2FAS Pass White Paper

Version: 0.1 (Jun 27, 2025)

Document Status

This is an early draft (v0.1) of the 2FAS Pass White Paper.
As our product evolves, this document will be updated to reflect new features, security improvements, and community feedback.

We welcome technical reviews and suggestions at:
https://github.com/twofas/

Changelog

➔ v0.1 (Jun 27, 2025): Initial draft release

Abstract

2FAS Pass is a local-first password manager designed to give users full control over their data. Unlike traditional solutions that depend on central servers, 2FAS Pass stores encrypted Vaults locally on the device by default.

For users who want synchronization, the app provides seamless, end-to-end encrypted sync using trusted platforms like iCloud (for iOS) or Google Drive (for Android). WebDAV integration enables cross-platform synchronization and secure storage with NAS systems like QNAP, Synology, or self-hosted setups.

The security model is zero-knowledge by design: all encryption happens on the user's device, and only the user has access to data. Vault keys are derived from a combination of the user's 15 Secret Words derived from randomly generated 160-bit entropy and a user-defined Master Password, ensuring strong encryption and fully local recovery.

1. Introduction

Password managers have become fundamental parts of digital security, and most existing solutions are built around centralized infrastructure. This model prioritizes ease of use by storing all data on providers' servers. However, this convenience comes at the cost of several structural limitations - particularly around availability, privacy, and in some cases, security.

Centralized Solutions

If access to a provider's service is interrupted - due to network outages, account suspension, geographic restrictions, or company shutdown - users may suddenly lose the ability to access their stored credentials. In high-dependency scenarios, this can result in complete lockout from essential services.

Privacy is another consideration. Centralized solutions typically require account creation and may collect personal data such as usernames, email addresses, IP addresses, or device identifiers. Additionally, they may gather metadata including access timestamps, synchronization frequency, or usage patterns. Even when encrypted content remains private, this metadata can reveal sensitive behavioral insights if misused or leaked.

Our Approach

By design, 2FAS Pass adopts a local-first architecture, eliminating centralized data aggregation and single points of failure. All encryption occurs on the user's device, and Vaults are stored locally by default. The app operates without mandatory accounts, respects user privacy, and treats synchronization as an opt-in feature.

When users choose to sync, they retain full control over storage location and infrastructure - whether using iCloud, Google Drive, WebDAV, or a self-hosted server. All Vaults remain end-to-end encrypted and inaccessible to intermediaries.

2. Design Principles

2FAS Pass was built with the following core principles:

Local-First Architecture

All encryption and decryption happens directly on the user's device. Vaults are stored locally by default, and with synchronization no data leaves the device in unencrypted form. This guarantees that the user remains the only one with access to their data, even when using third-party services for synchronization.

Zero-Knowledge by Design

No one but the user can access the contents of their Vault - and no one else even knows where the Vault is stored. There is no central server or user account. Because of this design, it is extremely hard to detect, monitor, or associate users with any stored data.

Optional and User-Controlled Sync

Users can choose whether to sync, where to store their Vault, and how it's done. Supported options include native cloud providers (iCloud for iOS, Google Drive for Android), or WebDAV - a standard protocol supported by many platforms. All synced Vaults remain fully encrypted, and 2FAS itself never has access to the content.

No Vendor Lock-In

Users retain full access to their Vault data regardless of the paid or free plan they have. As long as the user retains their Vault file, Secret Words, and Master Password, Vaults can be restored on any compatible device - without internet access.

Decentralized by Default

Because there is no central server and each user chooses where their Vault is stored, there is no single point of failure or attack. This decentralized model reduces systemic risk and removes the possibility of mass data breaches.

3. System Architecture

Overview

The core of the system is the Vault architecture. Each Vault holds all user data, including credentials, notes, metadata, and encryption parameters.

Vault Management

The app supports creating and managing multiple Vaults (this feature might not be available in early versions). Each Vault is:

- ➔ Identified by a unique UUID (VID)
- ➔ Encrypted with its own set of keys
- ➔ Completely independent from other Vaults
- ➔ Stored and synced as a separate encrypted file

Key Storage

2FAS Pass uses secure storage mechanisms provided by the operating system:

- ➔ For iOS - Keychain Services
- ➔ For Android - Android Keystore

These secure storage mechanisms keep Vault keys protected on the device. Users unlock with their Master Password by default, with optional biometric unlock available for convenience.

4. Cryptographic Design

The security model in 2FAS Pass is based on strong client-side encryption, deterministic key derivation, and per-Vault key separation.

4.1 Dual-Entropy System

Each Vault is encrypted using keys derived from two sources of entropy:

1. **Secret Words**: A randomly generated 15-word phrase (160 bits of entropy)
2. **Master Password**: A user-defined password (minimum 9 characters)

4.2 Secret Words Generation

When creating a new Vault:

1. Generate 160 bits of entropy **E** using the OS cryptographically secure random number generator (CSPRNG)
2. Compute Seed = SHA-256(**E**)
3. Append the first 5 bits of Seed as checksum to **E**, creating 165 bits
4. Split into fifteen 11-bit indices and map to the standard BIP-39 word list → 15 Secret Words

5. Concatenate the last 4 words and compute Salt=SHA-256 (concat). This approach ensures deterministic salt generation (always producing the same salt for given Secret Words) while maintaining high entropy for the key derivation process.

4.3 Master Key Derivation

The Master Key is derived using Argon2id:

- ➔ **Input:** Seed || NormalizedPassword (password normalized using Unicode NFKD)
- ➔ **Salt:** Derived from the Secret Words as described above
- ➔ **KDF:** Argon2id with adaptive parameters (time, memory, parallelism)
- ➔ **Output:** Master Key (256 bits)

The Argon2id parameters are stored inside the Vault to ensure consistent behavior during unlock and recovery.

