DASHLANE

Dashlane's Security Principles & Architecture

Table of Contents

This document is organized so readers can either follow it end to end or jump directly to the sections most relevant to their role. Each chapter includes a **Summary** for quick orientation.

- **Chapter 1 – Executive Summary**
- **Chapter 2 – Securing the Enterprise:** Explains how Dashlane fits into a modern enterprise security stack and how it mitigates common credential-based attack vectors.
- **Chapter 3 – Architecture Overview:** Details Dashlane's zero-knowledge architecture, encryption model, device trust, communication security, and use of confidential computing.
- **Chapter 4 – Credential Security in Detail:** Walks through core features of Dashlane: authentication flows, vault encryption lifecycle, multi-factor authentication, account recovery, secure sharing, and local data handling.
- **Chapter 5 – Enterprise Features:** Covers SSO, SCIM provisioning, role-based access control, activity logging and auditing, mass deployment, APIs, and proactive risk-reduction capabilities such as Credential Risk Detection, Nudges, and AI-powered anti-phishing.
- **Chapter 6 – Phishing Resistance:** Focuses on passkeys, passwordless authentication, FIDO2 security keys, and platform-level protections against phishing.
- **Chapter 7 – Attack Scenarios and Threat Model against Dashlane:** Analyzes realistic attack scenarios and contrasts Dashlane's architecture with more traditional approaches.
- **Chapter 8 – Security Operations:** Describes vulnerability management, incident response, bug bounty, penetration testing, and continuous improvement practices.
- **Chapter 9 – Compliance & Certifications:** Summarizes Dashlane's certifications, privacy commitments, and support for enterprise compliance.

---

1. Executive Summary

Credential-based attacks remain one of the most significant enterprise security threats. Dashlane is an **enterprise-grade credential security platform** designed to protect the entire lifecycle of credentials across your workforce, strengthening security, supporting compliance, and reducing risk exposure.

Dashlane provides a secure, encryption-based cloud storage solution, built on what we call a **zero-knowledge architecture**. Our technical design ensures only the user, not Dashlane or any third party, can decrypt their vault. Even if Dashlane's infrastructure is compromised, attackers should not be able to access stored credentials or secrets. The platform combines device-level encryption and cloud secure enclaves to protect data at rest, in transit, and in use.

Dashlane's security philosophy is grounded in one principle: Data must remain secure under all circumstances. Even if infrastructure, devices, or internal environments are compromised, our zero-knowledge architecture ensures vault data stays protected. We continuously evaluate and refine our threat model to defend against the most relevant attack vectors: application vulnerabilities, compromised devices, server attacks, internal IT breaches, and insider threats.

Through our secure-by-design approach, Dashlane embeds security into every phase of development. Formal threat modeling, code reviews, continuous third-party testing, and an active HackerOne bug bounty proactively strengthen defenses. Oversight from our internal Risk Committee ensures transparency, compliance, and continuous risk management at the highest level. Together, these practices enable enterprises to trust Dashlane as a platform that reduces exposure, maintains usability, and aligns with modern security and compliance standards.

For enterprises, Dashlane integrates seamlessly with existing identity and access management (IAM) frameworks through single sign-on (SSO) and SCIM provisioning. Administrative controls, role-based policies, detailed activity logs, and secure sharing empower IT and security teams to manage credentials at scale while giving employees flexibility to collaborate securely.

Download our "Security at a glance" one-pager

2. Securing the Enterprise

Summary

Dashlane reduces enterprise credential risk, including phishing, credential theft and shadow IT, without increasing the attack surface for the enterprise.

Dashlane aims to strengthen enterprises' security posture by actively mitigating a broad range of threats, including **phishing**, **credential stuffing**, **insider threats**, and **shadow IT/unmanaged sharing**. Dashlane delivers this protection while actively avoiding an increase in the enterprise's attack surface.

2.1 Mitigation to Credential Threats

All solutions are described in more detail in further sections of this document.

2.1.1 Phishing

Phishing is a social engineering attack that commonly uses a fake website to trick a user into entering sensitive information by impersonating a trustworthy organization.

The Dashlane extension prevents users from entering credentials on fraudulent domains, mitigates many click-jacking attacks, and provides users with cross-platform passkey support and a built-in, AI-powered phishing detection engine.

2.1.2 Credential Theft

Credentials are highly valuable targets for threat actors, making them critically vulnerable to attack vectors such as brute-force, credential stuffing, and large-scale data breaches.

Dashlane provides an easy way to generate strong and unique passwords for every single service users want to log in to, decreasing the risk of using weak or compromised credentials and the risk of credential reuse, where one compromised password exposes other accounts. Dark Web Monitoring features alert users so they can quickly react in case of theft or compromise.

2.1.3 Insider Threats

An insider threat refers to a security risk that originates within the organization. This person is often a current or former employee, contractor, or business associate who has or had authorized access to the organization's network, systems, or data and uses that access to maliciously or inadvertently compromise the organization's security.

Dashlane addresses insider threats through features such as Role-Based Access Control (RBAC), delegated admin controls, and detailed logs that monitor employee behavior, while its zero-knowledge encryption prevents unauthorized vault access.

2.1.4 Shadow IT and Unmanaged Sharing

Shadow IT/unmanaged sharing risks occur when employees use unauthorized or unsanctioned tools and methods (shadow IT) or share sensitive data through unsecured channels (unmanaged sharing). This creates security gaps that bypass corporate controls, increasing the risk of data loss, non-compliance, and exposure to unmonitored threats.

Dashlane mitigates shadow IT/unmanaged sharing risks by providing secure, policy-enforced credential sharing, which, combined with delegated administrative controls and event tracking, enables safe collaboration while strictly limiting unauthorized access.

2.2 Enterprise Integration

Dashlane is designed to integrate seamlessly into the modern enterprise security stack, strengthening credential protection without introducing additional complexity.

- **Fits into your ecosystem**: With standards-based SSO, SCIM provisioning, seamless integration with common MDM, and compatibility with leading identity and endpoint management solutions, Dashlane works with the tools you already rely on.

- **Streamlined for IT**: Deployment and administration are simple, reducing overhead for IT teams and avoiding the hidden costs of managing another siloed system.

- **Visibility and control for security leaders**: Detailed activity logs, real-time alerts, and granular role-based access give security teams the oversight they need to monitor, detect, and respond effectively.

- **User-centric protection**: Employees can securely manage and share credentials with a consistent experience that drives adoption and improves password hygiene.

By combining **seamless integration**, **user-first security**, and **enterprise-grade controls**, Dashlane elevates an organization's credential security posture while reducing operational burden across IT and security teams.

2.3 Zero-Overhead Security Posture

Dashlane is architected not only to solve credential threats but also to prevent itself from introducing new risk. Our defense-in-depth approach is centered on architectural safeguards aiming for a minimal footprint on the enterprise's attack surface:

- **Supply chain compromise:** Confidential computing and secure cloud enclaves isolate cryptographic operations from infrastructure, ensuring that keys and vault data remain protected even if the underlying cloud environment or Dashlane itself is compromised.
- **Man-in-the-middle attacks:** End-to-end encryption secures credentials at every stage, on device, in transit, at rest, and in use, with confidential computing providing defense even if network transport protocols are compromised.
- **Account takeover:** Multi-factor authentication, admin-assisted account recovery, automated SCIM deprovisioning, and secure device provisioning prevent account hijacking and enable rapid response to unauthorized access attempts.

3. Architecture Overview

Summary

Dashlane's architecture is built on a **zero-knowledge architecture** that protects user data at every layer, device, cloud, and network, ensuring that no one, including Dashlane, can access vault contents.

- **Zero-knowledge architecture:** All vault data is encrypted locally using AES-256-CBC-HMAC and decrypted only on authorized devices.
- **Device authentication:** Each device is uniquely bound to an account through a Device Key, enabling secure synchronization without exposing credentials.
- **Encryption model:** Vault encryption is separate from authentication, mitigating single-point failures.
- **Communication security:** End-to-end encryption and TLS with HSTS and forward secrecy protect data in transit.
- **Confidential computing:** Dashlane-hosted enterprise components, such as SSO integration, operate within **AWS Nitro Enclaves**, isolating cryptographic operations from host systems and reinforcing zero-knowledge guarantees.

Together, these controls deliver **enterprise-grade protection**, **verifiable confidentiality**, and **seamless integration** with modern identity and security stacks.

Dashlane is designed with a zero-knowledge architecture that ensures only the end-user can access their data and end-user devices decrypt vault data. The following sections detail the cryptographic, architectural, and operational measures used to protect user credentials at enterprise scale.

3.1 Zero-Knowledge Architecture

All vault data is encrypted before leaving the user's device and can only be decrypted locally. Neither Dashlane nor any third party can access, read, or recover the contents of a vault.

Here is an overview of how this zero-knowledge architecture is applied to critical flows. More details will be provided in subsequent chapters.

Strong Encryption

- Each individual's vault is encrypted with **AES-256-CBC + HMAC-SHA256** before storage on Dashlane's servers.

- Decryption requires either:

    - A **Master Password**, known only to the user, or

    - A hidden **Machine-Generated Master Password** for passwordless users

- The vault encryption key derived from one of these factors is never transmitted to Dashlane or stored on our servers in plaintext. Therefore, Dashlane will never be able to decrypt users' vaults.

Device Authentication

Each device accessing a Dashlane vault must be explicitly verified and bound to the user through a **User Device Key**. This ensures that even if credentials are stolen, an attacker cannot access vault data without control of an authorized device.

- **New account or new device setup**:
  When a user creates a new Dashlane account or enables an additional device for data synchronization, Dashlane verifies the identity of the account holder. This verification is completed by sending a one-time token to the registered email address or mobile number, after which the system auto-generates a **unique User Device Key**.
- **Passwordless login flow**:
  For passwordless users, the process is even more secure: Authorizing a new device requires approval from an already registered device. This eliminates reliance on email, which can be vulnerable to interception or account takeover.

Defense in Depth for Access

- **Decryption on the device**:
  When a user enters their Master Password in the Dashlane app (or uses passwordless authentication), vault data is decrypted **only within the memory of that authorized device**. Vault data is never decrypted on Dashlane's servers.

- Users can strengthen security with **two-factor authentication (2FA)** using TOTP Authenticator applications. Enabling the 2FA option at each login means that both the Master Password and the authenticator code are necessary for decrypting the vault.
- All communications between devices and Dashlane servers are secured using **TLS cryptographic protocols**.

Account Recovery

Dashlane provides recovery mechanisms that preserve zero-knowledge design while enabling continuity:

- **Account recovery key (ARK)**: Available for both Master Password and passwordless accounts, this single-use key allows secure self-service recovery without weakening encryption guarantees.
- **Biometric recovery on mobile:** Available for Master Password accounts only. Allows a user to reset their Master Password using their face/fingerprint on a mobile device if they forget their password.
- **Admin-assisted recovery**: Business admins can approve recovery requests via the Admin Console, acting as a trusted verifier of user identity.

3.2 Encryption Model: Secrets and Protections

Dashlane's encryption relies on distinct, purpose-scoped secrets. This separation of concerns ensures that a compromise of one element does not expose the vault data and authentication mechanisms at the same time.

Table 1: Dashlane Secrets Overview

Secret	Purpose	Origin	Storage	Notes
User Master Password (UserMP)	**Vault Encryption** Derives the vault encryption key	User-chosen	Not stored on servers; optional local storage (encrypted) if "Remember MP" enabled	Strength enforced with zxcvbn ($\geq 10^8$–10^{10} guesses)

Machine-Generated Master Password (MachineGeneratedMP)	**Vault Encryption** Replaces UserMP for passwordless accounts	Generated on device	Not stored on servers; transient in memory; on-disk only for web extension	40 chars, ~243 bits entropy
Intermediate Key (IntermediateKey)	**Vault Encryption** Supports local storage encryption	Generated on device	Stored locally, encrypted with derivative of UserMP	Random 32 bytes
User Device Key (DeviceKey)	**Authentication** Authenticates device to Dashlane servers	Generated server-side for each device	Stored locally in encrypted user data	Unique per device, separate from vault key
User Secondary Key (UserSecondaryKey)	**Vault Encryption** Adds second factor to vault encryption when 2FA enabled	Generated server-side upon 2FA activation	Released only after successful 2FA challenge	Combined with MP-derived material
Account Recovery Key (ARK)	**Vault Encryption** Enables account recovery while preserving zero-knowledge	Generated on device with Dashlane password generator	Held by user; not stored by Dashlane	28 chars, ~145 bits entropy, single-use
Mass Deployment Team Key	**Authentication** Authenticates logged-out users in enterprise mass-deployment flows	Generated by Dashlane servers	Stored on client device with restricted use	Random 32 bytes

3.2.1 Deep Dive on Key Secrets

Dashlane's encryption model relies on multiple secrets, each strictly separated by purpose. This design prevents any single compromise from exposing user vault data or authentication channels. Below are detailed explanations of the most critical secrets.

