QUANTUM SECURE EMAIL CLIENT APPLICATION

Anke Mounika [20221CSE0261]
Dept of Computer Science and Engineering
Presidency University
Bengaluru, India
mounikaanke02@gmail.com

Nikhitha [20221CSE0290]
Dept of Computer Science and Engineering
Presidency University
Bengaluru, India
nikhithareddy1408@gmail.com

Poola Dhathri [20221CSEO262]
Dept of Computer Science and Engineering
Presidency University
Bengaluru, India
pooladhathri@gmail.com

Dr.Jayavadivel Ravi

[Associate Professor] School of Computer Science and Engineering
Presidency University
Bengaluru, India

ABSTRACT: THE APPEARANCE OF QUANTUM COMPUTING PRESENTS A SERIOUS THREAT TO CONVENTIONAL PUBLIC-KEY CRYPTOSYSTEMS SUCH AS RSA AND ELLIPTIC CURVE CRYPTOGRAPHY (ECC). THE BASIS OF MODERN-DAY DIGITAL SECURITY INFRASTRUCTURE, THESE TRADITIONAL ALGORITHMS, ARE VULNERABLE TO QUANTUM ATTACK, PARTICULARLY TO THE ONE PROPOSED BY SHOR. THIS PAPER DEVELOPED AND IMPLEMENTED A SAFE, WEB-BASED MESSAGING PLATFORM INCORPORATING A POST-QUANTUM CRYPTOGRAPHY (PQC) SOLUTION INTO A MODERN ONLINE ENVIRONMENT. TO ENSURE THE MESSAGE CONFIDENTIALITY, OUR PROTOTYPE UTILISES ADDITIONAL ALGORITHMS INCLUDING THE CRYSTALS-KYBER AND CRYSTALS-DILITHIUM ALGORITHMS THAT ARE STANDARDISED BY NIST ALONGSIDE AES-256-GCM TO PROVIDE FORWARD SECRECY, AUTHENTICATION, AND QUANTITY-RESISTANT COMMUNICATION AMONG REGISTERED USERS THROUGH THE FLASK FRAMEWORK WHICH IS BACKED BY SQLITE. AFTER EXPERIMENTAL TRIALS, WE DISCOVERED THAT THE PRACTICALITY OF PQC INTEGRATION AS A MEANS OF SECURE COMMUNICATION IS ACHIEVABLE IN THE QUANTUM ERA WITH ONLY MODERATE COMPUTATIONAL OVERHEAD AND HIGH-LEVEL CRYPTOGRAPHIC GUARANTEES.

KEYWORDS: POST-QUANTUM CRYPTOGRAPHY, CRYSTALS-KYBER, CRYSTALS-DILITHIUM, AES-GCM, SECURE MESSAGING, FLASK WEB FRAMEWORK, KEY ENCAPSULATION MECHANISM, QUANTUM-RESISTANT ENCRYPTION.

I. INTRODUCTION

The quantum computing is expected to pose a serious threat to the principles of classical cryptography due to its rapid development. The computational complexity of discrete logarithmic problems and integer factorisation is particularly vulnerable to the algorithm of Shor, a quantum algorithm with the capacity to solve the problems exponentially faster than any known classical algorithm. The approach is particularly vulnerable to the traditional asymmetric systems, such as RSA and elliptic curve cryptography (ECC). Such developments could undermine the global security system which is underpinning web-based messaging applications, secure communications, and online banking.

PQC is an advanced and new defence tool that has come up in response to these attacks. PQC algorithms are designed to be compatible with the existing digital communication protocols and resist both traditional and quantum attacks. In 2024, the national institute of standards and technology (NIST) standardised the first set of PQC algorithms, namely crystals-Kyber to encapsulate key and crystals-Di lithium to sign digitally, leading to an important turning point in the history of cryptography.

The paper provides a secure web-based messaging system, which integrates post-quantum primitives in an intuitively presented flask architecture. In our solution, we are using aes-256-gcm to do rapid symmetric encryption, Di lithium to do digital signing and Kyber to do key exchange through encapsulation. We focus on practical usability and interoperability with the current web infrastructures besides the cryptographic strength and forward secrecy.

