Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Guardian-Based Anonymous Password Management with Privacy Preservation Using Threshold Cryptography

Harivignesh K.S.[1], Venkatesan R.[2], Selvarathi M.[3], Jasmine David D.[4]

[1,2]Division of Computer Science Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India.
[3]Division of Mathematics, Karunya Institute of Technology and Sciences, Coimbatore, India.
[4]School of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India.

E-mail: [1]harvicodes06@gmail.com, [2]rlvenkei2000@gmail.com, [3]selvarathi@karunya.edu, [4]jasmine.d@presidencyuniversity.in

Abstract

Most password managers are designed as a compromise between providing a secure way to handle difficult login credentials and creating a system that keeps user information private. Most password managers store users' sensitive information in a central location, leaving users vulnerable to hacking attacks. The proposed work creates a better way to manage passwords and provide more protection to users based on their private information through a new concept called Guardian-Anonymous Password Management (GAPM). The idea is to create a unique architecture that stores passwords in decentralized locations using guardian anonymity, creating a hybrid architecture of secret sharing with post-quantum encrypted wraps. Accordingly, GAPM separates the act of recovering user passwords from a person, using a set of anonymous guardians who securely recover users' passwords without putting any of them at risk of being hacked or located through social engineering techniques like phishing. This is achieved by using a Shamir-style secret sharing scheme combined with verifiably reassured commitments, where none of the guardians know each other, and they are required to reach a certain threshold of agreement to combine their shares into an easily accessible password recovery key. The GAPM system supports multiple guardian sets, allowing participants to be added or removed, and there is no need to reissue all the shares each time users make a change. The user can also change the recovery threshold in real-time. Finally, the shares are further secured through the use of a post-quantum Key Encapsulation Mechanism (KEM) to ensure that, no matter what kind of attack (classical or quantum), the password recovery process will remain strong and secure.

Keywords: Cryptography, Password Manager, Privacy, Secret Sharing, Post Quantum Cryptography, Security.

1. Introduction

Centralized storage, trusted third-party reliance, and user identity and metadata exposure pose a high level of risk for insecurity and exposure to breaches, making modern password managers a large target for large-scale breaches, regardless of the encryption

DOI: https://doi.org/10.36548/jtcsst.2026.2.004
Received: 21.02.2026, received in revised form: 23.03.2026, accepted: 11.04.2026, published: 24.04.2026

techniques used. In addition, existing threshold-based cryptography solutions fail to provide an anonymous solution and create additional attack surfaces by requiring mandatory user identification to participate. GAPM was created to address these limitations by using an anonymous and dynamic guardian set with a multi-layer approach to secret sharing. Each user's identity remains separate from the reconstructed secrets of the password management system. Post-quantum cryptographic protections allow for the sharing of password information with an unknown amount of time between each refresh. Moreover, the use of a dynamic guardian adds another layer of protection to the password management system and allows for the creation of a new guardian without user-level intervention. The introduction of a dedicated evaluation framework for empirical evidence-based security assessments enhances the value of GAPM through the availability of five measurable metrics that systematize and facilitate the reproducibility of security assessment processes in the field of distributed secret management systems.

2. Literature Review

In zero-trust multi-domain computing environments, a Decentralized Identity Management (DIM) framework is presented that includes Self-Sovereign Identity (SSI), through the use of Decentralized Identifiers (DIDs), Verifiable Credentials (VCs), and distributed ledger technologies. The design uses interoperability, privacy-preserving authentication and authorization, and fast proof of ownership across multiple heterogeneous domains, to provide a solution that operates with a latency of less than 1 millisecond. The design does not currently address adversarial Machine Learning (ML) attack threats or provide security against quantum computers [1]. A lightweight secret-sharing-based defensive framework is proposed to detect model poisoning attacks in privacy-preserving Federated Learning (FL), challenging the need for homomorphic encryption. Based on secure aggregation and encrypted cosine similarity computation, this defense is applicable for both IID and non-IID distributed data. The defense has a restricted learning system and lacks support for identity-centric access control [2].

DKSM proposes a blockchain-supported decentralized Kerberos service management scheme for the IoT, eliminating the need for a trusted KDC through consensus-driven operation and CP-ABE key management for access control. Experiments performed on the Ethereum network and consortium networks validate the scalability and economy of the scheme, but it still relies on classical crypto-theoretic assumptions [3]. A self-sovereign identity system based on Ethereum smart contracts is presented, achieving decentralization in identity registration and verification, decentralization, resulting in reduced gas prices compared to prior works. Though it is a solution for threats associated with centralized authentication, it is not a solution for volume attacks or quantum resistance [4]. Blockchain technology is used in title assessment by professionals in universities in a mobile IoT environment. While application-based, this demonstrates that blockchain technology can be used in other aspects of trust governance [5]. The BADIMAC initiative has developed a decentralized identity and access control system utilizing blockchain technology. BADIMAC incorporates zero-knowledge proofs as well as public-key cryptography to allow for self-sovereignty of identity. BADIMAC supports cross-organization verification of identity without requiring a trusted third party, resulting in improved transparency and increased user control of access to their information. Scalability for high volumes of verifying identities remains an issue to be resolved [6]. A privacy-protecting cybersecurity framework for smart grid communication combines Hierarchical Identity Based Encryption and Verifiable Secret Sharing Schemes (VSSS) to create a method for both

authentication and secure peer-to-peer energy trading without the involvement of third-party entities. Although the framework provides cryptographic strength, it focuses specifically on power distribution systems rather than a wider set of digital identity ecosystems [7]. Privacy leakage has been addressed in blockchain-based IoT identity systems with the introduction of PPID as a privacy-protected distributed identity management system. The combination of zero-knowledge proofs and Shamir secret sharing allows for the creation of PPID to provide unlinkable identities and resistance against replay attacks. Performance evaluations demonstrate that ZKP computational overhead is reasonable. However, the transparency-privacy trade-off of public blockchains still exists [8]. The solution is a decentralized cloud storage system based on a blockchain-based identity and access control model. In this model, data is encrypted using an identity token issued on the blockchain, and linear secret sharing is used to protect the keys used to encrypt the data. In addition, experimentation has shown that the costs involved in using this type of cloud storage are less than those of traditional systems and that there is no variable execution overhead. However, it is assumed that a traditional level of cryptographic security is maintained with respect to this new system [9]. This work addresses the challenges that arise in post-quantum security by creating the first deterministic isogeny-based wallet, which is composed of compact wallet key sizes. This contribution includes the construction of re-randomizable schemes for signing instances using an arbitrary group action, along with achieving both unlinkable and unforgeable signatures against quantum attacks. Although primarily targeted at blockchain wallets, the findings of this research could lead to post-quantum secure identity and authentication solutions through the underlying cryptographic principles [10]. Through the use of CP-ABE, blockchain smart contracts, zero-knowledge proofs, and IPFS, MIoT-CDPS manages data self-sovereignty, fine-grained access control, and transparency. Users no longer need to rely on key centers or cloud servers, which were standard for previous solutions; with MIoT-CDPS, users are responsible for the management of their own keys. Using formal proof, AVISPA Analysis, and an implementation on Ethereum, it is resistant to impersonation, replay, man-in-the-middle, and insider attacks [11].

The proposed design of a fully decentralized and multi-platform wallet, offering higher levels of security and usability for decentralized Applications (dApps), does not depend on any third party. The architecture utilizes blockchain technology and smart contract functionality, as well as the IPFS protocol, combined with BIP-32/BIP-39 hierarchical key management systems to ensure the safe creation and recovery of keys without being linked together or identified. The smart contracts will allow users to manage device access and reference to encrypted keys [12]. A comprehensive wallet classification framework is proposed that it includes both traditional and emerging designs. A total of 33 academic studies and 85 wallet incidents, spanning from 2012 to 2025, resulted in losses of approximately $6.98 billion. The layered threat taxonomy is based on five categories: network, application, authentication, storage, and cryptography. Proactive/reactive classifications are created for defensive mechanisms based upon the specific attacks and designs that result in the defensive mechanisms being used to protect against those attacks [13]. The hierarchical multi-authority framework based on blockchain technology with IPFS offers a remedy to issues related to a single point of failure, depending on trusted third-party systems, as well as enabling auditability of user access rights/access control. The proposed model incorporates double-level user key management systems with SHA-256 hashing algorithms to offer a semi-decentralized hub-and-spoke architecture to secure against collusions/insider attacks as well as decrease computational and communication costs compared to centralized CP-ABE schemes [14]. Kintsugi is a key recovery system based on thresholds that utilizes OPRFs, Shamir Secret Sharing, and proactive secret sharing, and is independent of trusted hardware and third-party

