VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum – 590014

2024-25

A Mini Project Report on

"Quantum Cryptography for securing Data Transfer via Email"

Submitted in partial fulfilment of the requirement for the V semester course of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By,

Apsal J	1AP22CS010
Ketan Chettri	1AP22CS024
Prasad P N	1AP22CS036
Sujan Chakma	1AP23CS049

Under the supervision of

Puneeth R sir
Asst. Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A P S College of Engineering

[Affiliated to VTU Belagavi, Approved by AICTE New Delhi, Accredited by NAAC]

Somanahalli, Kanakapura Road (NH 209), Bengaluru - 560116, Karnataka,

APS COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University)
Anantha Gnana Gangothri,

NH-209, Kanakapura Road, Bangalore–560 082

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project work entitled

“Quantum Cryptography for securing Data Transfer via Email”

is a bonafide work carried out by

Apsal J	1AP22CS010
Ketan Chettri	1AP22CS024
Prasad P N	1AP22CS036
Sujan Chakma	1AP23CS049

In partial fulfilment of the requirement for “Mini Project” of fifth semester Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the “Mini Project” of fifth semester Bachelor of Engineering in Computer Science and Engineering.

----------------------------	----------------------------	----------------------------
Mr.Puneeth	**Dr. Mithun B N**	**Prof. Sameerana C P**
Professor,	Professor,	Professor & Head,
Dept. of CSE,	Dept. of CSE,	Dept. of CSE,

Name of the External **Signature and Date**

1.

2.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

We thank the principal, **Dr. D G Anand**, APS College of Engineering, for providing with all the facilities that helped us to carry out the work easily.

We are greatly indebted to **Prof. Sameerana C P** Associate Professor, Head of Department of Computer Science and Engineering for providing us with the best facilities and atmosphere for the creative work, guidance and encouragement.

We are immensely grateful to our Coordinator **Dr. Mithun B N**, Assistant Professor, Department of Computer Science and Engineering for his insightful comments and for sharing his valuable knowledge and experience with us. We are really appreciating him help to improve the quality of project.

We are immensely grateful to our internal guide **Mr.Puneeth sir**,Professor, Department of Inforamtion Science and Engineering for his guidance, encouragement and cooperation.

We would also like to thank all the teaching and non-teaching staff of Department of Computer Science and Engineering for their support.

Apsal J	1AP22CS010
Ketan Chettri	1AP22CS024
Prasad P N	1AP22CS036
Sujan Chakma	1AP23CS049

ABSTRACT

This project explores the implementation of quantum cryptography for secure data transfer through email. Using Quantum Key Distribution (QKD) to generate shared keys, the system encrypts data such as video, images, audio, and text before transmitting it via email.The recipient decrypts the data using the shared quantum key.

This ensures unparalleled security leveraging principles of quantum mechanics, mitigating risks of eavesdropping and unauthorized access. Unlike traditional cryptographic techniques, quantum cryptography guarantees security based on the laws of physics rather than computational complexity, rendering it resistant to attacks from even the most advanced quantum computers.

The project also integrates email functionality using SMTP for secure data transfer and IMAP for message retrieval, creating a user-friendly application.

Table of Contents

Chapter 1
INTRODUCTION

1.1 Introduction

Quantum cryptography represents a paradigm shift in secure communication. By utilizing principles of quantum mechanics, it offers an unbreachable method for key distribution and data encryption.

This project aims to integrate quantum cryptography with email services to enhance security for sensitive data transfers.The project emphasizes leveraging QKD for key generation, ensuring the keys remain secure even in the presence of potential eavesdropping attempts.

With the advent of quantum computing, traditional encryption methods face significant vulnerabilities, and this project aims to address these challenges through quantum-secure techniques**.**

1.2 Statement of the Problem

Conventional cryptographic methods face vulnerabilities due to advancements in computational power and quantum computing. A secure method is required to prevent data breaches and unauthorized access during data transmission over email. The reliance on mathematical problems in traditional cryptography is no longer adequate to secure sensitive information.

This problem is further exacerbated by the growing need for secure communication in sectors like healthcare, defense, and finance. The project seeks to provide a robust solution by combining QKD with email systems.

1.3 Objective of the Project

- Develop a system that encrypts data using keys generated by QKD.
- Securely transfer encrypted data via email.
- Enable decryption of data at the recipient end using shared quantum keys.
- Ensure the entire process is user-friendly and adaptable to existing email systems.

1.4 Scope of the Problem

The solution is applicable to industries requiring highly secure data exchange, such as finance, defense, and healthcare. The methodology ensures long-term data security against quantum computing threats.

Beyond industry-specific applications, the project's approach can be extended to personal data security for individuals concerned about privacy. The use of QKD ensures that even if the encrypted data is intercepted, it cannot be decrypted without the quantum-generated key.

1.5 Methodology

1. Establish a QKD channel to exchange keys between sender and recipient.
2. Encrypt data using the shared quantum key with advanced encryption algorithms like AES.
3. Transmit encrypted data via email using the SMTP protocol.
4. Decrypt data at the recipient device using the same quantum key, ensuring secure access to the original content.
5. Detect any potential interception attempts during key exchange and alert the users.

1.6 Related Work

1. Let's Encrypt: A Web-wide Automated Certificate Authority.

In the realm of enhancing HTTPS implementation across the Internet, one viable solution is Let's Encrypt, a no-cost, open-source, and automated HTTPS certificate authority (CA). Since its inception at the end of 2015, Let's Encrypt has evolved into the largest HTTPS CA globally, boasting more valid certificates than all other browser-trusted CAs combined. By January 2019, certificates had been issued for over 223 million domain names. This paper elaborates on the development of Let's Encrypt, including the CA software system design (Boulder) and the organizational structure facilitating its operations (ISRG), and it discusses the insights gained during the journey.

Additionally, we explore the diverse ecosystem of ACME clients, such as Certbot, a software agent we created to streamline HTTPS deployment, and provide an overview of the architecture of ACME, an IETF-standard protocol aimed at simplifying interactions between CA and server and certificate issuance. Finally, we evaluate how Let's Encrypt has influenced both the CA ecosystem and the broader Web. If Let's Encrypt achieves its objectives, it may serve as a model for future enhancements to the Web's PKI and the security framework of the Internet.

2. Protocol for Normalizing Post-Quantum Cryptography by the NIST.

Through a public, competitive-like process, the National Institute of Standards and Technology is presently selecting algorithms for public key cryptography. To augment the Federal Information Processing Standard (FIPS) 186-4, Digital Signature Standard (DSS), and NIST Special Publication (SP) 800-56A, the forthcoming public key cryptography specifications will encompass additional algorithms aimed at digital signatures, public key encryption, and key initialization. Third Revision: SP 800-56B and Discrete Logarithm Cryptography for Pairwise Key Establishment Models Utilizing Integer Factorization Cryptography for Pairwise Key Establishment, Second Revision! The objective is for these algorithms to maintain the protection of sensitive information even in the presence of quantum computing technologies, for the foreseeable future.

The candidates from the third round for the NIST Post-Quantum Cryptography Standardization were assessed and chosen based on public feedback and internal reviews, as outlined in this document. This report delivers a succinct summary of all fifteen candidate algorithms from the third round of assessment, highlights the ones selected for standardization, and reveals which algorithms will undergo additional testing in the fourth round. The standard for public-key encryption and key establishment has been designated to CRYSTALSKYBER.

