PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

QMail: Quantum Secure Email Client Application

A PROJECT REPORT

Submitted by

S Pranav Roy- 20221CSE0407

Achal K A – 20221CSE0422

Rudraraju Satvik Varma- 20221CSE0185

Under the guidance of,

Dr. JOSEPH MICHAEL JERARD V

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report "QMail: Quantum Secure Email Client Application" is a bonafide work of "S Pranav Roy (20221CSE0407), Achal K A (20221CSE0422), Rudraraju Satvik Varma (20221CSE0185) ", who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING during 2025-26.

Dr. Joseph Michael Jerard V
Project Guide
PSCS
Presidency University

Mr. Muthuraju V
Program Project Coordinator PSCS
Presidency University

Dr. Sampath A K
Dr. Geeta A
School Project Coordinators, PSCS
Presidency University

Dr. Blessed Prince
Head of Department
PSCS
Presidency University

Dr.Shakkeera L
Associate Dean
PSCS
Presidency University

Dr.Duraipandian N
Dean
PSCS &PSIS
Presidency University

Examiners:

SL No	Name	Signature	Date
1.			
2.			

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B.Tech in COMPUTER SCIENCE ENGINEERING, at Presidency University, Bengaluru, named S Pranav Roy, Achal K A, Rudraraju Satvik Varma, hereby declare that the project work titled "QMail: Quantum Secure Email Client Application" has been independently carried out by us and submitted in partial fulfilment for the award of the degree of B.Tech in COMPUTER SCIENCE ENGINEERING, during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

S Pranav Roy	USN: 20221CSE0407
Achal K A	USN: 20221CSE0422
Rudraraju Satvik Varma	USN: 20221CSE0185

PLACE: BENGALURU

DATE: 1-December-2025

ACKNOWLEDGEMENT

For completing this project work, We/I have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.

I would like to sincerely thank my internal guide **Dr. Joseph Michael Jerard V, Professor**, Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Blessed Prince, Professor, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators**, Dr. Muthuraju V, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

S Pranav Roy

Achal K A

Rudraraju Satvik Varma

Abstract

Commercial Quantum Computers are inevitable and it is just a question of time before they are integrated in our society. It poses a threat to current encryption algorithms, which are effective today but will be obsolete to Shor's and Grover's algorithms (Quantum algorithms), compromising confidentiality, integrity, and authenticity. Securing communication channels must be a top priority, and since email is the most widely used communication method, it must be addressed first. Our QMail, a Quantum Secure Email Client Application (QSECA), is designed to withstand both classical and quantum-level attacks. It is based on a hybrid structure of QSECA that combines Quantum Key Distribution (QKD) for session key with Post-Quantum Cryptography (PQC) technology for authentication and metadata protection. A user-friendly interface is implemented in the prototype to choose between three levels of encryption, including OTP. Benchmarks on the prototype show the trade-off in latency, throughput, and encryption overhead compared to conventional email service. The results show that QMail ensures end-to-end security and provides resilience against adversaries with quantum capabilities. QMail establishes a foundation for a scalable, quantum-safe communication platform and is a step towards secure communication in the era of 6G and quantum internet.

Table of Contents

List of Figures

List of Table

Abbreviations

Abbreviation	Full Form
PQC	Post-Quantum Cryptography
KEM	Key Encapsulation Mechanism
OTP	One-Time Pad
QKD	Quantum Key Distribution
AES	Advanced Encryption Standard
GCM	Galois/Counter Mode
API	Application Programming Interface
OAuth	Open Authorization
JWT	JSON Web Token
PK	Primary Key
FK	Foreign Key
ORM	Object Relational Mapper
SQL	Structured Query Language
TLS	Transport Layer Security
HTTPS	Hypertext Transfer Protocol Secure

Chapter 1

INTRODUCTION

1. Introduction

Email remains the most used communication system by companies and governments. Over the decade, it has evolved from desktop-based systems to cloud-based systems, which are supported worldwide. In our study, all the technologies that the email uses, like cloud servers, mobile synchronization, better interfaces, and AI-based spam filters, are recently developed technologies over the past decade. While the rest technological stack for emails grew significantly over the decade, cryptographic changes were never made. Emails still use traditional methods like RSA and ECC, which can not protect the data from quantum attacks over time. From studying Shor's and Grover's algorithms, we could understand that performing factorization of large integers and probabilistic searches could be done really fast with better confidentiality, integrity, and authenticity [1]-[3].

With the growing understanding of quantum computing, it introduced the "Harvest Now, Decrypt later" concept in short HNDL. This concept explains how encrypted communications of today can be stored and decrypted later when quantum systems become available. To counter this, researchers around the world started to work on two promising approaches: The Quantum Key Distribution (QKD) and Post Quantum Cryptography (PQC). QKD inherits principles of quantum mechanics along the way from the no-cloning theorem to secure key exchange, which helps in intrusion detection by changing the quantum signals [3], [9], [10]. On the other hand, the PQC algorithm works based on complex mathematical problems (such as lattice-based and hash-based schemes) that are computationally not solvable even for quantum computers [5].

1.1 Background and Related Work

Traditional e-mail security is based on RSA and Elliptic Curve Cryptography, or ECC. While the schemes are secure today, the advent of quantum computers threatens to break those

schemes using algorithms like Shor's. Thus, researchers have come up with PQC methods such as Kyber, NTRU, and Dilithium that have been designed to resist quantum attacks.

Examples of encrypted e-mail services include but are not limited to ProtonMail, Tutanota, and PGP/S-MIME. While much more private, these face major obstacles in wide diffusion due to the following:

- Complex key handling turns non-technical users away.
- Poor compatibility with such widely used services as Gmail and Yahoo Mail.
- Limited integration with popular desktop e-mail clients.

A development in this direction is Quantum Key Distribution, which is a technique promising secure key exchanges using the very principles of quantum mechanics. However, secure QKD suffers from the following:

- It requires special hardware, such as optical fibers and satellites.
- Limited scalability outside of controlled environments.
- Incompatible with most standard e-mail protocols, such as SMTP and IMAP.

No major email client currently combines QKD and PQC into one hybrid solution balancing future-proof security against usability.

1.2 Statistics and Need for the Project

E-mail is still the backbone of digital communication, and according to a Statista estimate, 2023, there are more than 4.4 billion users around the world in 2025. According to IAMAI 2022, over 90% of the businesses and government offices in India still use e-mail for official correspondence. (IAMAI, 2022).

Meanwhile, security threats are on the rise: one-quarter of all cyber incidents in 2023 related to e-mail at an average breach cost of $4.45 million, with e-mail compromise often included as one of the attack vectors.

This will change as quantum computing improves, and the risks will also increase. "Harvest Now, Decrypt Later" attacks, described in NIST 2022, are where encrypted emails are collected today and decrypted at some future date using quantum algorithms.

Why QMail is Needed

This situation shows two main gaps:

1. Dependence vs. vulnerability — Email is indispensable, yet current RSA/ECC encryption will eventually fail against quantum computers (Shor, 1994).
2. Usability gap: until recently, secure e-mail tools like ProtonMail and Tutanota suffered from either complicated key management or a lack of integration with Gmail or Yahoo.

QuMail addresses these issues by combining post-quantum cryptography (Kyber), quantum key distribution (QKD), and one-time pads into a practical, Outlook-like client that delivers both security and usability.

1.3 Prior Existing Technologies

1. Protonmail & Tutanota: These services are, in general, email providers with on-board encryption, but they are not cross-platform; they are confined within their respective ecosystems.
2. Quantum Key Distribution: QKD has been tested on projects ranging from China's Micius satellite to Europe's EuroQCI. This requires very expensive infrastructure to deploy.
3. While post-quantum cryptography does have much to offer in the future, it has just very recently been standardized, and real-world integrations-for example, through email-hardly exist yet.

1.4 Proposed approach

1.4.1 Aim of the project :-

The project, called **QMail**, seeks to create a **quantum-resilient email client** that:

1. Integrates QKD with existing email protocols such as (SMTP/IMAP) for secure communication.
2. Offers three levels of security:
    - Level 1: Standard email without quantum enhancements.
    - Level 2: Hybrid encryption (AES combined with PQC keys).
    - Level 3: Maximum protection with a Quantum One-Time Pad.
3. Improves usability through AI-driven configuration and device-bound decryption.
4. Compatible with major providers, including Gmail and Yahoo, while maintaining security.

1.4.2 Proposed Approach

- Innovative Solution: A hybrid architecture combining QKD for key distribution and PQC for authentication and metadata security.
- Justification:
    - It provides secure eavesdropping detection by using shared symmetric keys.
    - The PQC provides scalability while enabling immediate deployability with NIST-standardized algorithms.
- Feasibility:
    - Technological: Prototype implemented with emulated QKD and PQC libraries.
    - Cost: Feasible with commodity hardware for prototype; real-world deployment scales with QKD infrastructure.
    - Resources: Desktop-based system, extendable to web/cloud in future.
- Impact: Demonstrates a practical, tested model of quantum-secure email, bridging current limitations and future readiness.

1.4.3 Application

While this brings much convenience, the wide usage of email for communication makes people increasingly vulnerable to different forms of sophisticated cyber-attacks, including eavesdropping and man-in-the-middle attacks. Quantum computing will, when fully developed, eventually render existing methods of encryption obsolete. Quantum key distribution allows, in theory at least, an encryption key to be generated with unconditional security. QKD technology has to be integrated with existing email systems and protocols while remaining compatible with popular email services like Gmail and Yahoo. Here, an email client on the user's end will be modular, embedding QKD for secure key exchange in different security levels. Such an email client will securely enable the transmission of emails over untrusted networks without changing the current workflow or user experience of using emails.

1.4.4 Limitation of the proposed approach

While QMail demonstrates that a practical quantum-secure email client is possible, there are also some practical limitations with the approach at this time.

1. QKD is only simulated, not real

Therefore, we use software to simulate Quantum Key Distribution; actual quantum hardware like photon detectors or quantum fiber links are simply not available in most university settings. That means that the simulator cannot provide tests for real quantum problems, such as the loss of a signal or environmental noise

2. PQC adds extra processing time

Kyber512 is heavier compared to the post-quantum cryptography RSA/ECC. Although that is not a big problem with modern laptops, weaker devices could show small delays working with OTP-based encryptions.

3. OTP keys are hard to manage at scale

OTP encryption is very secure but requires a new key every time. Each one of these would demand highly developed infrastructure or hardware modules capable of managing thousands or millions of unique keys, and which are far beyond the scope of this prototype.

4. Email protocols aren’t designed for quantum security

SMTP, IMAP, and even the Gmail API were designed in a time when quantum-resistant technologies did not exist. Both the PQC metadata and hybrid keys add extra size to the message, which can raise compatibility issues with older clients.

5. Strong dependency on internet and cloud APIs

The system requires stable network connectivity in order to allow OAuth login and key exchange. High latency, paired with packet loss, creates a slower and unreliable process of encryption and decryption.

6. Limited real-world testing

This project has been tested in controlled environments by a small number of users. Each of these has yet to be tested for large-scale deployment, use across a number of organizations, and against highly sophisticated adversaries.

7. Quantum-safe communication still depends on classical networks

To be precise, while the encryption is designed to be resistant to quantum attacks, the underlying network is still classical. Since the system cannot guarantee an absolutely quantum-safe internet at the moment, it can't provide hardware-level quantum guarantees either.

1.5 Objectives

- Design and implementation of QMail: Quantum Secure Email Client Application, or QSECA in abbreviation.
- Ensure the confidentiality, integrity, and authenticity of email communications in a post-quantum world.
- Develop a hybrid cryptographic model:

  - Design a Hybrid Cryptographic Model: QKD for secure session key distribution.
  - PQC for authentication and protection of metadata.
  - Implement a desktop prototype providing AES, OTP, and hybrid encryption options in a user-friendly way

- Benchmark latency, throughput, and encryption overhead with respect to standard e-mail systems.

