

**Table of Contents**

**1. Research Question & Problem Statement**

**2. Intended Users, Requirements, and Market Gap**

**3. System Requirements, Project Deliverables, and Final Outcome**

**4. Primary Research Plan**

**5.Initial Literature Review**

**6.Bibliography**

**7. Appendices**

**1. Research Question & Problem Statement**

**Research Question:**  
How can Post-Quantum Cryptography (PQC) be integrated into a Zero Trust Architecture (ZTA)-based web application to deliver a secure, usable, and privacy-preserving password wallet without relying on hardware components?

**Problem Statement:**  
The advancement of quantum computing threatens to break widely used public key cryptographic schemes, rendering traditional password vaults vulnerable. At the same time, Zero Trust Architecture (ZTA) principles demand continuous verification, strong identity management, and context-aware access control. While NIST has standardized PQC algorithms like Kyber and Dilithium, their integration into web-based security workflows especially in a way that preserves usability and compliance is underexplored. Most PQC applications depend on hardware (e.g. TPM, YubiKey), which limits deployability and accessibility. ZTA enforcement via software-based contextual policies (location, device posture, time) using Open Policy Agent (OPA), replacing hardware-bound attestation

This project will:

* Develop a web application password vault using Kyber (KEM) and Dilithium (signatures).
* Enforce continuous authentication policies using ZTA principles.
* Deliver a biometric-friendly, mobile-responsive UI without hardware dependency.
* Ensure GDPR compliance via privacy-by-design and cryptographic auditing.

**2. Intended Users, Requirements, and Market Gap**

**Target User Analysis:**

|  |  |  |
| --- | --- | --- |
| **User Group** | **Requirements** | **Expected Benefits** |
| Web Developers | API-first PQC integration, UI flexibility | Rapid adoption of PQC-secure credentials |
| Cybersecurity Teams | Secure password storage, ZTA enforcement | Central policy management, credential resilience |
| Privacy Advocates | No device tracking, no hardware usage | GDPR-compliant password storage and sharing |
| Academic Researchers | Experimental PQC+ZTA integration | Prototype for testing quantum-safe security flows |

**Explanation of Benefits:**

* **Web Developers** benefit from flexibility and developer-friendly integration using RESTful APIs and open-source libraries (liboqs, WebAuthn).
* **Cybersecurity Teams** gain real-time visibility and cryptographic assurance that aligns with Zero Trust policies and post-quantum resistance.
* **Privacy Advocates** are assured of strong data privacy as the system uses encryption without hardware binding or tracking keeping everything software-based and privacy-preserving.
* **Researchers** can benchmark cryptographic operations, user flows, and policy enforcements within a controlled testbed helping validate academic models.

**Market Gap:**

* No web-based PQC-ZTA password managers exist.
* Existing wallets rely on hardware (YubiKey, TPM) or legacy encryption (RSA, ECC).
* PQC migration strategies in current tools (e.g. Bitwarden, 1Password) lack ZTA enforcement.

**3. System Requirements, Project Deliverables, and Final Outcome**

**System Specifications:**

* **Frontend**: ReactJS + WebAuthn (FIDO2, optional biometric prompt)
* **Backend:** Flask + PostgreSQL + RESTful APIs
* **Cryptography**: Standardize to Kyber-1024 (NIST Level 5) and Dilithium-III (Level 3) for consistent quantum security
* **ZTA Enforcement**: JSON-based policy engine (Open Policy Agent)
* **Dashboard**: Streamlit for real-time credential access visualization.
* **Session Initialization:** Starts with HTTPS login (PQC-enabled TLS) and user authentication.
* **PQC Device Attestation:** Uses Dilithium signatures to validate client context.
* **Policy Enforcement:** Real-time risk scoring via the ZTA Policy Engine informs access decisions.
* **Credential Handling:** Secrets are encrypted using Kyber + AES-GCM and acknowledged with SPHINCS+.
* **Step-up Authentication:** Enforced for sensitive requests using biometric re-verification.
* **Audit Trail:** Every encrypted action is recorded with post-quantum signatures for GDPR compliance.

**Deliverables and Milestones :**

The operational architecture of the web-based PQC-ZTA password vault spans secure login, real-time policy enforcement, and quantum-safe credential handling. The system begins with the user loading the web application over a PQC-enabled HTTPS connection. Login credentials are processed through an Argon2id hash and sent for PQC-based device attestation using Dilithium signatures. Real-time risk analysis is performed by the ZTA policy engine, which informs session validation and access control.