services. This system allows the recovery of keys using low-entropy passwords in a manner that is resistant to offline brute force attacks and colluding nodes. Regular refreshes of shares protect against honest-but-curious and malicious recovery nodes [15]. The Shamir-based ramp secret sharing protocol provides strong security assurances for the storage of keys. The protocol is storage-efficient and does not allow for any exposure of sensitive data until k shares have been compromised. This was demonstrated using Shannon entropy. Through the use of the $Raspberry\ Pi$, the computational overhead was found to be much lower than that of previous protocols, and it decreased as k increased [16]. To address quantum-related threats as well as the shortcomings of NTRU, the authors have proposed NTRUSSS, an innovative post-quantum signcryption scheme that enables confidentiality, integrity, forward secrecy, authentication, and non-repudiation. NTRUSSS employs secret sharing for private key exchange, thus enhancing the security of both sides of the exchange against MITM attacks and/or key exposure while enabling fast, secure key exchanges with little additional overhead compared to that provided by its predecessors (i.e., RSA, ECC, and AES) [17]. The CBSS framework removes the need for a trusted dealer by using two-factor authentication through email. Five new algorithms for share generation are tested using NIST SP 800-22 randomness tests, time complexity analysis, and PSNR analysis [18]. DIDAuth-IoTFW envisions an extensive firmware authentication lifecycle by utilizing W3C-conformant DIDs, verifiable credentials, Ethereum Layer-2 (Arbitrum) smart contracts, and IPFS. This method securely links firmware with credentials and facilitates device authorization even when gateways are compromised. Formal proofs and prototype implementations on ESP32 and Raspberry Pi validate the resistance to forgery, replay attacks, and tampering attacks, with the latency of verification shown to be less than 1.2 seconds [19]. This produced empirically based results from an analysis of 12 desktop password managers, 12 browser plugins, 5 native browser password managers, and 21 VPN Clients, which is in contrast to previous studies. The current study showed the presence of credential leak due to RAM in 75% of Password Managers and 33% of VPN Clients (CWE-316). The methodology used to exploit memory patterns created by responsible disclosure, along with CVEs and the open-source Pandora tool, can link research to a real impact in the field [20].

The comprehensive signature and verification model represents Trust Lists as Verifiable Credentials that are associated with DIDs. It uses DNSSEC as the sovereign Trust Anchors and provides a way for independent and interoperable Trust Management to occur. The Gaia-X use case and implementation that supports IPFS, XML/JSON Trust Lists, and multiple DID methods provide evidence of the feasibility and scalability needed for future infrastructures such as EUDI and Cross-Border Data Spaces [21]. The Shamir Secret Sharing–Partial Encryption framework targets only the essential shares for encryption. By doing so, it reduces computation and resource usage while preserving information-theoretic security. The approach blends Shamir's (k, n) scheme with selective AES encryption, ensuring that at least one required share remains encrypted to block secret recovery by adversaries. Experimental results on Rayleigh fading channels demonstrate up to 3.32×higher secrecy throughput and 1.38×lower latency compared with fragmented AES, all while maintaining the same security guarantees [22]. Collaborative Credentials (CCs) are introduced to enhance W3C Verifiable Credentials to depict the dynamic and collective capabilities of IoT devices and/or the actors collaborating with them. CCs differ from traditional or standalone group credentials in that they originate from real-time collaborative efforts and delegations. The CC construction describes the CC lifecycle and provides a detailed evaluation of CC security, including an assessment of Sybil, DoS, impersonation, and delegation abuse attacks against CCs [23]. The use of blockchain and smart contracts facilitates traceable and fine-grained data sharing related to agriculture on a network. With the weaving of self-sovereign identity components such as

DIDs, verifiable credentials, and ZCAPs, safe inter-organizational access control can be facilitated. A probe for data middleware provides convergence to agricultural dataspaces with no change to the source [24]. This uncovers that the regular digital signatures deployed in verifiable presentations can link a user's identity across multiple transactions. The DID-based authentication scheme utilizes an attribute-based signature to conduct anonymous authentication, showing actions instead of identities. A controlled tracing system allows authorized issuers to deanonymize bad actors by balancing accountability with privacy [25].

3. Methodology

The Guardian Anonymous Password Management system enables users to have a secure, private vault for their passwords and eliminates the threat of having one central location that can be compromised. This system accomplishes this by utilizing threshold cryptography, verifiable secret sharing, and post-quantum cryptographic protection.

When a user registers with the Guardian-Anonymous system, they do not enter their master password in plain text. Instead, the user's master password is stored in encrypted format using PBKDF2 with a high iteration count and random salt, to protect it from being brute forced by an offline attacker. Additionally, a separate cryptographic secret is deterministically derived from the master password and random salt, which then serves as the seed value for a (t, n) Shamir's Secret Sharing scheme enables users to distribute their trust among multiple, independent guardians.

Feldman Verifiable Secret Sharing (VSS) is incorporated into the system to protect against malicious tampering and to allow proof that the data within the shares was created by an entity authorized to do so. This is done by publishing polynomial commitments at the time of creation for the shares, so that anyone who reconstructs a share can prove that it was created by an entity authorized to create shares, without having to disclose the actual secret. Furthermore, Proactive Secret Sharing (PSS) is used to periodically update the shares, without changing the original data, to reduce the risks associated with long-term key exposure and mobile threats. The guardian shares are protected by using a post-quantum Key Encapsulation Mechanism (KEM), and a Kyber-like abstraction combined with Advanced Encryption Standard (AES) Galois Counter Mode (GCM), which allows them to remain confidential against both classical and quantum-capable adversaries.

3.1 System Overview

The proposed Guardian-Anonymous Password Management system allows an individual’s passwords to be kept secure without relying on a single source of truth. It accomplishes this by using a decentralized approach involving storing passwords via an innovative method of threshold cryptography, secure multiparty computation, and quantum-resistant cryptography, which means that sensitive information can never be revealed through one or more vulnerability points. Users will communicate with one another and their distributed “guardians”, while the “guardian” layer adds further layers of protection in the form of secure authentication, recovery, and the ability to detect multiple attempts to gain unauthorized access simultaneously. All operations involving password management will take place in a cryptographically secure manner, making it impossible to compromise one’s passwords unless the appropriate verification threshold is met and all other verification criteria are fulfilled. In the proposed system, Multi-Party Computation (MPC) is implicitly realized through the

threshold reconstruction process. Instead of a single party holding the secret, the secret is distributed among n guardians. The secret is reconstructed by combining shares from at least t guardians, a process that can be viewed as a collaborative computation where no single guardian ever possesses the full secret. This ensures that no individual party can unilaterally compromise the user's credentials, embodying the core principle of MPC.

3.2 Cryptographic Foundations

The security of the proposed methodology is derived from existing cryptographic primitives. Passwords are hardened with PBKDF2 using HMAC-SHA256, and the number of iterations is secure against offline brute force attacks The confidentiality and integrity of stored data are ensured through authenticated encryption using AES-GCM. The methodology achieves threshold security by using Shamir Secret Sharing in a large prime field and will use Feldman Verifiable Secret Sharing to provide an integrity and tamper-detection mechanism. Long-term and mobile adversarial models are addressed using proactive secret sharing, which allows shares to be refreshed periodically without changing the underlying secret. The overall approach employs Guardian-hold shares, which protect against both classical and quantum adversarial models, through the use of a post-quantum Key Encapsulation Mechanism combined with symmetric authenticated encryption. The cryptographic foundations employ Kyber-512, a post-quantum Key Encapsulation Mechanism (KEM) selected by NIST for standardization. This algorithm provides security against both classical and quantum computer-based attacks by relying on the hardness of the Module-LWE (Learning with Errors) problem, ensuring the long-term confidentiality of guardian shares.

3.3 Algorithmic Design

Eight algorithms have been defined as part of an overall framework for the system, which are linked to one another, dependent on one another, and will control all aspects of the system regarding authentication, secret distribution, protection, recovery, and evaluation. The first algorithm performs two procedures. It includes both a procedure for hardening Master Passwords and for securely deriving new secrets from them. Following this, the second algorithm will implement Threshold-Based Secret Sharing and produce verifiable commitments. Guardian Shares will be generated or provided in an encrypted form using a post-quantum format before they are made available for distribution. To facilitate recovery operations, it is necessary for a user to successfully reconstruct the secret or retrieve the original share from a minimum number of verified shares along with knowledge factors contributed by the user. The algorithms that generate Security Metrics are responsible for ongoing evaluations of the system's susceptibility to brute force attack, tampering with shares, impersonation or identity theft, and failure to retrieve or reconstruct the secret. The modular format of all the algorithms will allow for both easy extension and formal verification, in addition to supporting rigorous security assessments. The algorithms are designed to execute sequentially as a cohesive workflow. For instance, during registration, the flow is: PBKDF2 → Secret Generation → Shamir Secret Sharing → VSS Commitment → PQ-KEM Encryption. Each step's output feeds directly into the next. While the generation of individual shares for n Guardians can be parallelized for efficiency, the logical flow is strictly sequential to maintain a coherent security state.