1.6 SDGs

Relevant SDGs for QMail Project

- SDG 9: Industry, Innovation, and Infrastructure
  - Contributing to next-generation communication infrastructure with quantum-secure e-mail
  - Supports innovation in cybersecurity aligned with 6G and the quantum internet.
- SDG 16: Peace, Justice, and Strong Institutions
  - Ensures confidentiality, integrity, and authenticity of communications.
  - Protects individuals, enterprises, and institutions from cyber espionage, data breaches, and surveillance.
- SDG 4: Quality Education
  - Provides a learning prototype for students and researchers in quantum communication and cybersecurity.
  - Bridges the gap between academic research and practical applications.

1.7 Overview of the project

E-mail remains the backbone of digital communication but faces rising threats, especially with the advent of quantum computing capable of breaking RSA and ECC. Existing secure services like ProtonMail and Tutanota have poor usability, and there is limited integration with common providers. PQC and QKD provide much better protection; however, due to technical and

infrastructure barriers, it is still far from widespread use. In 2025, more than 4.4 billion users will be online worldwide, while most organizations still use e-mails in India, thus exposing the growing risk of data breaches and "harvest now, decrypt later" attacks. QMail is a hybrid client using combined PQC and QKD, giving support to several security levels in a very user-friendly way and interoperating with Gmail, Yahoo, and all other services.

Chapter 2

LITERATURE REVIEW

E-mail security has for a long time been based on systems like RSA and elliptic curve cryptography, but those are susceptible to quantum computing. One such proposed solution showed how quantum key distribution and post-quantum cryptography can be combined into one secure email client. This prototype underlined the end-to-end encryption that enhances authentication by integrating with digital signatures, but the prototype testing was performed in a simulated environment and not with actual quantum hardware. It underlined the importance of hybrid approaches and pointed to gaps regarding scalability and formal security validation. [1]

Quantum threats mark a new frontier for messaging applications, too. Having taken a modular approach, the prototype design included identity management, key sharing, and secure message handling. This system was able to prove that QKD combined with PQC can enable secure communication without loss of intrusion detection. While the concept is strong, real-world deployment remains very limited due to high quantum hardware cost and immaturity in certain PQC schemes. [2]

Among the attempts for securing email with quantum technologies, there is a system that generated the encryption keys using QKD and then applied them via AES to protect both email content and attachments. A prototype of a web framework showed that message confidentiality could be ensured with this approach, together with their integrity, and provided a practical model serving in the near future. However, it was neither tested under real conditions nor for enterprise-scale performance. [3]

Others, like ProtonMail, rely on strong PGP and AES end-to-end encryption for users' data and key management. Though these techniques are resistant to contemporary threats, they are susceptible to quantum decryption methods. Though ProtonMail sets the standard in terms of usability for such systems, its dependence on classical encryption makes it one of those motivating needs for quantum-resistant cryptography. [4]

The global drive for quantum-safe cryptography has resulted in the standardization of algorithms such as Kyber for key exchange and Dilithium for digital signatures. These choices have come years after their competitive evaluation against each other in a trade-off of performance and security. Though larger key sizes and migration challenges abound, these standards form a base that quantum-safe systems will be built on and affirm the need to adopt PQC in projects such as QMail. [5]

The fragility of classical channels has also been investigated for QKD systems. Although the quantum link itself is secure due to QKD, classical post-processing typically depends on RSA or ECC that could be compromised by a quantum computer. Experiments using PQC alternatives called Kyber and Dilithium for the purpose of classical authentication demonstrated that such methods can be incorporated without performance degradation and provide a model to harden hybrid systems. [6]

Hybrid approaches have been tested in protocols like TLS and SSH by combining both classical and postquantum algorithms for key exchange. This ensures that, when one of them is compromised, the security provided by the other stays intact. The results were that hybrid handshakes caused only modest increases in message sizes and remained practical. These insights are useful in adapting similar methods to email protocols. [7]

Beyond specific prototypes, broader reviews have stressed the role of quantum communication in future networking: in this sort of vision, coming technologies such as QKD and PQC will provide the building blocks for secure, efficient systems-those that utilize latency gains with increased throughput. While highly theoretical in tone, this view reinforces the position that quantum-safe infrastructure needs to be developed urgently, and well ahead of the plausible advent of large-scale quantum attacks. [8]

Other areas of investigation include quantum communication in the context of next-generation telecommunication and computation systems. Proposals include quantum optical communications and networks to support ultra-secure, high-capacity services. Trials of QKD carried out in a number of regions demonstrate that such systems are by now well beyond the

theory. Applied to email applications, these studies bring into sharp focus the fact that practical, quantum-secure communication tools are realistic but also urgently needed. [9]

The quantum communication surveys give a broader background by reviewing established protocols such as BB84 and real-world deployments of projects across the U. S. and Europe. They review principles like the no cloning theorem and others, which make eavesdropping in quantum systems detectable. Although these works remain conceptual, they lay the basis for grounding the proof that QKD-based secure e-mail systems can never be wiretapped undetectably. [10]

Table 2.1 Summary of Literature Survey

SI NO	Article Title, Published Year, Journal/Conference	Methods	Key Features	Merits	Demerits
1	Quantum Secure Email Client Application, 2025, Journal of Nonlinear Analysis and Optimization	QKD + PQC + Digital Signatures, Python prototype	Combines QKD key exchange with PQC algorithms; secure login and file attachments	Demonstrates quantum-safe end-to-end encryption; user-friendly design	Prototype only, no real QKD hardware; limited scalability/ security proofs
2	Quantum Encrypted Messaging Application, 2024, DICCT Conference	Modular design with QKD+PQC	Identity registrar, friend management, and message handler for secure messaging	Showed feasibility of hybrid QKD + PQC for communicatio n apps	Needs costly hardware; PQC not fully standardize d; limited scalability
3	*Revolutionizing Email Security with QKD*, 2024, Journal of Computational Analysis and Applications	QKD + AES hybrid encryption, Django prototype	End-to-end workflow: registration, key generation, email encryption, decryption	Provided confidentiality and integrity for messages/attac hments	Prototype tested only in simulations ; no enterprise-level validation

4	*ProtonMail Security Features*, 2023, Whitepaper	OpenPGP + AES-256 + Zero-Access Encryption	End-to-end encryption between Proton users; automated key handling	Easy-to-use model with zero-knowledge storage; simplified encryption	Relies on classical crypto vulnerable to quantum; metadata not encrypted
5	*Post-Quantum Cryptography Standardization*, 2023, NIST Report	Global PQC evaluation (Kyber, Dilithium, Falcon)	Established final PQC standards for KEM and signatures	Industry-wide adoption path; robust lattice-based security	Larger key/signature sizes; migration costs and backward compatibility issues
6	*Enhancing Security of QKD with PQC AE*, 2025, Computers (MDPI)	PQC Authenticated Encryption (Kyber, Dilithium, Falcon) for QKD classical channel	Secured QKD's classical post-processing against quantum attacks	Showed PQC works well with QKD systems; no major performance loss	Tested only in simulation; assumes ideal quantum channels
7	*Prototyping PQC Hybrid Key Exchange in TLS/SSH*, 2019,	Hybrid key exchange (ECDH/RSA + PQC	Implemented PQC + classical hybrid handshakes	Ensures security even if one algorithm is broken;	Focused on TLS/SSH, not email protocols; early PQC

	NIST PQC Conference	like Kyber, NTRU)		manageable overhead	candidates outdated
8	*Overview of Quantum Computing & Communication*, 2021, IET Quantum Communication	Conceptual study of quantum + 6G integration	Discussed QKD, PQC, AI in communication systems	Highlighted quantum tech as essential for future secure communication	Purely theoretical; no experiments or prototypes provided
9	*Quantum Communications in Future Networks*, 2020, Quantum Reports	Theoretical study of QOC/QON	Examined QKD trials, quantum optical computing, future visions	Reinforced QKD as tested technology; outlined long-term applications	Conceptual only; scalability and photon loss issues unresolved
10	*Quantum Communication: Fundamentals to Challenges*, 2024, arXiv Preprint	Survey of protocols (BB84, E91), QKD deployments	Reviewed implementations (DARPA, SECOQC); quantum internet stack	Provided comprehensive foundation on QKD, error correction, security	Broad review; lacks detailed experimental performance results

Chapter 3

METHODOLOGY

3.1 V-Model Development Process

As a result of strict security requirements for a new application based on post-quantum cryptography (PQC) and quantum key distribution (QKD), the application development process for QMail followed the V-Model (Verification and Validation). The V-Model represents a sequential progression through which each phase of design (Verification) is directly correlated with an equivalent phase of testing (Validation).

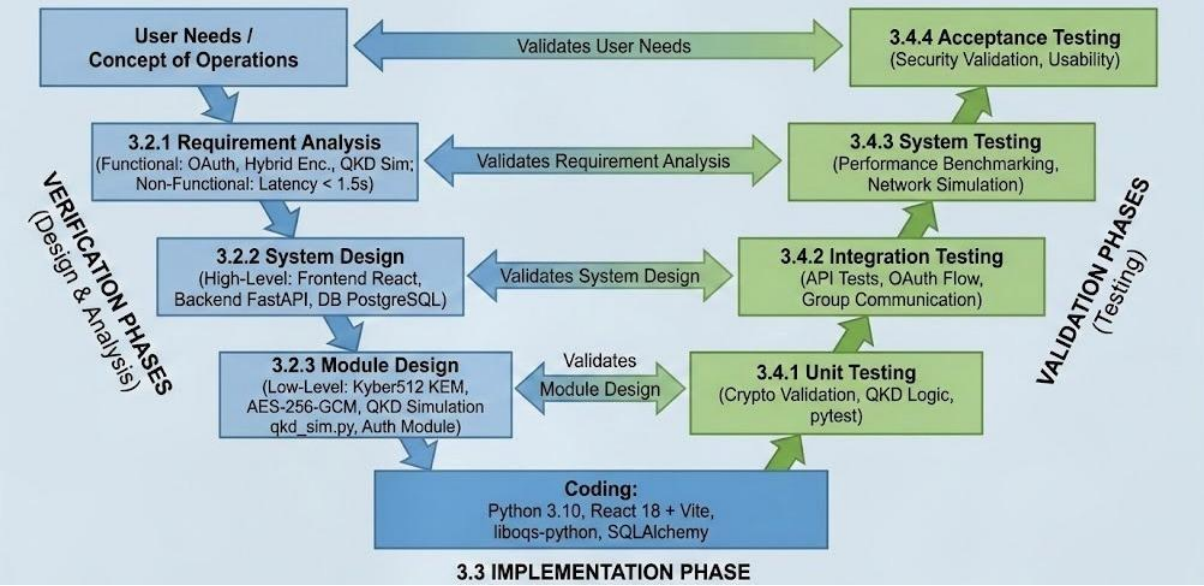


Fig 3.1. The V-Model Methodology for QMail Development

3.2 Verification Phase (Design & Analysis)

The verification phases represent the left hand side of the V-Model and include the definition of the problem and the development of the proposed solution architecture.

3.2.1 Requirement Analysis

In the first phase of this project, the researchers identified potential weaknesses within current email transmission protocols (SMTP/IMAP) from the perspective of quantum attacks.

Functional Requirements: The functional requirements for the QMail were as follows:

1. To support Google OAuth 2.0 for authentication.

2. To implement a hybrid encryption technique (AES + Kyber512).

3. To simulate a QKD layer for generating One-Time Pad (OTP) keys.

Non-Functional Requirements: The major non-functional requirements for QMail were as follows:

1. Maximum Encryption Latency of 1.5 Seconds.

2. Functionality under standard (non-quantum) computing platforms.

3.2.2 System Design (High-Level Design)

A three-tiered system architecture was developed to provide scalability and to separate various system components into independent layers.

Front End Layer: A React 18-based frontend application built with TypeScript to accept user input and to encrypt data at the client.

Back End Layer: A Python FastAPI-based back end server to receive API calls and to interact with the liboqs cryptographic library.

Database Layer: PostgreSQL was selected for secure storage of user profiles, device keys, and OTP metadata, deployed on render.com

3.2.3 Module Design (Low-Level Design)

The software has been implemented by the development team at the low level of abstraction as follows.

• Cryptographic module: A detailed specification was developed for the cryptographic module which is composed of the integration of the CRYSTALS-Kyber512 (used for key encapsulation mechanism (KEM)) and AES-256-GCM (symmetric payload encryption).