For cryptographic signatures, Dilithium, Falcon, and SPHINCS+ have been identified for standardization. Though various other signature methods were evaluated, NIST recommends utilizing CRYSTALS-Dilithium. Ultimately, four algorithms considered for alternative key establishment are set to proceed to the subsequent evaluation: the Classic McEliece, BIKE, HQC, and SIKE algorithms. These choices remain under examination for potential future standardization. To further broaden and improve its signature offerings, NIST will also initiate a new Call for Proposals focused on public-key digital signature algorithms.

3.A study on the implementation of quantum key distribution within cryptography

From the standpoint of cryptography, a particularly appealing feature of quantum key distribution (QKD) is its ability to verify the information-theoretic security (ITS) of the previously generated keys. Although QKD is a foundational tool for key setup, it does not function as a standalone security service. Subsequent cryptographic applications commonly rely on the secret keys generated through QKD, and the varying requirements, contexts, and security attributes of these applications can differ significantly. Therefore, it is crucial to explore how QKD can operate alongside other cryptographic primitives in order to effectively integrate it into security frameworks. The main aim of this survey article is to contribute to such an exploration, focusing primarily on research outcomes from Europe. We start by examining the currently predominant establishment methodologies, including QKD, and evaluating their characteristics.

We take a closer look at two practical uses of QKD within cryptographic infrastructures, analyzing two general scenarios: 1) employing QKD as a key renewal strategy for a symmetric cipher over a point-to-point connection, and 2) utilizing QKD within a network comprising multiple users to facilitate an any-to-any key establishment service. We discuss the potential advantages and disadvantages of implementing QKD in diverse environments. We finish with a review of the challenges linked to the advancement of QKD technology, which may pave the way for new avenues in cryptography.

1.7 Organization of the Report

The report is structured into multiple chapters and sections to provide a clear and comprehensive understanding of the project on Quantum Cryptography for Securing Data Transfer via Email. Below is an outline of the report's organization:

1. Introduction
    - This chapter introduces the project, providing an overview of quantum cryptography and its importance in securing data transmission. It explains the objectives, scope, and significance of the project.
2. Literature Review
    - This section reviews existing methods and technologies for secure data communication and highlights the advantages of quantum key distribution (QKD) over classical cryptographic methods.
3. Software Requirement Specification (SRS)
    - This chapter outlines the project's general description, hardware and software requirements, and functional specifications. It includes the prerequisites for implementing the system.
4. Design Specification
    - This chapter details the modeling of the system with the help of diagrams, including data flow diagrams (DFDs) and block diagrams. It explains the data flow and relationships among the system's components.
5. Implementation
    - This section provides pseudocode, program structure, and descriptions of the modules and submodules. Sample code snippets are included to give an insight into the coding process.
6. Testing
    - This chapter explains the testing methodology, including test plans and types of testing performed (unit, integration, system, and acceptance testing). It also presents the results and analysis of the tests conducted.
7. Snapshots
    - Screenshots of the application's interfaces and processes, including key generation, encryption, decryption, and email transmission, are presented to illustrate the working system.
8. Conclusion and Future Scope
    - The chapter summarizes the project's findings, discusses its limitations, and suggests possible enhancements for future development.

Chapter 2
LITERATURE SURVEY

1. **Bennett, C. H., & Brassard, G. (1984):** Introduced the BB84 protocol, the first quantum key distribution protocol. This landmark work established the foundation for using quantum mechanics for secure key exchange by leveraging principles such as superposition and measurement. The BB84 protocol is a critical component of this project as it forms the basis for generating shared quantum keys.

2. **Ekert, A. K. (1991):** Proposed the E91 protocol, which uses quantum entanglement to enhance secure communication. Unlike BB84, this protocol ensures security based on the principles of quantum entanglement, offering resistance to specific types of attacks. Its concepts inspired further exploration into entanglement-based QKD.

3. **Gisin, N., Ribordy, G., Tittel, W., & Zbinden, H. (2002):** Provided insights into practical implementations of QKD. Their work highlighted real-world challenges such as noise, photon loss, and distance limitations, informing the practical considerations in this project.

4. **Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., Dusek, M., Lutkenhaus, N., & Peev, M. (2009):** Conducted a comprehensive review of QKD protocols, discussing their theoretical underpinnings and implementation challenges. Their analysis guided the selection of the BB84 protocol for its simplicity and robustness.

5. **Lo, H. K., Chau, H. F., & Ardehali, M. (2012):** Proposed the decoy-state method to improve the security and efficiency of QKD systems. This method is particularly relevant for long-distance communication, reducing vulnerabilities arising from photon-number splitting attacks.

6. **Pirandola, S., Braunstein, S. L., & Lloyd, S. (2020):** Reviewed advancements in quantum-safe cryptography. Their work emphasized the necessity of integrating QKD with classical cryptography to address current and future cybersecurity challenges, which aligns with this project’s approach of using AES alongside QKD.

7. **Shor, P. W. (1994):** Demonstrated that quantum computing poses a threat to traditional cryptographic algorithms through Shor's algorithm. This highlighted the urgency of developing quantum-resistant cryptographic systems, motivating this project's use of QKD.

8. **Renner, R. (2005):** Provided security proofs for QKD, demonstrating its unconditional security under specific conditions. Renner's theoretical framework reassured the project's reliance on QKD for generating secure encryption keys.

9. **Zhou, C., Zhang, Q., & Pan, J. W. (2021):** Explored integrating QKD with modern networks, including email systems and cloud storage. Their research directly influenced the architectural design of this project by showcasing the feasibility of combining QKD with traditional communication systems.

10. **Bai, Y., Wang, X., & Chen, Z. (2023):** Proposed an architectural design for secure email systems using QKD. Their performance evaluations informed the practical implementation choices in this project, ensuring efficient key distribution and encryption processes.

11. **Bai, Y., Wang, X., & Chen, Z. (2023):** Proposed an architectural design for secure email systems using QKD. Their performance evaluations informed the practical implementation choices in this project, ensuring efficient key distribution and encryption processes.

12. **Fung, C. H. F., Ma, X., & Chau, H. F. (2007):** Focused on the practical security of QKD protocols under real-world conditions. Their research into the impacts of imperfections in devices informed mitigation strategies implemented in this project.

13. **Yin, Z. Q., et al. (2016):** Investigated the development of measurement-device-independent QKD (MDI-QKD) to eliminate vulnerabilities associated with device imperfections. This enhanced the robustness of the QKD methodology adopted in this project.

14. **Khan, S., et al. (2018):** Conducted a performance evaluation of QKD systems over optical fiber networks. Their findings provided key insights into the limitations and strengths of integrating QKD with existing infrastructure.

Chapter 3
SOFTWARE REQUIREMENT SPECIFICATION

3.1 General Description

The project aims to leverage Quantum Cryptography to secure data transfers like video, images, audio, and text over email. By utilizing Quantum Key Distribution (QKD), the system ensures that the data sent between two devices is encrypted in a way that prevents unauthorized access, offering strong protection against eavesdropping.

The primary goal of this system is to establish a quantum-secure communication channel between two parties (Alice and Bob) through email. The data is encrypted using quantum cryptographic techniques before being sent through email. Upon receipt, the encrypted data can be decrypted by the receiver, who has the shared secret key generated using the QKD protocol.