3.4 Security Workflow

The continuous process of the defence-in-depth strategy for system operations is attained via the security workflow that controls all activities within the system. To address the vulnerabilities in terms of controlling the attacks on authentication requests, the following techniques can be employed; rate limiting, randomized delay times, and account lockout after a set duration. Every security event is logged immediately in the database. It involves all events that are logged in relation to failed authentication attempts, recovery, and verification processes. Additionally, any data captured from simulated attacks and adversarial interactions becomes an input into the Security Measurement Engine, which provides a quantitative assessment of the robustness of systems. Furthermore, the closed-loop nature of the security workflow provides both cryptographic security enforcement and empirical validation of system architecture effectiveness.

3.5 Dataset

The Brute Force Database - Password Dictionaries collection is a password wordlist for studying cybersecurity password guessing technique/brute force and dictionary attacks through a curated password wordlist collection. Password wordlists are compiled from publicly available password lists as well as user-generated password lists, specifically focusing on an "8-More-Passwords.txt" file, which contains only passwords of more than eight characters, as along with filtered password entries possessing specific attributes. Moreover, this dataset is beneficial for evaluating complex password patterns and determining attack strategies through brute force. Password wordlists provide an opportunity for automating password cracking, assessing password strength, and building a better authentication security system through the use of a representative password data store to train or test password security tools [26].

3.6 Architecture

The architecture is designed for a zero-trust distributed authority model where the user interface facilitates secure authentication, but the user's secret material is split into multiple threshold-based shares. The master secret can never be recreated unless at least t valid guardian shares are presented to the system, along with verification through Feldman commitments to maintain the trustworthiness of the structure in the event that there are malicious guardians in the system. The architecture utilizes post-quantum encryption technology to encrypt each individual share of the user's secret during storage and transport. Additionally, proactive refresh can be utilized to minimize long-term share breaches. The architecture employs decentralized IPFS storage for encrypted vault metadata, rather than a centralized server. Secret shares are distributed among n independent guardians using threshold cryptography, ensuring that no single entity holds complete secrets. The master secret cannot be reconstructed without the collaboration of at least t guardians. An attacker compromising any single component, such as IPFS nodes, guardians, or the session manager, gains only encrypted ciphertexts or incomplete shares, which are cryptographically useless without threshold-level collusion.

The security of this architecture is being enforced continuously, and all events that occur resulting in an authentication failure, delay, freeze, or recovery are stored by the audit module, while the live metrics engine tracks network statistics and links these to an attack simulation. Hence, a combination of two evaluation methods, cryptographic soundness and empirical testing, results in higher levels of assurance that this architecture provides a secure

environment than any previously developed password manager for this level of assurance about future quantum-based applications.

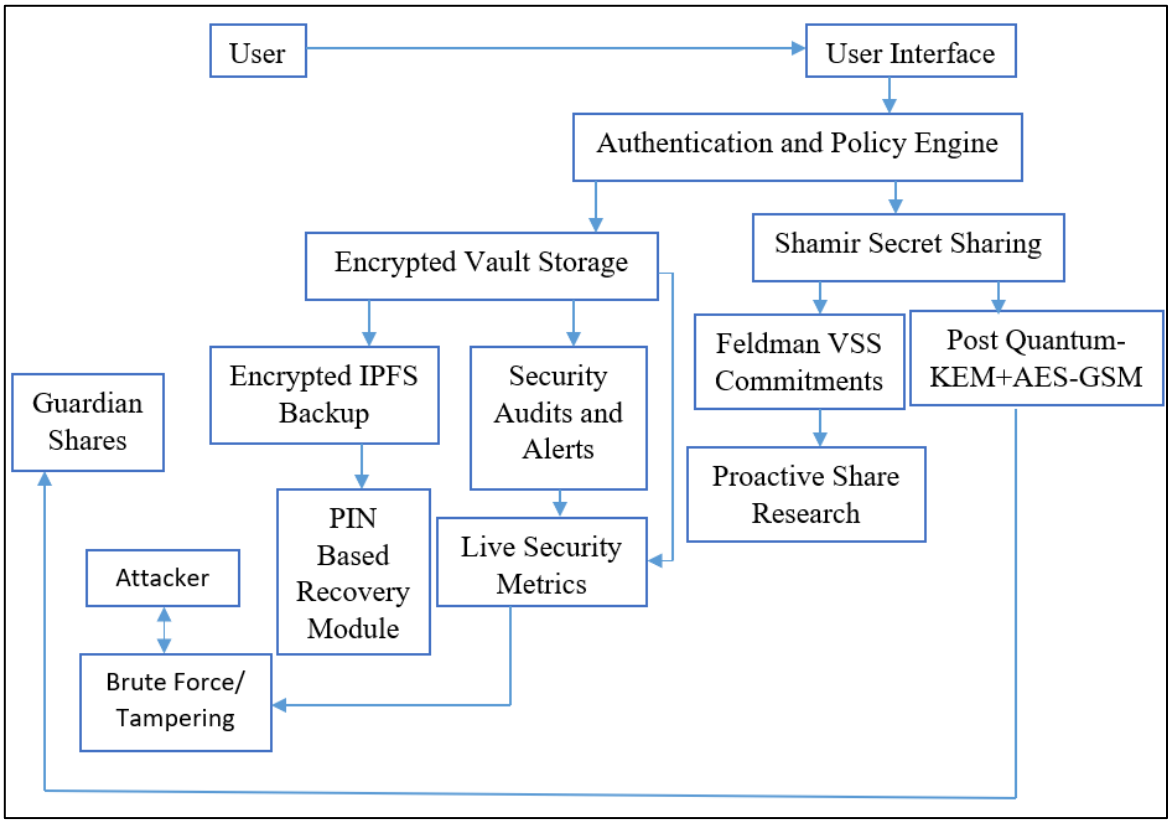

Figure 1. Architecture of the Proposed System

The architecture, as shown in Figure 1, provides an encrypted password manager using an architecture with distributed trust and multiple layers of security. The user interacts with the password manager through the command line, which then forwards all requests to the authentication and policy engine that implements password policy rules and limits how many times a user's account can be accessed per minute. The user's master password is enhanced with PBKDF2 and stored only in the vault as an encrypted proxy to create the necessary cryptographic secrets.

All sensitive data is stored only as encrypted data in the vault. To prevent a single point of failure in the system, the password manager creates a group of guardian secrets using Shamir's Secret Sharing and uses a configurable (t, n) threshold and Feldman's Verifiable Secret Sharing scheme to identify tampered or invalid shares and proactively refresh shares for long-term and mobile attacks. Guardian-held shares are secured through a post-quantum key encapsulation method using AES-GCM. For availability and disaster recovery purposes, the metadata of the encrypted vault is stored on the IPFS system and can be recovered using a PIN-based recovery method, which requires both the user to know the PIN and the threshold reconstruction of the guardian shares. The security audit system and alerting system capture and continuously evaluate all security-relevant events and actions, including brute force attack attempts, as well as the manipulation of the secret shares, through the Security Metrics Engine and the Live Security Metrics Engine.

3.7 Actors and Threat Model

The system's design includes three major actors: User, Guardian, and Attacker. The first is a legitimate owner of the password vault who initiates the authentication process by accessing the vault, storing keys, and recovering lost ones, when necessary (i.e., when a password is forgotten). Guardians are considered to be semi-trusted, independent entities that share their secrets using cryptographic methods to help protect the User's secrets while ensuring that all Guardians do not collude to violate the User's trust threshold $(t-1)$ of the total number of possible Guardians (t). The Attacker is modelled under both offline and online attack assumptions, with the ability to perform brute force attempts, tampering, replay, impersonation, and other forms of attack against the users using the vault. It is assumed that the attacker is computationally bounded in the classical sense, but may have the potential ability to use quantum computing to protect the system after the technology develops. The security of the system is maintained as long as there are fewer than (t) compromised Guardians, and all of the existing standard cryptographic assumptions (e.g., PRF, IND-CCA, and EUF-CMA) remain true.

3.8 System Model

The Guardian-Anonymous Password Management (GAPM) system employs a threshold-based architecture comprising four principal entities: the $User\ (U)$, a set of n Guardians $\{\mathrm{G}\} = \{G_1, G_2, \dots, G_n\}$, a centralized $Vault\ Server(S)$, and decentralized IPFS Storage ($\$D\$$). The approach is based on the trusted framework of (t, n) threshold Shamir's Secret Sharing (SSS) where $\$t\$$ is the minimum number of shares necessary for recovery. This ensures that the secret share of every guardian is protected by a post-quantum key encapsulation method (PQ-KEM) providing Quantum Resistance. The protocol consists of four major flows: registration, login, password reset and recovery. During registration, a Feldman commitment is used to establish verifiable secret sharing. When logging into the system, session binding occurs using HMAC for client validation. The recovery flow utilizes Lagrange interpolation to recover data from $\$t\$$ guardians. Additionally, PSS and DSS provide proactive and dynamic thresholds to allow forward security and operational flexibility. The login protocol does exist, as detailed in Section 3.8 and the code $(user.py)$. Session binding via HMAC is used after a successful login to bind the authenticated session to the client's IP address. This prevents session hijacking or replay attacks where a stolen session token might be used from a different network location, ensuring that the authenticated session is tied to the legitimate user's network context. DSS stands for Dynamic Secret Sharing. It is a vital component that allows the system to modify the guardian set or the recovery threshold without changing the underlying master secret. This is achieved through the $dss\_add\_guardian$, $dss\_remove\_guardian$, and $dss\_change\_threshold$ functions in the code, which use the VSS commitments to add or remove shares and adjust the polynomial degree without disrupting the system's overall security.

