Enpass Hub

Security Whitepaper

Last modified on: July 14, 2023
Released version 1.0

Contents

Introduction

Enpass Hub is a supplementary server solution designed to enhance the functionality and security of the Enpass password manager for organizations. Enpass password manager does not store users' password vaults or sensitive data on its own servers. Instead, the data is stored as vault files either on users' devices or on the business storage of a company, such as Microsoft 365 OneDrive or SharePoint.

Enpass Hub enables organizations to implement additional features that would otherwise be impossible without a central server storing some sensitive data of vaults. These features include:

1. **Access Recovery** – This feature enables users to reset their master password if the organization has deployed an Enpass Hub server. The app sends vault keys to the server, encrypted with the organization-wide master recovery public key. Users can later request recovery of their vault if they have forgotten the master password. A recovery admin can process the request and send a recovery link to the user.
2. **Vault Sharing** – This feature allows seamless sharing of vaults without manually passing vault keys. Vaults are shared via Microsoft Graph API, and the encrypted vault key for the corresponding vault is stored on the Enpass Hub server. This eliminates the need to explicitly share vault passwords when sharing a vault.
3. **Security Audit** – This feature enables admins of organization to view overall password health reports for the company in the Enpass Admin console.

An organization that wishes to utilize these features can self-host a server called Enpass Hub. This server will act as a central point to facilitate the above-mentioned features. It will store all sensitive data (vault keys) encrypted using asymmetric cryptography, specifically the RSA 3072-bit keys with OAEP padding scheme.

This document provides a detailed overview of the security measures and procedures involved in Enpass Application and Enpass Hub to provide these features.

Enpass Hub integration

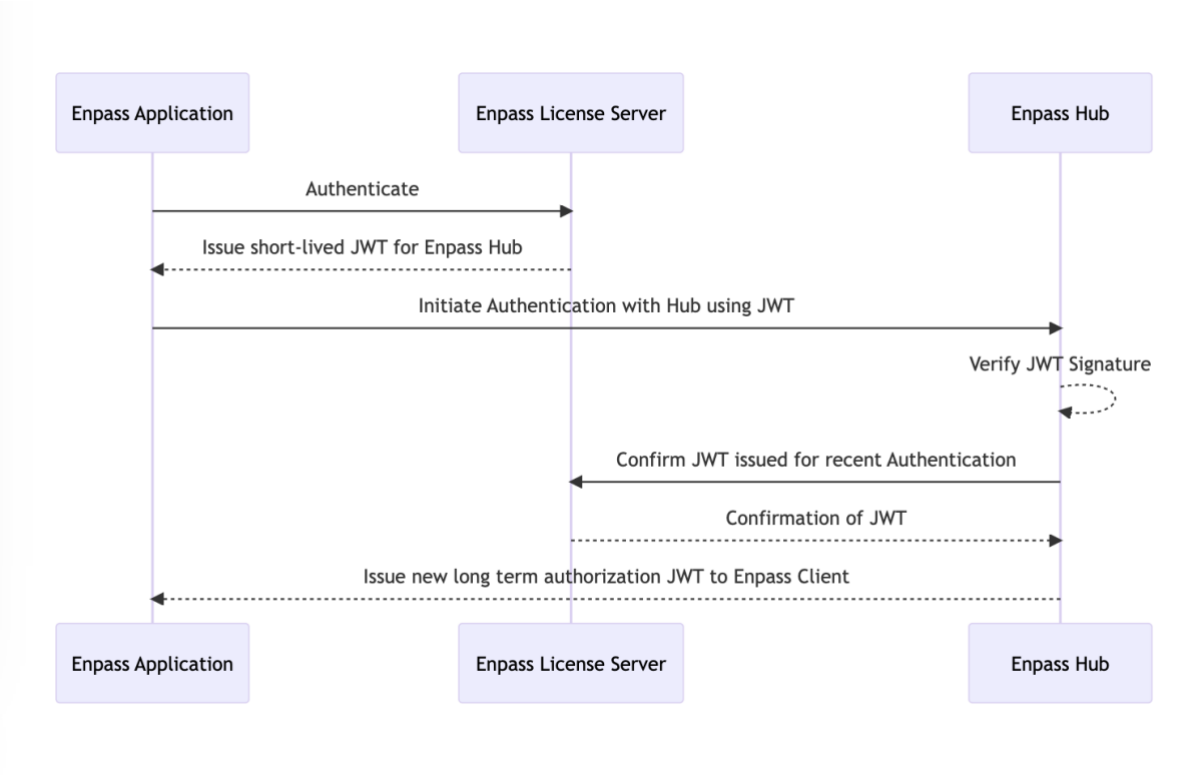
To make your hosted Enpass Hub features available to the users it must integrated with your organization’s Enpass account first. This integration makes the Enpass Hub's functionalities readily accessible to all users within the organization. It involves storing URL of your Hosted Enpass Hub and a Secret key that will act as a trust factor between Enpass server and Enpass Hub. Below are the security specific configurations:

1. **Secret Key**: The integration requires with the admin generating a long, random secret key on the Enpass Hub specific to the organization. This key is configured in the Enpass account of organization and it serves as a mutually agreed secret used to issue short-lived authentication tokens to app as part of the authentication process.
2. **SSL Certificate SHA-256 Fingerprint**: Optionally, the admin can provide the SHA-256 hash of the SSL certificate. The connected Enpass applications will utilize this

for certificate pinning, ensuring there is no Man-in-the-Middle or intercepting proxy between app and Enpass Hub.