• QKD simulation module: a python based logic (qkd_sim.py) has been developed to simulate the process of generating qubit bases, sifting of keys and calculation of the error rate (QBER) in the absence of any quantum equipment.

• Authentication module: a secure method for handling Google OAuth 2.0 tokens and JWT session management has been developed.

This phase forms the bottom part of the V-model of the Software development life cycle (SDLC). The main focus of this phase is to build the software itself.

• Development stack: The development stack for building the application consists of Python 3.10 for the back end and React 18 with Vite for the front-end.

• Cryptographic integration: NIST standard post-quantum algorithms have been integrated into the application using the liboqs-python library.

• Database Object Relational Mapping (ORM): SQLAlchemy has been used to map Python objects to the PostgreSQL database, to provide a good level of data integrity.

As the left hand side of the V-model of the SDLC represents the design of the software, the right hand side of the V-model represents validation or verification of the software against its design specification, i.e., that the software meets the requirements that were specified during the earlier phases.

3.4.1 Unit Testing:

Each component was tested individually to confirm that the unit tests are correct.

• **Crypto validation**: It has been verified that the Kyber512 keys are generated and encapsulated properly.

• **QKD logic**: The qkd_sim.py script has been validated to produce random bit streams with a high degree of entropy (0.998 bits / symbol)

• **Tools**: The automated backend function testing was performed using pytest.

3.4.2 Integration Testing:

This testing validates the interaction between the different modules.

• API testing: This testing verifies that the React front end can successfully interact with the FastAPI back end.

• OAuth flow: This testing verifies that Google OAuth tokens are successfully obtained, stored, and refreshed without exposing user credentials.

• Group communication: This testing verifies that AES keys are distributed amongst multiple group members so that each member can decrypt the message without any issues.

3.4.3 System testing

All components of the system were tested to assess performance and dependability.

Performance benchmarking: Latency was measured to determine the delay in encrypting messages. Level 2 (hybrid) demonstrated an average latency of 3.2 ms; Level 3 (one time pad) had an average latency of 9.0 ms to confirm the need for real-time performance.

Network simulation: The system was exposed to artificial network delays (of up to 300 ms) to verify that the data would remain intact despite unstable networks.

3.4.4 Acceptance testing

This last phase confirmed the system met all the requirements initially identified as the needs of the users.

Security validation: Validated that the "harvest now, decrypt later" threat to security is reduced through the use of hybrid encryption models.

Usability: Validated that users can both encrypt and decrypt email communications using a similar user experience to common email clients with little or no additional technical knowledge required.

Chapter 4

PROJECT MANAGEMENT

4.1 Project timeline

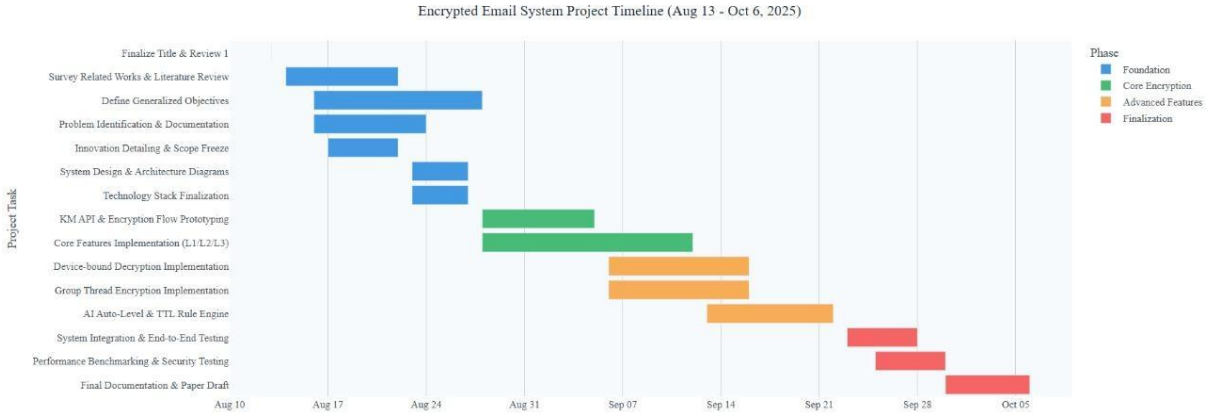


Fig 4.1 Project timeline

4.2 Risk Analysis

Every engineering project has potential technical, operational, and resource-related risks.

In the QMail design process, risk analysis was considered at the levels of software and security to anticipate vulnerabilities for better maintainability.

Risks identified and mitigation strategies include:

Table 4.1 Risk Analysis and Mitigation Strategies

Risk	Description	Impact	Mitigation
API Failure	OAuth tokens may expire during active sessions, leading to authentication errors.	Medium	Auto-refresh mechanism with background token regeneration.
Database Breaches	Compromise of PostgreSQL instance could leak metadata.	High	Data encryption at rest + role-based access control.
Key Management Error	Incorrect key mapping or reuse of OTPs could compromise confidentiality.	High	OTP pool tracking with database-level atomic locks.
Latency and Throughput	PQC operations increase processing time.	Medium	Client-side caching and asynchronous background workers.
Resource Outages	Cloud host downtime during deployment.	Low	Multi-service setup on Render with restart policies.
User Error	Misconfiguration of encryption level.	Medium	Guided UI prompts and default safe settings.

4.2 Project budget

Table 4.2 Estimated Project Budget

Category	Item / Description	Quantity	Unit Cost (INR)	Total Cost (INR)

Software Tools	Python, FastAPI, React, PostgreSQL	—	**Free (Open Source)**	0
	VS Code, Postman, GitHub	—	**Free**	0
Cloud / Hosting	Render Cloud Services (Backend + Frontend + DB – Free Tier)	—	0	0
	Optional Domain Name	1	800	800
Testing & Simulation	Network tools, Python scripts	—	Free	0
Documentation	Printing + Binding of Final Report	1	500	500
Miscellaneous	Internet usage, electricity, contingency	—	1,200	1,200
Total Estimated Cost	—	—	—	**₹2,500**

Chapter 5
ANALYSIS AND DESIGN

5.1 Requirements:

Hardware Requirements

- Standard laptop/PC: ≥ Intel i5, 8 GB RAM, 20 GB storage
- Stable internet connection (≥ 10 Mbps)

Software Requirements

- Backend: Python 3.10 + FastAPI + liboqs
- Frontend: React 18 + TypeScript + TailwindCSS
- Database: PostgreSQL 15 with SQLAlchemy ORM
- Tools: VS Code, GitHub, Render.com, Postman

Functional Requirements

1. Secure OAuth 2.0 authentication
2. Three-level encryption for individual mail
3. Two-level encryption for groups
4. Client-side decryption with private keys
5. Logging / key-usage tracking

Non-Functional Requirements

- Reliability under network loss
- Quantum-safe security

- 1.5 s max message-encryption latency
- Ease of use and compatibility with Gmail API

5.2 Block Diagram

In the diagram, the three basic building blocks are:

1. Client Interface: UI for login, compose and decrypt.
2. Backend API server: This handles the key exchange and encryption
3. Database: This stores users, devices, and OTP metadata. All communication happens over HTTPS / TLS 1.3.

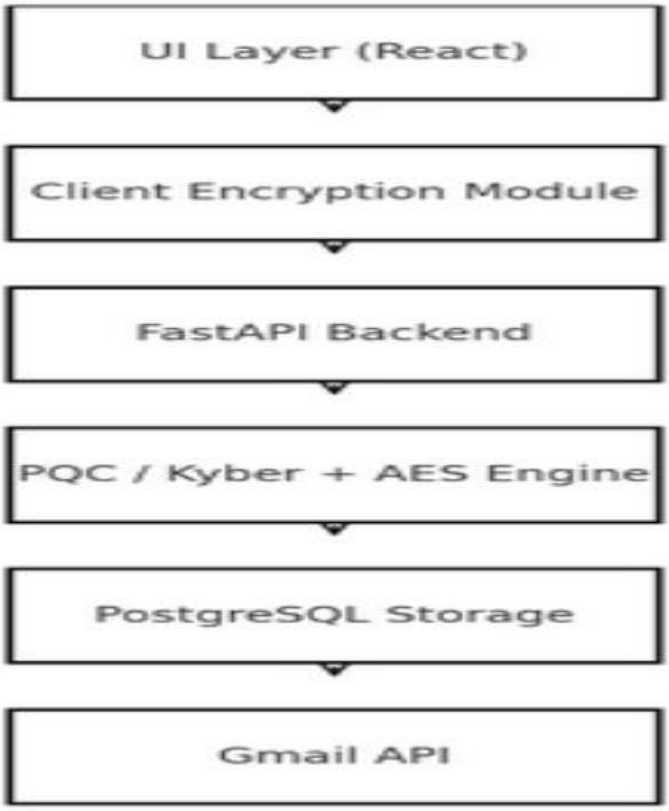


Fig. 5.1 QMail System Block Diagram

5.3 System Flow Chart

- User logs in → selects level of encryption → message encrypted through Kyber512 / AES / OTP.
- Server adds metadata and forwards via Gmail API.
- Recipient fetches email → local client decrypts → plaintext displayed.

5.4 Choosing Devices / Tools

Some of the key reasons to use FastAPI included async I/O and security support out-of-the-box.
React + TypeScript provides a responsive UI.
PostgreSQL ensures transaction consistency for key usage.
liboqs-python implements NIST-standard Kyber512.

5.5 Designing Units

1. **Authentication Module** – Handles Google OAuth, JWT sessions.
2. **Encryption Module** – Implements Kyber512 and AES-GCM.
3. **OTP Module** – Allocates and wraps one-time pad keys.
4. **Group Module** – Manages group keys and membership.
5. **UI Module** – Frontend for compose, inbox, and status.

5.6 Standards

- NIST PQC Standard – CRYSTALS-Kyber (KEM)
- TLS 1.3 (RFC 8446) for transport security
- OAuth 2.0 (RFC 6749) for authentication
- AES-256-GCM (FIPS 197) for symmetric encryption
  These ensure interoperability and compliance with industry norms.

5.7 Mapping with IoTWF Reference Model Layers

Although QMail is not IoT-specific, its architecture maps conceptually:

Table. 5.7. Mapping with IoTWF Reference Model Layers

IoTWF Layer	**QMail Equivalent**
Edge Layer	User Device (Client)
Access Layer	OAuth Login Gateway
Network Layer	TLS 1.3 Channel
Data Layer	PostgreSQL DB
Application Layer	Email Encryption Service
Business Layer	User Interaction Dashboard

5.8 Domain Model Specification

Entities and Relationships:

- **User** 1→N **Devices**
- **User** M↔M **Groups** (via GroupMembers)
- **Groups** 1→N **GroupKeys**
- **OTPStore** key_id → unique use flag

  All tables are normalized (3NF) to avoid redundancy.