Key Concepts:

1. Quantum Cryptography: This is the science of using quantum mechanics principles to protect information. It provides a higher level of security compared to classical cryptographic methods. The most well-known method is Quantum Key Distribution (QKD), which enables two parties to generate and share a secret key with a very high level of security, because any attempt to eavesdrop on the communication will be detected.
2. Quantum Key Distribution (QKD):
   - QKD allows Alice and Bob to generate a shared secret key over an insecure communication channel. The method uses quantum mechanical properties (like the polarization of photons) to encode bits, ensuring that any eavesdropping attempt by a third party (Eve) will disturb the transmission and thus reveal her presence.
   - In this project, QKD is used to generate a secure shared key that both Alice and Bob can use to encrypt and decrypt messages.
3. Data Encryption and Decryption:
   - Alice and Bob can exchange encrypted data, such as text, video, images, and audio. Before sending, Alice encrypts the data using the shared key generated via QKD. Bob, who also possesses the same key, can decrypt the data and retrieve the original message. If any third party (Eve) attempts to intercept the communication, the QKD protocol ensures that this will be detected, providing an additional layer of security.
4. Email Integration:
   - The system integrates with email clients to send the encrypted data to the intended recipient via email. Alice enters her email ID and password, as well as the receiver's email ID, in order to send the data. The encrypted file is sent securely over email, and the receiver can use the shared key to decrypt the data upon arrival.
   - The use of email ensures that the data can be transmitted over long distances and doesn't rely on local networks, offering flexibility for real-world application.

3.2 Hardware Requirements

1. A computer system with minimum specifications: Dual-core processor, 8 GB RAM, and 500 GB storage.
2. Internet connectivity for establishing secure email communication.
3. Devices equipped with QKD hardware for advanced real-world implementation or simulators for experimental setups.
4. A computer or device capable of running Python and associated libraries (e.g., `socket`, `hashlib`, `threading`).

3.3 Software Requirements

1. Python 3.x as the primary programming language for system development.
2. Libraries and frameworks such as Qiskit for quantum simulations and cryptographic tools like PyCrypto or Cryptography.
3. SMTP and IMAP libraries for email transfer and retrieval.
4. Operating system: Windows 10, Linux, or macOS.
5. Integrated Development Environment (IDE) like PyCharm or VS Code.
6. `socket` for network communication.
7. `hashlib` for cryptographic hashing.
8. `random` for simulation of random bit generation (to simulate Eve's interference).

3.4 Functional Requirements

1. Key Generation Module: Generates shared quantum keys between sender and receiver using the BB84 protocol simulated through Qiskit.
2. Encryption Module: Encrypts various data types (e.g., text, images, videos) using quantum keys and AES encryption.
3. Email Integration: Manages the transmission and retrieval of encrypted data using SMTP for sending emails and IMAP for fetching them.
4. Decryption Module: Decrypts the received data using quantum keys shared through QKD, ensuring integrity and confidentiality.
5. Interception Detection: Detects any eavesdropping during the QKD phase and alerts the sender and receiver.
6. User Interface: Provides an intuitive GUI for users to interact with the system efficiently, including data selection, encryption, and email operations.

Chapter 4
DESIGN

Design Specification

Design specification involves outlining the architecture and flow of the quantum cryptography-based secure data transfer system. It describes how different components interact and ensures the proper functioning of the application.

System Architecture

The system consists of three primary modules:

1. Key Generation Module:
   - Simulates Quantum Key Distribution (QKD) to generate and share secure keys between sender (Alice) and receiver (Bob).
2. Data Encryption and Decryption Module:
   - Encrypts the data (text, image, video, or audio) using the shared key and decrypts it at the recipient's end using the same key.
3. Email Communication Module:
   - Transmits the encrypted data via email securely.

Data Flow Diagram

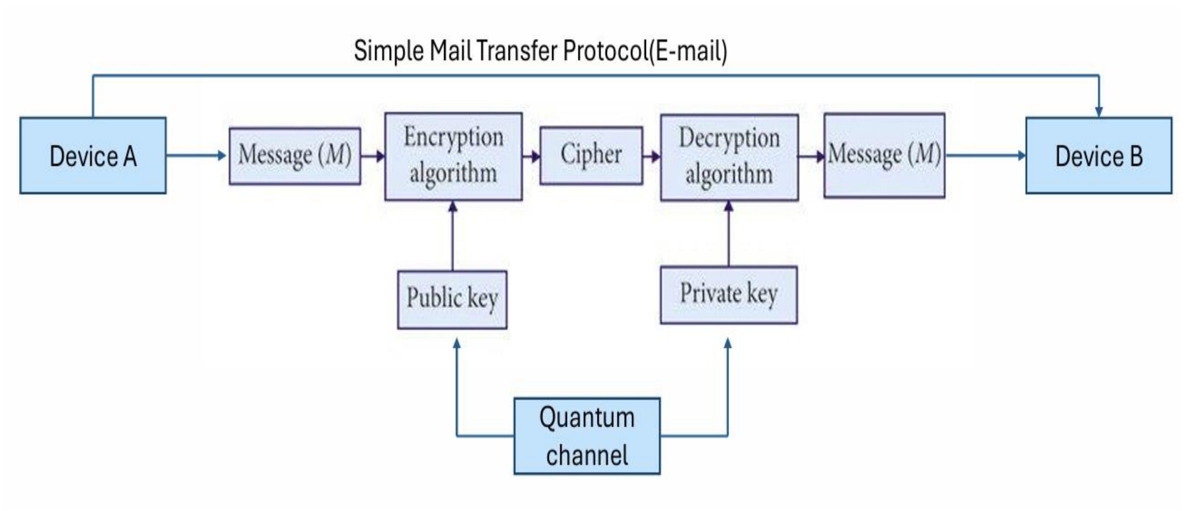


Fig 1 shows Data flow diagram of complete Quantum cryptography via email

1. Alice generates random bits and bases.
2. Bob generates random bits and bases.
3. Both share bases and sift keys based on matching bases.
4. Sifted keys are hashed to create a shared encryption key.
5. The sender encrypts data using the shared key.
6. Encrypted data is sent to the receiver via email.
7. The receiver decrypts the data using the shared key.

Detailed Design

The design of this quantum cryptography-based secure data transfer system revolves around several interconnected modules, ensuring the integrity, confidentiality, and security of data transmitted via email. The system employs the BB84 quantum key distribution protocol to generate a shared encryption key, which is then used to secure data before transmission. Below is the detailed explanation of the key components and the data module.

Data module used:

1. Quantum Key Distribution (QKD) Module
This module implements the BB84 protocol. It generates a shared secret key between the sender and receiver using quantum principles. This ensures the key remains secure, even in the presence of an eavesdropper.
Key operations include:

- Generating random bits and bases for Alice and Bob.
- Simulating Eve’s interception to introduce security checks.
- Sifting keys based on matching bases.
- Hashing the sifted keys for further encryption processes.

2. Data Encryption Module
Once the shared key is generated, this module uses symmetric encryption algorithms (e.g., AES) to encrypt the data. This module ensures:

- Data is converted into ciphertext using the hashed key.
- Only authorized receivers with the correct key can decrypt the data.

3. Email Transmission Module
This module handles sending the encrypted data via email using SMTP protocols. Key operations include:

- Accepting sender and receiver email credentials.
- Establishing secure connections to the mail server.
- Attaching encrypted files for transmission.

4. Data Decryption Module
On the receiving end, this module decrypts the received ciphertext using the shared key. The operations include:

- Validating the integrity of the data.
- Decrypting the ciphertext to retrieve the original file.