User Authorization to Enpass Hub

To use Enpass Hub features, a User's app must be authorized first, in order to maintain the security and privacy of the Enpass Hub. The authentication the process is as followed:



The steps involved in user authorization to Enpass Hub are as follows:

1. **Authentication with Enpass License Server**: When a user's Enpass app authenticates with the Enpass license server, the server issues a short-lived JWT (JSON Web Token) for Enpass Hub with information such as Hub URL and SSL certificate pinning hash (if configured). The JWT is signed (HS-256 algorithm) using a pre-agreed Secret Key between the Enpass license server and Enpass Hub, configured during integration.
2. **Initiating Authentication with Enpass Hub**: The Enpass app then initiates authentication with the Enpass Hub using the signed JWT received from the Enpass license server.
3. **JWT Verification:** Enpass Hub verifies the JWT's signature locally with its copy of Secret Key and on successful verification, it contacts the Enpass license server to confirm if it was issued for a recent authentication i.e. within 1 minute.
4. **App Token Issuance**: After successful verification and confirmation, Enpass Hub issues an new JWT token to the Enpass app. This token allows the app to access the necessary features and resources on the Enpass Hub securely on the user's behalf.

Key Generation and Storage

The foundation of Enpass Hub's security architecture is the generation and storage of cryptographic keys. These keys play a crucial role in ensuring the security of sensitive data while enabling the desired features for organizations. The key generation and storage process involve the following steps:

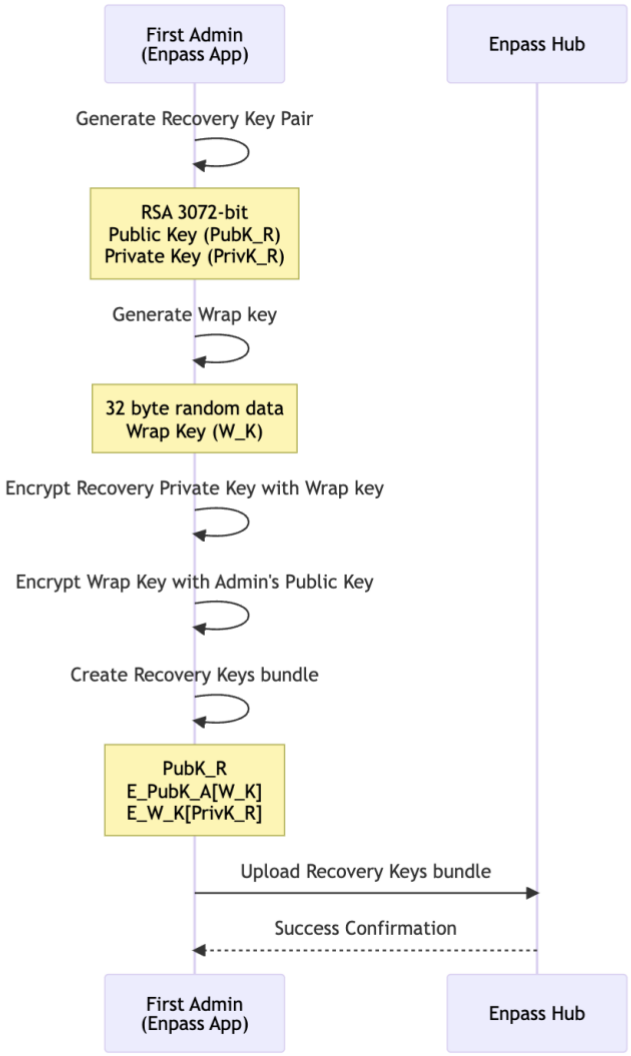
1. **User Keypair Generation**: When a user's Enpass app connects to Enpass Hub first time, a unique RSA 3072-bit keypair (public and private key) is generated for the user by the app.
2. **Private Key Storage**: The user's private key is securely stored in the user's Enpass vault. The private key never leaves the user's vault, ensuring that it remains protected from unauthorized access. This private key will be only used for decrypting data locally.
3. **Public Key Storage**: The user's public key is sent to the Enpass Hub and stored there. This public key will be used for encrypting sensitive data that can only be decrypted by the user's private key.

Access Recovery

Initial Admin Setup

The initial admin setup is a the first step in enabling the Access Recovery feature in Enpass Hub. During this process, the first recovery admin sets up the necessary keys and shares them with the Enpass Hub. The steps involved in the initial admin setup are as follows:

1. **Recovery Key Pair Generation**: The first recovery admin generates a RSA 3072-bit master recovery keypair. This key pair consists of a public key (PubK_R) and a private key (PrivK_R).
2. **Encryption of the Recovery Private Key**: The recovery admin encrypts the private key (PrivK_R) with a random AES-256 symmetric wrap key (W_K): E_W_K[PrivK_R]
3. **Encryption of the Wrap Key**: The recovery admin encrypts the wrap key (W_K) with their own public key (PubK_A): E_PubK_A[W_K]
4. **Sharing Encrypted Keys with Enpass Hub**: The encrypted private key (E_W_K[PrivK_R]), encrypted wrap key (E_PubK_A[W_K]), and public key (PubK_R) are sent to the Enpass Hub. The keys are securely stored on the Enpass Hub, enabling the recovery process when necessary.