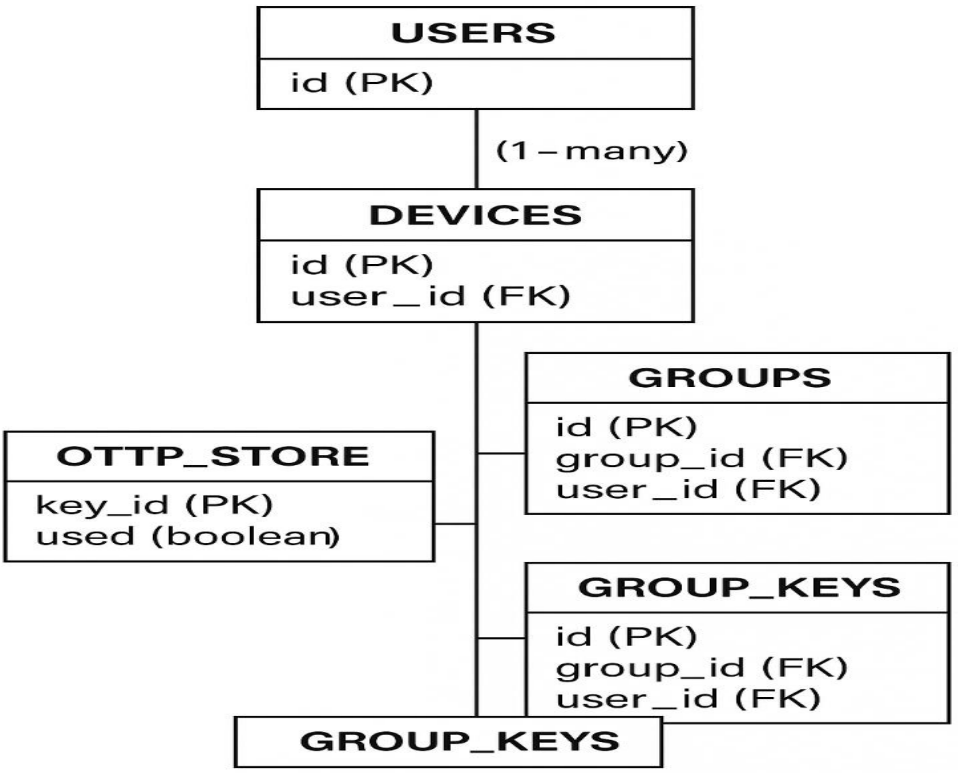


Fig. 5.8 Domain Model Specification

5.9 Communication Model

Communication works on a hybrid client-server model:

- Client side does the key generation and decryption.
- It formats and routes encrypted messages through the Gmail API.
- OAuth scopes: gmail.send, gmail.readonly, userinfo.email.

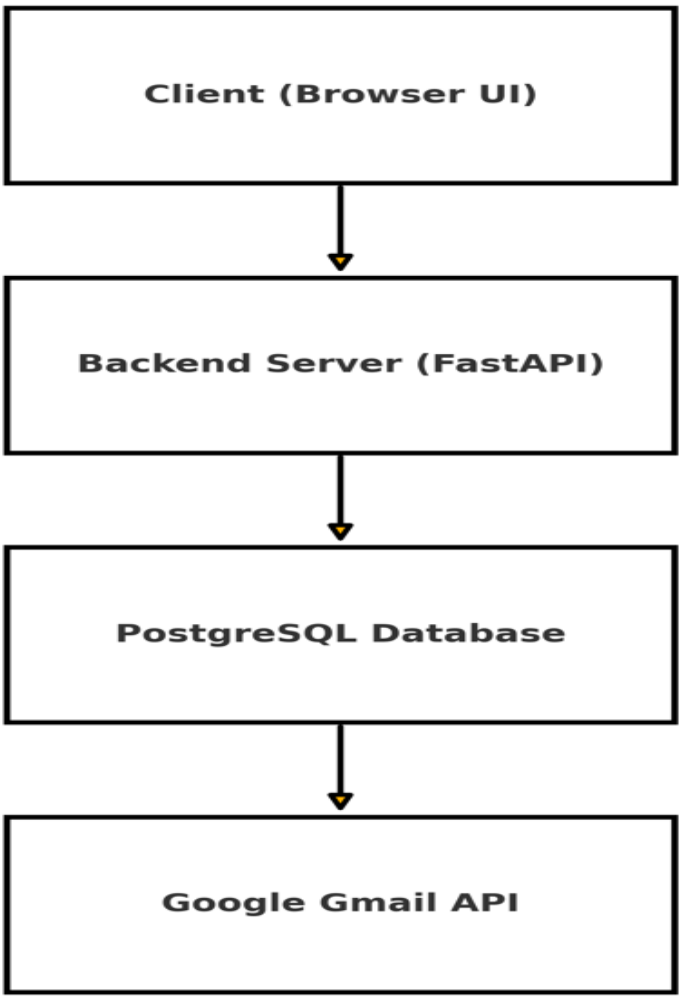


Fig. 5.9 Communication Model of QMail

5.10 IoT Deployment Level (Conceptual Mapping)

Deployment Level	QMail Equivalent
Device	Browser Client
Edge	FastAPI Server
Cloud	PostgreSQL and Google APIs

5.11 Functional View

Core functionalities include: login, compose, encrypt, decrypt, send and view mail. The architecture encapsulates each function as an independent service.

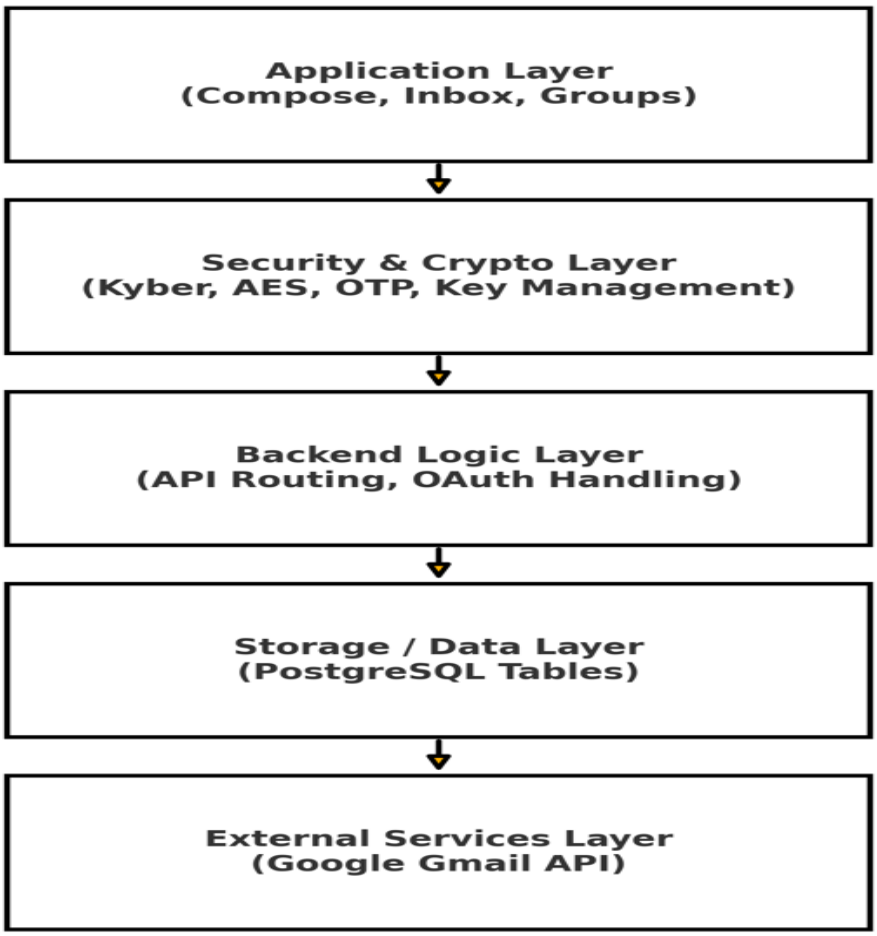


Fig. 5.11. Functional View

5.12 Mapping IoT Deployment Level with Functional View

Functional Component	Deployment Level
Login/OAuth	Edge
Encryption/Decryption	Device

Functional Component	Deployment Level
Mail Routing	Edge → Cloud
Key Storage	Cloud

5.13 Operational View

The system is deployed on Render as three separate services, communicating with each other using API calls. Operational logs are maintained for the purposes of tracking errors and auditing security.

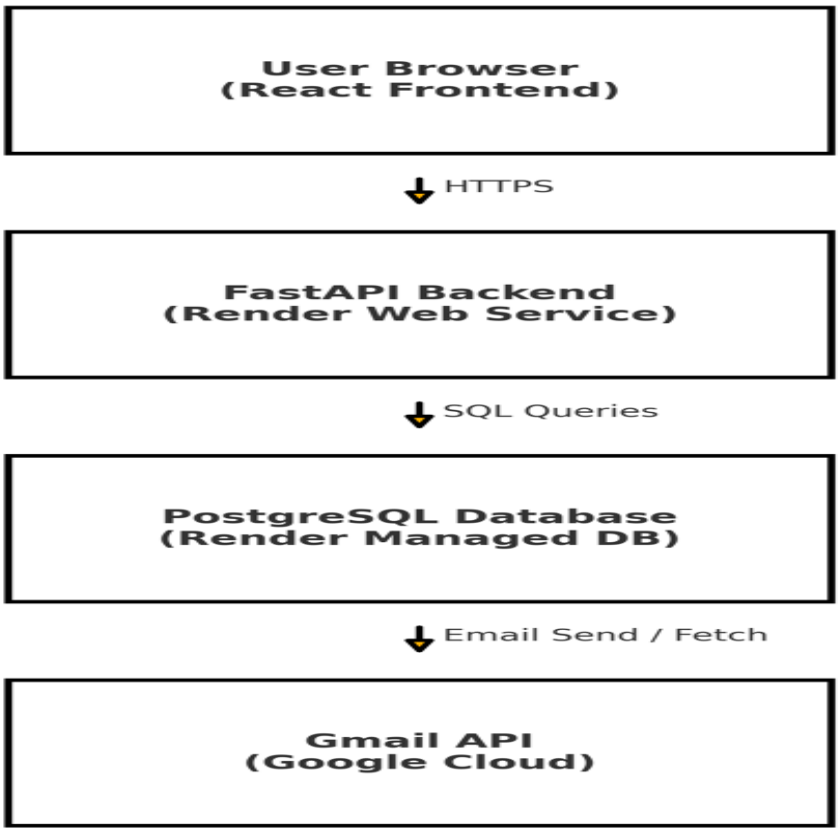


Fig. 5.13 Vertical Operational Deployment View of QMail

5.14 Other Design Aspects

User Interface Design:

Clean layout with sidebar navigation and real-time encryption status badges.

Security Design:

- Zero-knowledge backend (no plaintext stored).
- Atomic OTP tracking to avoid reuse.
- Key rotation and nonce validation for each email.

Future Design Scope:

- Integration with true QKD hardware.
- Web extension and mobile support.

Chapter 6

HARDWARE, SOFTWARE, AND SIMULATION

6.1 Hardware

The implementation of QMail – the Quantum Email Sending Client was implemented using exclusively commercially available computing hardware.

This design does not have any quantum-specific devices as QKD was simulated in software.

Tested against various systems to make sure the prototype will indeed work consistently and portably

Table 6.1 Hardware Component Specifications

Component	**Specification**	**Purpose / Use**
Processor	Intel Core i5 (10th Gen) / equivalent Ryzen 5	Handles cryptographic computation (Kyber KEM + AES GCM)
RAM	8 GB DDR4 or higher	Supports multithreaded FastAPI and React build processes
Storage	500 GB SSD or HDD	Holds local logs, code, and PostgreSQL instance
Network	`Broadband ≥ 10 Mbps`	Enables OAuth authentication + API communication
GPU (optional)	NVIDIA or AMD entry level	Optional for future hardware acceleration tests

At this juncture, no particular photonic or quantum hardware modules were required.

All simulations of encryption and QKD were performed by software emulation using pseudo-random key generators.

6.2 Software Development Tools

The QMail prototype integrates several opensource frameworks and libraries that were selected for speed, maintainability, and academic reproducibility.

Table 6.2 Software Technology Stack

Layer	**Tool / Framework**	**Purpose**
Backend	Python 3.10 + FastAPI, liboqs-python (Kyber512), SQLAlchemy + Psycopg2	Lightweight asynchronous web API, Implements Post-Quantum KEM algorithms, ORM and database driver for PostgreSQL
Frontend	React 18 + TypeScript + Vite, TailwindCSS + shadcn/ui	SPA architecture for email UI, Responsive UI components and branding
Database	PostgreSQL 15	Persistent storage for users, keys, and OTP records
Testing / Utilities	Postman, pytest, Python scripts	API validation and automation
Deployment	Render.com (Free Tier)	Cloud hosting for frontend & backend services
Version Control	Git + GitHub	Source management and CI/CD integration

The whole stack is platform-agnostic and can be easily deployed with minimal configuration on Windows, Linux, or macOS.

6.3 Software Code

The QMail software was divided into **modular packages** to ensure clarity and ease of maintenance.