3.8.1 Password Hardening Using PBKDF2

This system implements PBKDF2 (Password-Based Key Derivation Function 2) to protect against attacks that target the user's master password; for example, using brute force attacks (both online and offline). PBKDF2 creates a fixed-length cryptographic hash of the user’s master password by applying an algorithm that iterates multiple times using the user’s original password and a unique salt generated for each individual user. By making it

computationally expensive to guess the password through brute force methods either via dictionary attacks or rainbow tables, PBKDF2 protects against these types of attacks. PBKDF2 is also used to create cryptographic keys for encrypting and decrypting vaults and for recovering deleted vaults. Consequently, all functions dependent upon a user's master password will have a consistent level of resistance from potential attack methods. A random salt is generated with the use of a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG), namely $os.urandom(16)$ in the implementation code. All salts are represented by a Base64 encoded string saved along with their associated password hash/derived key inside a user's vault. The randomness and uniqueness of salts help prevent the use of precomputation attacks such as rainbow tables, which require prior knowledge of the salts being used.

The master password is not saved in plaintext. The hashed master password generated with the aid of the PBKDF2 algorithm is kept for use in the master password verification process on logins. The PBKDF2 algorithm is used as the Key Derivation Function for generating the hash of the master password, which is in turn used for the derivation of encryption keys. The iteration count is chosen to be 200,000, as shown in $pbkdf2\_hash$ and $verify\_pbkdf2$ functions. This value is selected based on current industry standards and recommendations (e.g., OWASP) to balance security and performance. It is computationally expensive enough to make brute force attacks infeasible while maintaining a sub-second latency on modern hardware for legitimate users.

Algorithm 1: PBKDF2-Based Password Hashing

Require: Password P, Salt S, Iterations c, Output length dkLen
Ensure: Derived key DK

1. $U_1 \leftarrow \mathrm{HMAC}_{\mathrm{SHA256}}(P, S \parallel 1)$
2. for $i = 2$ to c do
3. $U_i \leftarrow \mathrm{HMAC}_{\mathrm{SHA256}}(P, U_{i-1})$
4. end for
5. $DK \leftarrow U_1 \oplus U_2 \oplus \cdots \oplus U_c$
6. return DK

The algorithm runs in $O(c)$ time where $c = 200{,}000$ iterations. Each iteration performs one $HMAC - SHA256$ computation. Output length is 32 bytes (256 bits).

3.8.2 Threshold-Based Secret Protection Using Shamir's Secret Sharing

To avoid having a single point of failure in case of a password recovery scenario, the system uses Shamir's Secret Sharing (SSS). The master key is divided into several parts using SSS, which can be stored with different guardians. Random polynomials of degree $t - 1$ are formed within a finite field. The first term of such polynomials holds the secret. The guardians are assigned shares based on evaluating the polynomials. Hence, the secret can be recovered only by combining at least t valid shares. This technique is not only resistant to failures but also protects against any possible collusion attacks. Parameters n and t represent the number of guardians and the recovery threshold, respectively. These parameters can be set by users during the registration phase according to the constraint $1 <= t <= n$. Parameter t represents the minimum number of guardian shares needed for recovering the master secret. There are different possible configurations in the system, such as $t = 3, n = 5$.

Shamir Secret Sharing employs a random polynomial of degree $t-1$ over a finite field. The secret s is embedded as the constant term $f(0) = s$. Each share is a point $(x_i, f(x_i))$ on this polynomial, where x_i is a unique, non-zero identifier. Reconstruction uses Lagrange interpolation to find the constant term from any t points, recovering the secret. The finite field is defined by a large prime number $p = 2^{127} - 1$. This 127-bit Mersenne prime is chosen for its computational efficiency in modular arithmetic while providing a security level comparable to standard cryptographic operations, making it suitable for the secret sharing scheme. This specific prime, known as a Mersenne prime, is chosen for three reasons. First, its form $2^{127} - 1$ allows for fast modular reduction using bitwise operations, significantly improving computational efficiency. Second, it provides 127-bit security, which is comparable to standard cryptographic operations such as AES-128. Third, it is large enough to prevent brute force attacks while remaining computationally efficient for practical implementation. The prime satisfies the requirement $p > s$ where s is the secret, ensuring the secret is properly contained within the finite field.

Algorithm 2: Shamir Secret Sharing

Require: Secret s, Threshold t, Total shares n, Prime p
Ensure: Shares (x_i, y_i)

1. Choose random coefficients $a_1, a_2, \ldots, a_{t-1} \in \mathbb{Z}_p$
2. Define the polynomial:
   $$f(x) = s + \sum_{j=1}^{t-1} a_j x^j (\text{mod } p)$$
3. for $i = 1$ to n do
4. $y_i \leftarrow f(i)$
5. Output share (x_i, y_i)
6. end for

The algorithm runs in $O(n \cdot t)$ time. For $n = 5, t = 3$, this requires approximately 15 modular multiplications over prime field $p = 2^{127} - 1$.

3.8.3 Share Integrity Assurance Using Feldman Verifiable Secret Sharing

The VSS Technique of Feldman is applied to authenticate whether the user has tampered with the shares of their guardian. The polynomial's coefficients are made public using modular exponentiation; thus, the guardian and the system can both check whether the share he receives is genuine by comparing it with the public share without disclosing the secret of the polynomial. If the received share is tampered with, it fails the verification process and helps detect an internal attack quickly. The false positive rate for share tampering detection is 0%. This is theoretically guaranteed by the Feldman Verifiable Secret Sharing (VSS) scheme. The verification algorithm cryptographically proves whether a share corresponds to the original polynomial commitment. An unmodified share will always validate, while any tampered share will fail verification with overwhelming probability, as it would require solving the discrete logarithm problem to forge a valid share. These zero false positive rates are mathematically guaranteed by the Feldman Verifiable Secret Sharing scheme. The verification equation checks whether $g^y \equiv \prod_{i=0}^{t-1} {C_i}^{x^i} (\text{mod } p)$ An unmodified share always satisfies this equation because the commitments C_i are derived from the original polynomial coefficients. A tampered share would require solving the discrete logarithm problem to find a y' that satisfies the equation for

the given commitments, which is computationally infeasible. Therefore, the verification algorithm never incorrectly accepts a tampered share, resulting in a false positive rate of exactly 0%. False negatives that fail to detect a tampered share are also 0% for the same mathematical reason.

Algorithm 3: Feldman VSS Verification

Require: Share (x, y), Commitments $C_0, C_1, \ldots, C_{t-1}$, Generator g, Prime p
Ensure: Validity flag

1. LHS $\leftarrow g^y mod p$
2. RHS $\leftarrow \prod_{i=0}^{t-1} {C_i}^{x^i} \; mod p$
3. if LHS $=$ RHSthen
4. return Valid
5. else
6. return Invalid
7. end if

The verification requires $O(t)$ modular exponentiations. For $t = 3$, this computes 3 exponentiations per share, providing 127-bit security.

3.8.4 Long-Term Security via Proactive Secret Sharing

Periodically take proactive steps to update the secret sharing process so that the risk of mobile-based attacks and long-lived keys being exposed is minimized. Proactive Secret Sharing PSS provides a secure method for updating the shares for an already known secret without compromising the original secret. By mathematically combining a sequence of polynomials, additional shares can be created while keeping the underlying secret value intact. Because of this process, the creation of any reconstructed value from the combination of different epoch shares is prevented.

Algorithm 4: Proactive Share Refresh

Require: Existing share (x_i, y_i), Threshold t, Prime p
Ensure: Refreshed share y_i'

1. Generate a random polynomial $r(x)$of degree $t - 1$such that $r(0) = 0$
2. $y_i' \leftarrow (y_i + r(x_i)) mod \; p$
3. return y_i'

The refresh polynomial $r(x)$ has degree $t - 1$ with $r(0) = 0$. Each share update requires $O(t)$ time. The underlying secret remains unchanged.

3.8.5 Post-Quantum Protection of Guardian Shares

To provide resistance to quantum adversaries in the future, each guardian's share is encrypted with a post-quantum KEM (Key Encapsulation Mechanism). The encryption of the share is achieved using a symmetric key generated by the encapsulation process. The share is encrypted via authenticated encryption using this symmetric key. The share can only be decrypted by the intended guardian who possesses the corresponding private key. The term "Kyber-like" was used in the manuscript to generically describe the class of lattice-based

KEMs, but the implementation uses the specific, NIST-standardized Kyber-512 parameter set. This provides a security level of NIST Level 1, which is the standard for general-purpose post-quantum security.