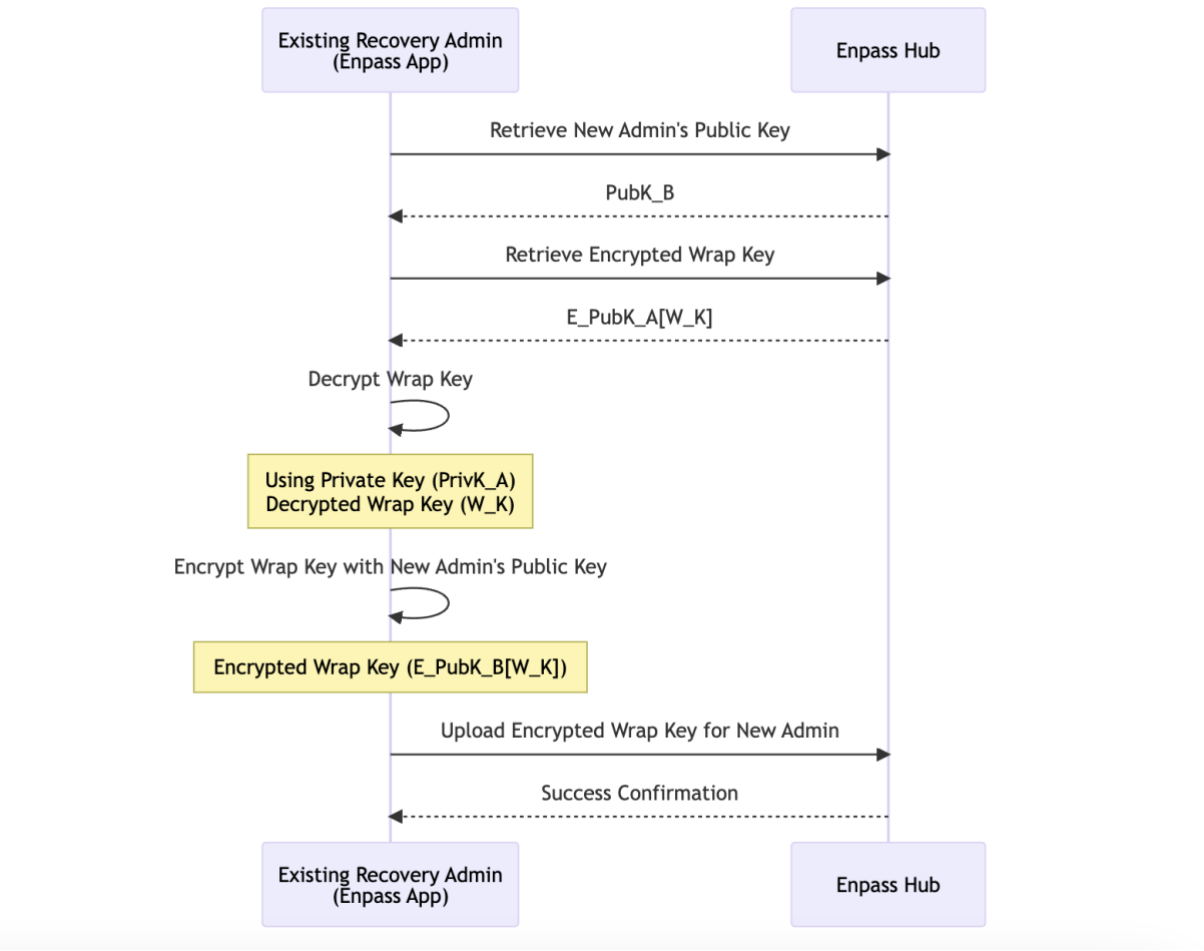
By completing the initial admin setup, the first recovery admin establishes the foundation for the Recovery feature by creating master recovery keypair. This process also ensures that only authorized admin has access to the master private key.

Adding a New Recovery Admin

Adding a new recovery admin is an important process to ensure redundancy and prevent lockouts in case of data loss or if an existing admin forgets their master password. An existing recovery admin with sufficient permissions can add another user as a recovery admin. The steps involved in adding a new recovery admin are as follows:

1. **Retrieve Encrypted Wrap Key**: The existing recovery admin retrieves the encrypted wrap key of the encrypted recovery private key (E_W_K[PrivK_R]) from Enpass Hub.
2. **Decrypt Wrap Key**: The existing recovery admin decrypts the encrypted wrap key (E_PubK_A[W_K]) using their private key (PrivK_A): D_PrivK_A[E_PubK_A[W_K]] = W_K

3. **Encrypt Wrap Key with New Recovery Admin's Public Key**: The existing recovery admin encrypts the decrypted wrap key (W_K) with the new recovery admin's public key (PubK_B): E_PubK_B[W_K]
4. **Share Encrypted Wrap Key with Enpass Hub**: The encrypted wrap key (E_PubK_B[W_K]) is sent to Enpass Hub and associated with the new recovery admin.



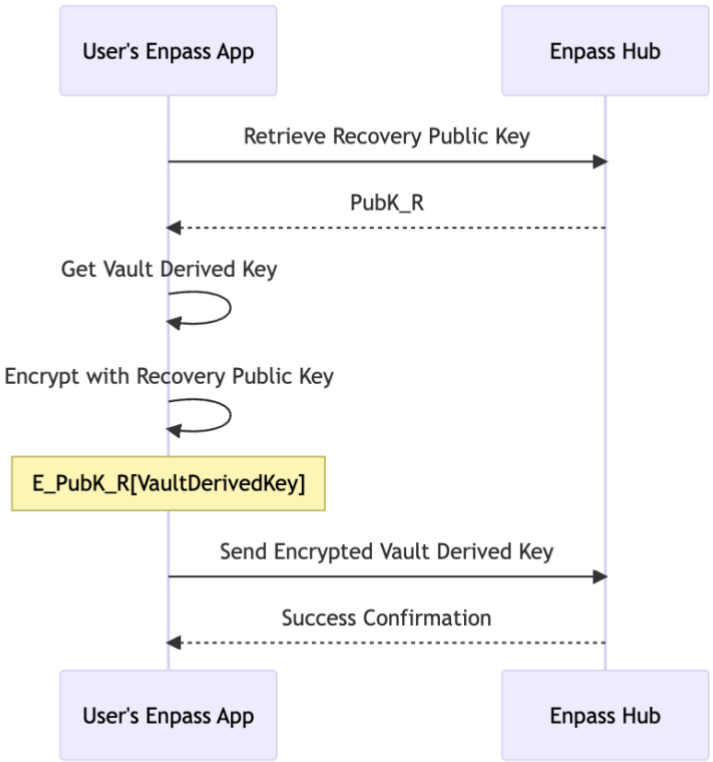
By following these steps, the new recovery admin can participate in the Recovery process, ensuring that the organization maintains redundancy and security in the password recovery process.

NOTE: ENPASS RECOMMENDS HAVING AT LEAST TWO ACTIVE RECOVERY ADMINS AT A TIME TO AVOID ANY LOCKOUT IN CASE OF DATA LOSS OR IF AN ADMIN FORGETS HIS MASTER PASSWORD, WHICH WON'T BE RECOVERABLE.