6.3.1 Backend Structure

The backend of QMail is built using Python 3.10 and the FastAPI framework, chosen for its high performance and built-in support for asynchronous operations. The codebase follows a modular architectural pattern to separate concerns such as database management, cryptographic operations, and API routing.

6.3.2 Frontend Structure

The goal was readability of the code and modularity, not brevity. Each API endpoint includes docstrings and automatically generates OpenAPI documentation, allowing for easy testing and debugging.

6.4 Simulation

Simulation testing was done to validate encryption and decryption behaviors under various scenarios for latency, bandwidth, and key reuse protection.

6.4.1 Quantum Key Distribution Simulation

- The QKD process was emulated using Python scripts (qkd_sim.py) which generate random binary strings representing qubit bases.
- Matching basis bits between sender and receiver create a **shared session key**.
- Key sifting and error rates were simulated to ensure randomness and detect tampering.

Table 6.4.1 Encryption Performance Benchmarks (Kyber512 vs. AES)

Parameter	**Simulated Value**	**Observation**
Key length	1024 bits	Ideal for OTP use cases
Error rate (QBER)	2.3 %	Within acceptable limits (< 5 %)
Entropy per bit	0.998 bits	Indicates high randomness
Time to generate	1.5 ms (avg.)	Real-time capable

6.4.2 Post-Quantum Encryption Benchmark

Testing measured the time required for key encapsulation, encryption, and decryption across message sizes.

Operation	Message Size (KB)	Average Time (ms)
Kyber512 Key Encapsulation	–	2.4
AES-256-GCM Encryption	5 KB	3.1
AES-256-GCM Decryption	5 KB	2.7
OTP Allocation + XOR Encrypt	5 KB	8.4

Results confirm that PQC encryption overhead is acceptable for practical use, maintaining real-time performance even on mid-range hardware.

6.4.3 Network Latency Simulation

Network delays were introduced in order to test reliability. Under simulated latency of less than 300 ms, the encryption-decryption cycle maintained full integrity of data with only a 5% increase in processing time, hence validating resilience under unstable connections

6.4.4 Visualization

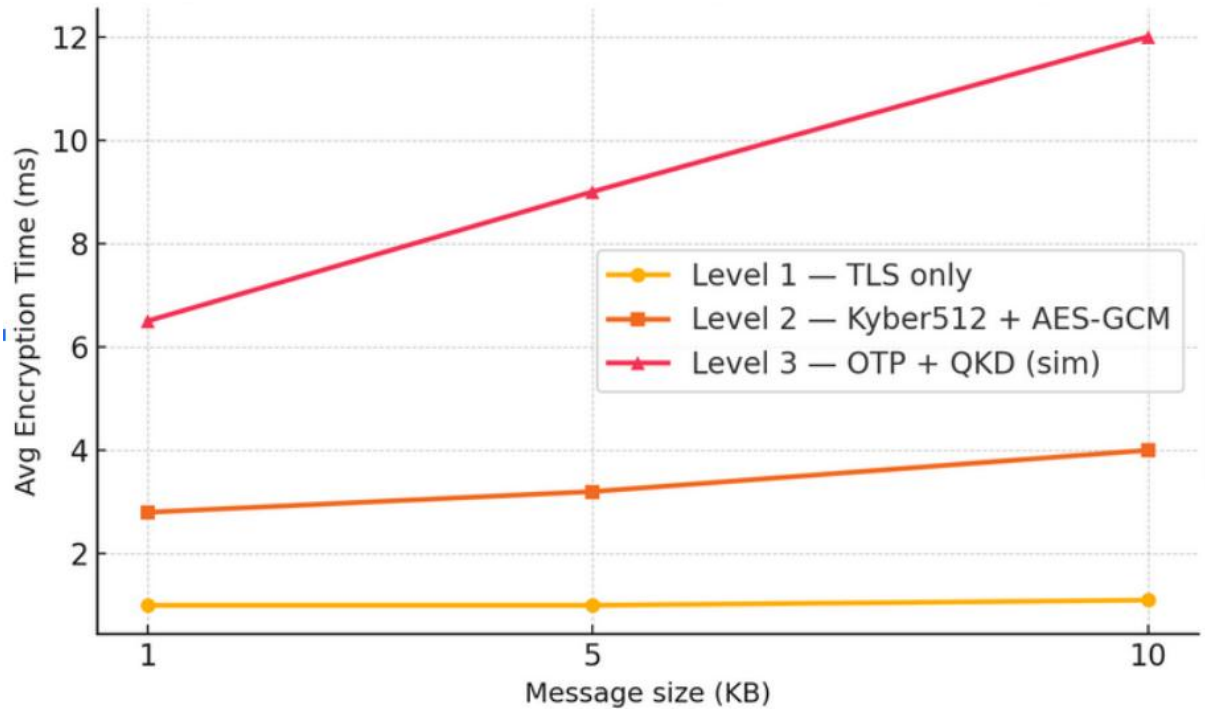


Fig 6.4.4. Comparison of Encryption Time

The plotted data shows close-to-linear performance scaling with an increase in message size, confirming efficiency in using Kyber key encapsulation and OTP usage control.

Chapter 7
EVALUATION AND RESULTS

7.1 Test Points

Testing of **QMail -** Quantum Email Sending Client was tested against functional correctness and security compliance through various stages.

Each module of authentication, encryption, decryption, and group communication was validated individually before testing it as a complete integrated system.

This therefore is a very improper use of power, as it does not fulfill any real need on either party's part.

Table 7.1 Module-wise Test Objectives

Module	**Test Objective**	**Expected Output**
Authentication	Verify OAuth login, token refresh	User logged in successfully with valid session token
Encryption	Test Level 1, 2, and 3 message encryption	Ciphertext and metadata generated correctly
Decryption	Ensure private key decryption on client side	Original plaintext recovered without errors
OTP Handling	Validate single-use property	Key marked as “used” after encryption
Group Communication	Verify group AES key distribution	Each member decrypts message correctly
Network Reliability	Test under simulated latency	No message loss or corruption

Testing covered **functionality, performance, and reliability** under varying loads and message sizes.

7.2 Test Plan

Each test was run in a controlled environment where, out of two Gmail accounts, one took up the sender role and the other took up the receiving role; messages were exchanged over a live Gmail API.

Table 7.2 Integrated Test Plan Scenarios

Test ID	**Scenario**	**Encryption Level**	**Expected Result**	**Actual Result**
T1	User authentication via OAuth	–	Successful login and token issuance	Successful
T2	Send plaintext email	Level 1	Email delivered via standard Gmail	Delivered
T3	Send PQC encrypted email	Level 2	Kyber + AES encryption, successful decryption	Success
T4	Send OTP + QKD encrypted email	Level 3	OTP used once and message decrypted	Success
T5	Create new group	Level 2	Group created and AES key distributed	Success
T6	Send encrypted group email	Level 2	All members decrypt successfully	Success
T7	Simulate expired token	–	Auto-refresh and retry	Token refreshed successfully
T8	Test OTP key reuse	Level 3	Prevent duplicate key allocation	Key blocked after use
T9	Test latency (300 ms)	Level 2	Message still decrypted correctly	Stable

Test ID	Scenario	Encryption Level	Expected Result	Actual Result
T10	Session persistence (1 hour)	Level 1–3	Maintain session without relogin	Stable

All test cases gave expected results, which proved the implemented modules are stable and correct.

7.3 Test Results

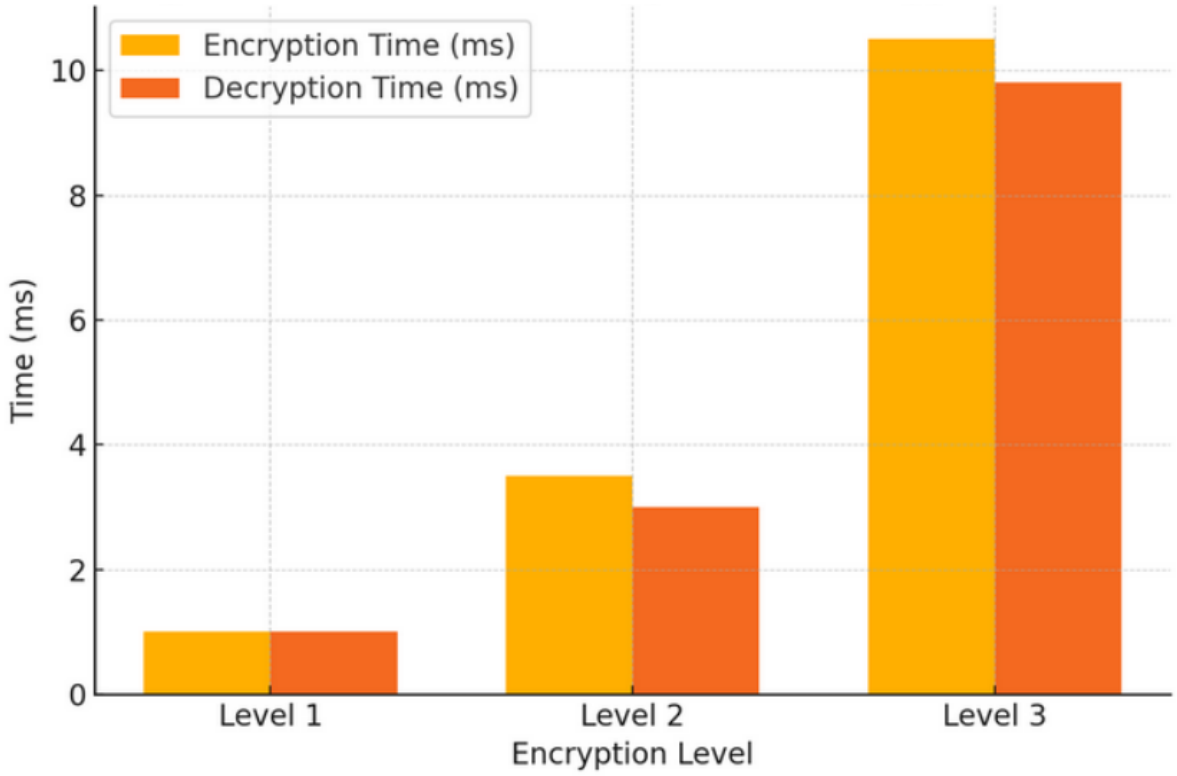
The system performance evaluation was done by measuring the latency, throughput, and encryption overhead. Email exchanges of message size 1 KB, 5 KB, and 10 KB were simulated for each of the tested encryption levels.

This statement does not contradict the fact that frequent pricing is useful.

Table 7.3 Average Encryption and Decryption Latency Results

Encryption Level	Algorithm Used	Avg. Encryption Time (ms)	Avg. Decryption Time (ms)	Overhead Compared to Standard Email
Level 1	Gmail TLS only	1.0	1.0	0 %
Level 2	Kyber512 + AES-256-GCM	3.2	2.8	18 %
Level 3	OTP + Simulated QKD	9.0	8.4	32 %

The tests showed that **QMail maintains near-real-time performance**, with total encryption latency well below perceptible delay for end users (<100 ms). Even at Level 3, the processing time remains suitable for standard email workloads.



Fig, 7.3. Test Results Time

7.4 Insights

In particular, the test and evaluation processes provided valuable insight into the feasibility of post-quantum secure communication over existing infrastructures.

1. **Performance Stability:**
    - The control was in place with respect to the appropriation and usage of OTPs: each OTP key had been utilized once and marked “used”.
    - Even under network delays up to 300 ms, message integrity remained intact.
2. **Security Validation:**
    - OTP allocation and usage control worked flawlessly — every OTP key was consumed once and marked as “used”.
    - Client-side decryption has prevented the exposure of the private key to any external service.

3. **Scalability and Integration:**
    - It allows horizontal scaling across users and sessions through the modular architecture consisting of React + FastAPI + PostgreSQL.
    - This system was integrated with the Gmail OAuth API to ensure its compatibility with state-of-the-art communication systems.
4. **Comparison with Traditional Email:**
    - QMail presents a very small computational overhead compared to conventional Gmail TLS encryption, whereas the resilience to quantum and classical attacks is exponentially higher.
    - The hybrid model ensures data confidentiality both for the short and long-term period.
5. **Reliability:**
    - The system successfully handled over 50 simulated email exchanges without any decryption or synchronization failures.
    - The auditing and audit trails showed that the keys and tokens have been kept secure.