User Master Password (UserMP)

- **Strength validation**: Dashlane uses the zxcvbn library to validate UserMP strength. The library analyzes the chosen password against multiple patterns (common passwords, dictionary words, keyboard patterns, complexity rules) and returns a score between 0 and 4:
    - **0**: Too guessable
    - **1–2**: Weak to moderate strength
    - **3–4**: Safely unguessable (estimated 10^8–10^{10} guesses required)
- **Enforcement**: Dashlane applications enforce a minimum score of **3**, ensuring users select sufficiently strong Master Passwords. Feedback is provided to help users improve weak choices.
- **Storage and transmission**:
    - The User Master Password is never sent to our server in plaintext; therefore, Dashlane will never be able to decrypt users' vaults.
    - Not stored locally by default; used only transiently to decrypt local files.
    - Optional encrypted local storage if the user enables "**Remember my Master Password**."

Machine-Generated Master Password (MachineGeneratedMP)

- **Purpose**: Alternative to the UserMP, used in passwordless accounts
- **Generation**: A strong, unique **40-character string** (~243 bits of entropy) generated using Dashlane's Password Generator
- **Storage and transmission**:
    - Never stored on Dashlane servers, nor are any derivatives

    - Not stored locally by default, except when logging in to the **web extension**
    - Never transmitted over the internet; only the encrypted vault is sent
- **Security advantage**: Machine-generated values eliminate human weakness (e.g., predictable passwords) and guarantee a high entropy

Intermediate Key

- **Purpose**: Used in certain local storage scenarios to strengthen the protection of encrypted files
- **Generation and use**: Random 32-byte value generated locally on the device
- **Protection**: Stored encrypted with a derivative of the UserMP
- **Design principle**: Adds another layer of defense so that one compromised key (e.g., through local device access) is insufficient to unlock vault data

User Device Key (DeviceKey)

- **Purpose**: Authenticates each device to Dashlane servers; prevents vault access from unauthorized devices, even if credentials are known
- **Generation**: Generated server-side for each new device linked to an account
- **Storage**: Stored locally, encrypted within user data.
- **Security advantage**: Separation of **vault encryption** (UserMP or MachineGeneratedMP) from **device authentication** ensures that vault compromise does not imply server compromise, and vice versa

3.2.2 Local Access to User Data

Access to a user's encrypted vault requires the **User Master Password (UserMP)** or, in the case of passwordless accounts, the **Machine-Generated Master Password (MachineGeneratedMP)**.

- **Vault key derivation**:

    - The UserMP (or MachineGeneratedMP) is used to derive a **256 bits key** that encrypts and decrypts vault data using **AES-256-CBC + HMAC-SHA256** locally on the user's device.

    - In passwordless flows, the MachineGeneratedMP is not exposed to the user; it is transported securely between devices during enrollment of a new device, and then functions identically to the UserMP for key derivation.

- **Cryptographic libraries**:

    - Dashlane uses the **Web Crypto API** for browser-based clients.

    - On mobile, Dashlane relies on **native cryptographic libraries** for iOS and Android.

    - For key derivation, Dashlane implements the **Argon2** reference library, compiled to WebAssembly (for browsers) or linked directly into the mobile applications.

    - Dashlane also supports PBKDF2 as a backward compatibility option for old devices. This support will be deprecated in 2026.

Our approach ensures that all cryptographic operations happen locally, using hardened and well-vetted primitives, with no exposure of sensitive material to Dashlane servers.

3.2.3 Local Data Usage After Decrypting

Once a vault has been decrypted—after the user provides their Master Password or validates their MachineGeneratedMP through a PIN code, biometric authentication, or a security key—data is handled under strict constraints to balance **usability** with **security**:

- **Session-limited decryption**:
  User data is only decrypted into **volatile memory** and never written back to persistent storage in plaintext.

- **Secure automation**:
  Dashlane processes individual credentials in memory to autofill them into websites and apps or to save new credentials, without requiring the user to re-enter their Master Password or MachineGeneratedMP each time.

- **Key-stretching defense**:
  Derivation of AES key from the UserMP (or MachineGeneratedMP) employs **Argon2d** (or PBKDF2 in fallback cases). These algorithms introduce computational latency by design, making brute force attacks significantly more expensive for attackers while remaining acceptable for legitimate users.

- **Memory protection**:
  The decrypted data in memory is protected by **leveraging pre-existing, platform-level safeguards** (detailed in 7.4 Memory Attacks), ensuring that exposure is minimized even in the event of device compromise.

3.3 Devices and Authentication

Dashlane uses a **device-based authentication model** to ensure that only verified devices can access user vaults, without ever relying on or transmitting a user's Master Password or Machine-Generated Master Password.

When a user creates an account or adds a new device for data synchronization, Dashlane generates a **User Device Key (DeviceKey)** composed of 40 random bytes using the OpenSSL `RAND_bytes` function. The first 8 bytes form the access key, while the remaining 32 bytes form the secret key. This DeviceKey uniquely binds each device to the user's account.

- **Local protection:** The DeviceKey is stored locally within encrypted user data, protected by the vault encryption mechanism.
- **Server protection:** On Dashlane servers, the secret key component is encrypted at rest, ensuring that employees or attackers cannot easily impersonate a legitimate device.
- **Independent from vault credentials:** Authentication to Dashlane servers uses only the DeviceKey, never the User Master Password or Machine-Generated Master Password.
- **Frictionless experience:** Once the user has unlocked their vault locally, Dashlane uses the DeviceKey for server authentication, enabling synchronization and cloud-based features without requiring further user input.

This design provides secure, password-independent authentication between user devices and Dashlane's infrastructure, maintaining zero-knowledge architectures while simplifying the user experience.

3.4 Communication Security

All communications between the **Dashlane application** and **Dashlane servers** are protected by modern transport security protocols. Dashlane enforces **HTTPS/TLS** across all endpoints, with the `dashlane.com` domain **HSTS-preloaded** to prevent protocol downgrades on any subdomain.

- **TLS best practices:** Cipher suites and protocol configurations are regularly updated to align with current industry standards and security recommendations.
- **Beyond transport security:** While HTTPS/TLS provides a strong baseline protection, Dashlane's design ensures that **data confidentiality does not depend on the transport layer**. Because vault data is **end-to-end encrypted**, even if the communication channel were compromised, an attacker would gain access only to encrypted ciphertext.

This layered approach to communication security preserves zero-knowledge guarantees while maintaining compatibility with enterprise security frameworks and modern web infrastructure.

3.5 Confidential Computing & Secure Enclaves

Enterprises require solutions that deliver both **security and operational transparency** when integrating with third-party systems, such as identity providers, and managing sensitive cryptographic operations. Dashlane uses **confidential computing** to meet these B2B requirements, ensuring that even when customers rely on Dashlane-hosted components, such as our SSO integration, **data remains fully protected from both Dashlane and external parties**.

Confidential computing establishes a **trusted execution environment** (TEE/secure enclave) that isolates data and code at runtime. In Dashlane's implementation, this ensures that encryption keys, SSO tokens, and authentication flows are processed in **cloud secure enclaves** that even cloud administrators cannot access.

3.5.1 Secure Enclaves at Dashlane

Dashlane leverages **AWS Nitro Enclaves** to isolate cryptographic materials and protect sensitive operations from exposure, even if the host infrastructure were compromised.

- **Isolated execution:** All encryption and decryption operations occur inside secure enclaves, separate from the host operating system and administrative layers.
- **Attestable integrity:** Each enclave generates a verifiable attestation, ensuring that only trusted code is executed and that no unauthorized changes can occur.
- **Confidential processing:** Enclave data and memory remain inaccessible to Dashlane personnel or any external process, reinforcing the zero-knowledge architecture.
- **Lifecycle assurance:** Enclave creation, execution, and destruction are verifiable, providing strong guarantees of code integrity throughout the operational lifecycle.

Our security model leverages **AWS Nitro Enclaves** as the foundational isolation layer for our confidential computing environment. While Dashlane operates on strict **Zero-Knowledge** principles, the integrity of our processing relies on the **AWS Nitro System**. The Nitro Hypervisor creates a logically isolated compute partition, which effectively eliminates administrative access and persistent storage. Our reliance is specifically on AWS's architectural claim that the Nitro Card and Hypervisor maintain a 'no-operator-access' environment. Crucially, our threat model acknowledges that our security guarantees are tethered to this logical isolation rather than relying on a Trusted Execution Environment (TEE) independent of the cloud provider's firmware, even as we use **Cryptographic Attestation** to verify our code's integrity.

This architecture ensures that even if infrastructure is compromised, **cryptographic keys and user data remain secure** through an isolated workload and attested trust.

To block unauthorized access, the enclave requires authentication via its attestation to access cryptographic material. This validation is managed by a **Key Management Service (KMS)**.

Upon authentication, each enclave instance performs two retrievals:

1. **Key Management Service (KMS):** Retrieves an **Enclave Local Key (ELKey)**
2. **Internal secret manager:** Retrieves an **Enclave Unseal Key (EUKey)**

These two keys are then combined and used to decrypt the Enclave Encryption Key (EEKey) as described in the following schema:

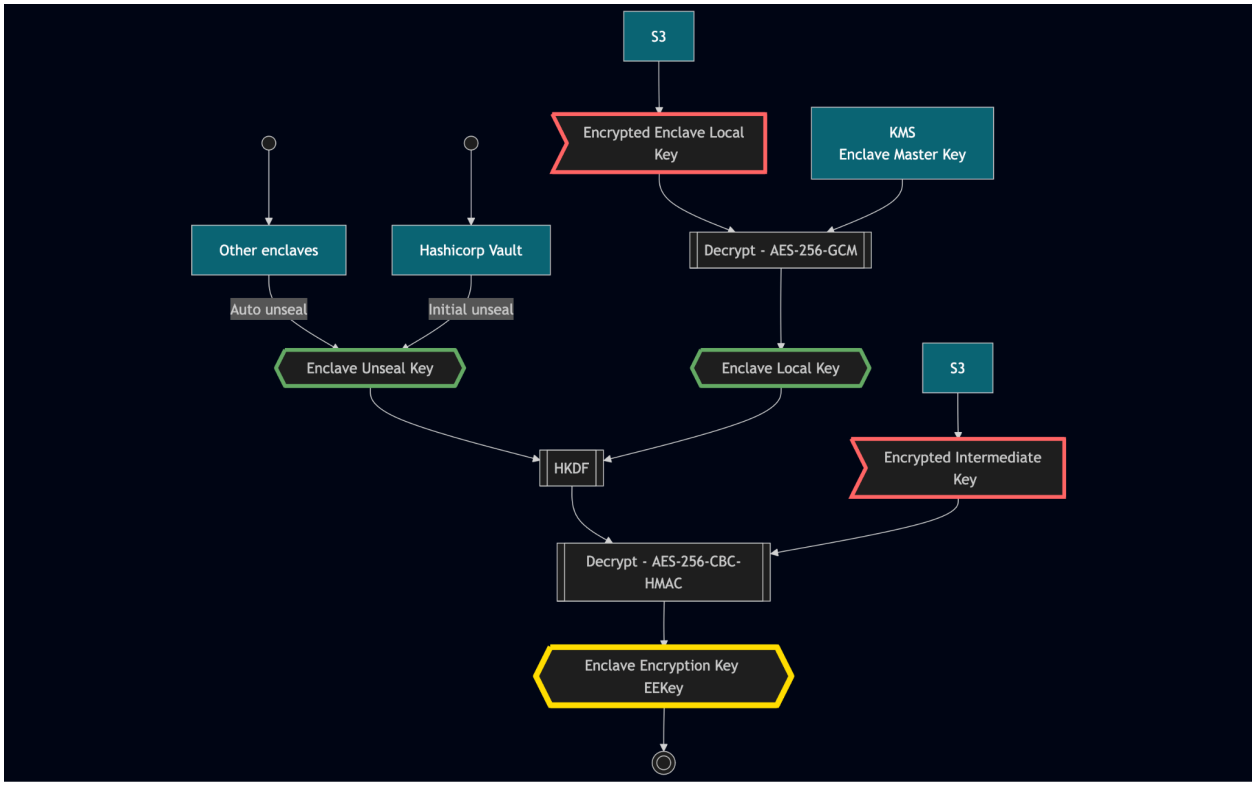


Figure: How keys are used to decrypt the Enclave Encryption Key

3.5.2 Enclave Workflows

The enclave operates through a series of secure workflows that govern initialization, storage, and all other operations related to the features that leverage them.

