CSE7101- Capstone Project
Review-1

QUANTUM SECURE EMAIL CLIENT APPLICATION

Batch Number : CSE_229

Roll Number	Student Nane
20221CSEO261	ANKE MOUNIKA
20221CSE0262	POOLA DHATHRI
20221CSE0290	S.NIKHITHA

Under the Supervision of,

Dr . Jayavadivel Ravi
Associate Professor
School of Computer Science and Engineering
Presidency University

Name of the Program : Computer Science And Engineering
Name of the HOD : Dr . Asif Mohamed
Name of the Program Project Coordinator : Dr . Jayavadivel Ravi
Name of the School Project Coordinators : Dr. Sampath A K , Dr. Geetha A

Content

- Problem Statement
- Objectives
- Background and Related work for title Selection
- Analysis of Problem Statement
- Innovation or Novel Contributions
- GiT-hub Link
- Timeline of the Project
- References

Problem Statement Number: PSCS_178_

Organization: Indian Space Research Organization (ISRO)

Category (Hardware/ Software) : Software

Problem Description:

As quantum computers advance, traditional encryption methods used in email communication (like RSA and ECC) are becoming vulnerable to quantum attacks. This creates a serious risk to the confidentiality and integrity of sensitive information. There is a growing need for a secure email client that uses post-quantum cryptographic algorithms to ensure long-term protection against both current and future quantum threats.

Problem Statement :

Current email systems largely rely on classical asymmetric algorithms that are at risk from future quantum attacks. There is a need for email software that:

- Establishes session keys using post-quantum algorithms so intercepted messages cannot be retroactively decrypted once quantum computers exist.
- Integrates post-quantum key exchange into a usable email flow (compose → send → receive → decrypt) with minimal UX friction.
- Provides secure, auditable key management and message storage (local or server) while supporting common features (attachments, metadata).
- Demonstrates practical tradeoffs (performance, message size, interoperability) and documents threats and mitigations.

Objectives :

1. Primary objectives:
2. Implement end-to-end encrypted email exchange between registered users using a post-quantum KEM for key establishment and AES-GCM (or equivalent AEAD) for message encryption.
3. Build authentication for users (registration/login) and secure key storage (encrypted at rest).
4. Provide a simple, usable UI for composing, sending, receiving, and viewing encrypted emails and attachments.
5. Evaluate performance (latency, ciphertext size) and security (threat model analysis, resistance to common attacks).

Secondary objectives:

1. Support key rotation and message versioning.
2. Produce documentation and a report comparing classical vs post-quantum workflow , including limitations and deployment considerations.

Background For Title Selection :

Email is one of the most widely used forms of communication for personal, professional, and governmental purposes. Traditional email security relies on cryptographic algorithms like RSA and ECC, which are secure against classical computers but vulnerable to attacks from emerging quantum computers. Quantum algorithms such as Shor’s can easily break these systems, creating a serious threat to the confidentiality of sensitive communications.

To overcome this, Post-Quantum Cryptography (PQC) provides algorithms that remain secure even against quantum computers. A Quantum Secure Email Client Application uses PQC-based key exchange, digital signatures, and strong encryption to ensure long-term privacy and integrity of emails. This not only protects sensitive information today but also future-proofs communication against quantum-era threats.

Analysis of Problem Statement

Technology Stack

1.Backend Framework:

- Flask (Python 3.11)

2.Frontend:

- Jinja2 templates + Bootstrap/Tailwind (UI)

3.Database:

- SQLite (for prototype, scalable later)

4.Crypto Libraries:

- python-oqs (for post-quantum KEMs, e.g., Kyber)

- cryptography (for AES-GCM, hashing)

5.Authentication:

- Flask-Login + Werkzeug password hashing

6.OS Support:

- Windows (development), extendable to Linux/Mac

Analysis of Problem Statement (contd...)

Requirements:

- **software Requirements:**

1. Operating System : Windows Only
2. Processor : 15 and above
3. Ram : 4gb and above
4. Hard Disk : 10 GB and above

Flow Chart :