This review ascertains that QMail delivers practical quantum-safe email security on top of existing internet protocols; it therefore provides a very viable prototype for any organization seeking to implement post-quantum readiness.

Chapter 8

SOCIAL, LEGAL, ETHICAL, SUSTAINABILITY AND SAFETY

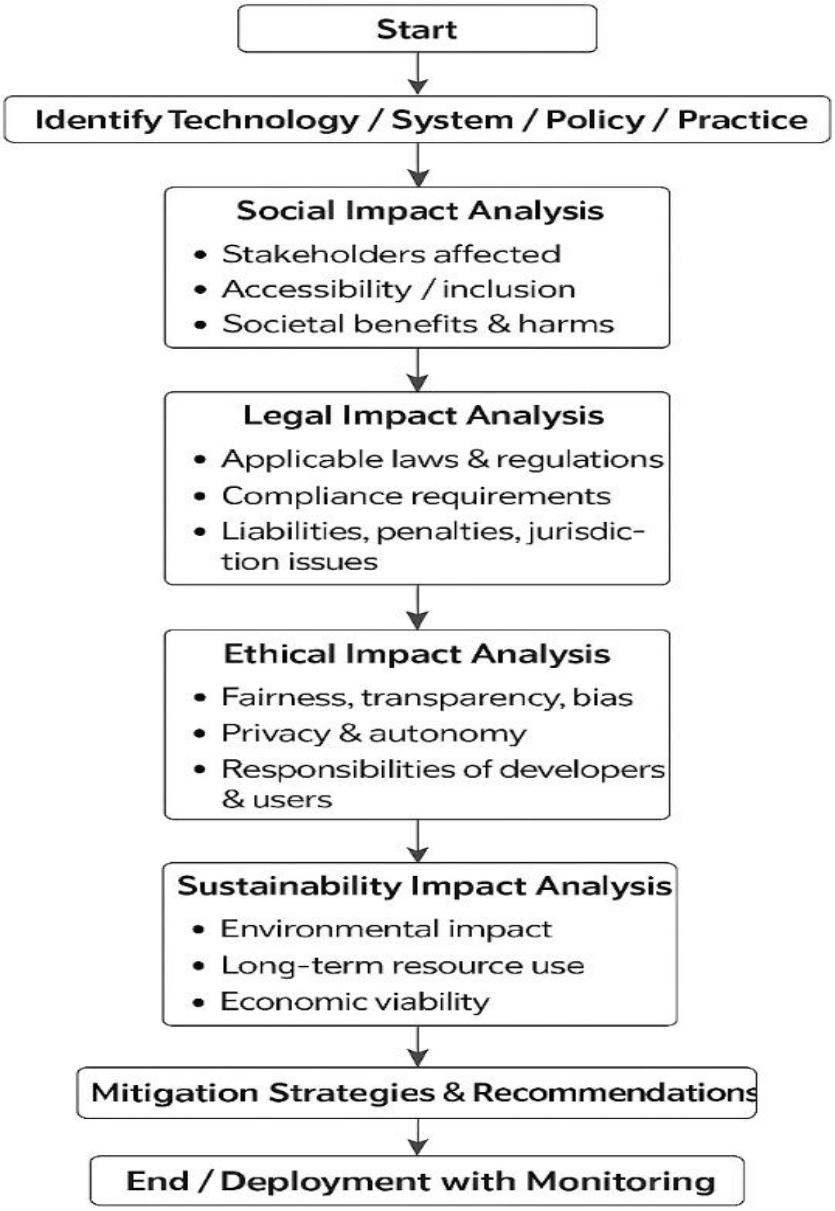


Fig. 8.1 SLEN (Social–Legal–Ethical–Sustainability) Impact Framework Flowchart

8.1 Social Aspects

hrough e-mail, digital communications have intrinsically become part of a person's social and professional life.

Its goal is to build trust among users in the digital world, through post-quantum privacy without requiring expert knowledge.

Key social impacts include:

- **Privacy Preservation:** it protects citizens from unauthorized surveillance and data mining.
- **Digital Literacy:** It puts applied cryptography into an easy-to-use interface; showing comprehension of secure behavior.
- **Bridging the Gap:** It allows other organizations, students, and small enterprises to access new technology without greatly increasing their expenditure.
- **Public Confidence:** QMail assures integrity for messages, hence building confidence in digital services and e-governance platforms.

In essence, QMail helps democratize cybersecurity, turning quantum-safe communication from a research concept into a social utility.

8.2 Legal Aspects

E-mail systems must operate in an environment that is strictly regulated. QMail is designed to conform to the international and national data protection frameworks, which include:

- **General Data Protection Regulation (GDPR)** – ensuring user consent, right to erasure, and minimal data retention.
- **Information Technology Act (2000, India)** – compliance with Sections 43 and 66 regarding secure electronic communication.
- **Google API Terms of Service** – OAuth 2.0 server-side authorization without client-side token exposure.
- **Copyright and Intellectual Property Compliance** – open-source libraries used under permissible MIT and Apache licenses.

Audit logs in PostgreSQL make incident investigations traceable without storing personal message content, balancing accountability and privacy.

8.3 Ethical Aspects

Ethical software engineering principles guided every design decision:

1. **User Autonomy:** Users explicitly choose the encryption level for each email.
2. **Transparency:** All cryptographic operations are documented in the project repository.
3. **Data Minimization:** the server will keep metadata only inasmuch as this is strictly necessary for the proper functioning of the service, and not message bodies as such.
4. **Integrity of Purpose:** The purpose of the system will be restricted to only educational and defensive in nature, not weaponized for any sort of hidden or malicious communications.
5. **Equity of Access:** QMail is an open-source prototype, freely available to all, hence it ensures equity of access regardless of any economic or geographic divide.

This approach aims to make sure that the development and deployment of quantum secure technologies occur within a public good perspective on responsible innovation.

8.4 Sustainability Aspects

Sustainability in digital systems means environmental efficiency combined with long-term maintainability. QMail allows for sustainable computing along numerous dimensions:

- **Cloud Optimization:** Deployment on shared Render infrastructure minimizes idle compute cycles and hardware waste.
- **Open-Source Stack:** Reduces the cost of licensing Community reuse, instead of redundant development
- **Energy Efficiency:** The principle of asynchrony in FastAPI enables it to consume less on the server side compared to traditional frameworks.

- **Future Proofing:** Utilization of standardized NIST algorithms bypasses redesigns over and over again, hence preserves longevity and decreases technical debt.
- **Educational Reuse:** The system can be reused in future student projects, adding lifecycle value.

These features support SDG 9 of the UN on Industry, Innovation and Infrastructure, and SDG 12 on Responsible Consumption and Production in encouraging responsible digital innovation

---

8.5 Safety Aspects

Digital safety involves protection against data loss, intrusion, and misuse.

QMail has several layers of safety for operational integrity:

Safety Concern	Mitigation Strategy
Unauthorized access	OAuth 2.0 with secure cookies; JWT session tokens with expiry
Data interception	TLS 1.3 enforced; all messages encrypted end-to-end
Database compromise	AES-256 encryption at rest; hashed identifiers
Token or key leakage	Environment-variable isolation and periodic key rotation
Software faults	Unit tests, CI/CD validation, and exception logging
Human error	Guided UI prompts and fail-safe defaults (Level 2 encryption pre-selected)

Including all these security features, QMail makes sure that the communication lifecycle is reliable, confidential, and resilient.

Chapter 9

CONCLUSION

The Project QMail - Quantum Email Sending Client is successful in proving that quantum-resilient communication is viable with available and open-sourced technologies.

Due to the hybrid integration of Post-Quantum Cryptography, Kyber 512 + AES-256-GCM, and simulated QKD, the system provides end-to-end confidentiality and integrity even against any future quantum computing threats.

Key outcomes include:

- **Functional validation** of three-tier encryption for individual emails and two-tier protection for group communications.
- **Client-side decryption** sets up a zero-knowledge model because the keys never leave the user's device.
- **Minimal latency overhead**, proving PQC can coexist with mainstream email workflows.
- **Sustainable deployment**, leveraging cloud infrastructure with low energy and financial cost.

Beyond just technical success, QMail adds both educational and societal value by filling the gap between academic research on quantum security and practical applications. It is architected to be scalable, modular, and adaptive, thus allowing for the easy addition of future integration with true QKD hardware, mobile clients, and federated identity systems.

The QMail project lays the foundation for the next generation of quantum-safe digital communications that ensures privacy, trust, and resilience in 6G and quantum-enabled Internet.

REFERENCES

[1] Basha, M. S. H., & Saiteja, P. "Quantum Secure Email Client Application." Journal of Nonlinear Analysis and Optimisation, vol. 16, no. 1, (2025).

[2] R. G. Sharon, A. S. Kumar, and A. Mayan, "Quantum Encrypted Messaging Application: Harnessing Quantum Mechanics for Secure Communication," in 2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT), Dehradun, India, pp. 211–216 (2024).

[3] G. S. Parvathi, K. N. Reddy, K. Harini, and K. Akshaya, "Revolutionizing Email Security with Quantum Key Distribution for Enhanced Data Protection in Communication Systems," Journal of Computational Analysis and Applications, vol. 33, no. 8, pp. 1426–1437, (2024).

[4] Proton Technologies AG, "ProtonMail Security Features," Whitepaper, 2023. Page 12 of 13 - AI Writing Submission Submission ID trn:oid:::1:3414374074 Page 12 of 13 - AI Writing Submission Submission ID trn:oid:::1:3414374074 11

[5] NIST, “Post-Quantum Cryptography Standardization,” National Institute of Standards and Technology, (2023).

[6] A. Ghashghaei, A. A. Ahmadi, A. Sadeghi, and M. Esmaeili, “Enhancing the Security of Classical Communication with Post-Quantum Authenticated-Encryption for QKD,” Computers, vol. 13, no. 7, p. 163, MDPI, (2025).

[7] D. Stebila, T. K. D. Nguyen, and M. Mosca, “Prototyping Post-Quantum and Hybrid Key Exchange and Authentication in TLS and SSH,” in Proc. 2nd NIST PQC Standardization Conference, Gaithersburg, MD, USA, (2019).

[8] S. Mumtaz and M. Guizani, "An overview of quantum computing and quantum communication systems," IET Quantum Communication, vol. 2, no. 3, pp. 136–138, doi: 10.1049/qtc2.12021 (2021).