Cryptographic materials

Key Name	Key Symbol	Description
Enclave Master Key	EMKey	Generated and stored within the KMS to encrypt and decrypt the Enclave Local Key (ELKey)
Enclave Local Key	ELKey	Generated within the KMS during the first enclave bootstrap and sent to the enclave to derive the

		Enclave Encryption Key (EEKey)
Enclave Unseal Key	EUKey	Generated by the deployment process at first bootstrap and sent to the enclave to derive the EEKey
Enclave Encryption Key	EEKey	Derived from ELKey ⊕ EUKey to encrypt the Service Provider Master Key (SPMasterKey)

Enclave Initialization

During initial deployment, the enclave requests the **Enclave Master Key (EMKey)** from the KMS, which grants access only to verified enclaves via attestation. The KMS generates an **Enclave Local Key (ELKey)** and returns two encrypted versions: one encrypted by EMKey and the other by an ephemeral enclave public key. The enclave stores only the encrypted ELKey, ensuring it is never available in plaintext outside the secure environment. When the enclave restarts, the KMS decrypts the ELKey using EMKey, reestablishing a secure state.

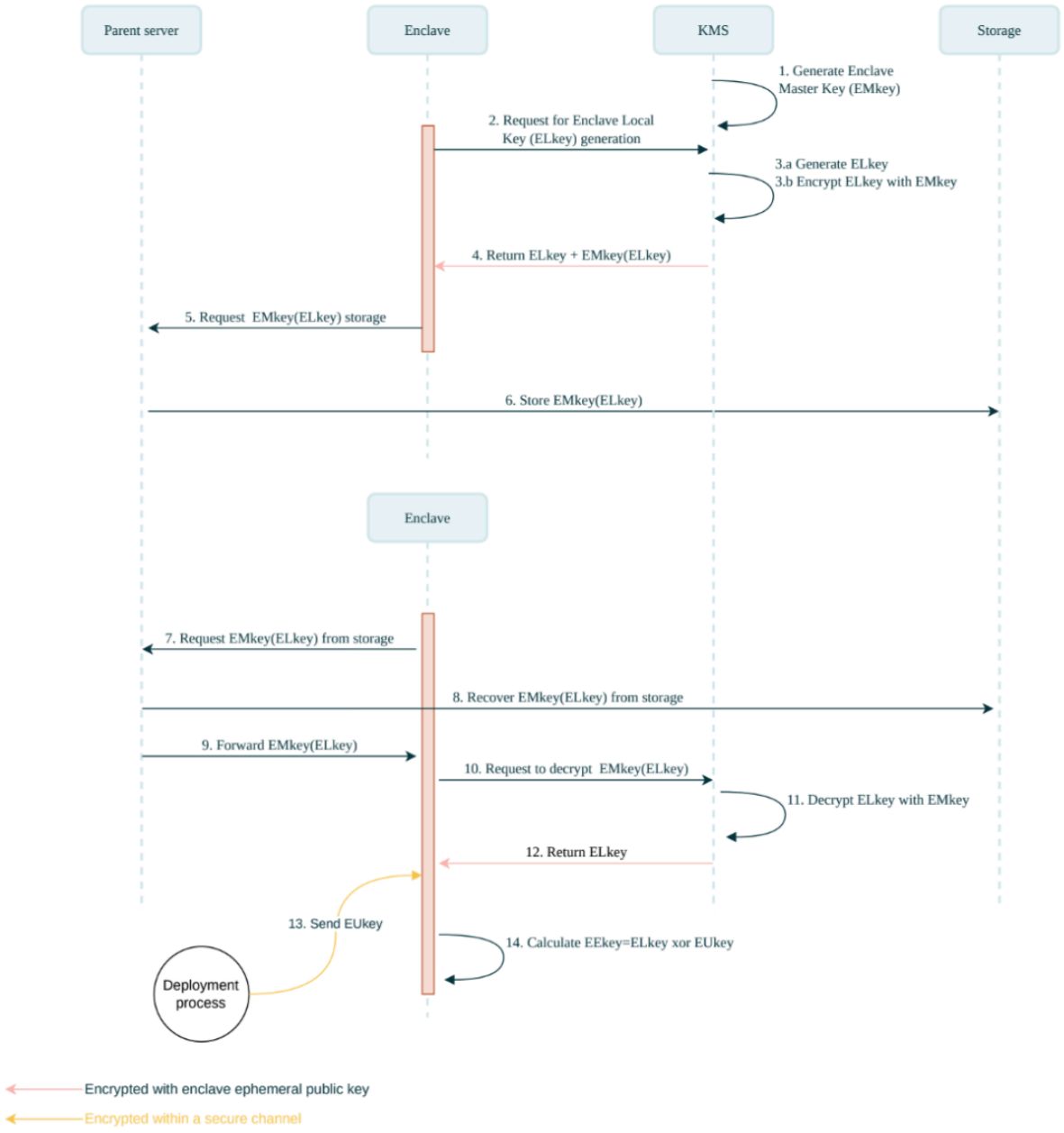


Figure: Enclave initialization

Storage and Key Management

Because enclaves lack persistent storage, all data leaving the enclave is encrypted before transmission as described in the following sequence diagram:

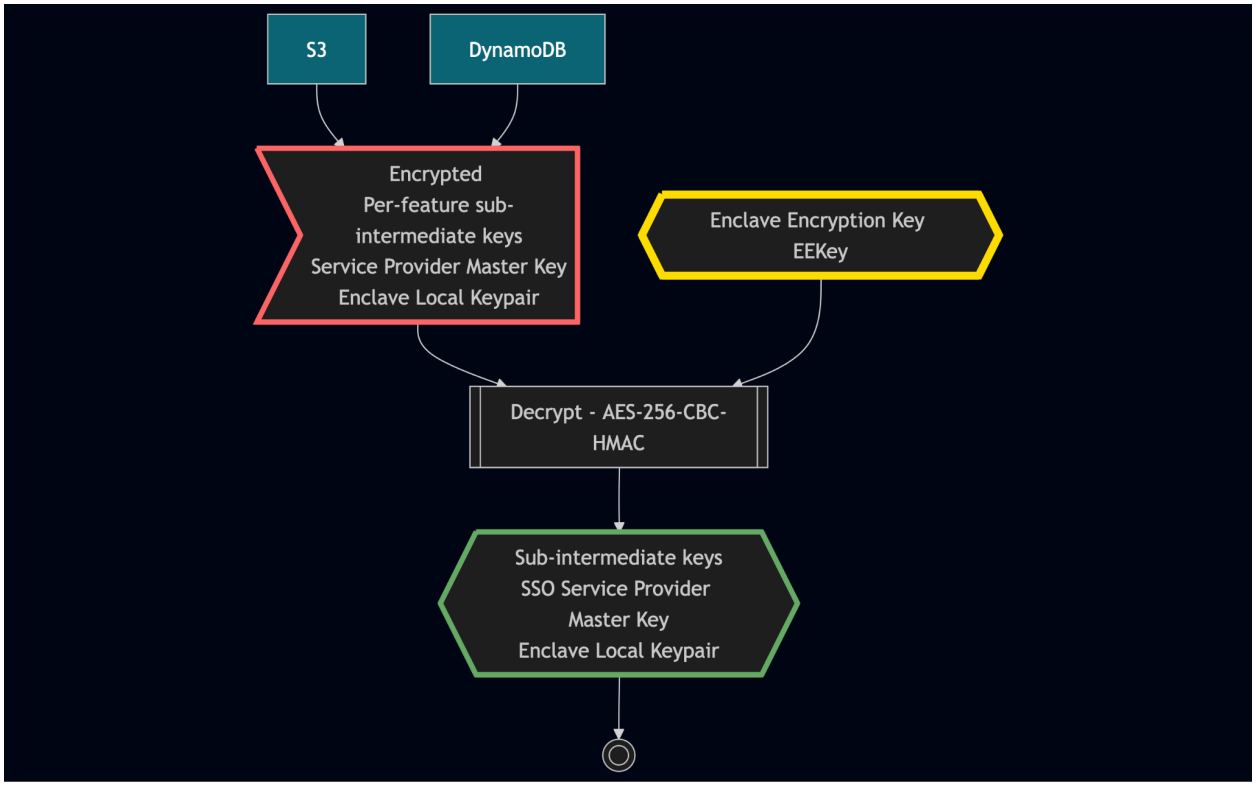


Figure: Data encryption before transmission

The Enclave Encryption Key encrypts the set of encryption keys for features operated by enclaves, as:

- SSO Service Provider keys: An encryption key is generated per team to encrypt the User ServiceProvider Keys at rest (outside of the enclave); these are the keys sent back to the clients to decrypt their vault.
- Sub-intermediate Keys: Encryption keys generated for specific features operated by the enclave, such as activity logs.

3.5.3 Secure Channels

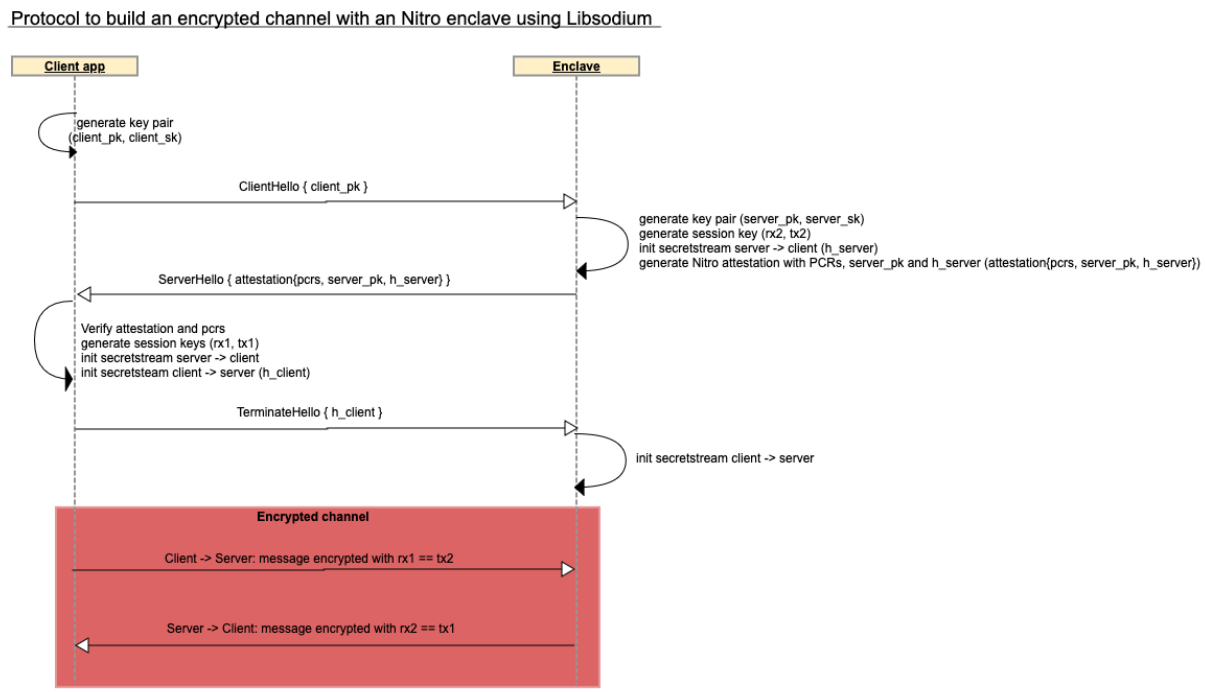


Figure: Protocol to build an encrypted channel using Libsodium

Secure channels are how clients securely communicate with enclaves. They are end-to-end encrypted channels between a client and enclaves, preventing a man-in-the-middle from eavesdropping on those communications.