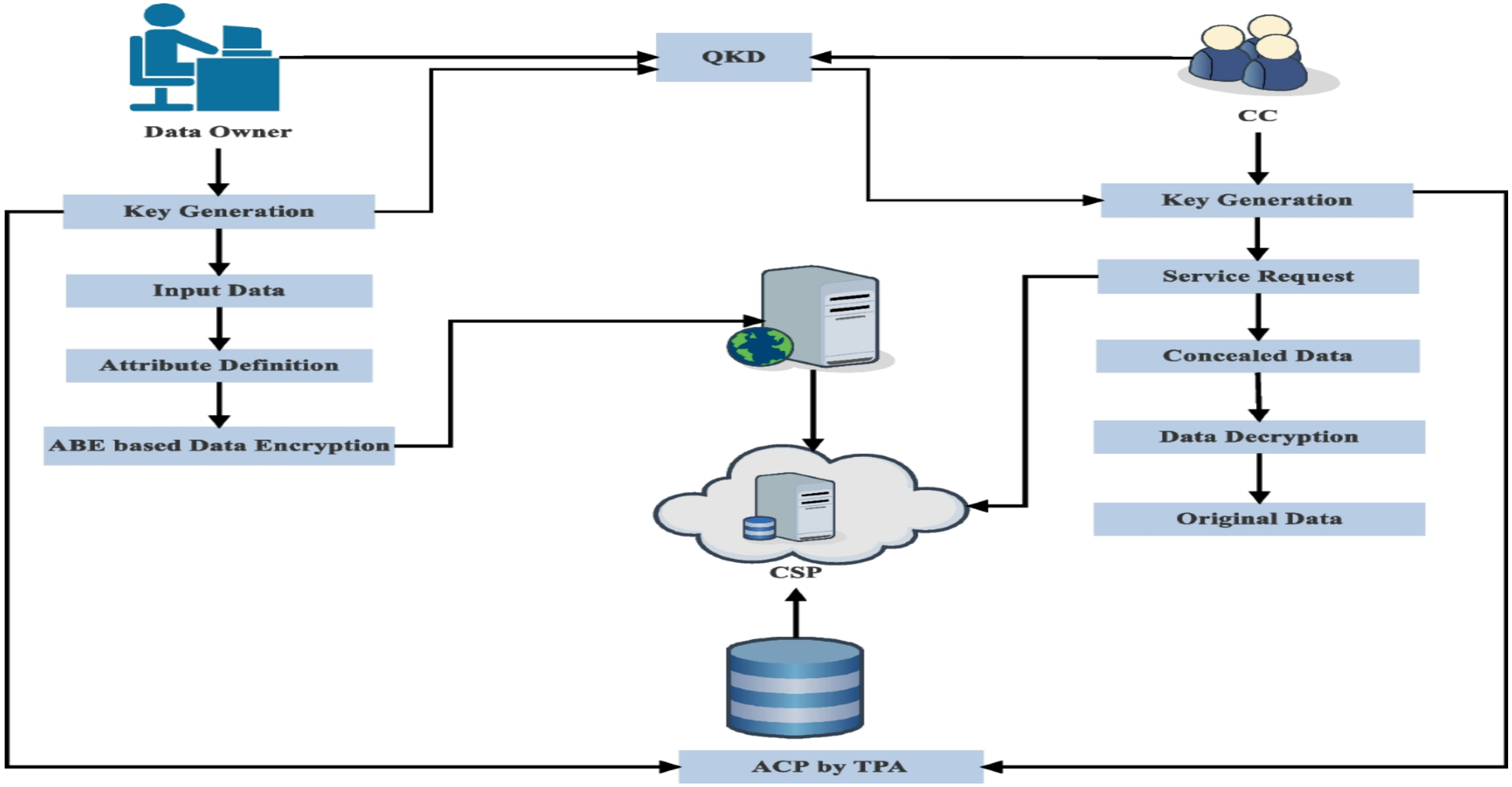
1.User Registration:
- Create account, system generates PQ keypair.
- Public key → DB, Private key → encrypted storage.

2.User Login:
- Authenticate with password.
- Unlock private key (decrypt).

3.Sending (Compose):
- Fetch recipient’s PQ public key.
- Perform KEM Encapsulation → shared secret + ciphertext.
- Derive AES-GCM key from shared secret.
- Encrypt email + attachments.
- Store encrypted message + KEM ciphertext in DB.

4.Receiving (Inbox):
- Fetch encrypted message from DB.
- Recipient uses private PQ key to decapsulate KEM ciphertext.
- Derive AES key → decrypt email.
- Display plaintext email securely.