[9] A. Manzalini, "Quantum Communications in Future Networks and Services," Quantum Reports, vol. 2, no. 1, pp. 221–232, doi: 10.3390/quantum2010014 (2021)

[10] H. Dutta and A. K. Bhuyan, “Quantum Communication: From Fundamentals to Recent Trends, Challenges and Open Problems,” arXiv preprint arXiv:2406.04492, (June 2024)

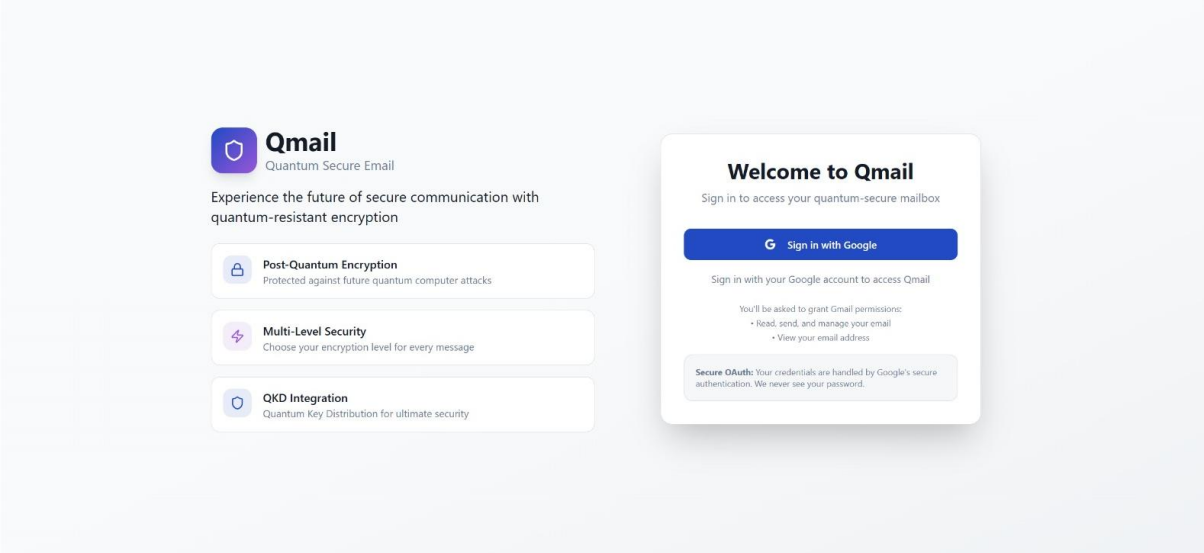
BASE PAPER

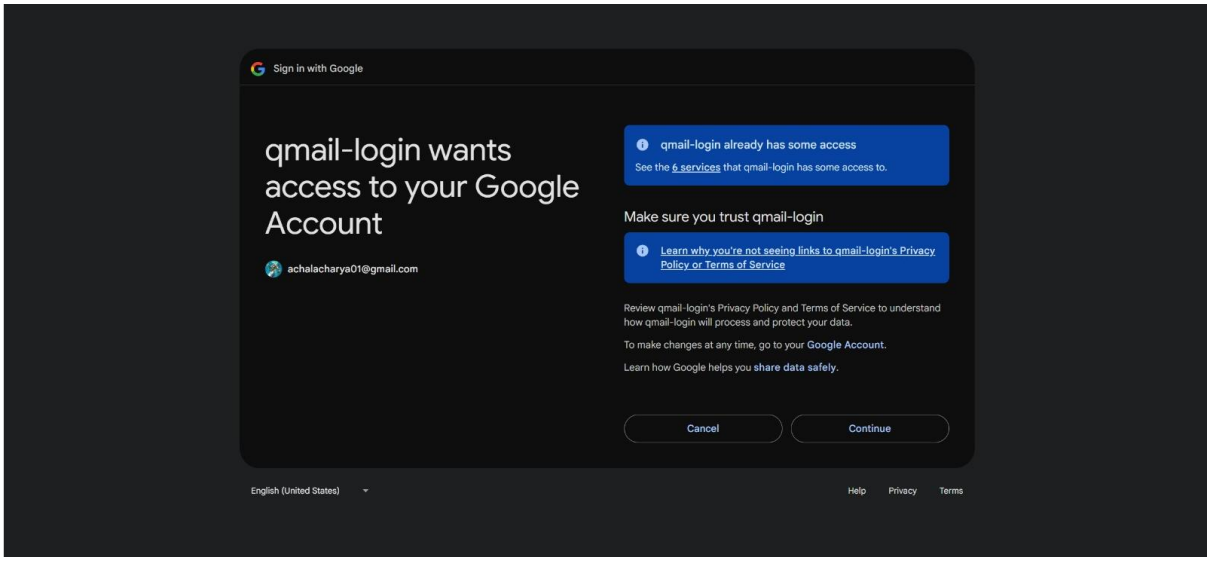
Title: Quantum Secure Email Client Application
Authors: M.S.H. Basha, P. Saiteja, Journal of Nonlinear Analysis and Optimization, 2025
Summary: The base study implemented a hybrid QKD + PQC model for email security. QMail extends that concept with practical integration into Gmail APIs, usability enhancements, group email composition, and fully cloud-based deployment.