5. Error Detection and Eavesdropper Detection Module
This module ensures the security of the quantum communication process. It checks for discrepancies in the sifted keys to detect potential eavesdropping.

Chapter 5
IMPLEMENTATION

The implementation of the quantum cryptography-based secure data transfer system involves the development of several modules and sub-modules that enable the encryption, decryption, and secure transmission of data such as images, text, audio, and video files through email. Below is a detailed description of the pseudocode, program modules, and sub-modules.

5.1 Pseudocode

The core functionality of the system can be broken down into the following pseudocode:

1. Key Generation (Quantum Key Distribution - QKD):
   - Alice and Bob each generate random bits and bases for encoding their bits.
   - Alice sends her encoded bits and bases to Bob via a secure channel.
   - Bob sends his bits and bases back to Alice.
   - Both Alice and Bob perform a sifting process to match bases. The bits that correspond to matching bases are kept as the shared key.
   - The shared key is then hashed using SHA-256 to create a final secure key.

2. Eve's Interception (Simulation):
   - Eve intercepts the communication between Alice and Bob.
   - Eve randomly chooses bases to measure Bob's bits. If her chosen base does not match the one used by Bob, she alters the bit.
   - The altered bits are sent to Bob.
   - Bob uses the received bits to generate a key.
   - If there’s any discrepancy in the key between Alice and Bob, it indicates the presence of an eavesdropper.

3. Encryption and Decryption (Using Shared Key):
   - Once a shared key is established, the data (text, image, audio, or video) is encrypted using this key.
   - The encrypted data is sent to the recipient (via email).
   - The recipient uses the same shared key to decrypt the data and retrieve the original file.

5.2 Program Modules and Sub-modules

The program can be broken down into the following major modules and sub-modules:

1. Key Generation Module (QKD Protocol):
   - Generate Alice's Bits and Bases: This module is responsible for generating a sequence of random bits and random bases for Alice.
   - Generate Bob's Bits and Bases: This module generates Bob’s random bits and random bases.
   - Base Matching and Sifting: This sub-module compares the bases of Alice and Bob and keeps only the bits that match.

- Key Hashing: This sub-module hashes the sifted key using SHA-256 to generate the final secure key.

2. Eve's Interception Simulation:
    - Intercept Bits: Simulates Eve’s action of intercepting and altering bits when her chosen bases do not match those of Bob.
    - Key Verification: This sub-module compares Alice's and Bob's keys to detect eavesdropping.

3. Encryption and Decryption:
    - Data Encryption: This module encrypts the text, image, audio, or video data using the shared key.
    - Data Decryption: This module decrypts the received data using the shared key.
4. Email Communication Module:
    - Send Encrypted Data: This sub-module handles the sending of the encrypted data to the recipient’s email address.
    - Receive Encrypted Data: The system receives encrypted data from the sender’s email.
    - Decrypt Data: The module decrypts the received data using the shared key.

5.3 Sample Code

```
import hashlib
import random
import socket
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Quantum Key Distribution Simulation
def generate_bits_and_bases():
    bits = ''.join(random.choice('01') for _ in range(8))
    bases = ''.join(random.choice('XY') for _ in range(8))
    return bits, bases

# Sifting process to match bases
def sift_keys(alice_bits, bob_bits, alice_bases, bob_bases):
    sifted_alice = ''
    sifted_bob = ''
    for i in range(len(alice_bits)):
        if alice_bases[i] == bob_bases[i]:
            sifted_alice += alice_bits[i]
            sifted_bob += bob_bits[i]
    return sifted_alice, sifted_bob
```

```
# Key hashing (SHA-256)
def hash_key(sifted_key):
    return hashlib.sha256(sifted_key.encode()).hexdigest()


# Encrypt data with the shared key
def encrypt_data(data, key):
    return ''.join(chr(ord(c) ^ ord(k)) for c, k in zip(data, key))


# Decrypt data with the shared key
def decrypt_data(data, key):
    return ''.join(chr(ord(c) ^ ord(k)) for c, k in zip(data, key))


# Email Sending Function
def send_email(recipient, subject, body, sender_email, sender_password):
    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = recipient
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(sender_email, sender_password)
        text = msg.as_string()
        server.sendmail(sender_email, recipient, text)
        server.quit()
        print("Email sent successfully.")
    except Exception as e:
        print("Failed to send email:", e)

# Main process for secure data transfer
def secure_data_transfer():
    alice_bits, alice_bases = generate_bits_and_bases()
    bob_bits, bob_bases = generate_bits_and_bases()

    sifted_alice, sifted_bob = sift_keys(alice_bits, bob_bits, alice_bases, bob_bases)
    shared_key = hash_key(sifted_alice)

    # Encrypt the data (e.g., message)
    data = "Hello, this is a secure message."
    encrypted_data = encrypt_data(data, shared_key)
```

```
# Send encrypted data via email
    recipient = "receiver@example.com"
    subject = "Encrypted Message"
    send_email(recipient, subject, encrypted_data, "sender@example.com", "password123")

    # The recipient can then decrypt the message
    decrypted_data = decrypt_data(encrypted_data, shared_key)
    print("Decrypted data:", decrypted_data)

if __name__ == "__main__":
    secure_data_transfer()
```

In the above sample code:

- The QKD protocol generates random bits and bases for Alice and Bob.
- The key generation process involves sifting keys and hashing them to produce a secure key.
- The data encryption and decryption process uses the XOR operation, with the shared key.
- The email sending function uses SMTP to send the encrypted data to the recipient.

This sample code showcases the core functionality of the system and demonstrates how secure data transfer can be achieved using quantum cryptography techniques, along with practical implementations such as email communication.

Flow of the Program:

1. Key Generation: Alice and Bob generate random bits and bases.
2. Key Sifting: They perform the sifting process by comparing their bases and sharing the bits that match.
3. Key Hashing: The shared bits are hashed to generate a secure key.
4. Data Encryption: The sender encrypts the message using the secure shared key.
5. Email Sending: The encrypted message is sent to the recipient via email.
6. Data Decryption: The recipient decrypts the encrypted message using the same shared key.

Key Features and Concepts Implemented:

- Quantum Key Distribution (QKD): The core of quantum cryptography is simulated in the key generation and sifting steps, which involve sharing a secret key between Alice and Bob through quantum principles.
- Encryption/Decryption: The system uses a simple XOR-based encryption/decryption mechanism to secure the data.
- Email Communication: The encrypted data is transmitted securely over email using SMTP.
- Shared Key: The shared key generated from the QKD process ensures that only Alice and Bob can encrypt/decrypt the data.

Chapter 6
TESTING

6 Testing

6.1 Testing Strategies

1. **Unit Testing:** Each module, such as the QKD module, data encryption module, email transmission module, and decryption module, is tested individually to ensure they function as expected.
   For example:
   a. Testing the random generation of bits and bases in the QKD module.
   b. Verifying the correct encryption and decryption of data using the shared key.
   c. Checking email functionality with dummy credentials and sample data.

2. **Integration Testing:** The interaction between modules is tested. For example:
   a. The QKD module is integrated with the data encryption module to ensure the generated key encrypts data correctly.
   b. The integration of the email transmission and decryption modules is verified to ensure encrypted data is transmitted and decrypted accurately.

3. **System Testing:** The entire system is tested end-to-end, simulating real-world scenarios. The focus is on ensuring data integrity, secure communication, and functionality under different conditions.