The client generates ephemeral key pairs and requests the enclave to create a secure channel. Upon being requested to mount a channel, the enclave generates its own key pairs and embeds the public key within a cryptographically signed attestation.

Nitro attestations are a set of properties of the runtime environment issuing the attestation, signed by the Nitro Hypervisor underlying the workload. Those properties are sealed as measurements called Platform Configuration Registers (PCRs)—basically hashes—into the attestation, to be verified by clients when checking the attestation.

Once the Dashlane application has verified the attestation and agreed on the PCRs within, the client is guaranteed that the public key embedded into the attestation is the one of a genuine enclave. Dashlane applications base their verification on two PCRs:

- PCR3: A contiguous measurement of the IAM role assigned to the parent instance; ensures that the attestation process succeeds only when the parent instance has the correct IAM role.
- PCR8: A measure of the signing certificate specified for the enclave image file; ensures that the attestation process succeeds only when the enclave was booted from an enclave image file signed by a specific certificate.

Therefore, the client can proceed with the key exchange and generate the cryptographic material to encrypt communication to the secure enclave.

Secure channels are based on the Libsodium library. As stated in the documentation of the library, the following primitives are used:

- X25519 and BLAKE2B-512 for the key exchange
- XChaCha20-Poly1305 for secretstreams

4. Credential Security in Detail

Summary

Dashlane protects credentials through a **layered, zero-knowledge architecture** that safeguards data across every stage of its lifecycle, from creation to recovery. Each credential is encrypted locally with AES-256-CBC + HMAC-SHA256 before synchronization, ensuring no plaintext data ever leaves the user's device.

- **Authentication and device management:** Secure flows verify device trust.
- **Multi-factor authentication (MFA):** MFA adds an additional layer of protection using TOTP at each login or when registering a new device, and enterprise policy controls reduce account takeover risks.
- **Account recovery:** Both **admin-assisted** and **self-service** recovery mechanisms maintain zero-knowledge guarantees.
- **Secure sharing:** Dashlane enables encrypted, policy-controlled collaboration across users and groups.
- **Password Health, Password Generator, and Dark Web Monitoring:** Continuous hygiene tools detect weak, reused, or compromised passwords, generate strong replacements, and alert users of exposed credentials

---

4.1 Authentication Flows and Vault Encryption Lifecycle

4.1.1 Registration

The initial registration for a user follows the flow described in the following figure.
As a reminder, neither the User Master Password nor the Machine-Generated Master Password are ever used to perform server authentication, and the only keys stored on our

servers are the User Device Keys.

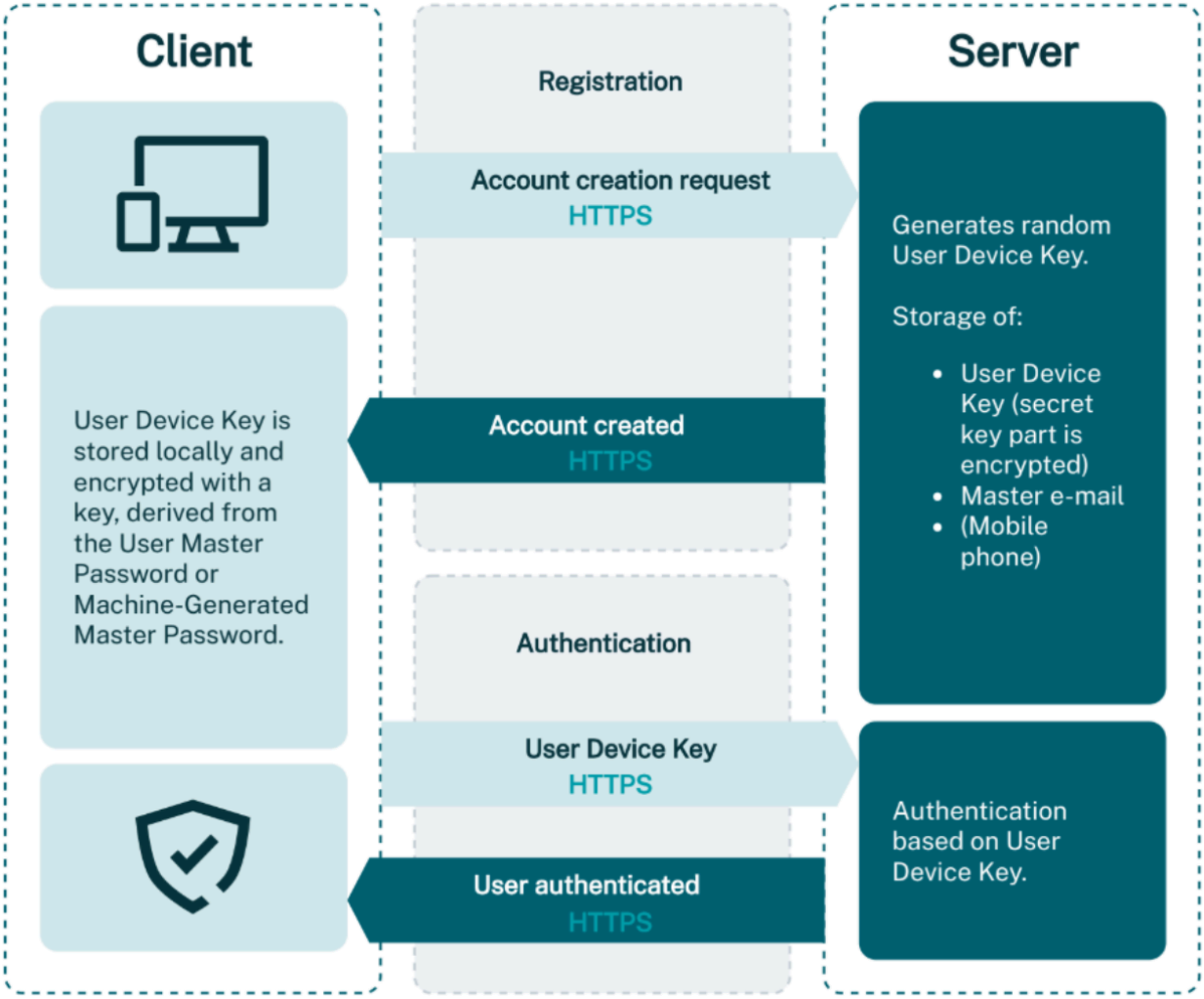


Figure: Authentication Flow during registration

Upon registration, the server generates a **User Device Key** and sends it back to the client application. The **User Device Key** consists of two parts:

- The **Device Access Key**: A public part acting as a key identifier (8 random bytes hexa encoded string)
- The **Device Secret Key**: A secret part (32 random bytes hexa encoded string)

When requesting the server, the application produces a signature with HMAC-SHA256 over the body and parts of the request headers and with the **Device Secret Key** as the signing key. The signature is set in the headers request along with the **Device Access Key**. This way, the **Device Secret Key** isn't exposed to the network after the registration workflow.

4.1.2 Adding a New Device for Master Password-based Users

When a user adds an additional device, Dashlane needs to ensure that the person adding the device is indeed the legitimate owner of the account. This is to provide additional protection in the event UserMP has been compromised and an attacker who does not have access to the user's already-enabled device is trying to access the account from another device.

When a user attempts to connect to a Dashlane account on a device that has not yet been authorized for that account, Dashlane generates a One-Time Password (OTP, or token) that is sent to the user to the email address used to create the Dashlane account initially.

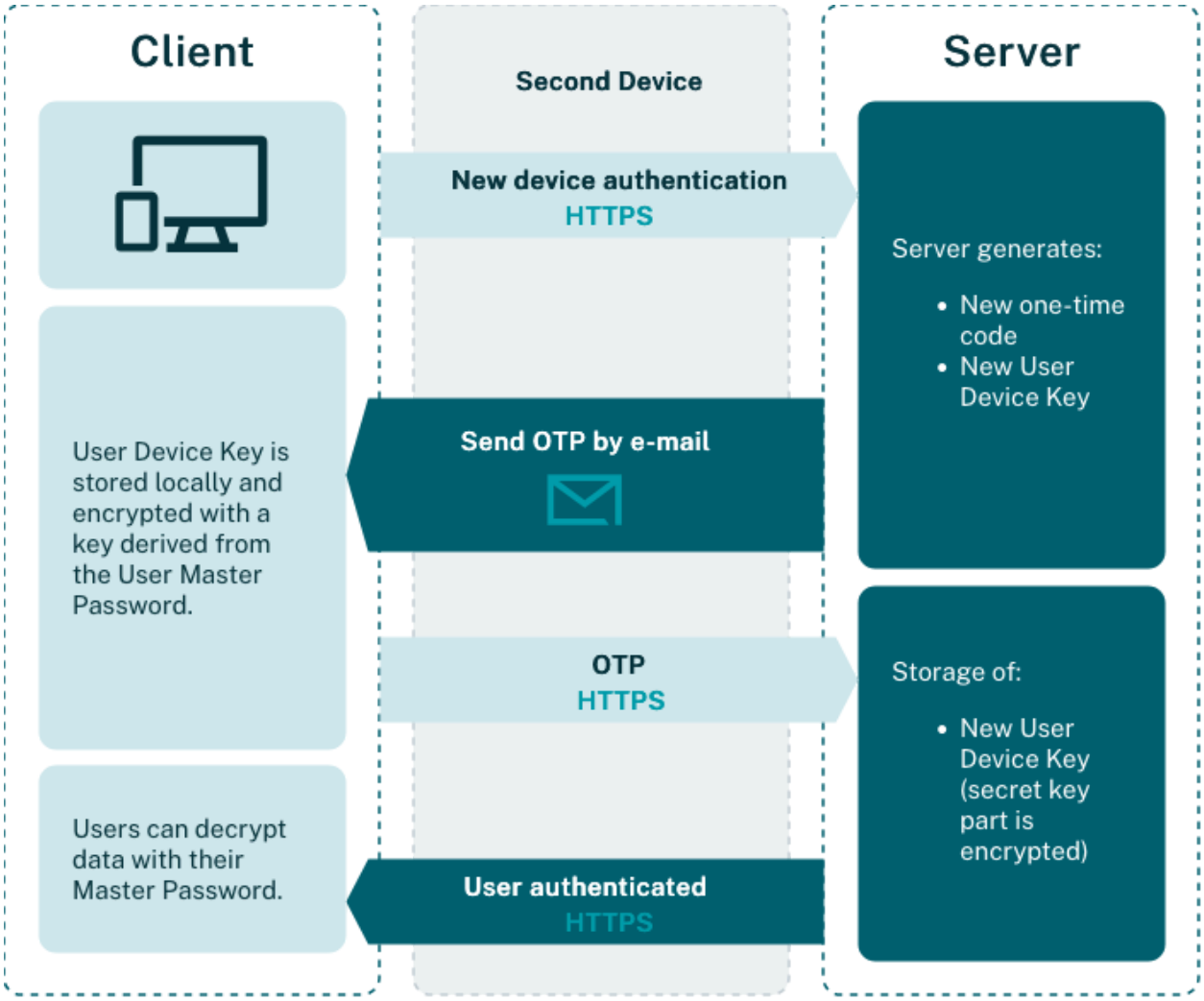


Figure: Authentication when adding a new device

Upon validation of the OTP by the server after the user enters it into the application, the device is provided with a User Device Key to authenticate to the server (cf 4.1.1 Registration).

