic Name: Quantum-Safe Communication using BB84.

Sr. No.	Name(s) of the student	Problem Statement/ Motivation/ Objectives/ Scope/ Feasibility Requirement (05)	Literature Survey (05)	Requirement Analysis, Modeling & Designing (10)	Planning & Prototyping and implementation (10)	Presentation & Question Answer (10)	Project Documentation &Partial Project Report (10)	Total (50)
1.	IT4005 Gayatri D. Bhosale	4	5	7	8	8	8	40
2.	Shrikant S. Hundekar	4	5	8	8	7	8	40
3.	Shriram S. Narkhede	5	5	8	8	9	7	42
4.								

Name and Signature of Evaluation Committee:

1. Prof. Pravin R. Kamble –
2. Prof. Dayanand Anagade – Offline

Examiners Feedback and Suggestions:

work on cryptographic Algo. in detail.
modified UML, & architechere diagram.
overall prescutation is good. more work needed

Signature of Guide

Dr. Vilas S. Gaikwad
Head of Department (IT)