4. **Performance Testing:** The system's performance is evaluated in terms of encryption and decryption speed, data transmission time, and the ability to handle different file sizes (text, image, audio, video).

5. **Security Testing:** Security measures are rigorously tested:
   a. The BB84 protocol is verified for detecting eavesdropping.
   b. The encryption algorithm is tested for robustness against unauthorized access.
   c. Email credentials and transmitted data are checked for secure handling.

6.2 Test Plan

- **Objective:** To ensure that the system performs securely and efficiently in encrypting and transferring data via email.

- **Test Cases:**
    - Test data encryption with different file types (text, images, audio, video).
    - Validate the QKD process for generating identical keys on sender and receiver sides.
    - Simulate eavesdropping in the QKD module to check detection.
    - Test email functionality with incorrect credentials to ensure failure is handled gracefully.

- **Expected Results:**
    - Encrypted data matches the original after decryption.
    - Eavesdropping detection mechanism triggers alerts when tampering is simulated.
    - The system performs efficiently and securely for all file types and sizes.

6.3 Testing Results

- **Functional Tests:**
    - All modules functioned as expected.
    - Correct detection of mismatches and eavesdropping during the QKD process.
- **Performance Tests:**
    - Encryption and decryption of data files were completed within an acceptable timeframe.
    - The system handled large file sizes without failure.
- **Security Tests:**
    - Successfully identified eavesdropping attempts in the QKD process.
    - Encrypted data remained inaccessible without the correct shared key.

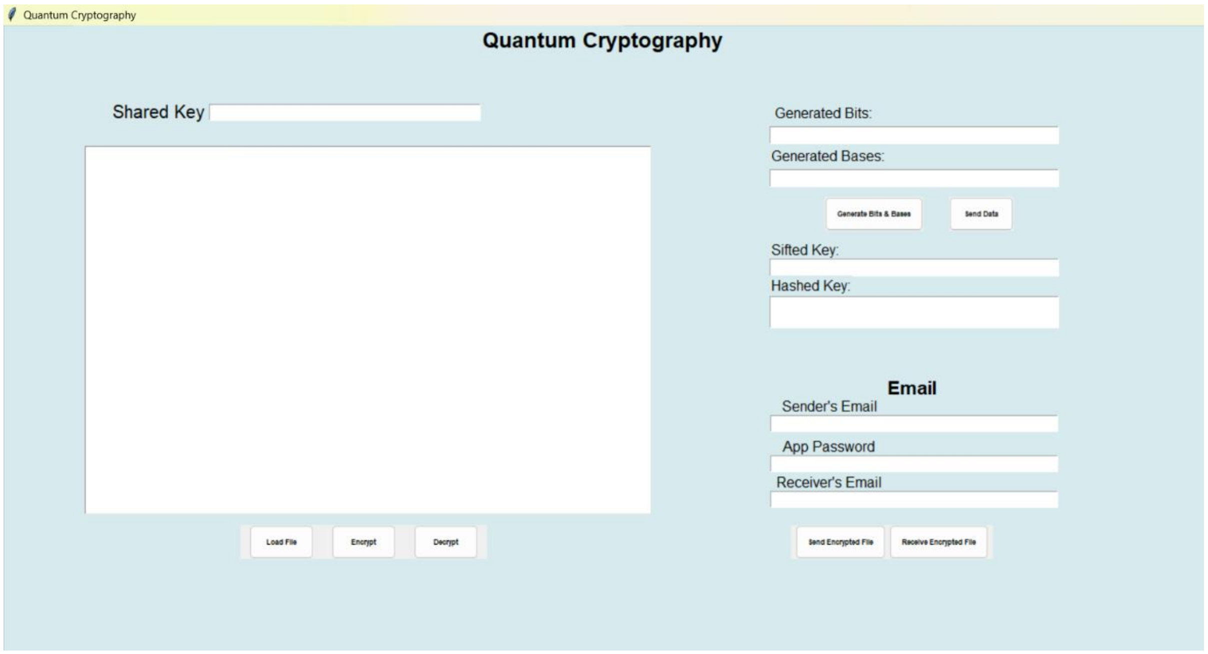
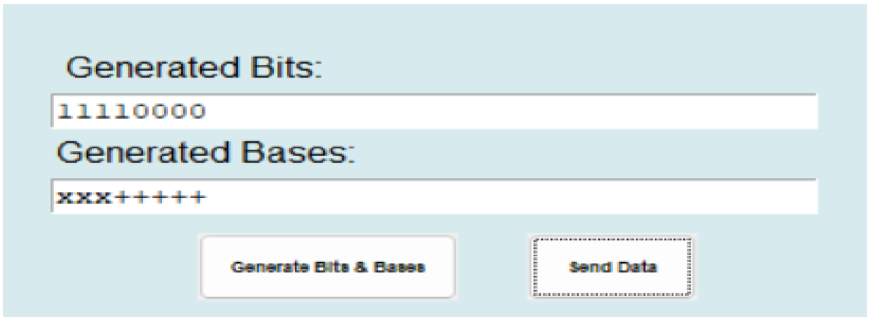


Fig 2 Shows the Desktop Application of Quantum Crypytography via Email

Chapter 7
SNAPSHOTS

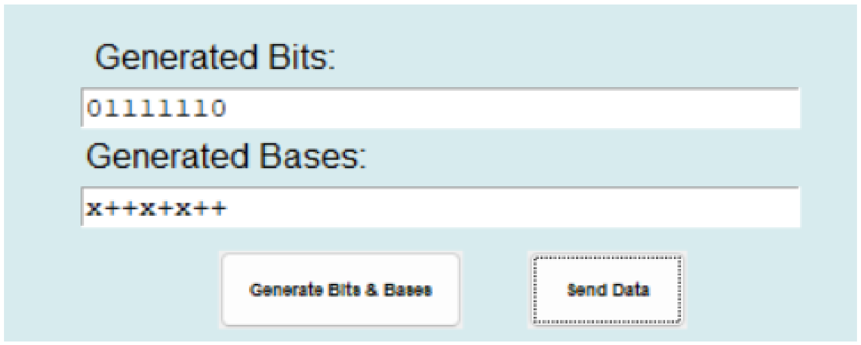
1. Bits and bases Generation (QKD Process):

- A screenshot showing the generation of random bits and bases by both sender (Alice) and receiver (Bob).
- Representation of matched bases between Alice and Bob and the creation of a shared key.