Recovery Data

Once the recovery public key (PubK_R) is available, Enpass app of each user, connected to the Enpass Hub, will send their business vault's derived keys encrypted with the master recovery public key to Hub. The steps involved in generating and sending the recovery data are as follows:

1. **Vault Derived Key Encryption**: Each user's Enpass application encrypts the derived key of their business vault and the identity info of the user, using the recovery public key (PubK_R): E_PubK_R[VaultDerivedKey]
2. **Sending Encrypted Vault Derived Key to Enpass Hub**: The encrypted vault derived key (E_PubK_R[VaultDerivedKey]) is sent to the Enpass Hub and securely stored. This ensures that only authorized recovery admins with the corresponding recovery private key can access and decrypt the vault keys in case of a recovery request.



The recovery data storage process ensures that the Enpass Hub can securely facilitate the Master Password Recovery feature, allowing recovery admins to access the necessary data for recovering a user's master password.

Access Recovery Process

The recovery process is initiated when a user forgets their master password and requests assistance from a recovery admin. This process is designed to securely restore the user's access to their Enpass vault while maintaining the confidentiality of the data. The steps involved in the recovery process are as follows:

1. **Recovery Request Initiation**: The user initiates the recovery process from their Enpass application, authenticates with their email, and sends a recovery request to the recovery admin.

2. **Request Verification**: The recovery admin receives the request and verifies its authenticity using external means such as phone calls, chats, or the company's support system.

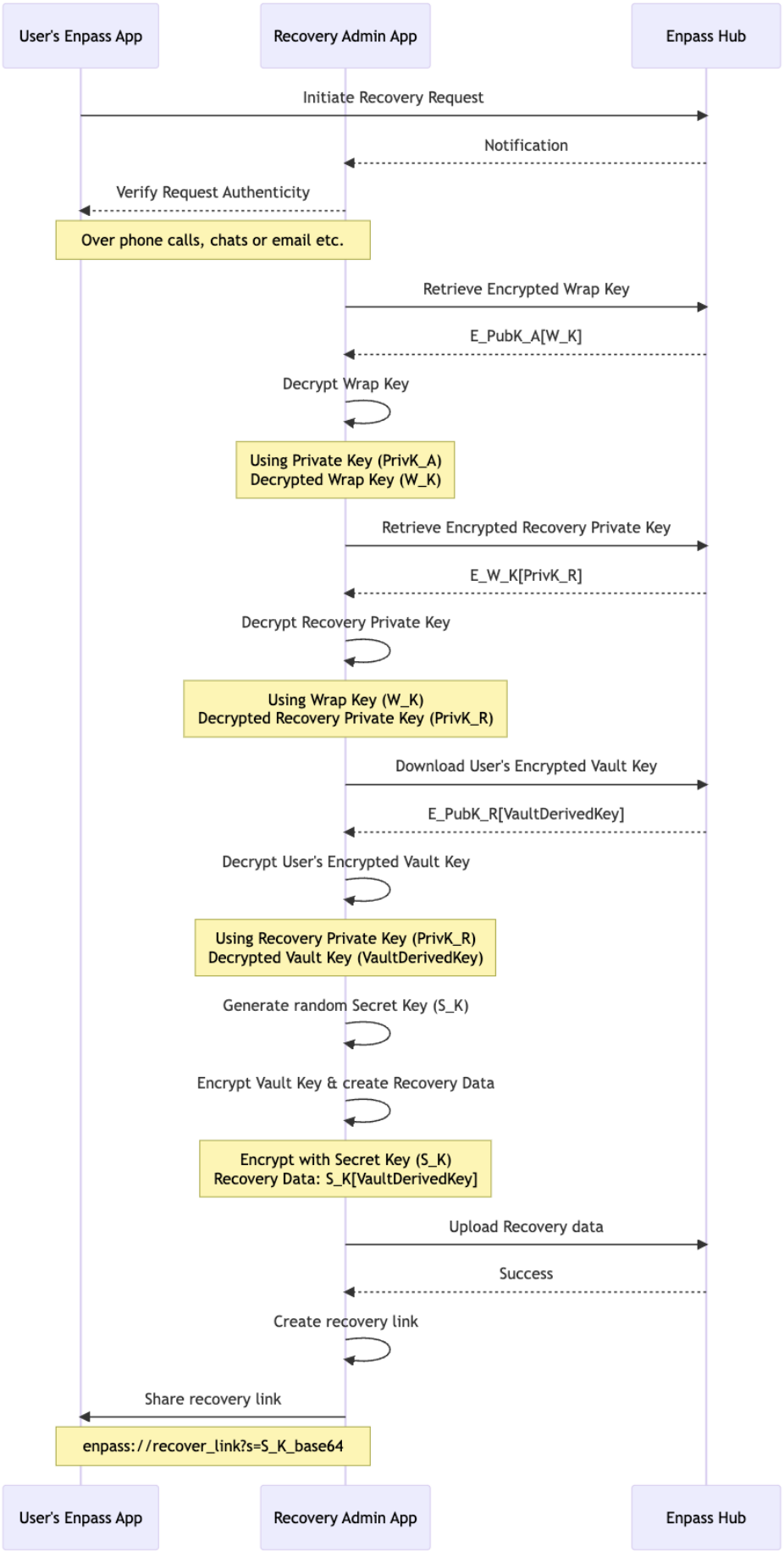
3. **Recovery Admin Decrypts Wrap Key and Recovery Private Key**: The recovery admins' app retrieves the encrypted wrap key (E_PubK_A[W_K]) and encrypted recovery private key (E_W_K[PrivK_R]) from Enpass Hub. The wrap key (W_K) is decrypted using the admin's private key (PrivK_A), and the recovery private key (PrivK_R) is decrypted using the wrap key (W_K).