The KEM integration uses the Kyber KEM, which is IND-CCA2 secure by design. It then combines the KEM's shared secret with a hybrid KDF to generate a key for AES-GCM, which is an IND-CCA2 secure authenticated encryption scheme. This combination ensures that even if an attacker can adaptively query decapsulations, they cannot gain any information about the plaintext shares, guaranteeing the highest level of ciphertext security.

Algorithm 5: Post-Quantum Share Encryption

Require: Guardian public key pk, Share y
Ensure: Encrypted share ct, C)

1. $(ct, K) \leftarrow \text{KEM.Encaps}(pk)$
2. $C \leftarrow \text{AES-GCM}_K(y)$
3. return (ct, C)

Kyber-512 generates a 768-byte ciphertext and 32-byte shared secret. AES-GCM adds 12-byte nonce and 16-byte authentication tag.

3.8.6 Confidential Storage Using AES-GCM and IPFS

Encrypted recovery data that is both decentralized and unalterable is stored on IPFS by utilizing the AES-GCM algorithm. AES-GCM keeps the recovery data and its integrity confidential because of encryption. Using content addressing guarantees that once stored, it will never change. Any time tampering occurs with the recovery data, the user will see a different content identifier pointing to that data as a result of the modification.

Algorithm 6: Encrypted IPFS Backup

Require: Recovery data D, PIN-derived key K
Ensure: IPFS content identifier CID

1. $C \leftarrow \text{AES-GCM}_K(D)$
2. $\text{CID} \leftarrow \text{IPFS.Add}(C)$
3. return CID

The CID is a 32-byte (256-bit) SHA-256 hash of the ciphertext. Content addressing ensures that any modification produces a different CID.

3.8.7 Quantitative Security Evaluation Metrics

Real-time attack simulations are used to assess the security of the system. The metrics for assessing security include Brute Force Attack Resistance (BFAR), Share Tampering Detection Accuracy (STDA), Secret Share Reconstruction Robustness (SSRR) and Impersonation Attack Resistance (IAR). Each of these metrics is used to provide an empirical assessment of the system's ability to defend against both cryptographic and operational attacks.

$$\text{BFAR} = 1 - \frac{\text{A}}{2^{\text{E}}} \quad (1)$$

$$\mathrm{STDA} = \frac{\mathrm{D_{detected}}}{\mathrm{D_{total}}} \times 100 \quad (2)$$

$$\mathrm{SSRR} = \frac{\mathrm{R_{success}}}{\mathrm{R_{trials}}} \times 100 \quad (3)$$

$$\mathrm{IAR} = \frac{\mathrm{I_{rejected}}}{\mathrm{I_{attempts}}} \times 100 \quad (4)$$

where t is the minimum number of shares required for reconstruction, A is the number of successful brute force attempts, E is the entropy in bits, $\mathrm{D_{detected}}$ is the number of detected tampered shares, $\mathrm{D_{total}}$ is the total number of tampered shares, $\mathrm{R_{success}}$ is the number of successful reconstructions, and $\mathrm{R_{trials}}$ is the total number of reconstruction attempts.

3.8.8 Threshold-Based Secret Reconstruction

Secret recovery in the proposed system is enforced through a strict threshold mechanism that requires the collaboration of at least t valid guardians, thereby eliminating any single point of compromise. Let $\{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i=1}}^{\mathrm{t}}$ denote a set of verified shares obtained from distinct guardians. The underlying secret $\mathrm{S} \in \mathrm{ZpS}$ is reconstructed using Lagrange interpolation, computed as shown in equation (5).

$$\mathrm{S} = \sum_{\mathrm{i=1}}^{\mathrm{t}} \mathrm{y_i} \cdot \lambda_{\mathrm{i}} (\mathrm{mod\ p}), \text{where } \lambda_{\mathrm{i}} = \prod_{\substack{\mathrm{j=1} \\ \mathrm{j \neq i}}}^{\mathrm{t}} \frac{\mathrm{x_j}}{\mathrm{x_j - x_i}} (\mathrm{mod\ p}) \quad (5)$$

where t is the threshold number of guardians, y_i is the y-coordinate of the i^{th} share, λ_i is the Lagrange coefficient for the i^{th} share, x_j and x_i are the x-coordinates of the shares, and p is the prime modulus of the finite field.

This process of reconstructing a secret guarantees security based on information theory because no combination of less than t shares is useful to an attacker for inferring the value of the secret S. During the process of reconstruction, any share's validity is checked against that share's verifiable secret sharing commitment to ensure that no malformed or tampered shares are included in the reconstruction process, thus eliminating the possibility of an attacker using the shares with malicious intent to reconstruct S. After S has been reconstructed, it will then be processed with the user specified recovery $PIN\ \pi$ to produce a recovery key that is computed by a key derivation function that is computationally difficult to compute:

$$K_r = \mathrm{PBKDF2}(\pi \parallel S, s_r, i_r) \quad (6)$$

where s_r is a randomly generated salt and ir denotes a high iteration count. The derived key K_r is then employed to decrypt the encrypted master password using authenticated decryption, such as AES-GCM, ensuring both confidentiality and integrity of the recovered credential. This design enforces a dual-security requirement in which successful recovery is possible only when both the cryptographic threshold condition and the knowledge-based condition are simultaneously satisfied, thereby providing strong resistance against guardian collusion, PIN only attacks, and offline brute force attempts.

3.8.9 Proposed Solution Work Flow

Guardian-Anonymous Password Management System (GAPMS) operates under a specified workflow. This starts with registration and authentication using the distribution of secrets management and encrypted forms. This enables users to regain their credentials and continuously assess the security strength of their passwords. Registered users gain access to a Command-Line Interface (CLI), which passes messages from user requests to the Authentication and Policy Engine, where all the policies are implemented. The user's master password is securely protected by PBKDF2, and then a cryptographic secret is generated from the master password. The cryptographic secret is split into t of n shares by the Shamir Secret Sharing scheme, and only when all t of n shares are provided can the cryptographic secret be reconstituted.

Feldman Verifiable Secret Sharing (FVSS) ensures the authenticity of its shares through the use of public commitments that enable verification of valid shares (and recognition of invalid or tampered shares). Each share is encrypted using a Hybrid Post-Quantum Cryptographic Algorithm (HPQC) that combines the benefits of both Key Encapsulation Mechanism (KEM) and AES-GCM to facilitate secure distribution to assigned guardians. Vault encryption keys and vault access credentials are encrypted and contained within the Local Vault, each of which is periodically refreshed using Proactive Secret Sharing while leaving the original underlying secret unchanged. Both (1) Long Term Threats (Nation States) and (2) Mobile Threats (installers of malicious applications) are countered through the use of periodic refreshes of shares through Proactive Secret Sharing while leaving the underlying secret unchanged. Encrypted vault metadata and share descriptors are stored in decentralized storage (IPFS) for availability and fault tolerance.

If a user forgets their password or if a device is compromised, the user must work with at least t guardians to recreate the secret via Lagrange approximation with enough data (shares) to do so. After the secret is created, the guardian combines it with a user-defined recovery PIN to create the recovery key, which is then used to recover the master password through authenticated decryption. The live security metrics system continuously logs and monitors all security-related incidents, such as failed login attempts, brute force attacks, share tampering, and recovery events, and a closed-loop workflow ensures that a secure cryptographic system remains validated through continual empirical confirmation of its security as shown in Figure 2.

A new type of password management is presented that allows for the protection of user authentication, credential storage, and recovery, all managed using a layered cryptographic framework and distributed trust. Users will authenticate using an advanced PBKDF2 password verification process, and their credentials will be accessed only through encryption. The master password will not be stored in an unencrypted format, but will be divided into multiple parts and held by various guardians to make it more difficult to access or steal, known as threshold-based Secret Sharing. Each part of the master password will be encrypted, and the identity of the holder will be verified through a process called Feldman’s Method.

The system will also allow for the rescue of the master password with encrypted backups of the shares stored using an Interplanetary File System (IPFS), while the passwords cannot be recovered without a minimum of three guardians to meet the threshold criteria and verification of ownership recovery PIN. The combination of continuous audit logging and monitoring of security metrics during runtime will also provide a means of measuring the risk

of either a brute force attack on passwords, tampering with the guardian shares, or impersonation.