Generated Bits:
11110000
Generated Bases:
xxx+++++

Fig 3 shows generation of bits and bases by person A

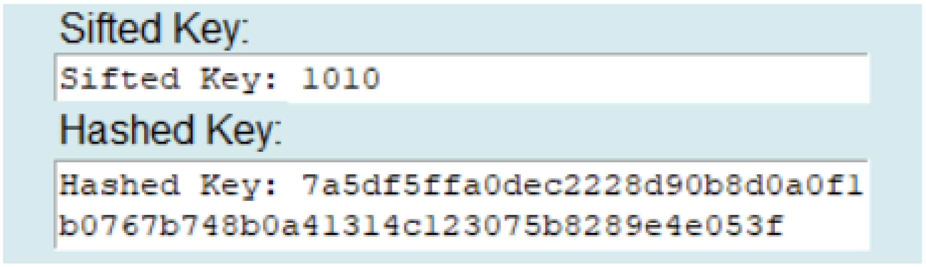


Generated Bits:
01111110
Generated Bases:
x++x+x++

Fig 4 shows generation of bits and bases by person B

2. Sifted key and Hash value Generation :

- The sifted key is used as a shared secret for encrypting and decrypting sensitive data, such as images, text, audio, or video files.
- The hash value ensures that any tampering with the sifted key can be detected

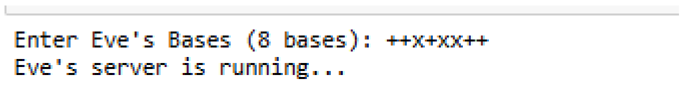


Sifted Key:
```
Sifted Key: 1010
```
Hashed Key:
```
Hashed Key: 7a5df5ffa0dec2228d90b8d0a0f1
b0767b748b0a41314c123075b8289e4e053f
```

Fig 5 shows generation of sifted key and hash value

3. Bases Generation by Eavesdropper:

- Just like Alice and Bob, Eve generates a sequence of random bases (rectilinear or diagonal).
- These bases are independent of the bases used by Alice or Bob.

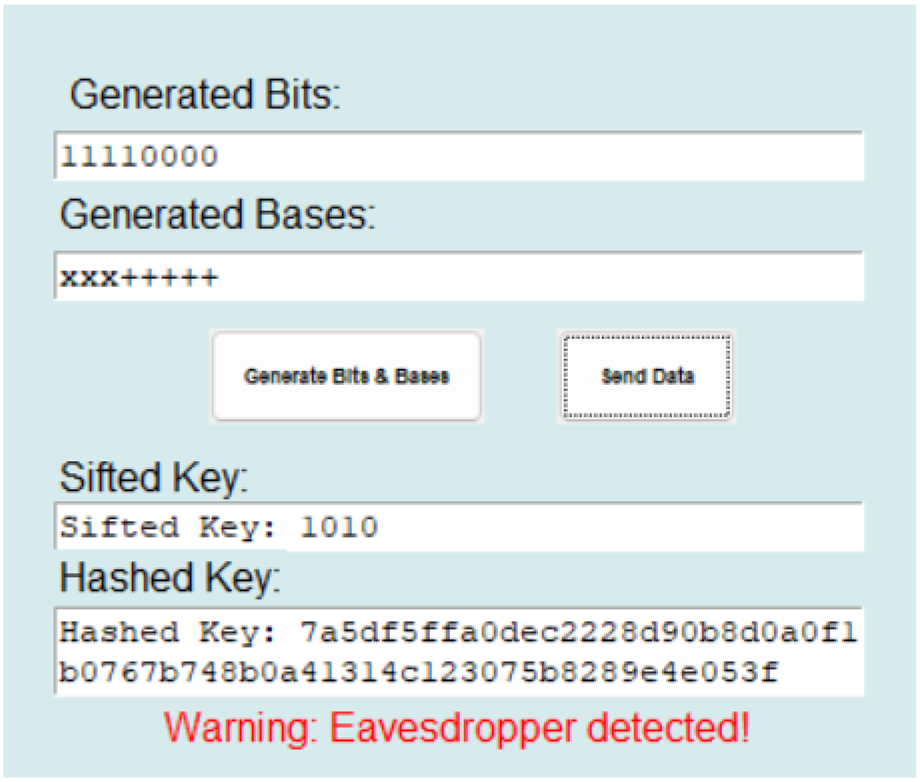


```
Enter Eve's Bases (8 bases): ++x+xx++
Eve's server is running...
```

Fig shows generation of bases by eavesdropper

4. Eavesdropping Detection:

- Eve intercepts the qubits sent by Alice and measures them using her randomly chosen bases.
- If Eve's chosen basis matches Alice's encoding basis, she measures the correct bit value.
- If the bases do not match, Eve’s measurement introduces errors into the communication.



Generated Bits:

```
11110000
```

Generated Bases:

```
xxx+++++
```

Sifted Key:

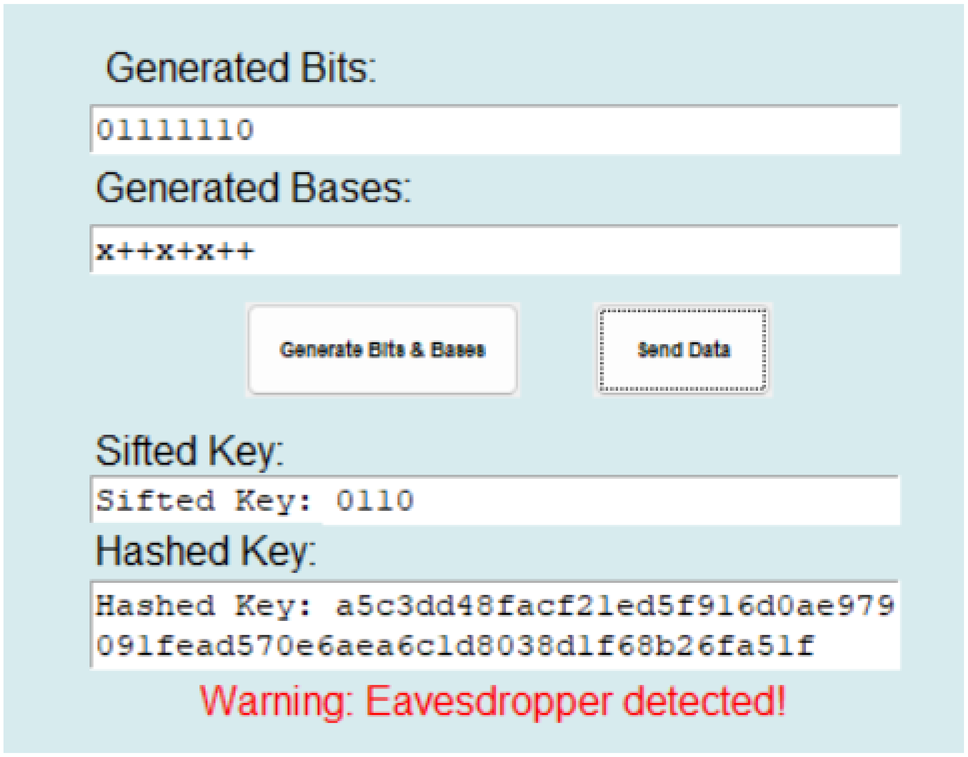
```
Sifted Key: 1010
```

Hashed Key:

```
Hashed Key: 7a5df5ffa0dec2228d90b8d0a0f1
b0767b748b0a41314c123075b8289e4e053f
```

Warning: Eavesdropper detected!

Fig 6 shows Eavesdropper detected on person a application


Generated Bits:

01111110

Generated Bases:

x++x+x++

Sifted Key:

Sifted Key: 0110

Hashed Key:

Hashed Key: a5c3dd48facf21ed5f916d0ae979091fead570e6aea6c1d8038d1f68b26fa51f

Warning: Eavesdropper detected!

Fig 7 shows Eavesdropper detected on person B application

5. Shared Key

The **shared key** is the outcome of the sifting and hashing processes between Alice and Bob. After comparing their bases and filtering out mismatched bits, the remaining matched bits form the sifted key. This key is then hashed (e.g., using SHA-256) to generate a secure cryptographic key. The shared key plays a central role in encrypting and decrypting data securely.