4. **Download and Decrypt User's Encrypted Vault Key**: The user's encrypted vault key (E_PubK_R[VaultDerivedKey]) is downloaded from the Enpass Hub. The recovery admin decrypts the encrypted vault key using the recovery private key (PrivK_R).

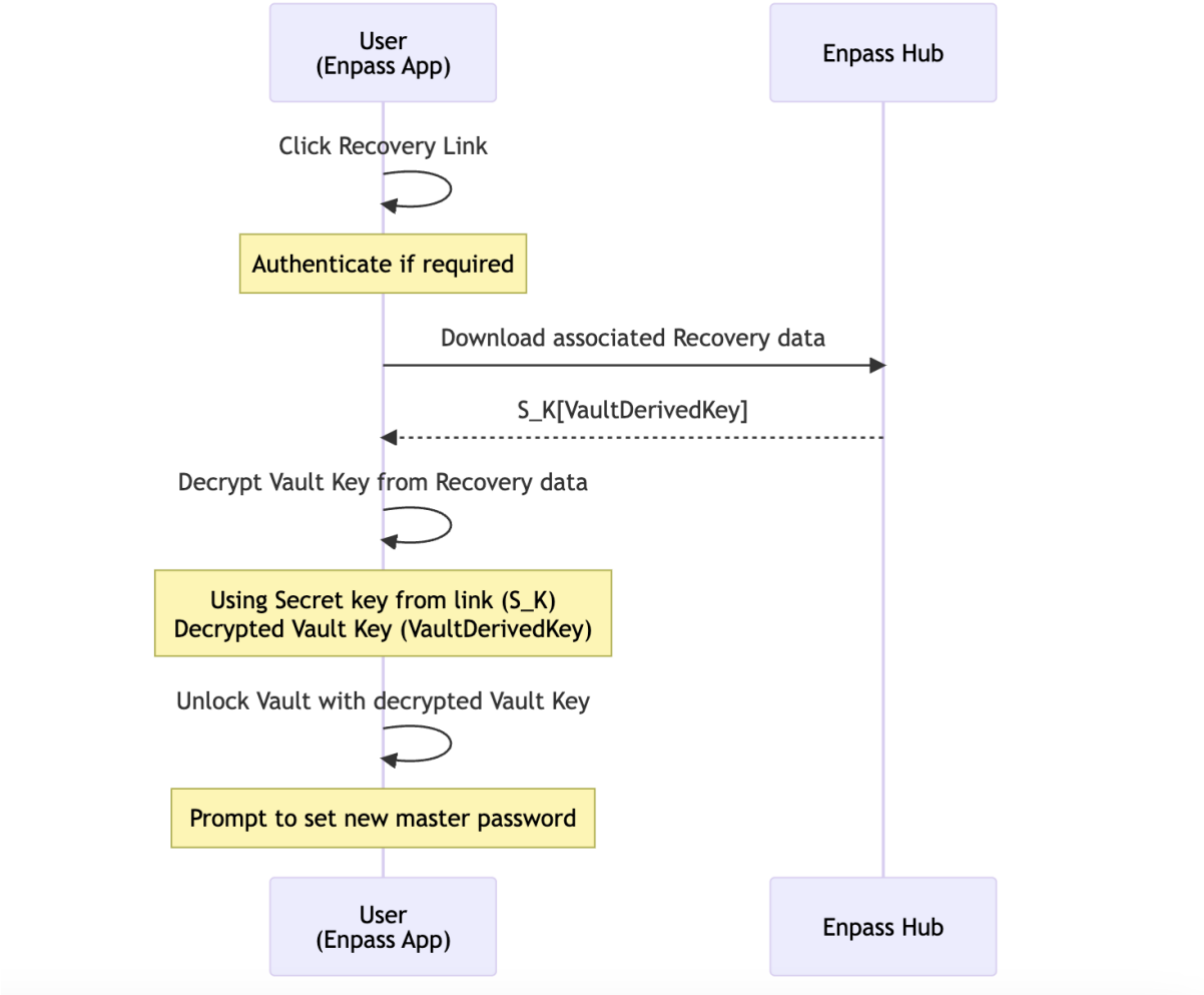
5. **Generate Temporary Secret and Encrypt Vault Key**: The recovery admin's app generates a random secret (s) and encrypts the vault key with it. The encrypted vault key is stored as a temporary accessible item on Enpass Hub, and a link is generated containing the random secret:
enpass://recover_link?s=<secret-base64-encoded>.

6. **Share Recovery Link with User**: The recovery admin shares the link with the user through an external means. The link is valid for 2 hours by default. Super admin of self-hosted Enpass Hub can adjust this links validity according to the their organization security policy.

   NOTE: THE SECRET IS NEVER STORED OR RELAYED THOUGH HUB. ADMIN SHOULD USE AN APPROPRIATE CHANNEL TO SEND THIS LINK TO USER.


User's Enpass App
Recovery Admin App
Enpass Hub
Initiate Recovery Request
Notification
Verify Request Authenticity
Over phone calls, chats or email etc.
Retrieve Encrypted Wrap Key
E_PubK_A[W_K]
Decrypt Wrap Key
Using Private Key (PrivK_A)
Decrypted Wrap Key (W_K)
Retrieve Encrypted Recovery Private Key
E_W_K[PrivK_R]
Decrypt Recovery Private Key
Using Wrap Key (W_K)
Decrypted Recovery Private Key (PrivK_R)
Download User's Encrypted Vault Key
E_PubK_R[VaultDerivedKey]
Decrypt User's Encrypted Vault Key
Using Recovery Private Key (PrivK_R)
Decrypted Vault Key (VaultDerivedKey)
Generate random Secret Key (S_K)
Encrypt Vault Key & create Recovery Data
Encrypt with Secret Key (S_K)
Recovery Data: S_K[VaultDerivedKey]
Upload Recovery data
Success
Create recovery link
Share recovery link
enpass://recover_link?s=S_K_base64
User's Enpass App
Recovery Admin App
Enpass Hub

7. **User's Access Recovery with Recovery Link**: The user processes the link in their Enpass app (after authenticating with email if not already), app downloads the encrypted vault key from the server, decrypts it with the secret from the link, and unlocks their vault. Enpass prompts the user to set a new master password.