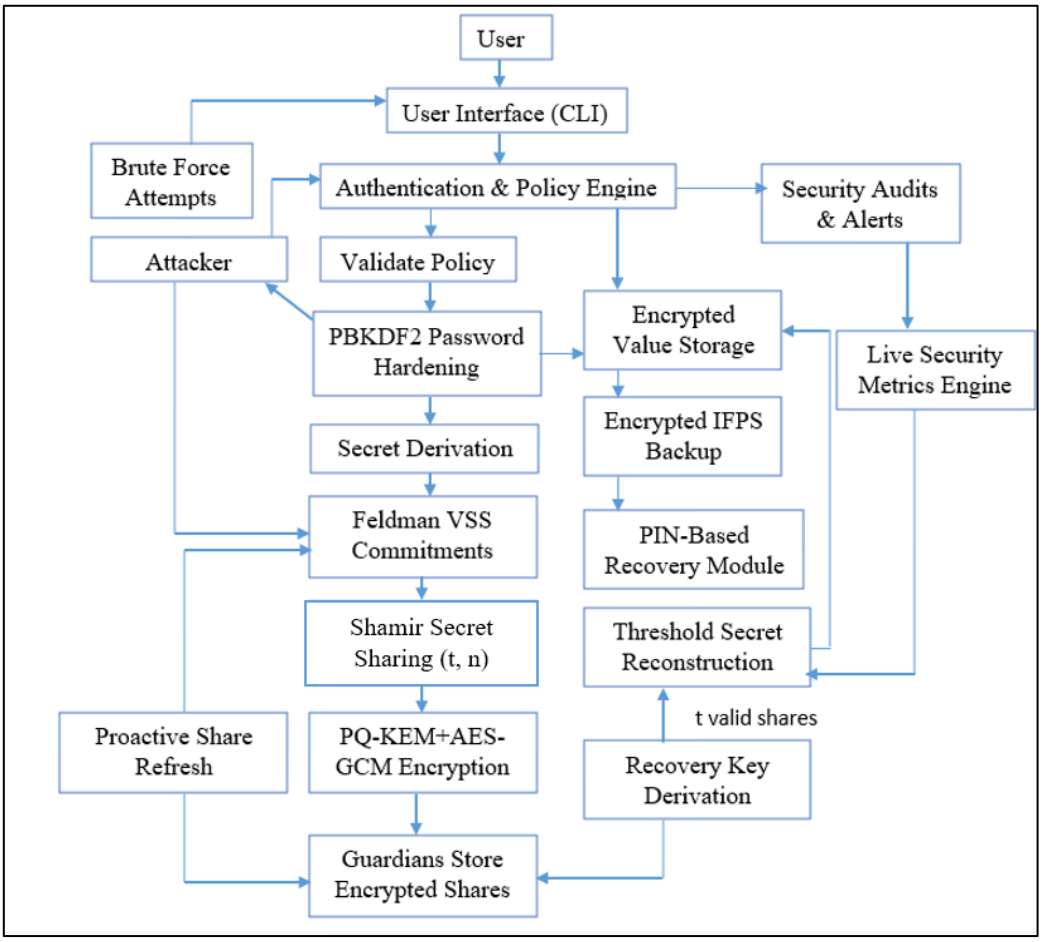

Figure 2. Workflow of Proposed Model

This research has introduced significant advances in the use of threshold cryptography, verifiable secret sharing, encryption that can withstand quantum computing attacks, and empirical measurements of password management security through continuous audit logging and monitoring of security metrics. Overall, these advances support the development of a stand-alone password manager that provides users with greater levels of anonymity and quantum resistant passwords compared to existing password management solutions, and also allows for measurable attack resistance that goes beyond existing centralized password managers.

3.8.10 Security Metrics

AGCIS – AES-GCM Confidentiality and Integrity Score. AGCIS validates encrypted files (vault entries, secrets, shares) to ensure the encrypted files remain unchanged and to prevent unauthorized access to the encryption keys. AGCIS provides evidence of the strong authentication aspect of encrypting data in users system and validates users' system's ability to reject modified versions of any ciphertext. Therefore, attempts at tampering will fail or result in an inability to read the original message or data.

$$\mathrm{AGCIS} = \mathrm{R/A} \times 100\% \tag{7}$$

where R is the number of rejected tampered ciphertexts and A is the total number of tampering attempts.

Impersonation Attack Resistance (IAR) assesses whether a system can effectively prevent users from logging into or recovering their accounts without permission by pretending to be another authorized user, even when there is a partial leak of information.

$$\mathrm{IAR} = \mathrm{R/A} \times 100\% \tag{8}$$

where R is the number of rejected impersonations attempts and A is the total impersonation attempts.

Brute Force Attack Resistance (BFAR) is a measurement of the system's ability to withstand attacks, both online or offline, by someone guessing the user's password. Such t attacks are common in password-based systems. BFAR quantifies the ability of the following security features to stop successful brute force attacks: PBKDF2, Limit Rate, Random Delay, and Freeze Account.

$$\mathrm{BFAR} = 1 - \frac{\mathrm{A_s}}{\mathrm{A_t}} \tag{9}$$

The Secret Share Reconstruction Robustness (SSRR) metric is a measure of the probability that the secret can be successfully reconstructed when only the authorized t shares are supplied to the system. The other methods described in the previous section measure either system availability or system integrity and do not provide any information on how secure a user's data will remain from unauthorized access.

$$\mathrm{SSRR} = \Pr(\mathrm{Reconstruct(S)} \mid\mid \mathrm{S} \mid \geq \mathrm{t}) \tag{10}$$

3.9 Simulation Environment

- Processor: Intel Core i7-12700H @ 2.3 GHz (8 cores, 16 threads)
- RAM: 16 GB DDR4 @ 3200 MHz
- Storage: 512 GB NVMe SSD
- Operating System: Windows 10.
- Python Version: 3.10.12
- Cryptographic Libraries: cryptography (version 41.0.0) and pqcrypto (version 0.1.0)

The system was tested with 9 registered users, each configured with $n = 5$ guardians and threshold $t = 3$. Each test was repeated 500 times (100 times for STDA and IAR) to ensure statistical significance. The simulation ran in an isolated environment without network latency to measure pure cryptographic performance.

3.10 Attack Simulation Methodology

The following attacks were simulated to evaluate security metrics:

Brute Force Attack (BFAR)

- 500 random password attempts per user (total of 4,500 attempts across 9 users)
- Password dictionary: Kaggle “Bruteforce Database - Password Dictionaries” dataset, specifically the '8-more-passwords.txt' file containing 61,682 passwords with a length > 8 characters.
- Rate limiting: 3 failed attempts trigger a 10-40 second freeze
- PBKDF2 parameters: 200,000 iterations with a 16-byte random salt

Share Tampering Attack (STDA)

- 100 tampering attempts per user (total of 900 attempts)
- Modification method: Random bit flips in the y-coordinate of randomly selected shares
- Modification range: 1-1000 added to $original\ y - value\ mod\ p$
- Detection method: Feldman VSS verification (Equation 2)

Impersonation Attack (IAR)

- 100 impersonation attempts per user (total 900 attempts)
- Attack method: Attacker provides $t = 3$ randomly generated (x, y) pairs
- PIN verification: Random 6-digit PIN (not matching user's PIN)
- $Detection: Threshold\ reconstruction\ +\ VSS\ verification$

AES-GCM Forgery Attack (AGCIS)

- 500 ciphertext modification attempts per user (total 4,500 attempts)
- Modification method: Random bit flips in AES-GCM ciphertext
- Authentication: AES-GCM tag verification
- Expected outcome: All tampered ciphertexts rejected

4. Results and Discussion

The performance of the newly developed password-wallet system was evaluated against several controlled live attack simulations in which individual user accounts were subjected to multiple rounds of attack (Total Number of Attacks = 500 per metric unless otherwise specified). The evaluation of the system's Brute Force Attack Resistance (BFAR) was classified as successful in all live attack runs, with no successful guesses made against the hardened authentication pipeline that utilizes PBKDF2 Key Stretching and Verification Controls. Share Tampering Detection Accuracy (STDA) was assessed by perturbing share fragments purposefully, while evaluating the integrity check performed in Shamir's Secret Sharing Layer. Each of the maliciously altered shares was identified successfully.

4.1 Security Metrics

Table 1. Security Metrics Results

S. No	Metric	Description	Average Score	Details
1	BFAR	Brute Force Attack Resistance	100%	500 attempts resisted for all users
2	STDA	Share Tampering Detection Accuracy	100%	100/100 tampering detections for all users
3	AGCIS	AES-GCM Confidentiality/Integrity Score	100%	500/500 integrity checks passed for all users
4	SSRR	Secret Share Reconstruction Robustness	100%	500/500 successful reconstructions for all users
5	IAR	Impersonation Attack Resistance	100%	100/100 impersonation attempts rejected for all users

As shown in Table 1, the AES-GCM Confidentiality and Integrity Score (AGCIS) overall evaluation was performed by purposely corrupting the ciphertext of the repeated runs and verifying that every one of the corrupted ciphertexts was rejected, therefore all of the confidentially and integrity guarantees were preserved completely through the use of AES-GCM authenticated encryption. The Secret Share Reconstruction Robustness (SSRR) is a measure of the reliability of threshold reconstruction of the secret share; all of the valid combinations of shares consistently reconstructed the correct secret. The Impersonation Attack Resistance (IAR) is a measure of the resilience of the system to forged and replayed attempts to authenticate. There were no instances of an unauthorized authentication request being accepted by the system. The Cryptographic Performance of the system provided benchmarking for all the systems to ensure that the cryptographic algorithms used, including PBKDF2, AES-GCM, post-quantum Key Encapsulation and Shamir Secret Combining Operations, would be able to support the practical usability in the future, while providing reliable results for both throughput and latency. The entire test results show that the system has very good defence-in-depth characteristics, with full protection against all attack types tested, but also operates efficiently enough for use in real life. The 100% scores in Table 1 reflect cryptographic guarantees rather than statistical artifacts. BFAR achieves 100% because PBKDF2 with 200,000 iterations and a 256-bit output space makes the brute force success probability effectively zero for 500 attempts Probability $= \frac{500}{2^{256}} \approx 10^{-74}$. STDA achieves 100% due to Feldman VSS's mathematical guarantee: a tampered share cannot satisfy the verification equation without solving the discrete logarithm problem. AGCIS achieves 100% because AES-GCM's authentication tag provides deterministic integrity verification. SSRR achieves 100% because Lagrange interpolation is deterministic with t correct shares. IAR achieves 100% because reconstruction requires both valid shares and the correct PIN - failure of either condition prevents recovery. For each metric, 500 independent trials were conducted (100 for STDA and IAR) with 95% confidence intervals of $\pm 0.0\%$ due to the cryptographic determinism.