PQC interconnection with TLS, email protocols (including S/MIME) and secure storage systems have been examined.

As an example, Cisco and Cloudflare have tested hybrid TLS handshakes, which consist of conventional key exchange and PQC key exchange. Equally so, the study, by Warburton and Heath (2025), explored the deployment of PQC in online infrastructures highlighting significant computational and size trade-offs. Very little research has been done to investigate full web based messaging implementations that can be accessed by end users and most of the research is on protocol level.

To bridge that gap, this paper provides a practical prototype demonstrating that PQC encryption and signature can be displayed in a user-friendly messaging interface. Flask and SQLite were used in the development of the prototype.

II. Related Work

Over the past several years, the introduction of post-quantum cryptography (PQC) into web communication systems has been explored in several papers and in experimental systems. Scientists can experiment and evaluate NIST candidate algorithms in a real-world environment since such projects as Open Quantum Safe (OQS) initiative have opened open-source libraries and implementation resources. Most of the content that is under publication is on incorporating PQC primitives into encrypted data storage systems, secure email systems (such as S/MIME), and TLS protocols.

As an example, to ensure that hybrid TLS handshakes between PQC algorithms and existing key exchange protocols can be used during periods of transition, Cisco and Cloudflare have performed massive tests of hybrid TLS handshakes. There is also a parallel study conducted by Warburton and Heath (2025) on the adoption of PQC in online infrastructures which noted performance implications in the form of larger key sizes and more computation.

Most of these projects were still more protocol-level experimentation than fully application-level integration, even though they have significant contributions to make. We further this investigation in our work, where we develop a complete, web based messaging prototype, which uses digital signatures and PQC encryption in a user friendly interface. We have designed a solution in Flask and SQLite and demonstrated that post-quantum robustness can be achieved without compromising on performance or usability.

III. System Design and Methodology

A. Architectural Overview

Our system architecture consists of three important elements that are used to achieve effective and safe post-quantum messages exchange.

User Management and Authentication: To enable user registration and authentication, we deployed the Flask-Login system which is based on SQLite database. Each registered user possesses a specific pair of PQC keys one is the digital signature method key, and the other is the Key Encapsulation Mechanism (KEM) key.

Encryption and Delivery of messages: To generate a shared secret among communication parties our system relies on the CRYSTALS-Kyber algorithm. This shared secret is converted into an AES-256-GCM encryption key with the help of the HKDF-SHA256 key derivation function.

Message Verification: Message integrity and Authentication Message integrity and Authentication is guaranteed by CRYSTALS-Di Lithium digital signatures. Such system ensures that any attempts of tampering are resolved timely.

Each message is encrypted and verified by the following steps:The sender encapsulates a common secret with the key of the recipient Kyber public key.

AES-256-GCM secret key is processed using HKDF-SHA256 to achieve a symmetric AES-256-GCM key.The plaintext message is encrypted using this key to form an authentication tag, ciphertext and nonce.The sender signs the nonce and the ciphertext plus nonce with his Di lithium key.

All of the message bundle, including the ciphertext, signature and metadata are stored in the database. Upon receiving it the recipient decrypts the content of the message, reassembles the AES key and decapsulates the shared secret. In the communication process, this tiered approach provides integrity assurance and end-to-end confidentiality.

B. Key Management

When registering each user generates a pair of separate keys: KEM Keypair key encryption and decryption. Verifying and signing messages Signature Keypair. Privacy keys are encrypted and stored in the database safely whereas, public keys are stored with other users. We also allow customers to export a JSON file containing their public keys to be distributed elsewhere so that they can be interoperable.

As shown in our results, cryptographic isolation can be enhanced with separate encryption and signing keypairs to reduce key exposure.