4.4 Per-Vault Key Derivation

For each Vault with UUID VID[i], three keys are derived using HMAC-SHA256:

1. **tKey_i** = HMAC_SHA256(Master Key, VID[i] || "/tKey") → Tier: Secret
2. **sKey_i** = HMAC_SHA256(Master Key, VID[i] || "/sKey") → Tier: Highly Secret & Top Secret
3. **eKey_i** = HMAC_SHA256(Master Key, VID[i] || "/eKey") → External sync encryption

These derived keys are wrapped with the device-specific App Key and stored in secure storage. After derivation, the Master Key and Salt are wiped from memory.

4.5 Encryption Method

All data is encrypted using **AES-256-GCM,** which provides both confidentiality and authenticity.

4.6 Tiered Security Model

2FAS Pass implements a flexible security model where each stored item can be assigned one of three security tiers:

Security Tier	Description	Use Case
Secret	All fields encrypted with tKey_i	Standard protection for everyday accounts, with full auto-fill and extension support for maximum convenience.
Highly Secret	Most fields encrypted with tKey_i, but sensitive fields like passwords use sKey_i	When passwords need extra protection, requiring the app to be unlocked and authenticated to access them.
Top Secret	All fields encrypted with sKey_i	High-value accounts, not available in auto-fill and extensions, ensuring that even with an unlocked device, the most sensitive credentials cannot be accessed without explicit user action within the app.

This model allows users to balance between security and usability based on the sensitivity of each item.

5. Synchronization

Available Options

2FAS Pass supports three synchronization methods:

- ➔ iCloud (iOS only): Uses CloudKit for seamless integration
- ➔ Google Drive (Android only): Uses Google Drive
- ➔ WebDAV: Universal protocol for custom storage solutions

Additional sync options are planned for future releases.

Sync Security

When synchronization is enabled:

1. The entire Vault file is encrypted with eKey_i (as described in 4.4) before upload
2. Only encrypted data is transmitted to the sync provider
3. Conflict resolution happens locally on the device using a merge strategy

Conflict Resolution

The app uses a deterministic merge strategy based on modification timestamps:

- ➔ Each entry tracks its last modification time
- ➔ During sync, the most recent change always wins
- ➔ To prevent sync conflicts due to incorrect device time settings, the app calculates and saves a time delta when internet is available. This delta is then applied during sync to correct for any device clock drift

Offline Operation

The app is fully functional without internet connection. Synchronization occurs opportunistically when connectivity is available, with all changes queued locally until successful sync.

6. Recovery Process

Recovery Requirements

To recover a Vault, users must have:

1. The encrypted Vault file (from local storage or sync provider)
2. The 15 Secret Words
3. The Master Password

Users can generate a **Decryption Kit** - a PDF document containing the 15 Secret Words and optionally the Master Password hash (in QR code).

Recovery Steps

1. User inputs Secret Words and Master Password
2. App derives the Seed from Secret Words and Salt from the last 4 words
3. Master Key is derived using the stored Argon2id parameters
4. Per-Vault keys (tKey, sKey, eKey) are derived using HMAC-SHA256
5. Vault is decrypted and access is restored

Recovery can be performed offline if the Vault is stored locally, or requires internet connection when retrieving from a sync provider.

Important: Without the encrypted Vault AND valid recovery details, recovery is impossible. Valid recovery details means either:

- ➔ 15 Secret Words + Master Password, or
- ➔ Decryption Kit with Master Password hash

The zero-knowledge design ensures that no one, including 2FAS, can recover lost credentials.

7. Threat Model

Protected Against

- ➔ Server compromise: No central server by default
- ➔ Mass surveillance: Decentralized storage, no user account needed
- ➔ Device theft: No fallback from biometric to device PIN, encrypted storage
- ➔ Network interception: All sync traffic is end-to-end encrypted
- ➔ Sync provider compromise: Providers only see encrypted data

Assumptions

- ➔ User device is not compromised
- ➔ OS secure storage is trustworthy
- ➔ User protects Secret Words and Master Password
- ➔ Cryptographic primitives remain secure

8. Conclusion

2FAS Pass represents a new approach to password management, prioritizing user sovereignty and privacy without sacrificing usability. By combining local-first architecture with optional, user-controlled synchronization, it offers a robust alternative to centralized solutions.

The zero-knowledge design ensures that users maintain complete control over their data, while the Tiered Security model provides flexibility in protecting different types of information. With no vendor lock-in and full offline capability, users retain access to their passwords under any circumstances. 2FAS Pass delivers on its promise: your passwords, your control.

Feedback and Contributions

Technical Feedback

We welcome technical feedback on this white paper.
Join our community on Discord where you can discuss implementation details, suggest improvements, and engage with our team: https://2fas.com/discord/

Security Vulnerabilities

For security vulnerabilities, please use responsible disclosure:

- ➔ Email: security@2fas.com
- ➔ PGP Key: keys.openpgp.org

We strongly encourage encrypting sensitive reports using our PGP key.
To help us address issues quickly, please include:

- ➔ The specific product affected (e.g., 2FAS Pass iOS, 2FAS Pass Android, Browser Extension)
- ➔ Type of vulnerability (e.g., unauthorized data access, privilege escalation, encryption bypass)
- ➔ Detailed steps to reproduce the issue
- ➔ Environment details (device model, OS version, browser version)
- ➔ Potential impact and proof-of-concept code, if available

We follow a 90-day disclosure timeline and will keep you informed throughout the process.

Disclaimer

This white paper represents our current design and implementation plans. As 2FAS Pass is under active development, some features and technical details may change. We are committed to maintaining the core security principles while refining the implementation based on security audits and user feedback.