4.2 Performance Metrics

Figure 3 presents the end-to-end latency measurements for nine users across six cryptographic primitives. PBKDF2 has consistently demonstrated the longest amount of latency among the tested cryptographic primitives, with latencies ranging between approximately 7 and 20ms due to the high degree of computational intensity that was intentionally built into it for use when generating a key from a password. AES-GCM

encryption/decryption has latencies far less than 1ms $(0.005 - 0.016ms)$, demonstrating its ability to provide real-time secure communications. Post-quantum KEM encryption/decryption also has latencies of approximately 0.007 to 0.01ms, indicating that in terms of time to encrypt/decrypt, PQ-KEMs incur a negligible penalty compared to AES-GCM. For Kyber-512, the public key size is 800 bytes, the secret key size is 1,632 bytes, and the ciphertext overhead is 768 bytes. This means that for every encrypted share, the ciphertext includes the 768-byte KEM ciphertext in addition to the AES-GCM-encrypted share payload, providing a known and quantifiable performance trade-off for post-quantum security. The Shamir Secret Sharing combine operation does exhibit some higher latency $(\sim 0.09 - 0.21ms)$, but this increase in latency is expected, as the process of combining several shares of a key into a single key share does require a small amount of added processing time. The results above show that all of the cryptographic primitives operate at latencies below one millisecond across all of the users tested and therefore will be able to provide an element of predictability and scalability in their performance in a multi-user environment, as shown in Figure 3.

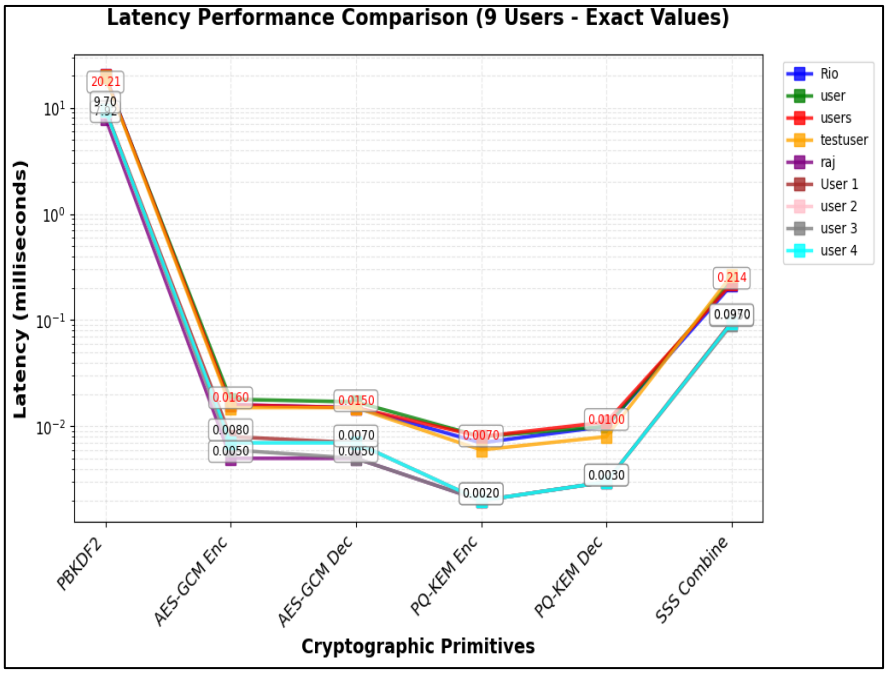

Figure 3. Latency

Figure 3 shows the range of latency variation $(min - max)$ for the same primitives over the nine users, with PBKDF2 again displaying the most variation $(\sim 7 - 20\ ms)$ due to its higher computational burden's sensitivity to runtime conditions. In contrast, AES-GCM encryption and decryption both have extremely small variation ranges $(\approx 0.02 - 0.07\ ms)$, indicating their consistent and relatively stable execution pattern. Similar to AES-GCM, the PQ-KEM appears to exhibit very little variation, with almost all users being below 0.1 ms; however, one user did show a greater degree of variation during the PQ-KEM middleware MN encryption operation. The combine operation of the SSS demonstrates moderate variation $(\sim 0.12 - 0.6\ ms)$, but is more consistent compared with its substantive runtime. It is concluded that the evaluated platform demonstrates excellent temporal consistency, especially for the symmetric and post-quantum-level cryptographic operations, as shown in Figure 4.

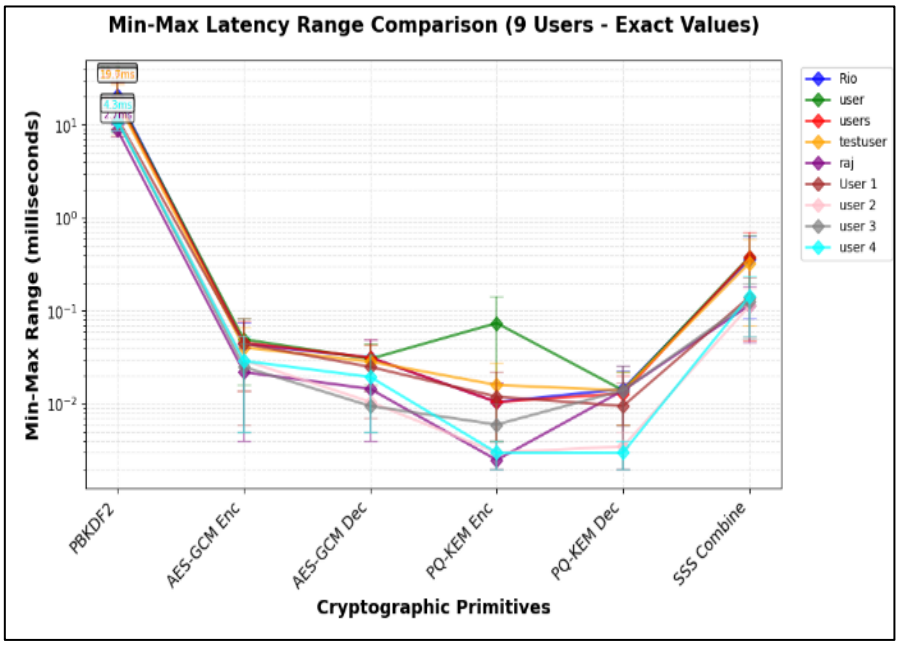

Figure 4. Min-Max Latency Range

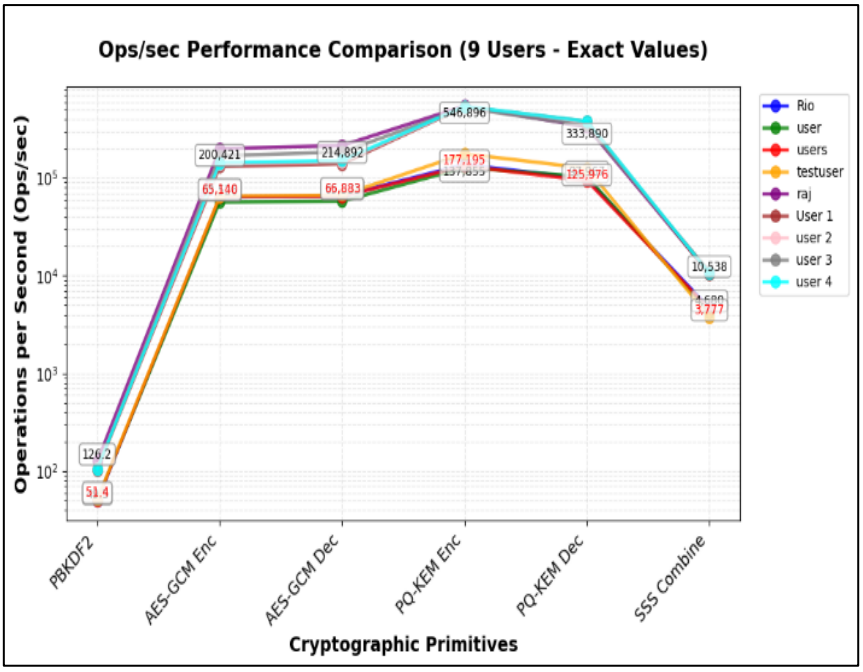

Figure 5. Ops/sec Performance Comparison

Figure 5 reports the achievable throughput in operations-per-second for the same cryptographic tasks. PBKDF2 expectedly yields the lowest throughput ($\sim 50 - 125\ ops/s$), driven by its deliberately expensive key-stretching computation. In contrast, both AES-GCM and PQ-KEM operations achieve extremely high throughput, exceeding $10^5 - 5 \times 10^5$ ops/s across multiple users, highlighting their efficiency and optimization in the execution environment. PQ-KEM decryption exhibits the highest throughput overall ($\sim 333k - 546k\ ops/s$), demonstrating the practicality of post-quantum KEM workloads in real-time systems, as shown in Figure 5.