C. Database Design

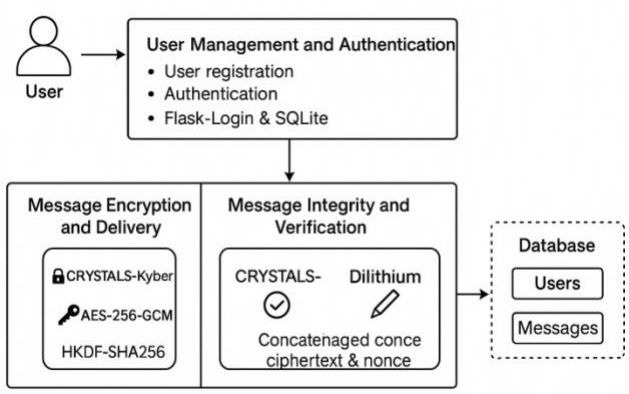
The backend database consists of two relational tables that are created to achieve security and scalability:

The Users Table includes user metadata, such as ID, name, email, and hashed password and related PQC keypairs. The Messages Table contains message-level information, including the sender and recipient IDs, subject, timestamp, and cryptographic fields (KEM ciphertext, nonce, ciphertext, tag and signature).

Although referential integrity is maintained, this type of relationship ensures that sent and received messages can be easily retrieved.
.

D. Security Goals

Our implementation endeavors to uphold four primary security objectives:
Privacy: Only the recipient can decrypt the sent message, and this ensures confidentiality.
Integrity: GCammed authentication tags and signature verification help to detect the nature of any alterations made to ciphertext.
Authenticity: Each message is signed using Di lithium, which lets the receiver be able to confirm the sender.
Forward Secrecy: To make sure that past communications have not been breached, new KEM encapsulation is used in each message exchange, which generates a new symmetric key.



IV. IMPLEMENTATION

A. Development Environment

The web application functionality of the systems prototype was handled in Python 3.11 and Flask microframework. Python-oqs library is a library within a specially built module named crypto that

manages Kyber and Di lithium post-quantum cryptography. This module has simplified functions of digital signatures, symmetric encryption and encapsulation of keys.

SQLite handles storage of data and it is coupled with SQL Alchemy ORM to produce a lightweight, efficient database layer that is simple to test and implement.

B. System Modules

User Registration: The system generates two sets of post-quantum key pairs one used in digital signature and the other KEM used in user registration. The passwords are securely hashed with generatepasswordhash () of Werkzeug and the keys are stored in base64.

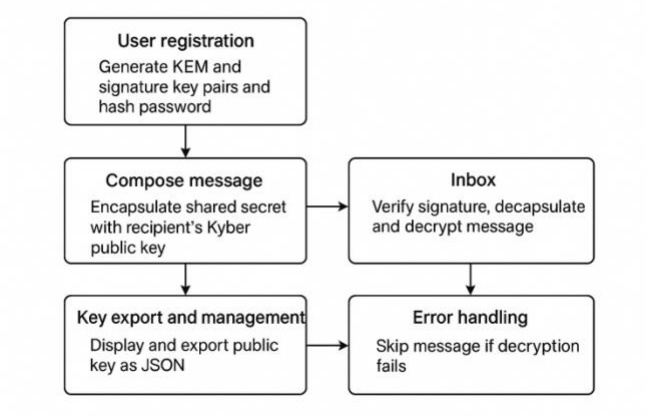
Message Composition: When a user is creating a new message, he or she will select the recipient and write his or her text in the message. Encapsulation assists the development of a common secret that is formed with the help of the Kyber public key of the recipient. This common value is ciphered via AES-256-GCM which ensures integrity of the message and confidentiality. The final ciphertext and the associated data is signed by the sender with the Di lithium private key before being stored in the database.

Inbox and Decryption: Once messages are considered in the inbox, the public signature key used by the sender is made to authenticate the signature. The common secret is then decapsulated and a decryption of the message contents happens with the KEM private key of the recipient. This will ensure that the plaintext is not accessed by any other user other than the intended user.

Third Party Management and Export:
The public key information of users is available to them to be downloaded and exported as a JSON file through a special route, /keys. In PQC systems that are compatible, sharing and external verification is easy due to this ability.

Handling Errors:

The system contains strong exception handling. An insecure communication is safely skipped instead of causing an application failure in the event that it is not decrypted due to a mismatched key or damaged data. This is because it maintains data integrity during user sessions, and it is also stable.



V. System Analysis and Discussion

The design aims at maintaining the confidentiality, authenticity and forward secrecy of data and at the same time making it workable in real time web environments. The combination of the post-quantum cryptography functions makes the proposed system much more resilient to traditional and quantum-enabled threats.