By following these steps, the recovery process ensures a secure method for restoring a user's access to their Enpass vault without compromising the confidentiality of their data.

Vault Sharing

Vault sharing is an essential feature for organizations that allows users to securely share access to their vaults with other users. The process ensures that only authorized users can access the shared vault and is based on public-key cryptography. It involved following processes:

Enabling Sharing of Vault

When the first manager or creator of a vault initiates the sharing process for the first time on their Enpass application, vault is registered on the hub for sharing. The process involves following steps:

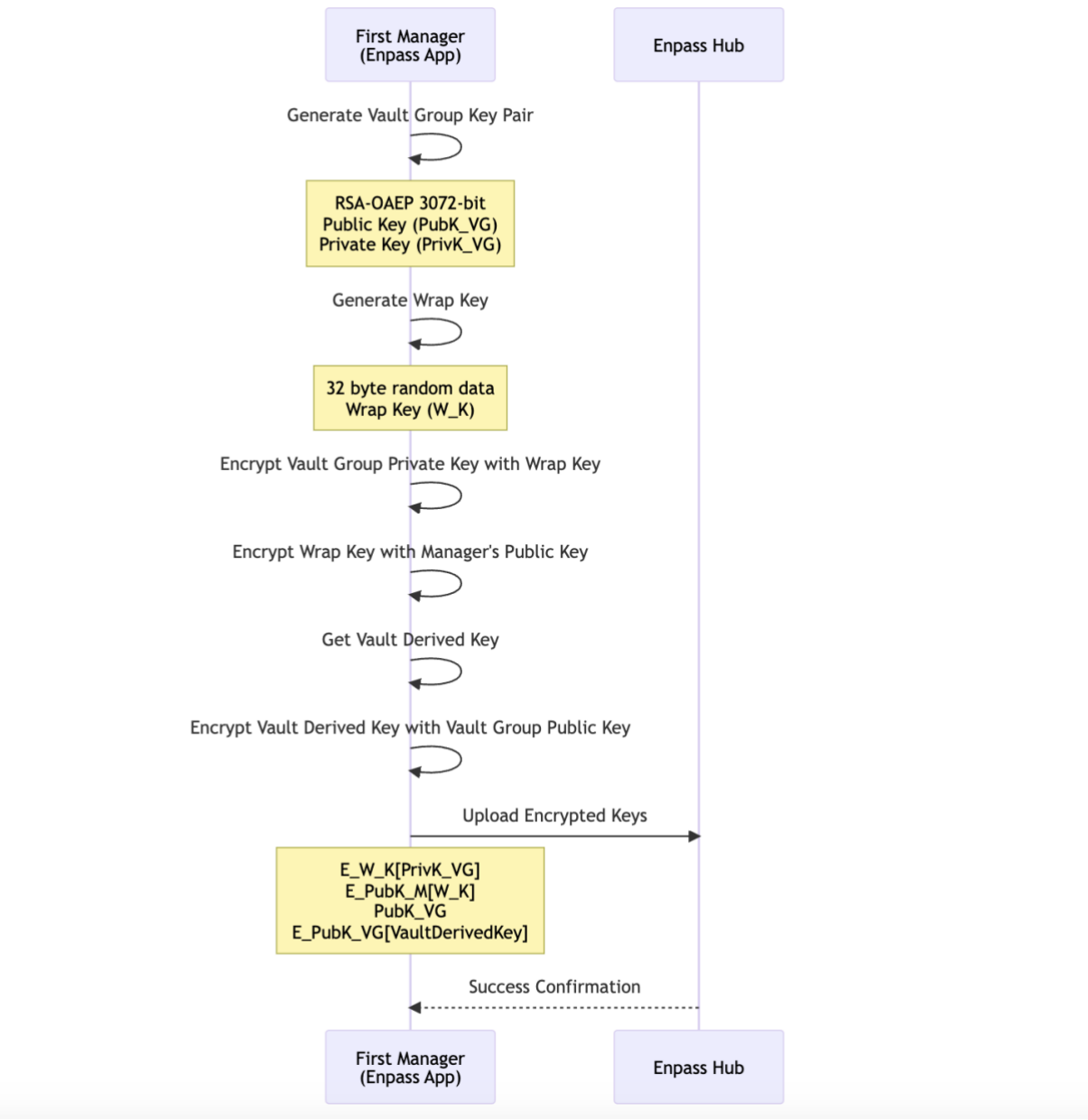
1. **Vault sharing group Key Pair Generation**: The first manager generates a vault group key pair using the RSA 3072-bit algorithm. This key pair consists of a public key (PubK_VG) and a private key (PrivK_VG).

2. **Encrypting Vault Group Private Key**: The vault group private key is encrypted with a random AES-256 symmetric wrap key (W_K): E_W_K[PrivK_VG]

3. **Encrypting Wrap Key**: The wrap key (W_K) is encrypted with the manager's public key (PubK_M): E_PubK_M[W_K]

4. **Encrypting Vault Derived Key**: The vault-derived key is encrypted with the group's public key (PubK_VG): E_PubK_VG[VaultDerivedKey]

5. **Sharing Encrypted Keys with Enpass Hub**: The encrypted private key (E_W_K[PrivK_VG]), encrypted wrap key (E_PubK_M[W_K]), group public key (PubK_VG), and encrypted vault key (E_PubK_VG[VaultDerivedKey]) are sent to Enpass Hub.