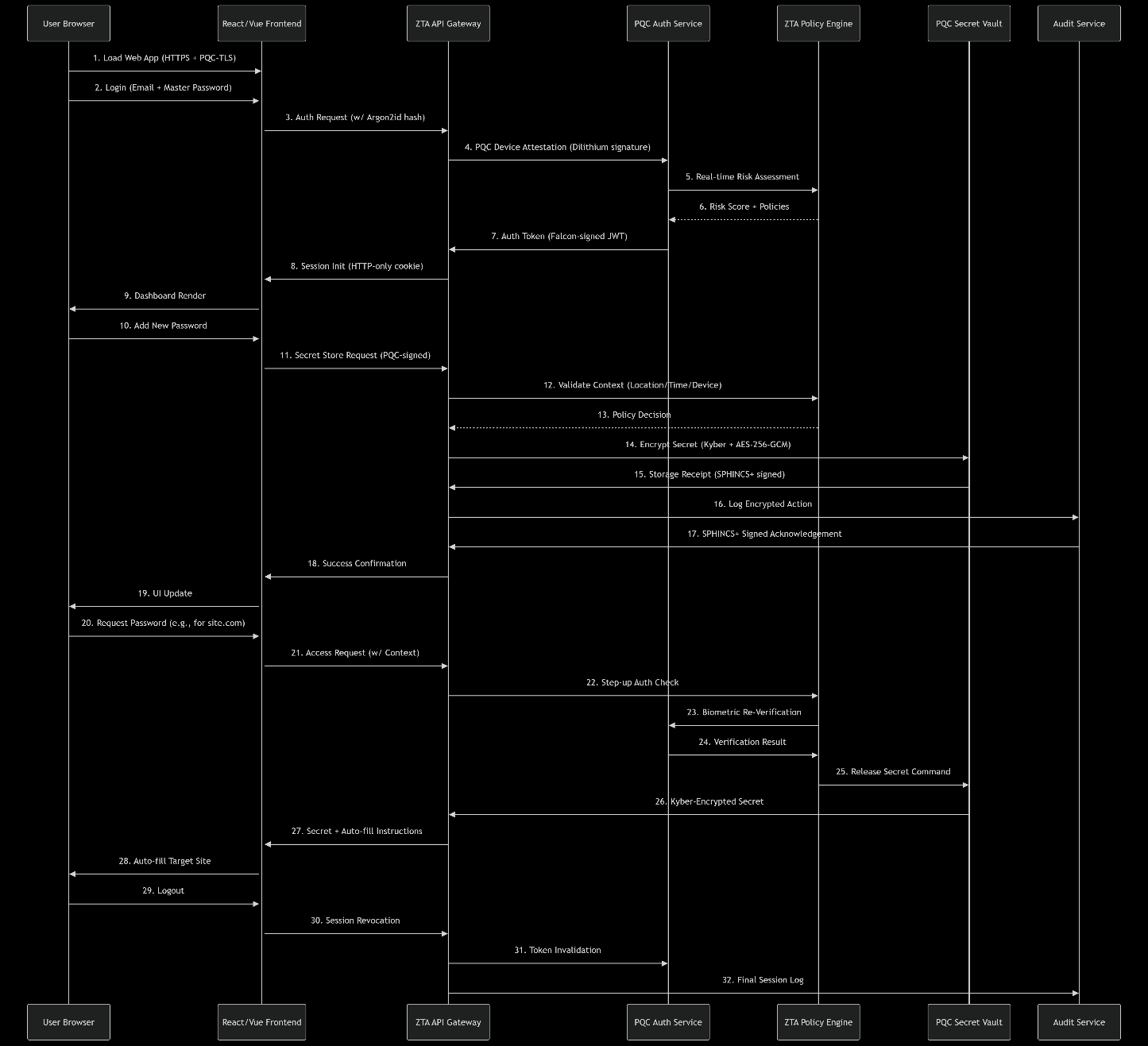
Once authenticated, an authorization token (Falcon-signed JWT) is issued and validated against policies defined by the Open Policy Agent. Secrets are encrypted using Kyber combined with AES-256-GCM for hybrid efficiency and stored in the secret vault. The storage receipt is signed using SPHINCS+ to ensure non-repudiation, and every action is logged cryptographically for GDPR compliance.