Flow chart :


Data Owner
QKD
CC
Key Generation
Input Data
Attribute Definition
ABE based Data Encryption
Key Generation
Service Request
Concealed Data
Data Decryption
Original Data
CSP
ACP by TPA

Innovation or Novel Contributions :

Title	Author	Key contribution
Hybrid Signal protocol for post-quantum email encryption*	Sara Stadler, Vitor Sakaguti, Harjot Kaur, Anna Lena Fehlhaber*	Hybrid adaptation of Signal protocol for email encryption with post-quantum security*, demonstrating how to adapt secure messaging protocols for email in a PQC context.
Secure Composition of Quantum Key Distribution and Symmetric Key Encryption	Kunal Dey, Reihaneh Safavi-Naini	Formal security analysis combining Quantum Key Distribution (QKD) with symmetric key encryption*, introducing quantum-enabled KEM and hybrid encryption suitable for email content and key exchange.
An Experimentally Validated Feasible Quantum Protocol for Identity-Based Signature with Application to Secure Email Communication	Tapaswini Mohanty, Vikas Srivastava, Sumit Kumar Debnath, et al	Quantum identity-based signature protocol experimentally validated*, enabling secure email authentication without relying heavily on traditional PKI or certificates.

Innovation or Novel Contributions :

Quantum-resistant End-to-End Secure Messaging and Email Communication	Christoph Döberl, Wolfgang Eibner, Simon Gärtner, et al	Integration of PQC and QKD in end-to-end secure messaging and email communication*, providing a case study (Delta Chat) with insights into implementation tradeoffs.
Anonymous, Robust Post-Quantum Public Key Encryption	Paul Grubbs, Varun Maram, Kenneth G. Paterson	Design of anonymous and robust post-quantum public key encryption schemes*, addressing metadata privacy and sender identity protection.
A Hybrid Encryption Framework Combining Classical, Post-Quantum, and QKD Methods*	Amal Raj, Vivek Balachandran	Framework combining classical cryptography, PQC, and QKD in hybrid encryption*, useful for transition strategies in email systems toward post-quantum readiness.
Post-Quantum Encryption Algorithms	Peter Pekarčík, et al	Comprehensive survey of post-quantum encryption algorithms* like BIKE, Classic McEliece, HQC, helping in algorithm selection for encryption and signatures.

Innovation or Novel Contributions :

The Impact of Quantum Computing on Present Cryptography	Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang	Background on quantum computing's impact on classical cryptography*, explaining vulnerabilities of RSA, ECC and overviewing post-quantum alternatives for threat modeling.
Quantum-Secured Email: Revolutionizing Confidential Communication with Quantum Key Distribution*	Y. Vasantha, T. Ravinder, A. Srikanth	Use of QKD for email encryption and authentication*, exploring how QKD can secure emails and attachments in a quantum-secured communication environment.
Quantum-Secure Email: A Post-Quantum Cryptographic Approach	R. Kumar, S. Gupta, and A. Sharma	The paper analyzes quantum computing threats to classical encryption and introduces post-quantum cryptographic algorithms to secure email communication. It proposes hybrid encryption models for seamless integration and discusses practical implementation challenges for building quantum-secure email clients.

Innovation or Novel Contributions :

- A Quantum Secure Email Client Application is designed to future-proof communication against the threat of powerful quantum computers, specifically focusing on defeating the "Harvest Now, Decrypt Later" attack.
- Uses a combination of a Post-Quantum Cryptography (PQC) Key Encapsulation Mechanism (KEM) (e.g., NIST-standardized ML-KEM / CRYSTALS-Kyber) and a classical key exchange (e.g., ECDH).
- The session key is a combination of both secrets, ensuring the connection remains secure even if one of the underlying algorithms is broken. This prevents the "Harvest Now, Decrypt Later" attack.
- Employs a PQC Digital Signature Algorithm (e.g., NIST-standardized ML-DSA / CRYSTALS-Dilithium) to authenticate the sender's identity.
- This ensures the recipient can verify that the email truly came from the claimed sender, as classical signatures (like RSA and ECC) are vulnerable to Shor's algorithm.
- While symmetric crypto is less vulnerable than public-key crypto, the doubled key length is a mitigation strategy to counter the speed-up provided by Grover's quantum search algorithm.

Github Link

The Github link provided should have public access permission.

GitHub Link : https://github.com/Dhathri-Poola-22

Security Configuration :

1. Cryptographic Configuration (Quantum Security)

. **The primary security measure is the use of Hybrid Cryptography to provide protection against both current classical attacks and future quantum attacks (the "harvest now, decrypt later" threat).**

. **The session key is a combination of both secrets, ensuring the connection is secure as long as *at least one* algorithm holds up.**