Once the device is authenticated to our server, the device can download the user's vault in its encrypted form (cf 3.2 Encryption Model: Secrets and Protections). Then, the user can decrypt their vault by providing their Master Password.

This workflow enforces a multi-factor way of accessing the plaintext version of a user's data.

4.1.3 Adding a New Device for Passwordless Users

When a passwordless user adds a new device, they use an existing, logged-in device to complete the setup. Depending on the trusted device type, the setup can be finalized either through a **QR code scan** (proximity method) or a **security challenge** (remote method). In both cases, the goal is to securely transfer the **Machine-Generated Master Password (MachineGeneratedMP)** from an already trusted device to a new one.

This key exchange is based on **Elliptic Curve Cryptography (Curve25519)**, ensuring strong security and protection against interception.

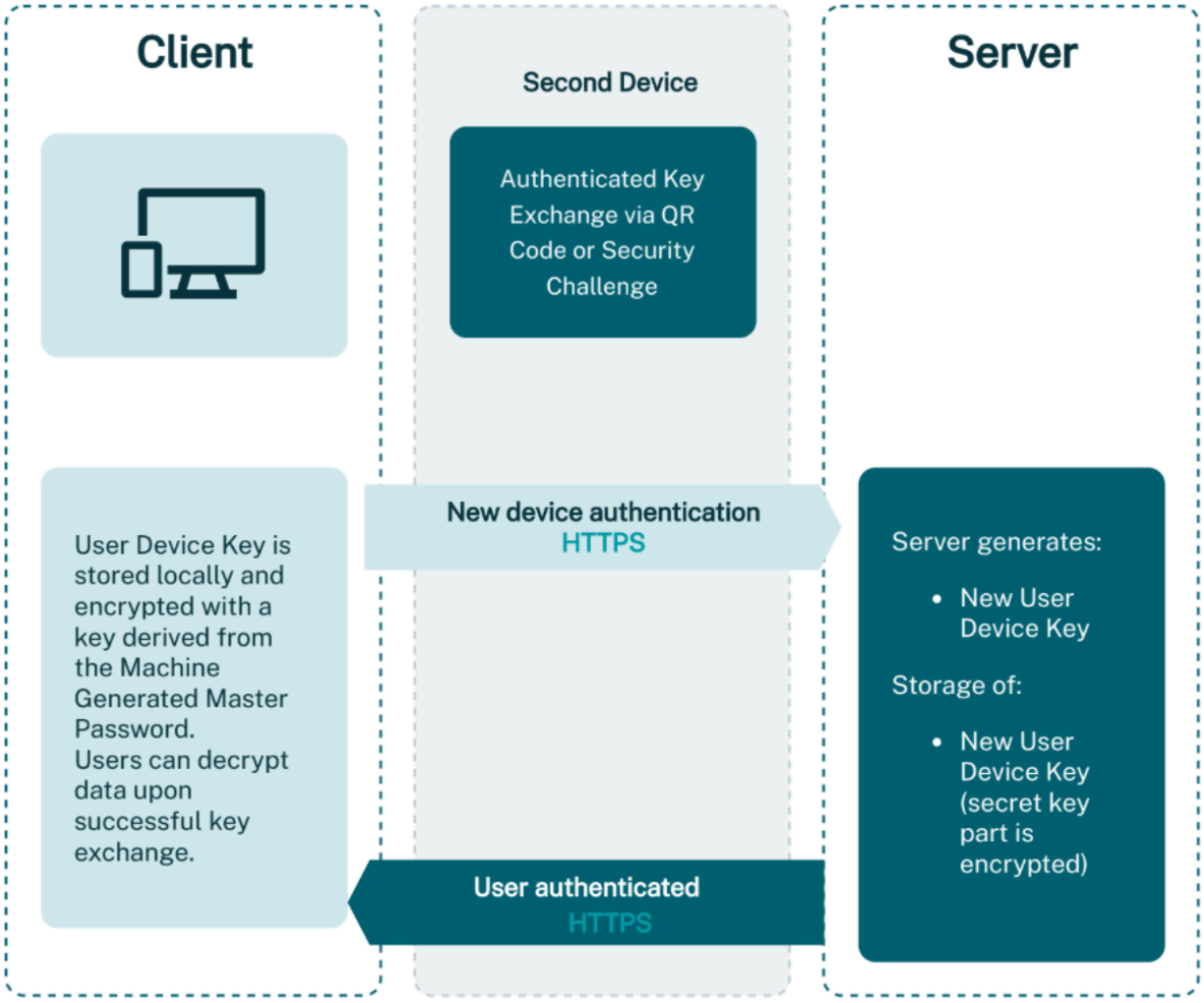


Figure: Adding a new device for passwordless users

Proximity Transfer with QR Code Scan

If the user has a logged-in mobile device, a QR code can be used to authorize the new device. When the user enters their email on the untrusted device, it generates an **X25519 key pair** and displays its public key as a QR code. The trusted device scans this code, completing the key exchange. Both devices then derive the same **shared secret**, which becomes a cryptographic key used to encrypt and transmit the MachineGeneratedMP. Once received, the new device decrypts the vault locally and becomes authorized.

Exchange via Server with Visual Verification

If a user cannot perform a QR scan (for example, if they lack a camera-enabled trusted device), the exchange occurs through Dashlane's servers using a **Short Authenticated**

String (SAS) verification. This process authenticates the key exchange and ensures integrity:

- From the shared secret, Dashlane derives a key used to select **five random words** from the EFF large word list.
- Both devices display these words; if the lists match, the exchange is authentic. The user is asked to confirm by entering a missing word on the trusted device.
- A **Public Key Commitment** mechanism adds further protection. The untrusted device first sends a hash of its X25519 public key before revealing the key itself, preventing an attacker from manipulating the exchange in real time.

Upon successful verification, the **MachineGeneratedMP** is securely transmitted to the new device, allowing it to decrypt the vault locally. This design provides strong protection against **man-in-the-middle** attacks while maintaining a seamless user experience.

4.1.4 Adding a New Device With a Security Key (FIDO2)

When the account is protected with a security key, registering a new device is based on a WebAuthn flow with the passkey stored in the Security Key. Upon validation of the WebAuthn flow, the device is provided with a **User Device Key**.

Then, the device can download the encrypted vault from the server. The decryption is based on the PRF extension of the WebAuthn standard, letting the Security Key issue a symmetric key from the passkey to decrypt the MachineGeneratedMP of the account (downloaded along with the encrypted vault).

4.2 Multi-Factor Authentication (MFA)

For Master Password-based accounts, Dashlane can strengthen account protection through **multi-factor authentication (MFA)**, which adds an additional layer of security beyond the Master Password. MFA ensures that even if one factor is compromised, unauthorized access to the vault remains highly unlikely.

By combining multiple authentication factors with local encryption, Dashlane ensures that access to user vaults and administrative functions remains secure, even in the event of stolen credentials or compromised devices.

Users can enable MFA at any time from the Dashlane web extension or mobile app. Dashlane offers two levels of MFA.

4.2.1 MFA for New Device Setup

When connecting a new device, instead of sending a one-time token via email, Dashlane can prompt for a **one-time code** generated by the linked 2FA authenticator app. After the code is validated, a new **Device Key** is securely registered for that device, maintaining zero-knowledge architectures.

4.2.2 MFA at Each Login

Users can link their Dashlane account to an authenticator app (for example, Google Authenticator or Microsoft Authenticator). Each login requires both the Master Password and a one-time verification code.

When activated, both the locally stored data and the synchronized vault data are **re-encrypted** with a new symmetric AES-256 key derived from the combination of the user's primary credential (Master Password) and a **User Secondary Key** generated on Dashlane's servers: The output of the Argon password-hashing function is XOR-ed with the high entropy key generated on Dashlane's servers.

The User Secondary Key is transmitted to the client only after successful MFA verification, ensuring that both factors are required to decrypt the vault.

4.2.3 Enterprise MFA Controls

For Dashlane business accounts, administrators can enforce MFA through policy settings in the **Admin Console**.

4.3 Account Recovery

Dashlane provides secure, zero-knowledge account recovery options to ensure that users can regain access to their vaults without compromising encryption or privacy. Different complementary methods are available:

- **Account Recovery Key (ARK)** and **Biometric Recovery on Mobile** for all users
- **Admin-Assisted Account Recovery** for business users

4.3.1 Account Recovery Key (ARK)

The **Account Recovery Key (ARK)** provides a self-service fallback for individual users and supports zero-knowledge design principles. The ARK is a unique, single-use, 28-character alphanumeric key generated by Dashlane's password generator during setup.

- During configuration, the ARK is confirmed and saved by the user. A derivative key encrypts the Master Password locally using AES-256 and stores only the encrypted value on Dashlane servers.
- The ARK can be disabled at any time from the user's security settings, invalidating the existing key.
- In the event the user forgets their Master Password or loses access to all devices, they can initiate recovery by verifying their identity via email code or 2FA token. After successful verification, the server releases the encrypted Master Password, which is decrypted locally using the ARK. The user is then prompted to create a new Master Password.

Once the process is complete, the old ARK becomes invalid. A new key must be generated and confirmed in the user's security settings. The ARK is also automatically disabled when the Master Password is changed or a user switches from **Master Password login to SSO, passwordless, or a security key**.

These recovery mechanisms combine usability and security, empowering users to regain access independently while maintaining Dashlane's zero-knowledge guarantees.

4.3.2 Biometric Recovery on Mobile

Biometric Recovery provides a device-specific recovery mechanism for mobile users, strictly adhering to zero-knowledge design principles. This feature leverages hardware-backed security capabilities to allow Master Password replacement without server intervention. The implementation varies between platforms to leverage native security features while maintaining equivalent security guarantees.

On Android, when the user enables Biometric Recovery, a copy of the Local Key, a 64-byte randomly generated symmetric key that encrypts the vault data, is encrypted and stored locally. The encryption uses a cryptographic key generated within the Android Keystore System, which is backed by the device's Trusted Execution Environment (TEE) or Secure Element. Access to this key requires successful biometric authentication (fingerprint or face recognition).

On iOS, a copy of the Master Password is encrypted and stored in the iOS Keychain with biometric access control, leveraging the Secure Enclave when available. The Master Password is used to decrypt the Local Key. In both implementations, cryptographic secrets never reach Dashlane servers during setup or recovery.

Recovery Process

When a user forgets their Master Password, they initiate recovery on the device where the feature was enabled:

- The user authenticates via biometric prompt (fingerprint, face recognition, Touch ID, or Face ID). Upon successful authentication, the secure storage releases the cryptographic secret:
    - Android: The Local Key is decrypted into memory.
    - iOS: The Master Password is retrieved and used to decrypt the Local Key.
- The user creates a new Master Password, which re-encrypts the Local Key.
- The updated structure synchronizes to Dashlane servers, enabling access from other devices with the new Master Password.

Throughout this process, raw cryptographic secrets never leave the device, preserving zero-knowledge architecture.

Security Safeguards

Biometric Recovery can be disabled at any time, immediately purging encrypted material from secure storage. Both platforms protect against unauthorized access when device biometrics change:

- Android: The operating system automatically invalidates Keystore keys when new biometric templates are enrolled (for example, a new fingerprint is added), immediately disabling recovery.
- iOS: The application monitors biometric enrollment changes via the Local Authentication framework's domain state hash. When changes are detected (for example, new Face ID enrollment), the feature automatically deactivates and removes the stored Master Password.

In both cases, users must explicitly reenable and reauthenticate to restore functionality after biometric changes, ensuring new individuals cannot gain unauthorized vault access.

4.3.3 Admin-Assisted Account Recovery for Business

Admin-assisted account recovery allows **Dashlane business** users who log in with a Master Password to securely reset it while preserving zero-knowledge guarantees. Dashlane’s

patented recovery process ensures that Master Passwords are never stored on servers or transmitted in any form. The zero-knowledge architecture ensures that account recovery can only be initiated from a device on which the user has previously logged in.

When admin-assisted recovery is enabled and the user first logs in, the user's **Local Key** is encrypted with a locally-created **User Recovery Key**. The resulting **Encrypted Local Key** is stored in the device's local storage.