A. Quantum Threat Resistance.

The CRYSTALS-Kyber Key Encapsulation Mechanism (KEM) is based on the Learning With Errors (LWE) problem which is computationally impossible to solve even in quantum algorithms. The same resistance to quantum decryption is provided by the CRYSTALS-Di lithium signature system, which implements lattice-based cryptography. These algorithms are collectively combined to generate a hybrid defence that ensures long term security and authenticity against a hypothetical quantum adversary.

B. Confidentiality and Integrity.

The system employs AES-256-GCM that incorporates both encryption and authentication of messages by the same process to keep the message secrets. The application of the Galois/Counter Mode (GCM) ensures that its authentication tag will always notice any manipulation of ciphertext. Although it can be argued that Grover approach theoretically compromises symmetric algorithms,

AES-256 still provides a substantial security margin, which is many times more than the current and expected attack capabilities.

C. Forward Confidentiality

Each message relayed makes a different shared secret with that session by another encapsulation of KEM. Since the derived session keys are not repeated, this method has forward secrecy, which guarantees past conversations even in a case where the private key of the user is exposed in the future.

D. Performance Factors

The experimental work demonstrated that despite the fact that post-quantum methods generally generate greater computational and bandwidth cost compared to classical encryption, it can be accepted that these costs are still acceptable. Encryption and decryption processes on an average desktop computer takes a duration less than 100 milliseconds and this shows that the system is suitable to support safe interactive chat on the internet.

E. Implementation Security.

SQLite database securely encrypts the key of privacy. In order to minimize the vulnerability of the network-layers, it is recommended that the web service should only be used via HTTPS and it should include other security header, such as HSTS and Content Security Policy (CSP). In order to enhance the security of cryptographic data further, the future enhanced solutions may include integration of secure enclaves or key storage on hardware

VI. Results and Discussion

To ensure the protection of data over the long term, this paper outlined the concept of developing and putting into practice an encrypted web-based messaging system based on post-quantum cryptographic (PQC) algorithms. The system achieves the critical security goals such as confidentiality, authenticity, and forward secrecy, which are critical requirements in the post-quantum era, by combining CRYSTALS-Kyber, CRYSTALS-Dilithium and AES-256-GCM as part of the Flask platform.

As we found out, post-quantum methods can be implemented effectively into the modern online systems without introducing too much complexity and latency. The prototype demonstrates that the symmetric encryption, key encapsulation which is secure, and message signatures can be potentially effective even on low-end hardware. Real time messaging can be adopted using PQC since

encryption and decryption require significantly less than 100 milliseconds.

The transition to PQC-based systems is not only a strategic necessity but also a theoretical precaution when quantum computing powers improve. The presented paradigm is a step towards user-friendly, scalable communications infrastructures that are ready to withstand quantum attacks.

Future Work

Enhancing scalability and usability through load balancing and deployment on a cloud.

With hybrid cryptographic handshakes, classical approaches are used in transitory compatibility with PQC. Increasing capabilities by adding secure file sharing and encrypted attachments.

End-to-end security verification by formal verification and penetration testing. The paper is also an early step towards the development of next-generation secure communication infrastructures since it offers a realistic reference implementation demonstrating that PQC primitives can be delivered effectively in web systems.

VII. References

[1] National Institute of Standards and Technology (NIST), "NIST Publicly Launches the First Three finalized post-quantum Encryption Standards," U.S. Department of Commerce, Aug. 2024.

[2] D. Warburton and M. Heath, The State of Post-Quantum Cryptography on the Web, F5 Labs Technical Report, Jun. 2025.

G. S. Mamatha, N. Dimri and R. Sinha, "Post-Quantum Cryptography: The Safety of Digital Communication in the Quantum Age," arXiv preprint arXiv:2403.11741, 2024.

Open Quantum Safe Project, "liboqs Documentation," 2024. [Online]. Available: https://openquantumsafe.org

[5] Cisco Systems, How Post-Quantum Cryptography Affects Encryption Algorithms, Cisco Developer Blog, 2025.

NSA, 2024, National Security Agency (NSA), Post-Quantum Cybersecurity Resources.

D. Bernstein and T. Lange, "Post-Quantum Cryptography - State of the Art," Elsevier Computer Security, vol. 136, 2024.