2. Key Management and Storage Security

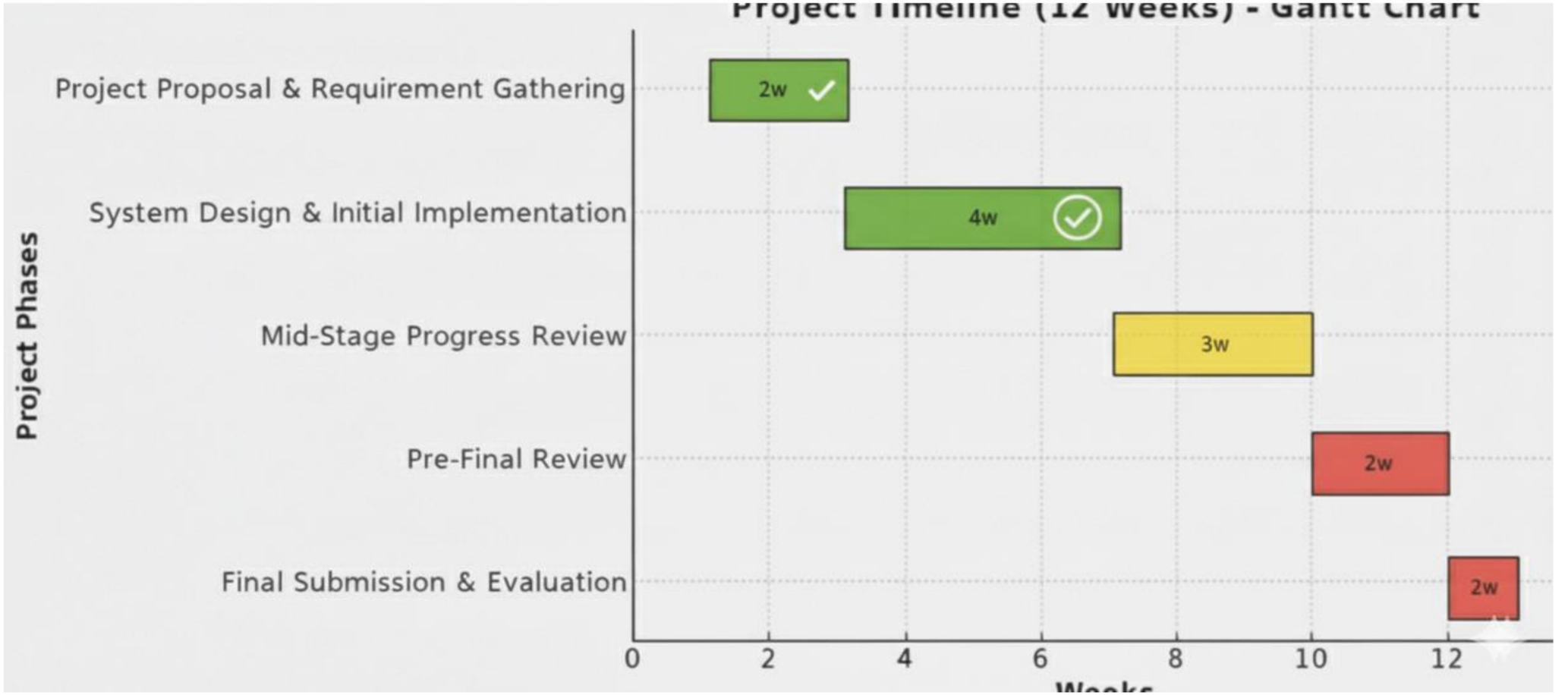
. **Private Key Protection: All PQC private keys (for ML-KEM decapsulation and ML-DSA signing) must be stored encrypted at rest on the user's local device, ideally protected by a master password derived from the user's login passphrase .**

. **Hardware Security Module (HSM): For enterprise or high-security deployments, integrate support for HardwareSecurity Modules (HSMs) to store and perform cryptographic operations with the private keys, preventing their extraction.**

3. Application and Software Security

. **Authentication:** Mandate **Multi-Factor Authentication (MFA)** for user login to protect the account that guards the local private keys.

Timeline of the Project (Gantt Ch)