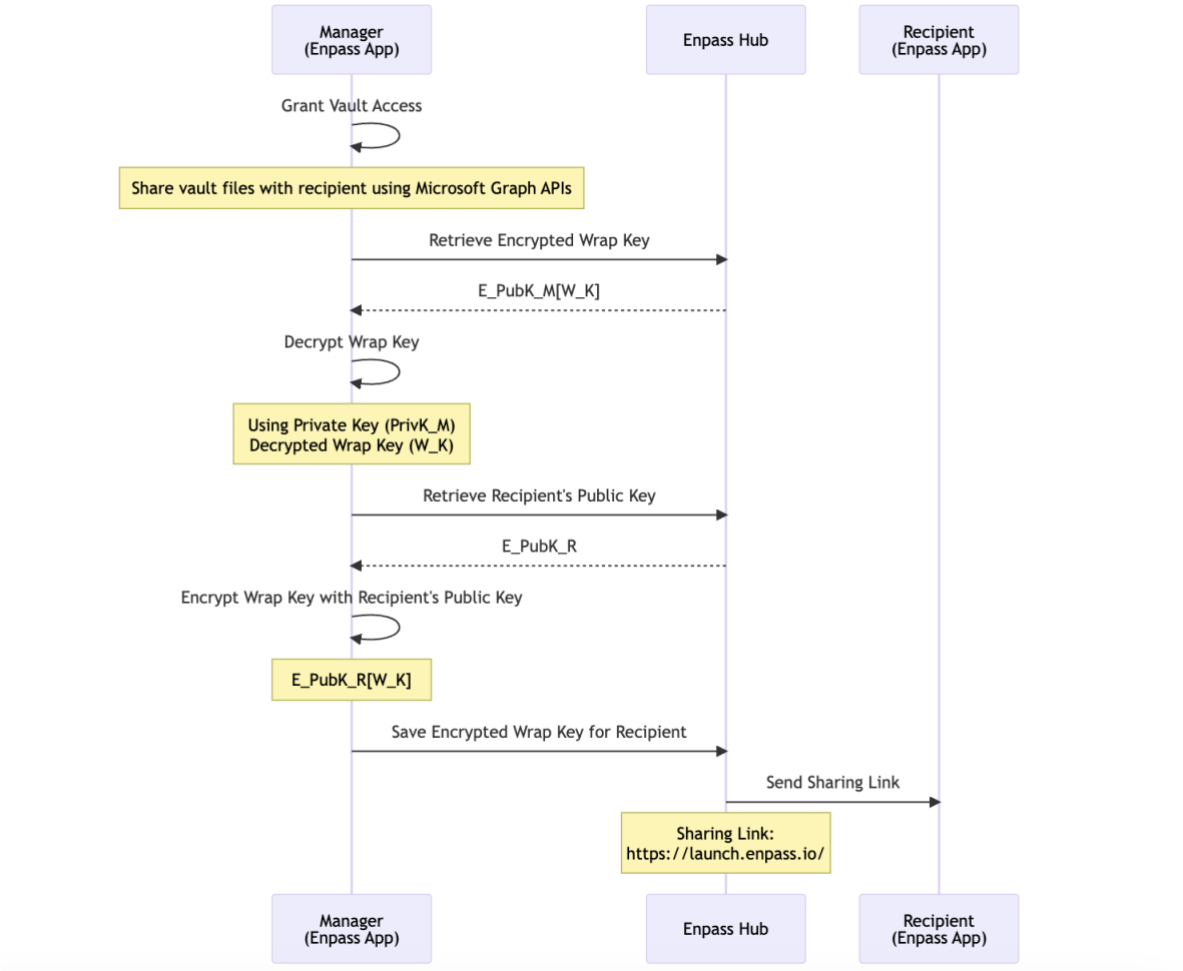
The vault is now ready to be shared securely with other users.

Sharing with Other Users

The manager can now share the vault by adding more users to the vault sharing group. The process involves following steps:

1. **Granting Vault Access**: The manager's app shares the vault files with the intended recipient using Microsoft Graph APIs, ensuring that the recipient can access the vault files on Microsoft 365 OneDrive or SharePoint.
2. **Retrieving and Decrypting Wrap Key**: The manager's app retrieves the his copy of encrypted wrap key of the vault sharing group (E_PubK_M[W_K]) from Enpass Hub and decrypts it with his private key (PrivK_M): D_PrivK_M[E_PubK_M[W_K]] = W_K
3. **Encrypting Wrap Key with Recipient's Public Key**: The manager encrypts the decrypted wrap key (W_K) with the new receiving party's public key (PubK_R): E_PubK_R[W_K]

4. **Saving Encrypted Wrap Key**: The encrypted wrap key (E_PubK_R[W_K]) is now sent to Enpass Hub, associated with the new user and appropriate permissions.