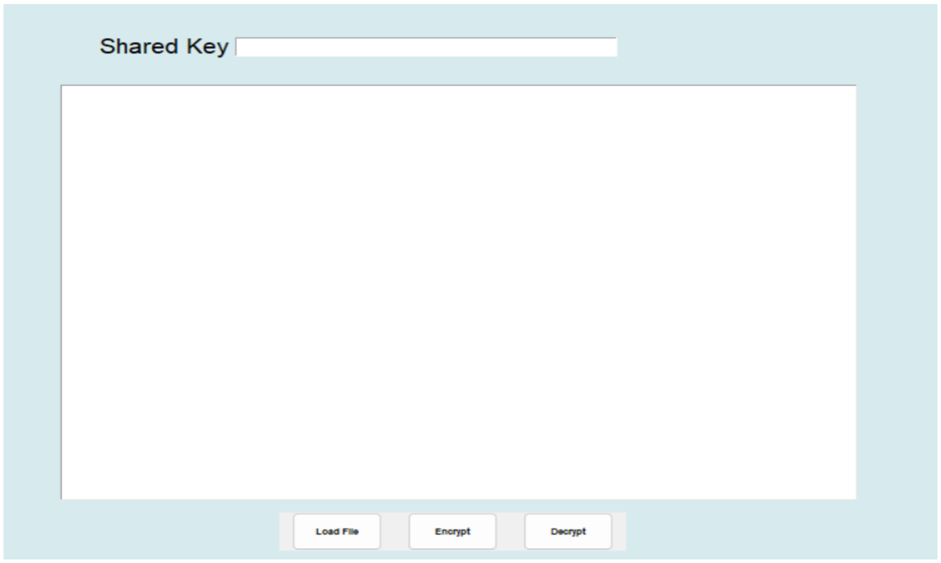
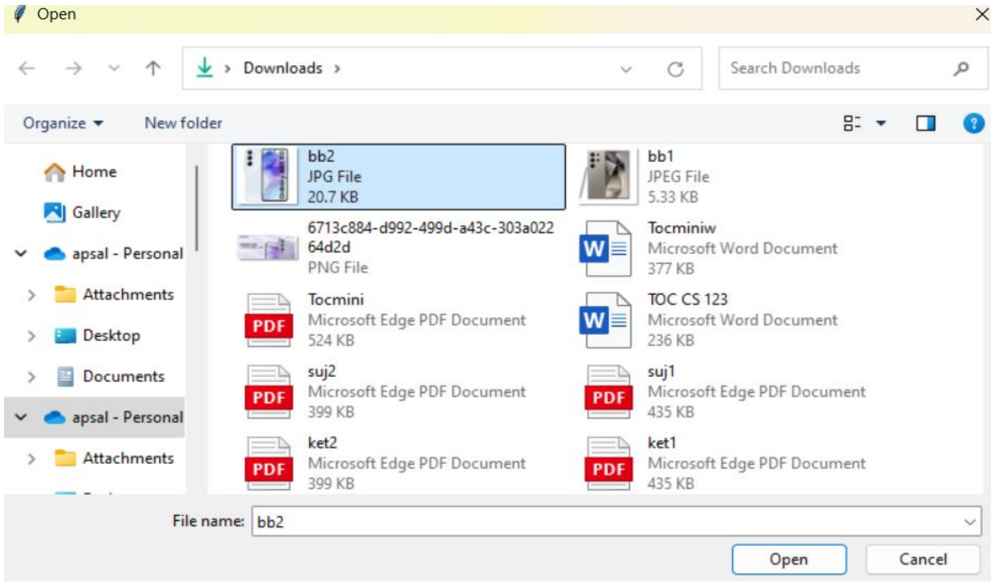


Fig 8 shows entering of shared key

6. Encryption Process:

- The user selects a file (text, image, audio, or video) to encrypt.
- The shared key generated via QKD is used in an AES encryption algorithm.
- AES operates on blocks of data, ensuring secure transformation from plaintext to ciphertext.
- The encrypted file is saved in a secure format, ready for transmission.



Open

Downloads › | Search Downloads

Organize ▾ New folder

Home
Gallery
apsal - Personal
Attachments
Desktop
Documents
apsal - Personal
Attachments

bb2 JPG File 20.7 KB
bb1 JPEG File 5.33 KB
6713c884-d992-499d-a43c-303a02264d2d PNG File
Tocminiw Microsoft Word Document 377 KB
Tocmini Microsoft Edge PDF Document 524 KB
TOC CS 123 Microsoft Word Document 236 KB
suj2 Microsoft Edge PDF Document 399 KB
suj1 Microsoft Edge PDF Document 435 KB
ket2 Microsoft Edge PDF Document 399 KB
ket1 Microsoft Edge PDF Document 435 KB

File name: bb2

Open Cancel

Fig 9 shows selecting the data to be encrypted

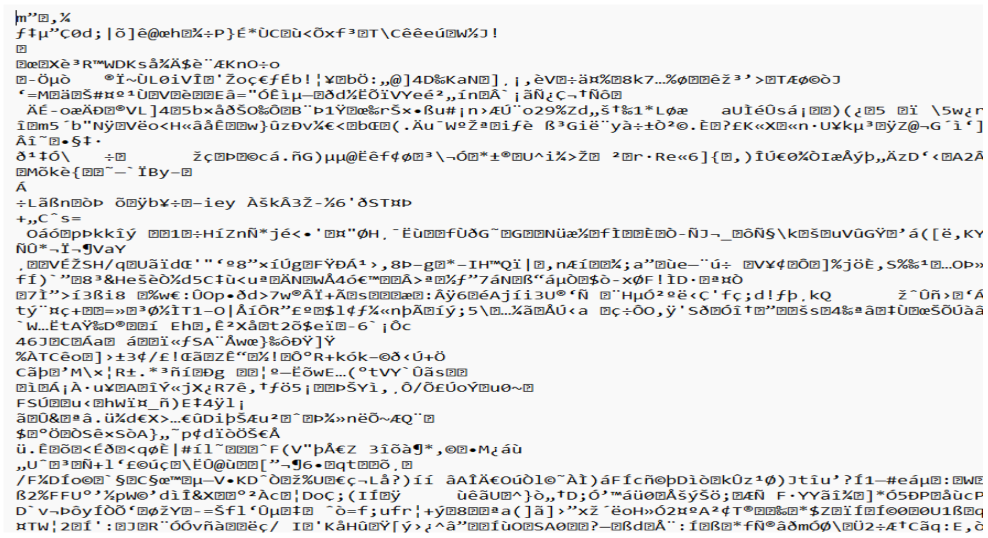
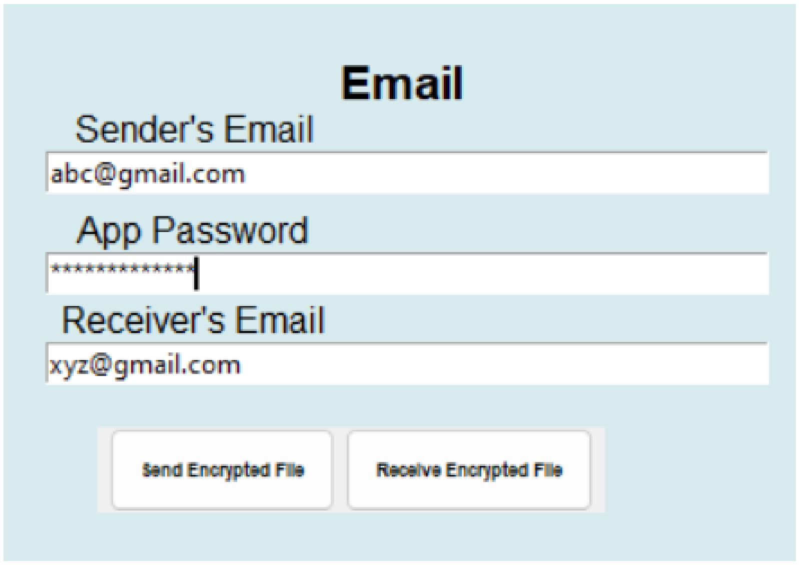