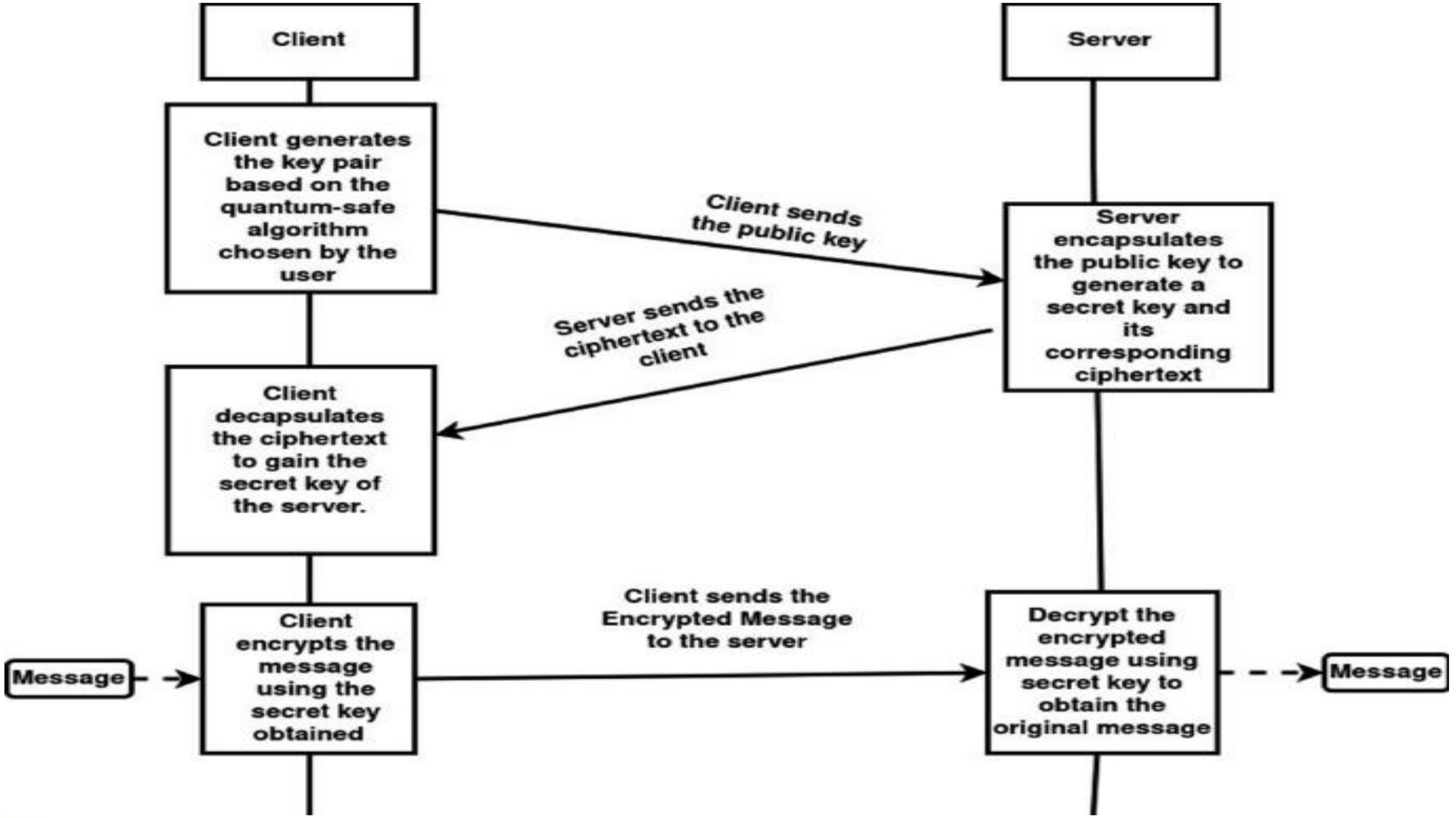
References :

1. https://www.jetir.org/papers/JETIR2504198.pdf
2. https://ijcnis.org/index.php/ijcnis/article/view/8024/2208
3. https://ijcnis.org/index.php/ijcnis/article/view/8024/2208
4. https://www.youtube.com/watch?v=vDcQ2vc4yQI
5. https://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf
6. https://thequantuminsider.com/2024/03/18/q-got-mail-tuta-launches-post-quantum-crypto
7. https://www.google.com/search?q=https://ieeexplore.ieee.org/abstract/document/101037
8. https://www.mdpi.com/1999-4893/16/2/97
9. https://kinsta.com/blog/secure-email-providers/
10. https://cyberinsider.com/email/best-encrypted-email/
11. https://eviden.com/solutions/cybersecurity/post-quantum-security-pqc/
12. https://www.ibm.com/quantum/quantum-safe
13. https://www.encryptionconsulting.com/services/post-quantum-cryptographic-advisory-serv
14. https://quantum.microsoft.com/en-us/vision/quantum-cryptography-overview
15. https://pqshield.com/

References Countinue…

16. https://cloud.google.com/security/resources/post-quantum-cryptography
17. https://www.idquantique.com/quantum-safe-security/quantum-key-distribution/
18. https://content.pqshield.com/secure-messaging-in-a-post-quantum-world
19. https://www.entrust.com/solutions/post-quantum-cryptography
20. https://www.ssh.com/blog/quantum-safe-email-security-with-salax-secure-mail

Block diagram :




Thank
You!