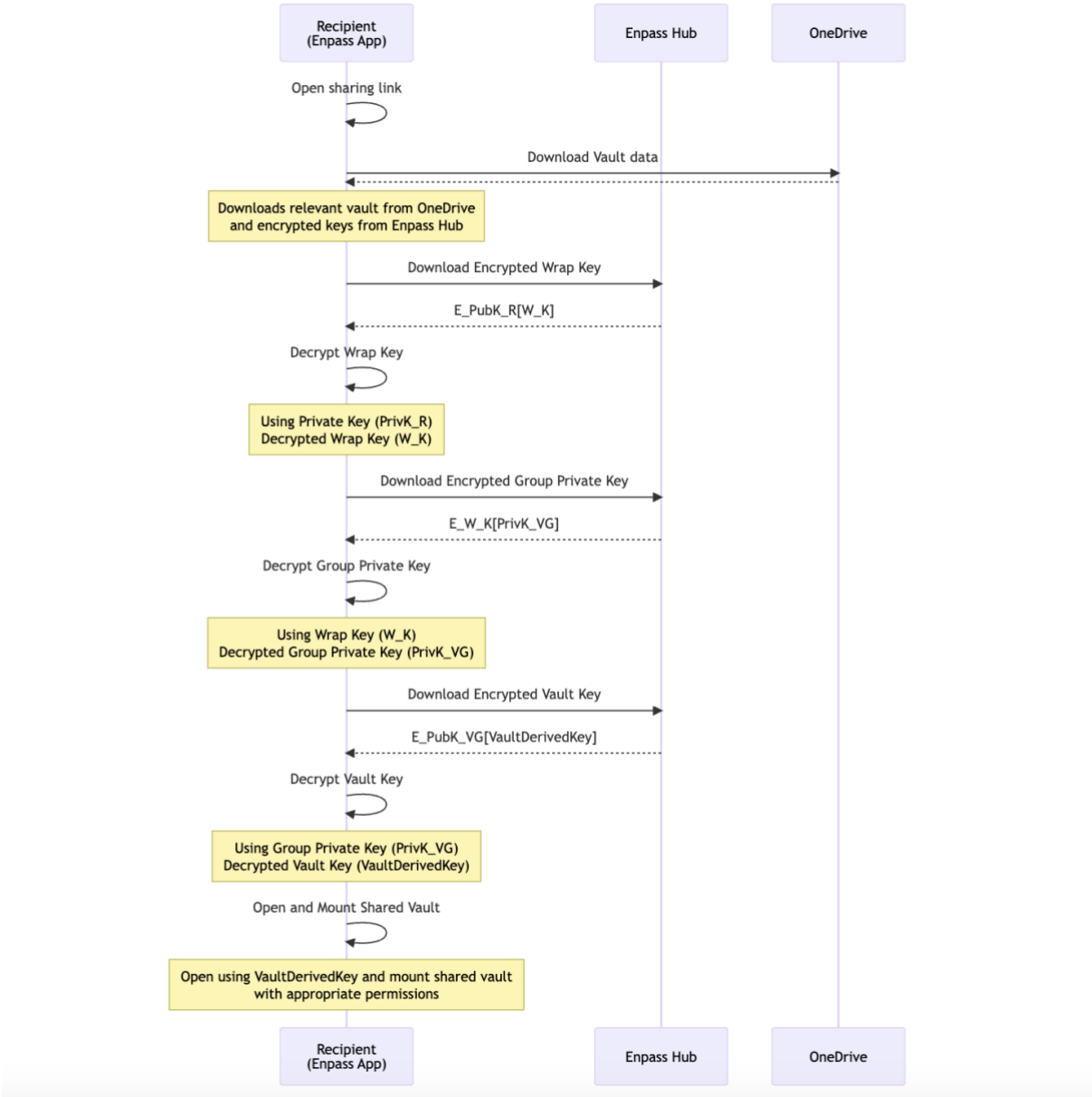


5. **Generating Sharing Link**: A sharing link is generated and sent to the recipient via email, containing the identification of the shared vault: https://launch.enpass.io/#/<share-id>

Mounting the Shared Vault:

1. **Opening Enpass App**: The recipient clicks the sharing link, opening their Enpass app.
2. **Downloading Vault and Encrypted Keys**: The app downloads the relevant vault from OneDrive, the encrypted vault group private key (E_W_K[PrivK_VG]), the encrypted vault key (E_PubK_VG[VaultDerivedKey]), and his copy of encrypted wrap key (E_PubK_R[W_K]) from Enpass Hub.
3. **Decrypting Wrap Key**: The recipient decrypts the wrap key (E_PubK_R[W_K]) with his private key (PrivK_R): D_PrivK_R[E_PubK_R[W_K]] = W_K
4. **Decrypting Group Private Key**: The decrypted wrap key (W_K) is used to decrypt the encrypted vault key group private key (E_W_K[PrivK_VG]): D_W_K[E_W_K[PrivK_VG]] = PrivK_VG
5. **Decrypting Vault Key**: The decrypted PrivK_VG is used to decrypt the encrypted vault key (E_PubK_VG[VaultDerivedKey]): D_ PrivK_VG [E_PubK_VG[VaultDerivedKey]] = VaultDerivedKey

6. Opening and Mounting Shared Vault: The shared vault is opened and mounted with appropriate permissions using the decrypted vault derived key.



By following these steps, the vault sharing process ensures that only authorized users can access the shared vault while maintaining the confidentiality and integrity of the data.

Handling Key-Pair Loss

In an unlikely event that a user loses their access to vault and he will need to start everything from scratch. This is because all the access keys shared to the user were encrypted using his registered public key on the Enpass Hub and now he has no longer access to the corresponding private key. To start everything from scratch, the following

steps must be taken to ensure the user can regain access to shared vaults and, if applicable, their recovery admin status:

1. **Reset Public Key Registration**: The user must ask the administrator to reset their registered public key on the Enpass Hub server. This will remove the old public key associated with the user.
2. **Register new Public Key**: The user will now start the Enpass app and create a new vault if not already. The new public key will be generated and registered with Enpass Hub as described in "Key Generation and Storage" section.
3. **Loss of Shared Vault Access**: The user's access to all previously shared vaults will be lost due to the absence of their original private key. Vault owners will need to re-share access to the affected vaults with the user.
4. **Re-adding a Recovery Admin**: If the user was a recovery admin, they will need to be re-added by another recovery admin.

Security Audit

Security audits are crucial for maintaining the overall security of the organization's password management system. Enpass Hub provides a feature that allows administrators to review the password health of each user's vault and the organization as a whole. This helps identify potential vulnerabilities and areas for improvement. The steps involved in the security audit process are as follows:

1. **Gathering Vault Audit Information**: Each user's Enpass application sends their business vault's audit information to the Enpass Hub. This data includes numerical data about the total items, total passwords, the number of breached websites, weak, duplicated, and compromised passwords, among other metadata of vault. However, any of the vault content like passwords or URLs etc. are not sent to the Enpass Hub.
2. **Aggregating Audit Data**: The Enpass Hub aggregates the received audit information to calculate an overall security score for the organization and individual scores for each user.

The administrator can now see the security scores. Based on the security audit results, he can decide on the necessary actions to improve password security, such as enforcing stricter password policies or providing user training on password best practices.