Fig 10 shows selected data is converted to cipher text

7. Email Transmission:

- **User Authentication:**The sender logs in using their email ID and app password to ensure authenticity.
- **Attachment Preparation**:The encrypted file is attached to the email.
- **Email Sending**:The email is sent using Python libraries like smtplib or modern email APIs.
- **Recipient Notification:**The recipient is informed to retrieve the email and it is saved in specific folder and decrypt the attachment using the shared key.



abc@gmail.com

****

xyz@gmail.com

Fig 11 shows email functionality

8. Decryption Process:

- **Retrieve Ciphertext:**The recipient downloads the encrypted file attached to the email.
- **Input Shared Key:**The recipient uses the shared key generated during the QKD process.
- **Symmetric Key Decryption**:Using the same AES algorithm, the ciphertext is decrypted into plaintext.
- **Output Original Data:**The decrypted data (text, image, audio, or video) is displayed or saved.

Chapter 8
CONCLUSION AND FUTURE SCOPE

Conclusion:

The project on quantum cryptography for securing data transfer via email has successfully demonstrated the application of quantum key distribution (QKD) in encrypting and securely transmitting data such as text, images, audio, and video between devices. The system uses QKD to generate a shared secret key between the sender and the receiver, which is then used to encrypt and decrypt the data, ensuring confidentiality and integrity.

By integrating email functionality, the system provides a user-friendly interface for transmitting encrypted data, making it practical for real-world applications. The implementation ensures that even in the presence of an eavesdropper, any interception will be detected, thus safeguarding sensitive information from unauthorized access.

Future Scope:

The future scope of quantum cryptography for securing data transfer is vast and promising. As quantum technologies continue to advance, there are opportunities to expand the system's capabilities to address growing security needs. One significant area for future development is scalability. While the current system functions well for communication between two devices, it must evolve to support multiple users, enabling secure group communications.

Additionally, integrating advanced quantum cryptographic protocols like BB84 and E91 would enhance security and robustness, providing stronger encryption mechanisms for sensitive data.

As quantum computing becomes more powerful, it may eventually break traditional cryptographic algorithms. To mitigate this risk, incorporating post-quantum cryptographic algorithms, such as lattice-based cryptography, will ensure that the system remains secure even in the face of quantum computing advancements.

Furthermore, optimizing the speed of key generation and distribution will be crucial for real-time applications, ensuring that encrypted communication, such as video or voice calls, can be processed without delays.Enhancing user experience through an intuitive interface, integrating digital signatures for data authenticity, and supporting cloud platforms for encrypted file sharing are key areas for improvement.

Inclusion of cross-platform compatibility and blockchain for secure data logging could make the system more accessible, versatile, and trustworthy, paving the way for its use in various industries like finance, healthcare, and government communications.

References

[1] **S Pirandola, UL Andersen, L Banchi… -** Advances in optics …, 2020 - opg.optica.org
https://opg.optica.org/abstract.cfm?uri=aop-12-4-1012

[2] **C Portmann, R Renner -** Reviews of Modern Physics, 2022 – APS, Security in quantum cryptography https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.94.025008

[3] **D Joseph, R Misoczki, M Manzano, J Tricot**… - Nature, 2022 - nature.com, Transitioning organizations to post-quantum cryptographyhttps://www.nature.com/articles/s41586-022-04623-2

[4] **BM Gupta, SM Dhawan**… - Science & Technology …, 2021 - Taylor & Francis, Quantum cryptography research: A scientometric assessment of global publications 2019
https://www.tandfonline.com/doi/abs/10.1080/0194262X.2021.1892563

[5] **F Grasselli -** Quantum science and technology. Cham: Springer, 2021 - SpringerQuantum cryptography https://link.springer.com/content/pdf/10.1007/978-3-030-64360-7

[6] **J Yin, YH Li, SK Liao, M Yang, Y Cao, L Zhang, JG Ren**… - Nature, 2020 - nature.com
Entanglement-based secure quantum cryptography over 1,120 kilometres
https://www.nature.com/articles/s41586-020-2401-y

[7] **G Alagic, G Alagic, J Alperin-Sheriff, D Apon, D Cooper**… - 2019 - tsapps.nist.gov
https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927303

[8**] AP Bhatt, A Sharma -** Journal of Electronic Science and Technology, 2019 – Elsevier
https://www.sciencedirect.com/science/article/pii/S1674862X19300345

[9] **B OzÇakmak, A Ozbllen**… - 2019 IEEE international …, 2019 - ieeexplore.ieee.org
https://ieeexplore.ieee.org/abstract/document/9006238/

[10] **M Kumar, P Pattnaik -** 2020 IEEE High Performance Extreme …, 2020 - ieeexplore.ieee.org
https://ieeexplore.ieee.org/abstract/document/9286147/

[11] **Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J. Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, Seth D. Schoen, and Brad Warren**. 2019. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. In ACM CCS 2019, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 2473 2487. https://doi.org/10.1145/3319535.336319

[12] **Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob**

Lichtinger, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. 2022. NISTIR 8413: Status Report on the Third Round of the NIST Post-Quantum Cryptography StandardizationProcess. https://doi.org/10.6028/NIST.IR.8413-upd1

[13] **Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann.** 2021. Post-quantum WireGuard. In 2021 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 304–321 https://doi.org/10.1109/SP40001.2021.00030

[14] **Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila**. 2022. Post-quantum Asynchronous Deniable Key Exchange and the Signal Handshake. In PKC 2022, Part II (LNCS), Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe (Eds.), Vol. 13178. Springer, Heidelberg, 3–34. https://doi. org/10.1007/978-3-030-97131

[15] **Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé.** 2018. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018. IEEE, 353 367.
https://doi.org/10.1109/EuroSP.2018.00032

[16] **Leo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé**e 2018. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. IACR TCHES 2018, 1 (2018), 238–268. https://doi.org/ 10.13154/tches.v2018.i1.238-268
https://tches.iacr.org/index.php/TCHES/artic le/ view/839.
https://doi.org/10.1109/SP.2015.22 232–249.

[17] **Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsa tiansup, Tanja Lange, Adrian Marotzke, Bo Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang.** 2020. NTRU Prime. Technical Report. National Institute of Standards and Technology. available at 7 IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol.14, No 3 Sep 2024 https://csrc.nist.gov/projects/post-quantumcryptography/post-quantumcryptographystandardization/round-3-submissions.

[18] **Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe.** 2019. The SPHINCS+ Signature Framework. In ACM CCS 2019, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACMPress,2129–2146.
https://doi.org/10.1145/3319535.3363229

[19**] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila**. 2017. Tran_sitioning to a Quantum-Resistant Public Key Infrastructure. In Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Tanja Lange and Tsuyoshi Takagi (Eds.). Springer, Heidelberg, 384 405.
https://doi.org/10.1007/978-3-31959879-6_22