When a user accesses stored credentials, context (such as time, location, and browser ) is verified. If risk is elevated, a step-up authentication using WebAuthn is triggered. Approved requests result in secrets being decrypted and auto-filled, followed by token revocation and session logout. This full cycle ensures continuous verification, privacy, and quantum resilience without relying on hardware components.

**Outcome:**

The final deliverable will be a fully operational, web-based, quantum-secure password wallet system that implements advanced cryptographic and architectural principles tailored to the evolving cybersecurity landscape. Built without dependence on external hardware devices, this platform will:

* Be accessible via standard web browsers across platforms (desktop/mobile), supporting cross-device usage without requiring installation of browser extensions or client-side modules.
* Use lattice-based PQC algorithms (Kyber-768 and Dilithium-II) to achieve robust, forward-secure encryption and digital signing, ensuring full resistance to both classical and quantum computational attacks.
* Replace classical RSA/ECC encryption mechanisms entirely with NIST endorsed PQC schemes to eliminate vulnerability to future quantum adversaries.
* Enable biometric-friendly, passwordless login through WebAuthn integration, supporting facial recognition or fingerprint-based authentication through built-in device authenticators.
* Employ continuous verification and contextual access control using Zero Trust Architecture (ZTA) principles, ensuring real-time session validation and enforcing per-request trust decisions based on behavioral and environmental factors.
* Maintain full GDPR compliance by leveraging privacy-preserving audit logging (via SPHINCS+ signatures), token-based access revocation, and encryption-at-rest without requiring payload inspection.
* Provide real-time policy enforcement using Open Policy Agent (OPA), delivering a flexible and modular rules engine to secure user identity workflows and access privileges.
* Deliver high usability scores (System Usability Score >80) while maintaining operational performance (auth latency <100ms) through lightweight architecture and optimized cryptographic libraries (liboqs, OQS-OpenSSL).



**4. Primary Research Plan**

**Methodology :**

1. **Design Phase:**
   1. Sketch the overall idea of how the system works.
   2. Identify different user types (e.g., privacy-focused users, cybersecurity teams).
   3. Create diagrams that show the flow between frontend, backend, and cryptographic modules.
2. **Development Phase:**
   1. Use the liboqs library to include post-quantum cryptography algorithms (Kyber for encryption and Dilithium for signatures).
   2. Build a secure backend using Flask, and connect it to a PostgreSQL database.
   3. Design a user-friendly frontend using ReactJS, where users can register, login, and manage passwords.
3. **Policy Control using Zero Trust Architecture (ZTA):**
   1. Use Open Policy Agent (OPA) to enforce security rules.
   2. The system will make real-time access decisions based on context (location, time, device, etc.).
   3. This replaces the old “trust once, trust always” model with “always verify before access.”
4. **Testing and Optimization:**
   1. Check how fast each cryptographic function runs (target is under 100 milliseconds).
   2. Simulate attacks or odd behaviors (e.g., logins from unknown places) and make sure the system reacts correctly.
   3. Gather user feedback to ensure it’s easy to use and secure (aim for System Usability Score above 80).
5. **Design:** User persona creation, UI wireframes, architecture diagrams
6. **Development:** Integrate liboqs into Flask APIs; implement Kyber/Dilithium-based crypto
7. **Policy Control:** Use Open Policy Agent (OPA) for ZTA session validation
8. **Testing:**
   1. Latency (<100ms target for all crypto ops)
   2. Anomaly detection (login frequency, location changes)
   3. Usability metrics (System Usability Score >80)

**Data Sources:**

* **Synthetic Credentials:** Fake usernames and passwords generated with tools like Faker to test login and encryption securely.
* **Simulated Attacks:** Tools like OWASP ZAP and Burp Suite used to mimic real-world threats like brute-force or replay attacks.
* **Browser Storage Testing:** Local storage and session behavior checked using browser dev tools to ensure secure token handling.
* **Usability Simulation:** Puppeteer and Selenium used to automate login/logout flows and measure speed and ease of use.
* **OPA Logs:** Decision logs from Open Policy Agent help verify how the system handles rules during risky login behavior.

**5.Initial Literature Review**

A comprehensive understanding of Post-Quantum Cryptography (PQC) and Zero Trust Architecture (ZTA) requires examining both the theoretical foundation and the current state of applied research in quantum-resilient systems, credential management, and privacy-preserving web applications.