The **User Recovery Key** is then encrypted with a **Server Recovery Key** (known only by the server and the user's client devices), producing the **Server Protected Recovery Key**. The client then encrypts this **Server Protected Recovery Key** with each admin's **Public Key** to generate the **Server and Admin Protected Recovery Keys**. If the team has N admins, there will be N of these keys.

Only the **Server and Admin Protected Recovery Keys** are sent to the server. Crucially, the **User Recovery Key** and the **Server Protected Recovery Key** (the intermediate key) never leave the user device. Once the server has received the encrypted keys, admin-assisted account recovery is fully enabled for the user.

When the user initiates a recovery flow, they must first verify their identity and define a new **Master Password**. Upon definition of the new **Master Password**, a public/private key pair is created on the user device. The private key is encrypted with the new **Master Password** and stored locally, while the public key is sent to the server along with the recovery request.

The admin then **verifies** the request, acting as a trusted third party to confirm the action. Once approved, the admin's **Private Key** (associated with the public key used at activation) is used to decrypt its **Server and Admin Protected Recovery Keys**. The result, the **Server Protected Recovery Key**, is then encrypted locally with the user's new Public Key to create a **Server and User Protected Recovery Key**, which is returned to the server. This process ensures that the **Server Protected Recovery Key** is never sent to the server in an unencrypted form, preserving Dashlane's zero-knowledge guarantee over the **User Recovery Key**.

After the admin's approval and cryptographic operation, the user can authenticate with the new **Master Password**. This **Master Password** is used to retrieve the user's Private Key, which then decrypts the **Server and User Protected Recovery Key**. The result is the **Server Protected Recovery Key**, which is then decrypted using the **Server Recovery Key** to yield the original **User Recovery Key**. This final key is used to decrypt the **Encrypted Local Key**, recovering the **Local Key**. This step allows the extension to access the user's local vault data. Finally, the vault is re-encrypted with the new **Master Password** and synced with the server.

After recovery, all other devices must be reregistered with Dashlane to maintain secure synchronization. Admin-assisted recovery can be enabled or disabled by administrators from the **Admin Console**.

> **Privacy note:** Because the admin serves as a trusted verifier, if they have access to both the user's device and account email, they could theoretically trigger a recovery and access the vault. Organizations should restrict such privileges to trusted personnel only.

4.4 Secure Sharing

Dashlane enables users to share credentials and Secure Notes, maintaining its **zero-knowledge architecture** throughout the process. Shared data is always encrypted end-to-end, ensuring that neither Dashlane nor any third party can access the content.

- **Zero-knowledge architecture:** Dashlane servers handle only encrypted data; the decryption keys never leave user-controlled environments.
- **Granular access control:** Users can grant different levels of rights when sharing credentials.
- **Traceability:** For business users, every sharing action is logged in **Activity Logs** for enterprise visibility and compliance.

4.4.1 Sharing Mechanism

Dashlane's sharing system relies on **asymmetric encryption**. Upon account creation, each user generates a unique pair of **RSA 2048-bit public and private keys**:

- The **public key** is stored on Dashlane's servers and used to encrypt shared data for the intended recipient.
- The **private key** remains encrypted in the user's vault and is never shared or transmitted.

When a user (Alice) shares an item with another user (Bob):

1. Dashlane retrieves Bob's public key.
2. Alice generates a random 256-bit AES key (the **ObjectKey**) to encrypt the item.
3. The ObjectKey is encrypted using Bob's public key and sent to Bob through Dashlane's servers.
4. Bob decrypts the ObjectKey with his private key, then decrypts the shared item locally.

4.4.2 Group and Collection Sharing

Dashlane also supports sharing with **groups** or **collections** **of items**, using an additional layer of symmetric encryption:

- A **GroupKey** (AES-256) is generated for each group or collection.
- Each member's public key encrypts the GroupKey, ensuring that only authorized users can access shared credentials.
- The GroupKey's corresponding **RSA private key** is used to sign and manage shared items within that group, providing auditability and integrity.

4.4.3 Link-based Sharing

Dashlane enables users to share credentials with anyone, including recipients who do not have a Dashlane account or the Dashlane extension installed. This feature preserves Dashlane's zero-knowledge architecture by ensuring encryption keys are never transmitted to Dashlane servers.

Overview

Link-based sharing allows users to generate a time-limited, access-controlled URL that displays credential details (username, password, and TOTP codes if configured) on a secure landing page. The link automatically expires based on configurable conditions: time elapsed or number of views. Once any expiration condition is met, the link becomes permanently invalid.

Encryption and Key Management

When a user initiates link-based sharing:

- The Dashlane application generates K1, a 64-byte cryptographic key split into two 32-byte components: one for AES-256-CBC encryption and one for HMAC-SHA256 authentication.
- The credential is encrypted locally using this key pair before being transmitted to Dashlane's servers.
- Dashlane's server generates a unique itemUuid and stores the encrypted payload on AWS S3.
- The application constructs a sharing URL in the format: https://share.dashlane.com/{itemUuid}#{K1}

For instance:

```
https://share.dashlane.com/0c90af73-d077-43ca-9fc7-32a74cfa79cc#6f642
db043b3bf775294772516785686ba93371a5cd592b600e1b857f4aff17d1268358012
f8b523a4553abe20a71306122ff8366e4bcba5c9fc80415904b31a
```

where :

Component	Value	Purpose
Domain	share.dashlane.com	The sharing landing page
Path	0c90af73-d077-43ca-9fc7-32a74cfa79cc	Item UUID (server-generated identifier)
Fragment #	6f642db0...04b31a	**Encryption key (64 bytes, hex-encoded)**

The URL uses a fragment identifier (#) to separate the itemUuid from K1. By design, URL fragments are never transmitted to servers, ensuring that Dashlane never sees the decryption key, even in transit.
This preserves zero-knowledge guarantees throughout the sharing process.

Link Access and Decryption

When a recipient opens the link:

- The landing page extracts the itemUuid from the URL path and K1 from the fragment.
- A Cloudflare Turnstile challenge is completed to prevent automated enumeration attacks.
- The landing page requests the encrypted payload from Dashlane's servers using only the itemUuid.
- Dashlane's server validates expiration conditions (time and remaining views) and decrements the view counter atomically.
- If valid, the encrypted payload is returned to the landing page.
- The landing page decrypts the credential locally using K1 and displays the contents to the recipient.

Expiration and Automatic Cleanup

Link-based shares enforce strict expiration controls to minimize exposure:

- Time-based expiration: Links expire after 24h.
- View-based expiration: Links can be only accessed one time.

First-condition-met: the link expires as soon as any single condition is satisfied.

Server-side, Dashlane periodically scans for expired items and:

- Deletes the encrypted payload and shared item entry from storage
- Generates an audit log entry recording the expiration event

When a user's account is deleted, all link-based shares created by that user are immediately destroyed, regardless of their configured expiration. The encrypted payloads are removed from S3, and corresponding audit logs are generated.

Enterprise Controls

Link-based sharing is governed by team settings in the Admin Console:

- Sharing functionality dependency: Link-based sharing requires the "Secure sharing for logins, Secure Notes and Secrets" policy to be enabled.
- External sharing dependency: Link-based sharing requires the "Allow sharing outside company" policy to be enabled.
- Link sharing policy: Administrators can enable or disable link-based sharing for their organization.

These controls ensure that organizations can restrict link-based sharing based on their security and compliance requirements.

Auditability

For enterprise customers, link-based sharing generates Activity Log entries for full traceability:

- Link created: Records the user who created the link, the item identifier, expiration settings, and originating IP address.
- Link accessed: Records each access attempt, including the recipient's IP address and timestamp.
- Link expired: Records when a link expires, whether due to time, view limits, or manual deletion.

These logs are available to administrators through the Admin Console and support compliance and incident response workflows.

4.5 Password Health Score

The **Password Health score** helps users and organizations measure and improve the overall strength of their stored credentials. Each password in the vault is evaluated based on key criteria—**strength, reuse, and exposure in known breaches**—to produce an overall score.

- **Reused or weak passwords** reduce the score and generate actionable alerts.
- **Compromised passwords** detected via Dark Web Monitoring are flagged for immediate update.

Admins and individual users can access detailed reports through their **Security Dashboards**, which include trend tracking and prioritized remediation recommendations. By quantifying credential hygiene, the Password Health score provides a clear metric for improving enterprise password security and reducing exposure to credential-based attacks.

4.6 Password Generator

Dashlane's built-in **Password Generator** helps users create strong, unique credentials and **passphrases** that balance complexity with memorability. Users can customize generation settings to meet personal or enterprise security policies:

- **Password options:** Choose length (4–40 characters), include or exclude letters, digits, symbols, or similar characters, and ensure compliance with site-specific rules.
- **Passphrase mode:** Generate easy-to-remember yet highly secure phrases composed of random words, separated by spaces or special characters. Passphrases typically offer higher entropy while maintaining usability. Length can be defined between 4 and 8 words.
- **Automatic integration:** The generator is integrated into the autofill experience and the web extension, enabling quick credential generation and replacement during sign-up or password updates.

By promoting the use of high-entropy passwords and customizable passphrases, Dashlane's Password Generator significantly reduces password reuse and strengthens overall credential hygiene across individual and enterprise accounts.

4.7 Dark Web Monitoring

4.7.1 Dark Web Monitoring for Vault Credentials

This feature alerts Dashlane users if their monitored email addresses appear in a known data breach. To determine if an email is compromised, Dashlane checks it against databases compiled from verified third-party breach sources.

Upon subscription to Dark Web Monitoring alerts, the user will receive an email with a link. The link activates the monitoring. The user will be redirected to a page on the website (`www.dashlane.com`) with a summary of the known data leaks, if there are any, for the given email. The summary includes the following information: affected company/domain, date of the breach, data type affected (credit cards, emails, passwords, identity, etc).

Whenever new breaches occur, the user will be notified in the Dashlane application if one of the monitored emails appeared in the breach.

Users can monitor up to five different email addresses.

4.7.2 Dark Web Monitoring for Master Password

This feature alerts Dashlane users if their **Master Password** appears in a known data breach. To determine if a password is compromised, Dashlane checks it against databases compiled from verified third-party breach sources.

All breach data collected through partner APIs is **hashed using the Argon2d v1.3 function** before storage on Dashlane servers. When a user enters their Master Password in the mobile or web app, the client applies the **Argon2d algorithm with a unique salt** specific to this feature (different from the salt used for vault encryption) to produce a 32-byte hash:

Algorithm	Iterations	Memory	Parallelism	Threads	Hash length
Argon2d v1.3	3	32,768 KB	2	2	32 bytes

To preserve **zero-knowledge architecture**, Dashlane uses a **k-anonymity** model: The full hash never leaves the user's device. Only the **first three bytes** of the hash are sent to Dashlane's servers to check for potential matches in breach datasets. If matches are found, the server returns a list of candidate hashes for local comparison.

The Dashlane application then compares the full hash locally; if it matches one from the breach list, the user is immediately alerted that their Master Password may have been compromised and should be changed. This ensures that **no plaintext passwords or full hashes** ever leave the device, maintaining privacy even during monitoring.

This privacy-preserving process enables Dashlane to provide real-time breach alerts and maintain its **zero-knowledge promise**, ensuring that no one, including Dashlane, can ever access or reconstruct a user's Master Password.