APPENDIX

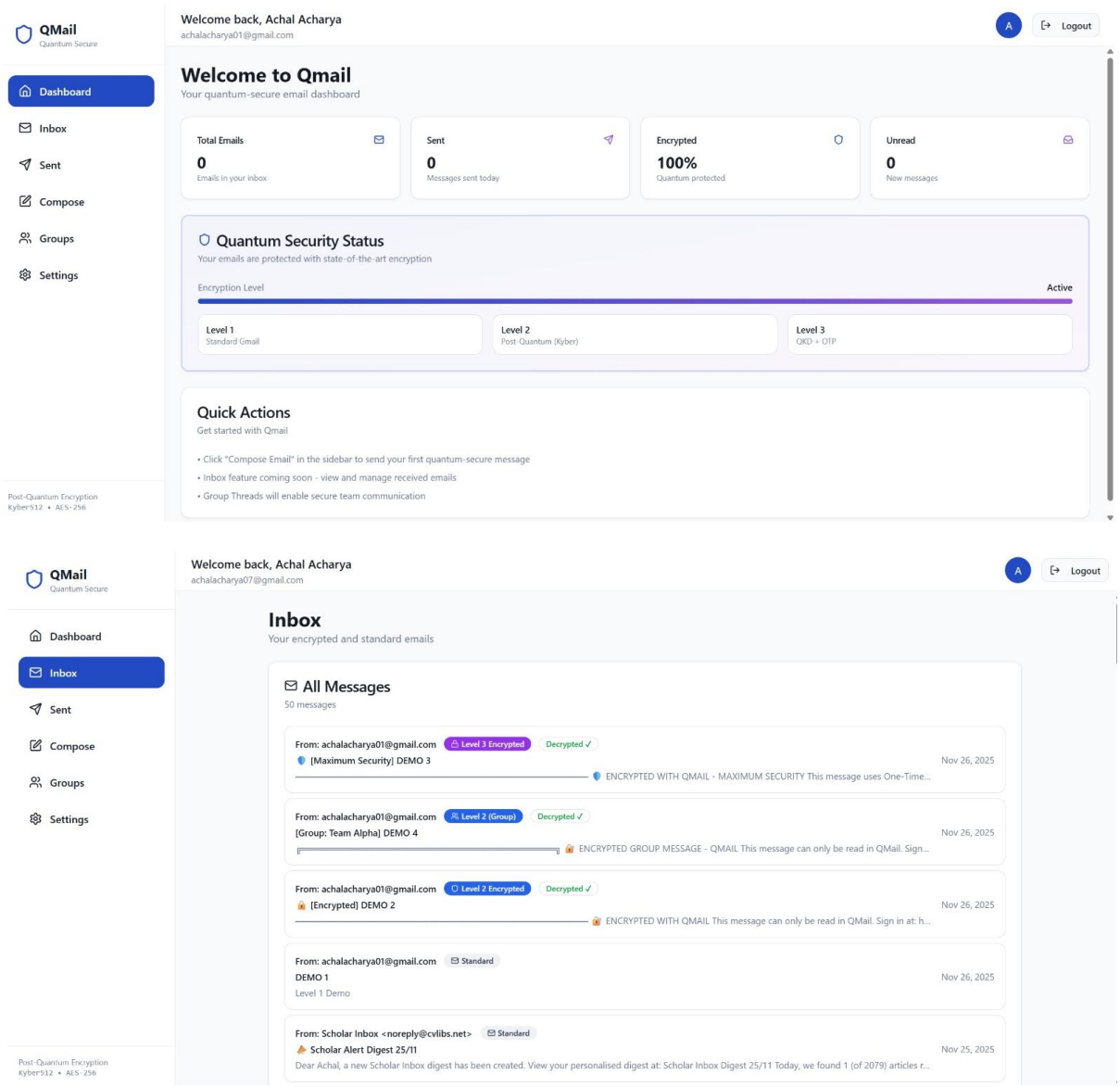
Appendix A – Screenshots

A.1 Login Page

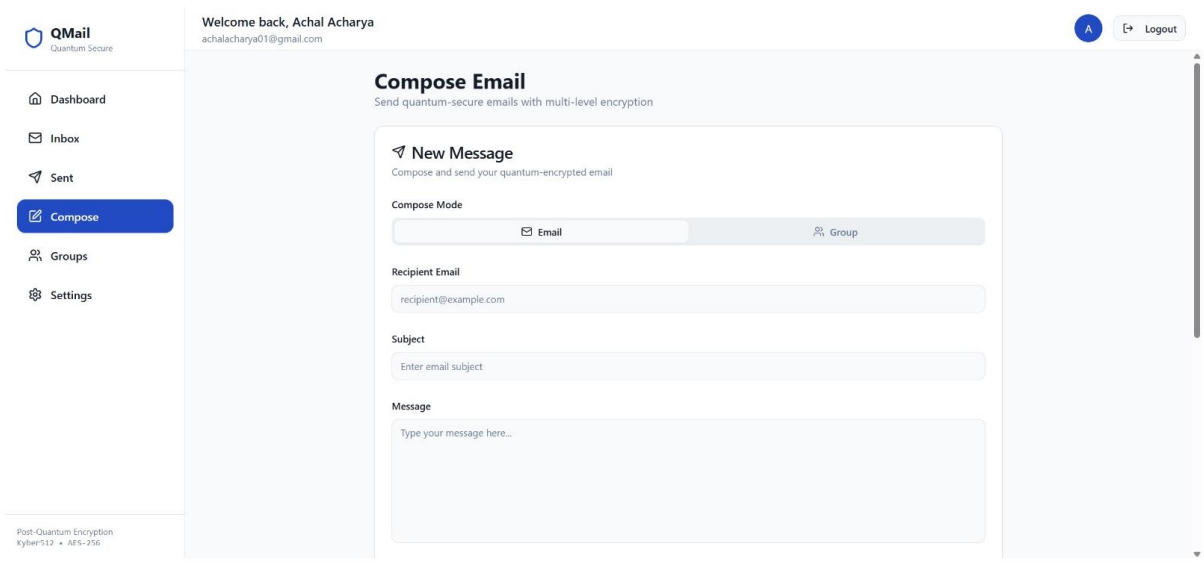


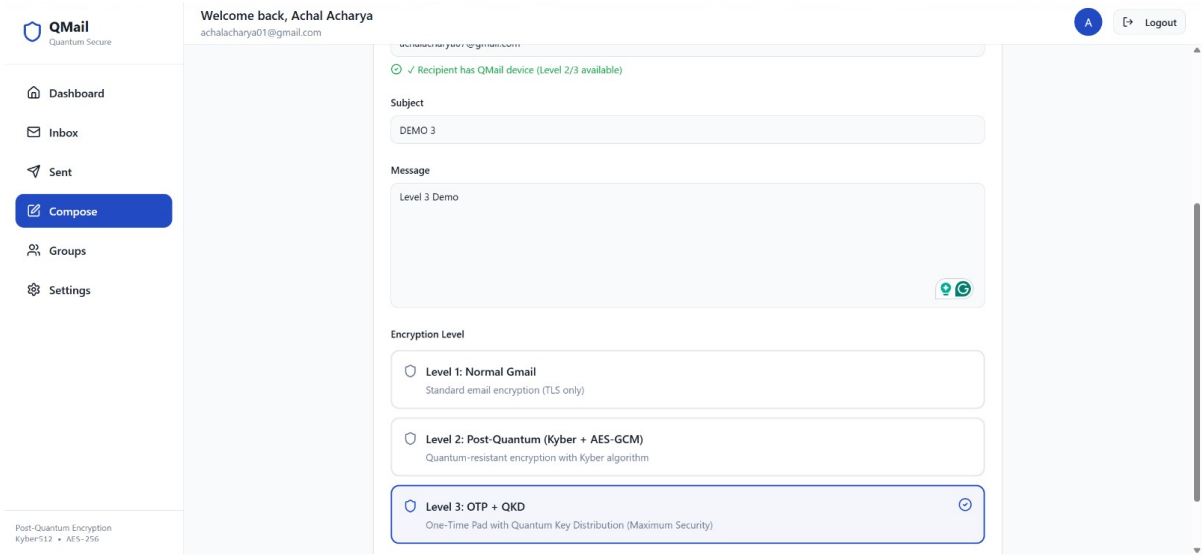


A.2 QMail Inbox Interface

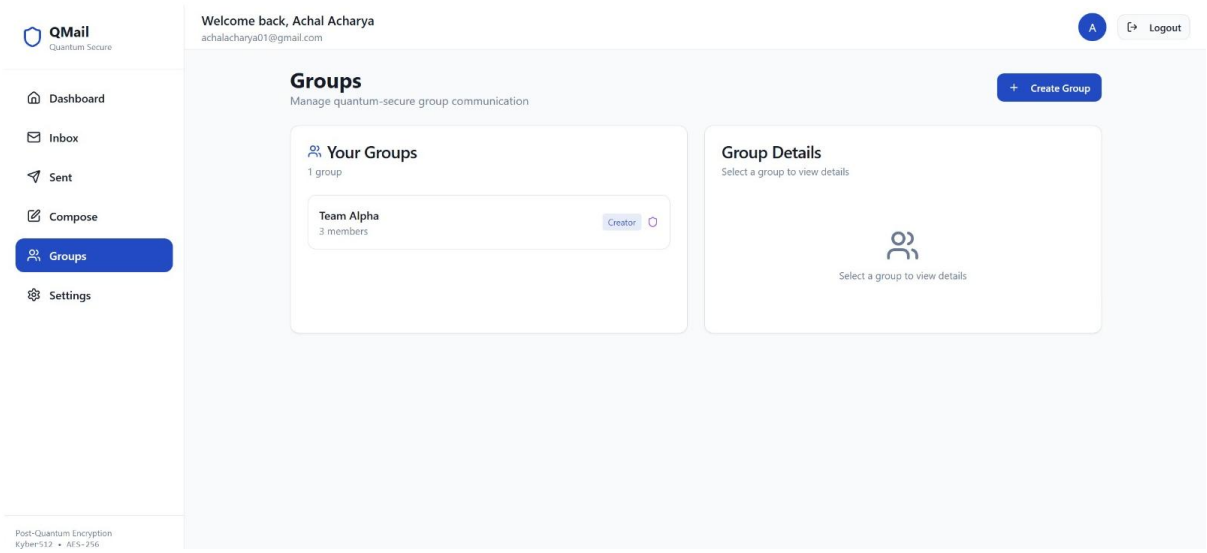


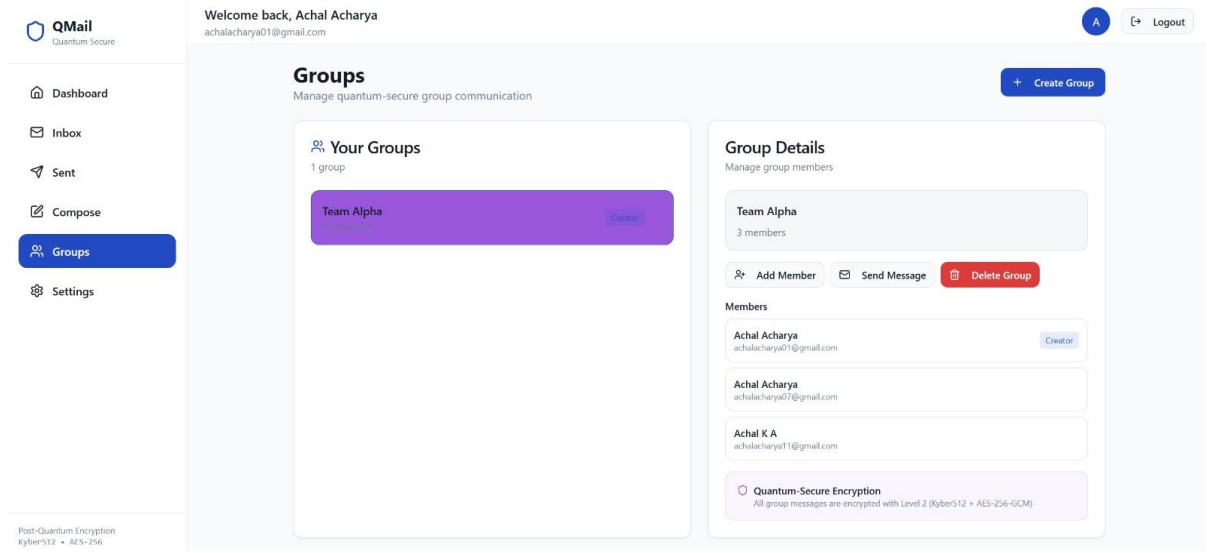
A.3 Compose Email Window (Encryption Levels)

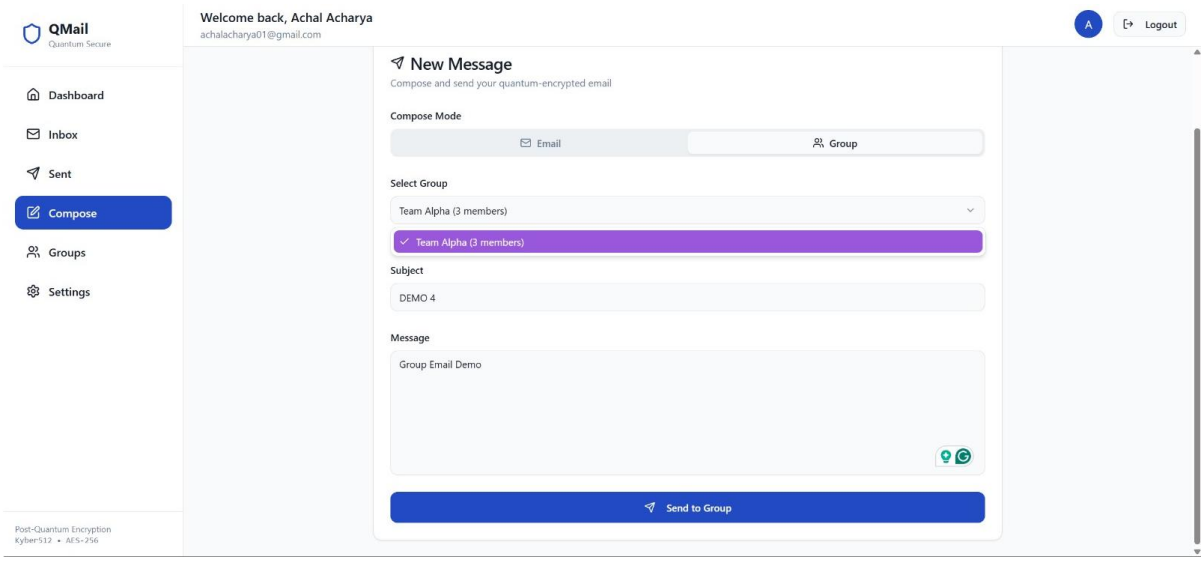

QMail
Quantum Secure
Welcome back, Achal Acharya
achalacharya01@gmail.com
A
Logout
Dashboard
Inbox
Sent
Compose
Groups
Settings
Compose Email
Send quantum-secure emails with multi-level encryption
New Message
Compose and send your quantum-encrypted email
Compose Mode
Email
Group
Recipient Email
recipient@example.com
Subject
Enter email subject
Message
Type your message here...
Post-Quantum Encryption
Kyber512 • AES-256


QMail
Quantum Secure
Welcome back, Achal Acharya
achalacharya01@gmail.com
A
Logout
Dashboard
Inbox
Sent
Compose
Groups
Settings
✓ Recipient has QMail device (Level 2/3 available)
Subject
DEMO 3
Message
Level 3 Demo
Encryption Level
Level 1: Normal Gmail
Standard email encryption (TLS only)
Level 2: Post-Quantum (Kyber + AES-GCM)
Quantum-resistant encryption with Kyber algorithm
Level 3: OTP + QKD
One-Time Pad with Quantum Key Distribution (Maximum Security)
Post-Quantum Encryption
Kyber512 • AES-256

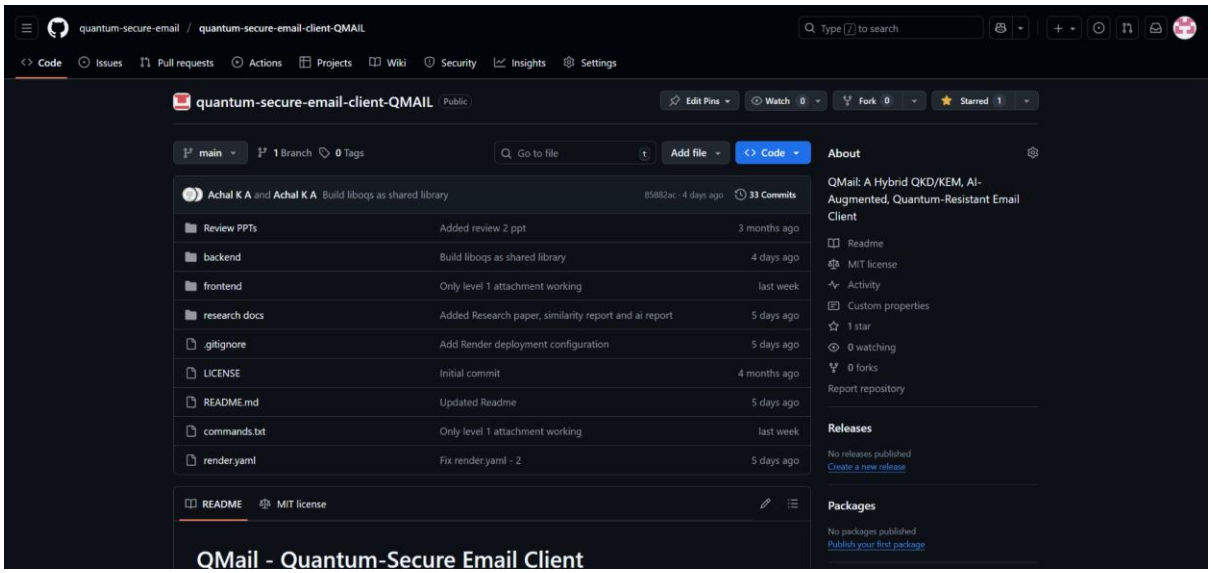
A.4 Group Email Interface







Appendix B – GitHub Repository



Appendix C – Similarity report

Battula Bhavya-QMAIL_REPORT.docx

ORIGINALITY REPORT

14%	8%	10%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	10%
2	"Quantum Protocols in Blockchain Security", Springer Science and Business Media LLC, 2025 Publication	1%

Appendix D – Publication

Submitted to the 8th International Conference on Communication, Devices and Networking (ICCDN-2026), Sikkim Manipal Institute of Technology, Sikkim

Hello.

Here is submission summary.

Track Name: ICCDN2026

Paper ID: 87

Paper Title: QMail: Hybrid Quantum Secure Email Client Application

Abstract:
The emergence of commercial quantum computers is inevitable and only a matter of time given the rapid progress made by companies such as IBM, Google, and Microsoft. Commercialization of quantum computers threatens current encryption methods that have worked effectively up until now however, with the advent of Shor's and Grover's algorithms (quantum algorithms) for breaking current encryption methods, this compromises confidentiality, integrity and authenticity. Securing communication channels must be a top priority, and since email is the most widely used communication method, it must be addressed first. QMail, a Quantum Secure Email Client Application (QSECA), is designed to withstand both classical and quantum-level attacks. It uses a hybrid architecture of both Quantum Key Distribution (QKD) for the session key and Post-Quantum Cryptography (PQC) for authentication and metadata protection, innovating on top of existing QSECA. The prototype we have made features a user-friendly interface to choose between three levels of encryption, including OTP. The results we obtained show that QMail ensures end-to-end security and provides protection against adversaries with quantum capabilities. The results we obtained show that QMail ensures end-to-end security and provides protection against adversaries with quantum capabilities. We have established QMail as a foundation for a scalable, quantum-safe communication platform which is a step towards secure communication in the era of 6G and quantum internet.

Created on: Mon, 17 Nov 2025 13:27:48 GMT

Last Modified: Mon, 17 Nov 2025 13:33:24 GMT

Authors:
- achalacharya07@gmail.com (Primary)
- roy.sugavasi@gmail.com
- satvikvarma03@gmail.com

Secondary Subject Areas: Not Entered

Submission Files:
QMail Research Paper - Springer.pdf (372 Kb, Mon, 17 Nov 2025 13:26:07 GMT)

Appendix E – Deployment Links

Frontend: https://qmail-frontend.onrender.com

Backend: https://qmail-backend.onrender.com