* **Alagic et al. (2022)** provide the core foundation for PQC through the standardization of Kyber and Dilithium as part of the NIST PQC project. Kyber is a lattice-based key encapsulation mechanism (KEM), and Dilithium is a lattice-based digital signature algorithm. These algorithms form the backbone of the cryptographic components used in the proposed password wallet system.
* **Rose (2020)** details the principles of Zero Trust Architecture in NIST SP 800-207. ZTA emphasizes the need for continuous verification, device posture checks, and contextual access controls, which are essential for secure cloud-native applications like password managers.
* **Boneh (2024)** explores the implications of quantum computing on classical cryptographic systems and highlights how most current password vaults using RSA or ECC are vulnerable. His work justifies the urgency of adopting PQC in authentication and encryption workflows.
* **Choudhury et al. (2024)** analyze the limitations of traditional network monitoring in encrypted environments and promote machine learning for anomaly detection. Their insights support integrating risk scoring and adaptive policy enforcement in Zero Trust systems.
* **Jietal. (2024)** present a literature review on AI-based anomaly detection in encrypted traffic, proposing privacy-respecting behavior-based analysis techniques. This aligns with the audit and compliance layer of the proposed system, where data minimization and behavioral analytics replace invasive payload inspection.

**Identified Gaps:**

The integration of post-quantum cryptographic methods with real-world, user-facing web applications especially in the domain of password management remains highly underdeveloped. The following gaps, drawn from a synthesis of academic literature and market research, highlight the motivation and uniqueness of this proposed project:

1. **Application-Layer PQC Scarcity**: Most research and industry focus on PQC is confined to infrastructure level implementations such as encrypted VPN tunnels, TLS protocols, firmware updates, and key management systems. There is an evident lack of effort toward developing browser based or web application-layer solutions that serve individual users without specialized hardware.
2. **Reliance on Classical Cryptography**: Popular credential managers (e.g., 1Password, LastPass, Bitwarden) continue to rely on RSA and ECC for key exchange and digital signatures. These are known to be susceptible to quantum attacks (e.g., Shor’s algorithm), creating future risks for all stored secrets, yet no mainstream provider has implemented lattice-based PQC standards.
3. **Hardware Dependency in PQC Adoption**: Experimental implementations of PQC often rely on hardware-bound security like TPM chips, YubiKeys, or custom secure enclaves. This limits the accessibility and affordability of such solutions for the general population or in environments where hardware tokens are impractical.
4. **Lack of PQC-WebAuthn Synergy**: Although WebAuthn is designed for strong, passwordless authentication and is a part of FIDO2, it is primarily tied to platform authenticators or external security keys. Integrating WebAuthn with PQC-based signing and verification workflows (e.g., using Dilithium instead of ECDSA) remains unexplored in live web systems.
5. **Missing Policy-Enforced PQC Identity Layers**: ZTA mandates policy-driven access decisions and continuous trust evaluation. Open Policy Agent (OPA) is emerging as a standard for policy enforcement in cloud-native environments, but its usage in enforcing quantum-resilient session controls, identity verification, and context-aware policies in password management systems is virtually nonexistent.
6. **Absence of Unified, Open-Source PQC-ZTA Platforms**: There is no available platform open-source or commercial that combines Kyber (key encapsulation), Dilithium (signatures), WebAuthn (authentication), and OPA (policy enforcement) in a fully functional, GDPR-compliant password vault system that is web-native and hardware-independent.

**6.Bibliography**

* **Alagic, G. et al. (2022)** – Provided the official NIST recommendations for post-quantum cryptographic standards, specifically introducing Kyber and Dilithium as the primary algorithms for key encapsulation and digital signatures. These algorithms are central to the cryptographic layer of the password wallet.
* **Boneh, D. (2024)** – Detailed how emerging quantum algorithms such as Shor’s and Grover’s can effectively break classical encryption schemes like RSA and ECC, supporting the case for adopting lattice-based post-quantum alternatives.
* **NCSC (2023)** – Published national-level quantum security guidelines which underscore the urgency for post-quantum preparedness in critical infrastructure, influencing the compliance and architecture planning for the system.
* **Rose, S. (2020)** – Authored NIST SP 800-207 on Zero Trust Architecture, defining principles such as continuous verification, context-aware access control, and policy enforcement, which directly informed the design of the ZTA layer in this project.
* **GDPR Article 32 (2018)** – Stressed the importance of data protection by design and by default, reinforcing the need for encryption, auditability, and minimal data exposure all of which are achieved through PQC-enhanced cryptographic logging and session control mechanisms.

**7. Appendices**