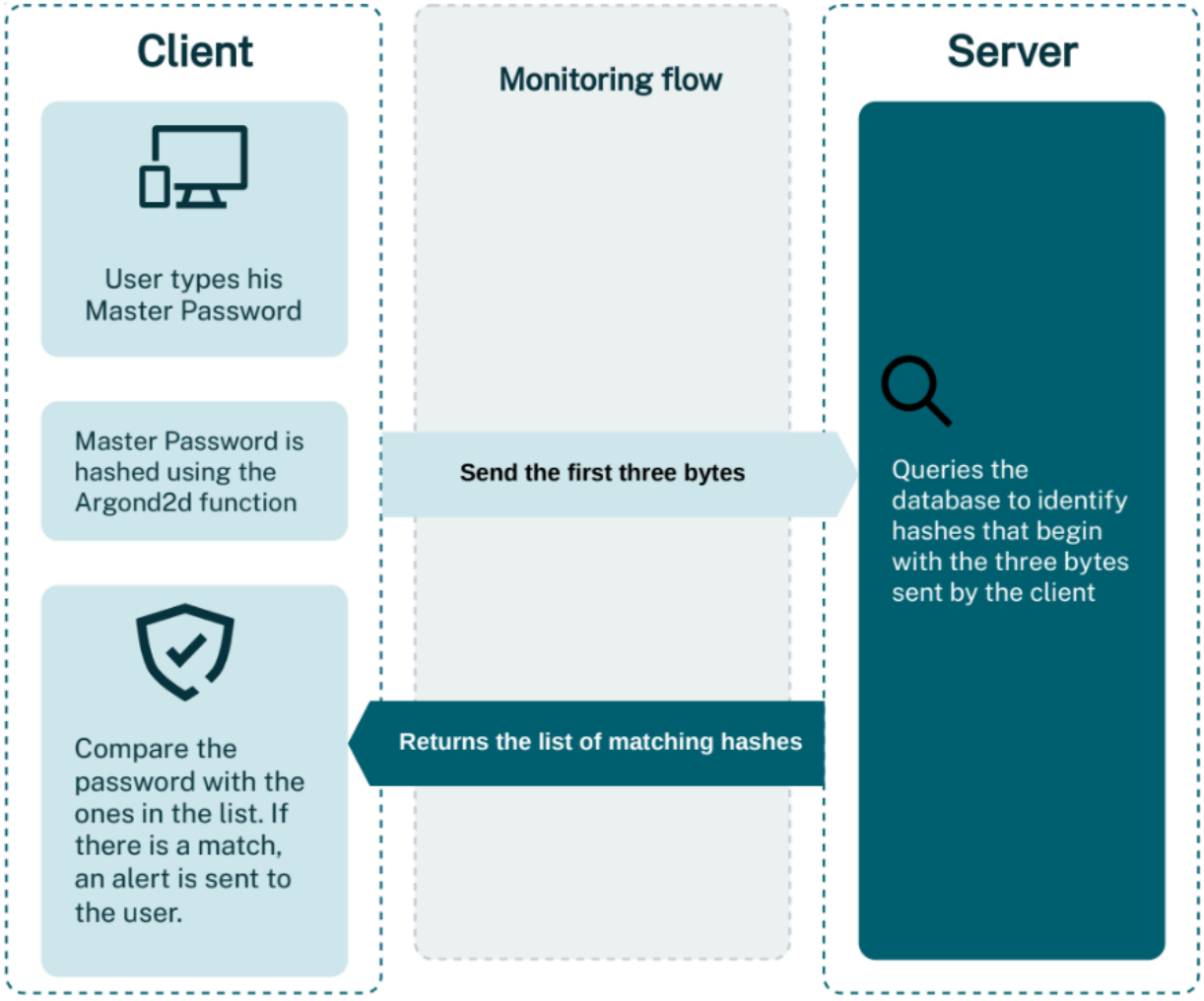


Figure: Dark Web Monitoring for Master Password

4.7.3 Dark Web Insights for Businesses

Dark Web Insights expands Dark Web Monitoring to cover the entire organization, including employees without Dashlane accounts. It continuously scans the dark web for employee

email addresses associated with verified company domains, enabling detection of domain-wide breaches and exposed organizational data.

When an admin verifies a company domain for Dark Web Insights, all corresponding work email addresses are automatically enrolled and monitored. Employees who are already Dashlane plan members continue to manage their individual exposure through Dark Web Monitoring and receive alerts when their data appears in a breach.

Dashlane compares verified domains against breach datasets sourced from trusted third-party providers. Any credentials matching the organization's domains are aggregated in the **Dark Web Insights dashboard**, with results also available via CSV export. Reports include the breached services, breach dates, affected email addresses, and the types of exposed data (such as usernames, passwords, or payment information).

To drive better security hygiene, admins can invite some or all non-plan employees identified in Dark Web Insights to create a Dashlane account, helping extend protection and remediation across the workforce.

4.7.4 Credential Risk Detection

Credential Risk Detection is a business feature that applies the same monitoring as Dark Web Monitoring for Master Password, but on credentials autofilled, copy-pasted, or manually typed by employees on company-managed browsers where Dashlane is deployed. See 5.7.1 Credential Risk Detection section for more details.

4.8 Import and Export

Dashlane supports secure import and export of credentials to help individuals and organizations onboard efficiently or transition their data when needed.

4.8.1 CSV Import

Credentials can be imported from other password managers or browsers using CSV files, a widely supported industry format. During import, credentials are processed locally on the user's device and encrypted immediately according to Dashlane's zero-knowledge architecture before being stored or synchronized. This ensures that plaintext credentials are never exposed to Dashlane servers.

4.8.2 Export

For export, users can choose between a standard CSV format or Dashlane's proprietary encrypted export. CSV exports provide interoperability with third-party tools but contain credentials in plaintext and should be handled with care.

Dashlane's proprietary export preserves end-to-end encryption, ensuring that credentials remain protected and can only be decrypted by an authorized Dashlane client. These options allow consumers and organizations to balance interoperability, security, and operational needs while maintaining control over sensitive credential data.

4.8.3 Credential Exchange and Portability

Dashlane supports **FIDO Credential Exchange** to ensure credential portability (including passkeys) across platforms and devices without sacrificing security. This protocol enables secure import and export using a short-lived asymmetric key exchange validated through QR code proximity or visual verification.

All operations occur using **elliptic-curve cryptography (Curve25519)** and ephemeral session keys. The private key never leaves the secure hardware enclave of the original device. The exchanged payload is encrypted and signed before being relayed through Dashlane's servers, which act solely as a transport layer and have no visibility into the underlying data.

Credential Exchange is currently only supported on iOS and Android.

5. Enterprise Features

Summary

Dashlane delivers an enterprise-ready platform that integrates with your existing security stack and raises your security baseline without extra IT overhead. Confidential SSO leveraging AWS Nitro Enclaves, SCIM provisioning, and RBAC anchor access control; Activity Logs provide auditability; and Omnix (Credential Risk Detection, Nudges, AI phishing alerts) drives continuous risk reduction. Central policies, mass deployment, and CLI/Public API integrations round out a scalable program.

5.1 Single Sign-On (SSO)

Dashlane integrates with **identity providers** (IdPs) that use the **SAML 2.0** standard, including **Microsoft EntraID**, **Okta**, and **Google Identity Management,** to allow employees to unlock their Dashlane vaults using corporate credentials instead of a Master Password. This integration preserves Dashlane's **zero-knowledge architecture**, ensuring that neither Dashlane nor any third party can access user encryption keys.

5.1.1 Dashlane Confidential SSO

Dashlane Confidential SSO operates within **AWS Nitro Enclaves**, leveraging **confidential computing** to maintain zero-knowledge guarantees while providing easier deployment and management.

For general information about Dashlane's use of confidential computing and cloud secure enclaves, please refer to 3.5 Confidential Computing & Secure Enclaves

During team setup, administrators configure the IdP certificate and domain ownership through a **DNS challenge**, which is verified directly by the enclave before allowing configuration. Once the domain is validated, the enclave creates and stores the **SPMasterKey** and associated metadata (as described in 3.5.2 Enclave Workflows). For each user login, a **User Service Provider Key (UserSPKey)** is generated, encrypted with the SPMasterKey, and stored securely. During subsequent logins, the enclave decrypts and

returns the UserSPKey to the Dashlane client via a secure channel after verifying the SAML assertion signed by the IdP.

These workflows ensure that:

- **Encryption keys never leave the enclave** in plaintext.
- **SAML assertions** are verified and processed only in attested environments.
- **Team creation, user login, and key provisioning** are fully auditable and verifiable.

Through this architecture, Dashlane provides a highly secure, verifiable, and zero-knowledge SSO solution that integrates seamlessly with enterprise identity systems.

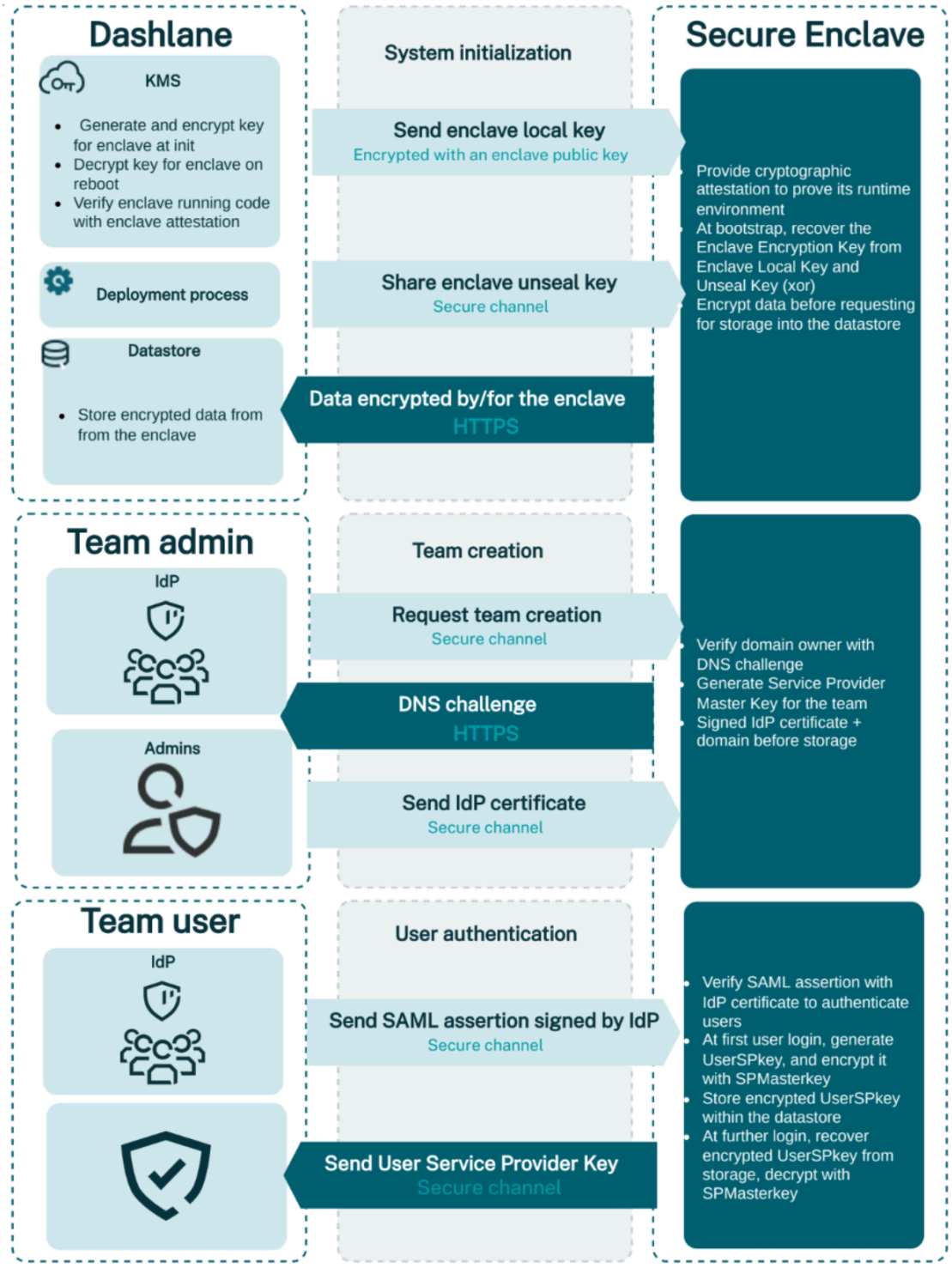


Figure: Confidential SSO overview

Team Creation

When an organization configures SSO, the administrator provides the **IdP certificate** to

verify SAML assertions for users within the organization's domain. The enclave verifies ownership of the claimed domain via a DNS challenge before registration. Once verified, the enclave generates a **SPMasterKey**, an admin authentication token, and stores both encrypted by the EEKey. The admin token authenticates future SSO administrative operations.