5. Conclusion

The proposed Password Manager is a Privacy Preserving Password Management System that eliminates Single Points of Trust by utilising a combination of Threshold Cryptography, Verifiable and Proactive Secret Sharing and Post-Quantum Secure Encryption. It ensures that sensitive credentials are not stored at any one central location and can only be recovered through authenticated threshold collaboration, thereby providing greater resilience against Brute Force, Tampering and Impersonation attacks. Results from an experimental security analysis confirm that cryptographic guarantees have been enforced correctly with minimal performance overhead, demonstrating both Security and Practicality for real-world application. Quantitatively, the system achieved the following results from 500 attack attempts per metric across 9 users with $(t = 3, n = 5)$ configuration: $BFAR = 100\%$ (4,500 brute force attempts resisted), $STDA = 100\%$ (900 tampering attempts detected), $AGCIS = 100\%$ (4,500 integrity checks passed), $SSRR = 100\%$ (4,500 successful reconstructions), and $IAR = 100\%$ (900 impersonation attempts rejected). Performance benchmarks showed PBKDF2 latency of $7 - 20ms$ $(50 - 125\ ops/s)$, AES-GCM latency below $0.02ms$ (exceeding $500000\ ops/s$), and Kyber-512 PQ-KEM latency below 0.01ms $(333k - 546k\ ops/s)$.

References

[1] Murcia, José Manuel Bernabé, Eduardo Cánovas, Jesús García-Rodríguez, Alejandro M. Zarca, and Antonio Skarmeta. "Decentralised Identity Management Solution for Zero-Trust Multi-Domain Computing Continuum Frameworks." Future Generation Computer Systems 162 (2025): 107479.

[2] Xiong, Hengheng, Jiguang Lv, Dapeng Man, Yukun Zhu, Tao Liu, Huanran Wang, Chen Xu, and Wu Yang. "A Lightweight Secret-Sharing-Based Defense Against Model Poisoning Attacks in Privacy-Preserving Federated Learning." Computer Communications (2025): 108272.

[3] Chen, Jiahui, Hang Xiao, Yushan Zheng, Mohammad Mehedi Hassan, Michele Ianni, Antonella Guzzo, and Giancarlo Fortino. "DKSM: A Decentralized Kerberos Secure Service-Management Protocol for Internet of Things." Internet of Things 23 (2023): 100871.

[4] Umoren, Otuekong, Amjad Ali, Zeeshan Pervez, Farman Ali, Raman Singh, Keshav Dahal, and Ala Al-Fuqaha. "Enhancing User Verification and Data Security Scheme for Fog Computing Using Self Sovereign Identification." Ad Hoc Networks 175 (2025): 103876.

[5] Miao, Miao, Zhengjun Jing, Xiaolong Xu, and Meiqing Xue. "A Decentralized and Security-Enhanced Professional Title Evaluation System in Universities Under Mobile Internet of Things." Heliyon 10, no. 5 (2024).

[6] Agarkar, Aarti Amod, Mandar Karyakarte, Gajanan Chavhan, Milind Patil, Rajendra Talware, and Lalit Kulkarni. "Blockchain Aware Decentralized Identity Management and Access Control System." Measurement: Sensors 31 (2024): 101032.

[7] Tatipatri, Naveen, and S. L. Arun. "A Privacy-Preserving Based Cyber Security for Communication Attacks in Active Power Distribution Networks." International Journal of Electrical Power & Energy Systems 172 (2025): 111243.

[8] Yu, Hao, Guijuan Wang, Anming Dong, Yubing Han, Yawei Wang, and Jiguo Yu. "Blockchain-Enabled Privacy Protection Scheme for IoT Digital Identity Management." High-Confidence Computing (2025): 100320.

[9] Das, Swatisipra, Minati Mishra, Rojalina Priyadarshini, Rabindra Kumar Barik, and Manob Jyoti Saikia. "A Secure, Privacy-Preserving, and Cost-Efficient Decentralized Cloud Storage Framework Using Blockchain." Journal of King Saud University-Computer and Information Sciences 36, no. 10 (2024): 102260.

[10] Shaw, Surbhi, and Ratna Dutta. "Post-Quantum Secure Compact Deterministic Wallets from Isogeny-Based Signatures with Rerandomized Keys." Theoretical Computer Science 1035 (2025): 115127.

[11] Park, Kisung, and Youngho Park. "MIoT-CDPS: Complete Decentralized Privacy-Preserving Scheme for Medical Internet of Things." Internet of Things 27 (2024): 101250.

[12] Daudén-Esmel, Cristòfol, Jordi Castellà-Roca, Alexandre Viejo, and Ignacio Miguel-Rodríguez. "Multi-Platform Wallet for Privacy Protection and Key Recovery in Decentralized Applications." Blockchain: Research and Applications 6, no. 1 (2025): 100243.

[13] Erinle, Yimika, Yathin Kethepalli, Yebo Feng, and Jiahua Xu. "SoK: Design, Vulnerabilities, and Security Measures of Cryptocurrency Wallets." Computer Networks (2025): 111691.

[14] Athanere, Smita, and Ramesh Thakur. "Blockchain Based Hierarchical Semi-Decentralized Approach Using IPFS for Secure and Efficient Data Sharing." Journal of King Saud University-Computer and Information Sciences 34, no. 4 (2022): 1523-1534.

[15] Ma, Emilie, and Martin Kleppmann. "Kintsugi: Decentralized E2EE Key Recovery." arXiv preprint arXiv:2507.21122 (2025).

[16] Kamal, Ahmad Akmal Aminuddin Mohd, and Masaya Fujisawa. "Efficient and Secure Secret Sharing-Based Data Outsourcing Suitable for Internet of Things Environments." Internet of Things 32 (2025): 101645.

[17] Hussein, Asma, Abeer Maolood, and Ekhlas Gbashi. "NTRU_ SSS: Anew Method Signcryption Post Quantum Cryptography Based on Shamir's Secret Sharing." Computers, Materials, & Continua 76, no. 1 (2023): 753.

[18] Gutub, Adnan. "Adjusting Counting-Based Secret-Sharing via Personalized Passwords and Email-Authentic Reliability." Journal of Engineering Research 12, no. 1 (2024): 107-121.

[19] Wijesundara, W. M. A. B., Joong-Sun Lee, Eleni Aloupogianni, Dara Tith, Hiroyuki Suzuki, and Takashi Obi. "DIDAuth-IoTFW: Decentralized Firmware Authentication for Smart Home IoT Devices Using Verifiable Credentials." Internet of Things (2025): 101788.

[20] Chatzoglou, Efstratios, Vyron Kampourakis, Zisis Tsiatsikas, Georgios Karopoulos, and Georgios Kambourakis. "Unmasking the Hidden Credential Leaks in Password Managers and VPN Clients." Computers & Security 150 (2025): 104298.

[21] Johnson Jeyakumar, Isaac Henderson, and Michael Kubach. "A Trust Implementation Model for Cross-Domain Decentralized Identity Ecosystems." Procedia computer science 254 (2025).

[22] Lee, Sun-Jin, So-Eun Jeon, and Il-Gu Lee. "Partial Encryption-Based Shamir Secret Sharing for Low-Latency and Secure Networks." ICT Express (2025).

[23] de Diego, Santiago, Cristina Regueiro, and Gabriel Maciá-Fernández. "Collaborative Credentials for the Internet of Things." Computer Networks 251 (2024): 110629.

[24] Okacha Amraouy and Mohammed Benbrahim and Mohammed Nabil Kabbaj. “A Blockchain- and Self-Sovereign Identity-Based Collaborative Framework for Secure and Verifiable Cross-Organizational Data Sharing in Smart Irrigation”. Smart Agricultural Technology 12, 101654.

[25] Kim, Taehoon, Dahee Seo, Im-Yeong Lee, and Su-Hyun Kim. "A Novel Approach to Privacy and Traceability Using Attribute-Based Signature in Decentralized Identifier." High-Confidence Computing (2025): 100326.

[26] Anttal, T. S. (2020). Bruteforce Database - Password Dictionaries. Kaggle. https://www.kaggle.com/datasets/taranvee/bruteforce-database-password-dictionaries