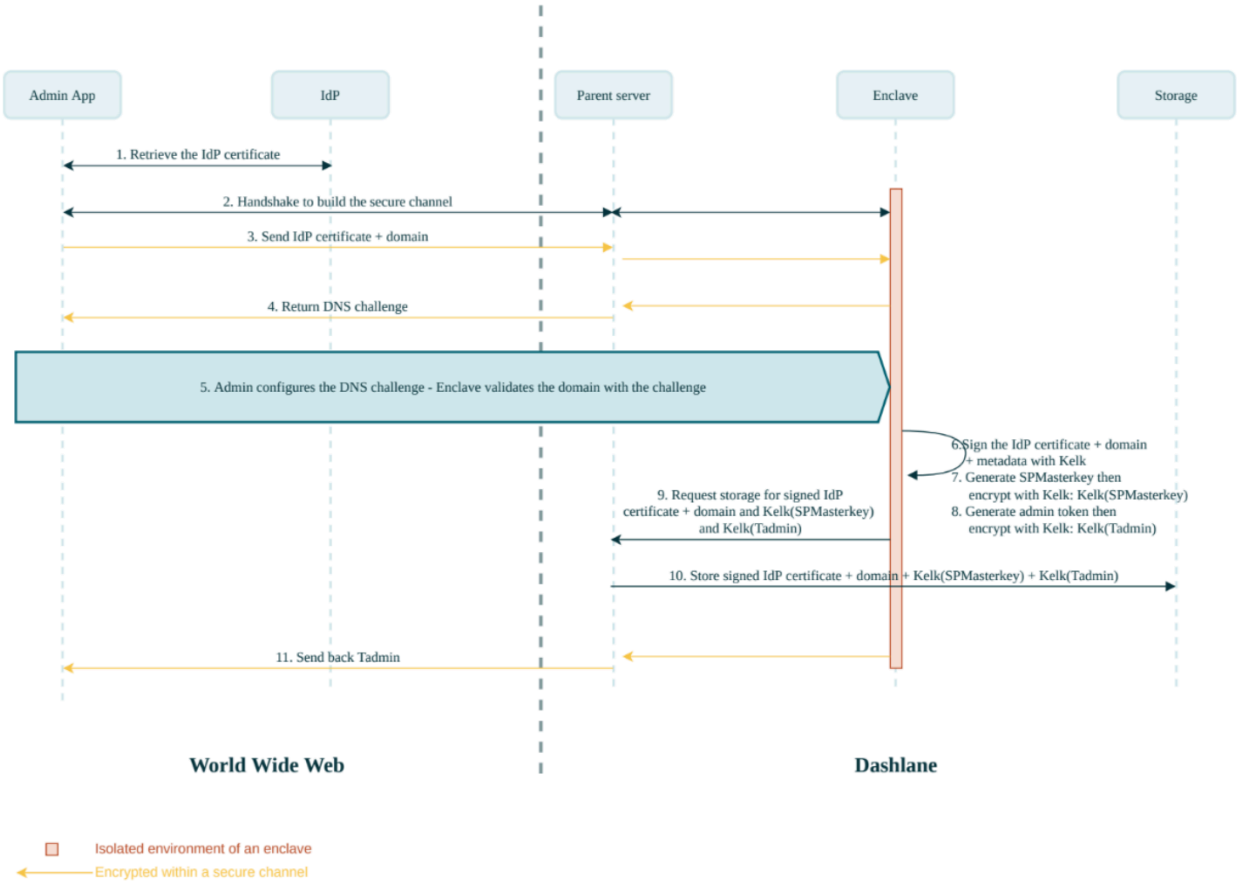


Figure: Confidential SSO: Team creation flow

User Authentication

At login, the user authenticates via their enterprise IdP, which returns a signed SAML assertion. The Dashlane client forwards this assertion to the enclave through a secure, attested channel. The enclave verifies the signature using the stored IdP certificate, decrypts the SPMasterKey with the EEKey, and either generates or retrieves the UserSPKey for that user. This key is then securely returned to the client to unlock the vault. Subsequent logins reuse the stored, encrypted UserSPKey to minimize overhead while maintaining zero-knowledge guarantees.

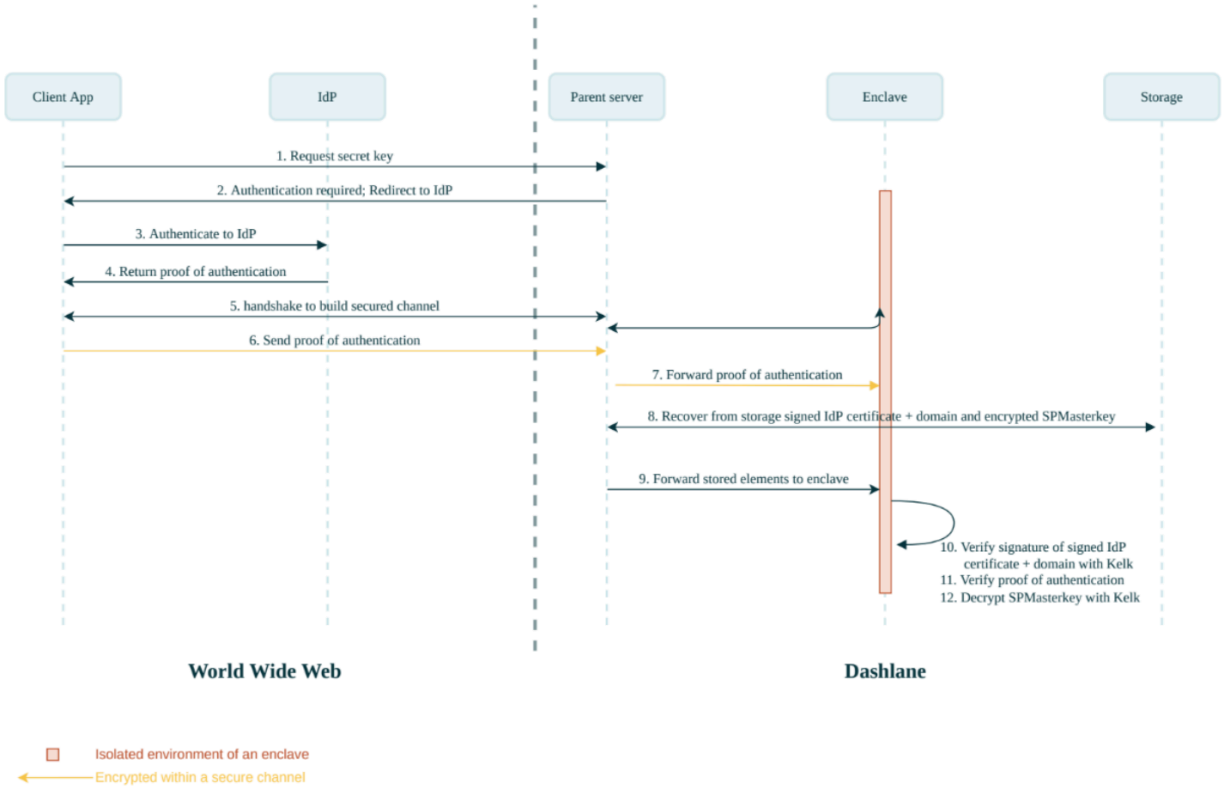


Figure: Confidential SSO: User login flow

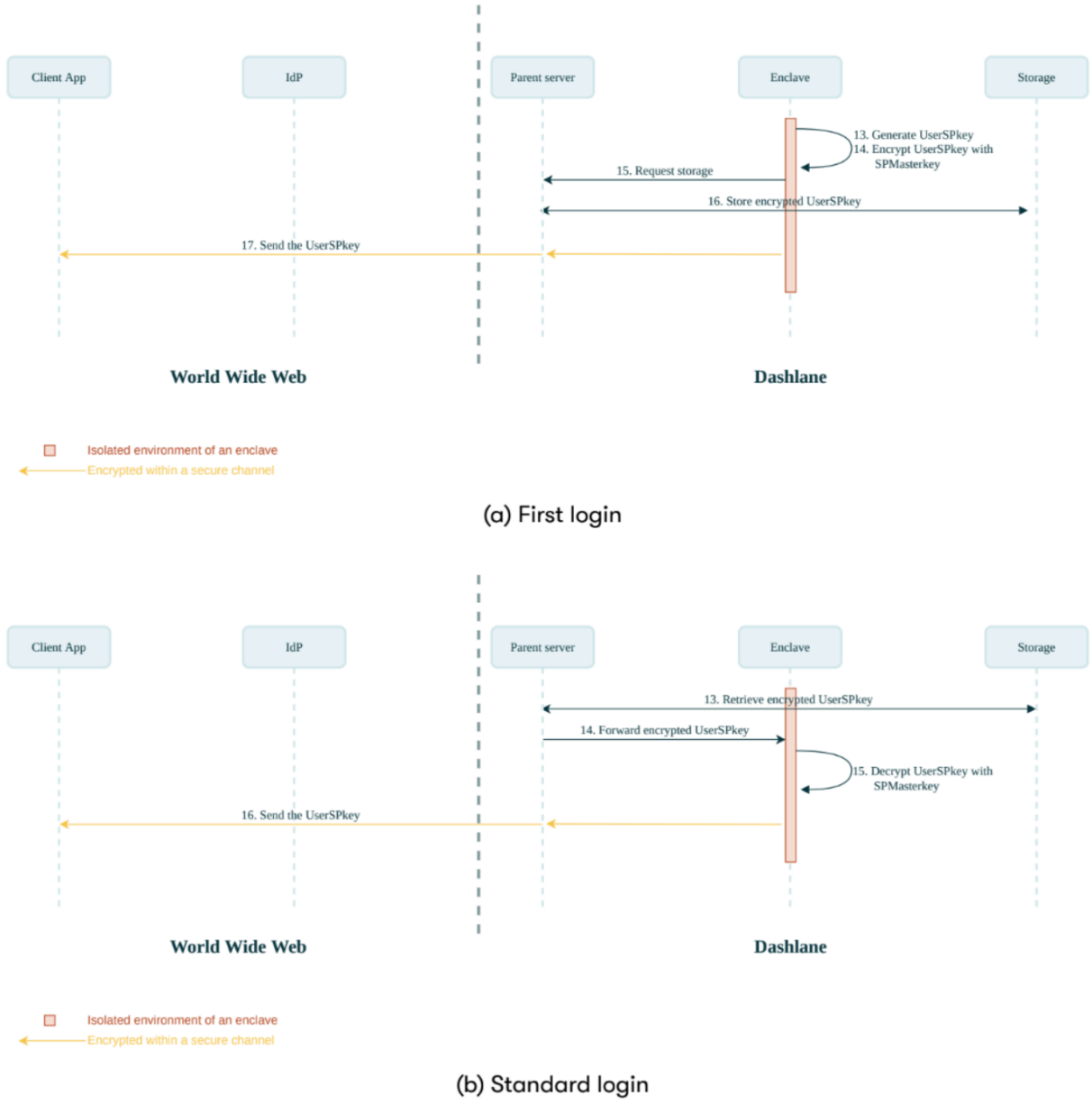


Figure: Confidential SSO: First login and standard logins

These steps ensure that key management, user provisioning, and authentication occur only within the enclave, providing verifiable protection and auditability throughout the SSO process.

5.2 Provisioning

Dashlane's confidential computing infrastructure supports **SCIM (System for Cross-domain Identity Management)** for automated, secure user provisioning and deprovisioning.

5.2.1 User Provisioning

Administrators can configure confidential user provisioning directly from the **Admin Console**. When set up, the Dashlane extension generates a **bearer token** (UUIDv4) that is transmitted securely to the enclave, where it is encrypted and stored. The administrator then adds this token and a team-specific SCIM endpoint URL into their identity provider (IdP).

Once configured, user creation, updates, and deletions are automatically synchronized via HTTPS requests from the IdP to the enclave. The enclave validates the SCIM bearer token before forwarding operations to the Dashlane servers. Upon user creation, the enclave generates a unique **SCIM user ID (scimId)**, shared between the IdP and Dashlane to maintain consistency across systems. This guarantees a reliable, auditable linkage between identity data and user vaults.

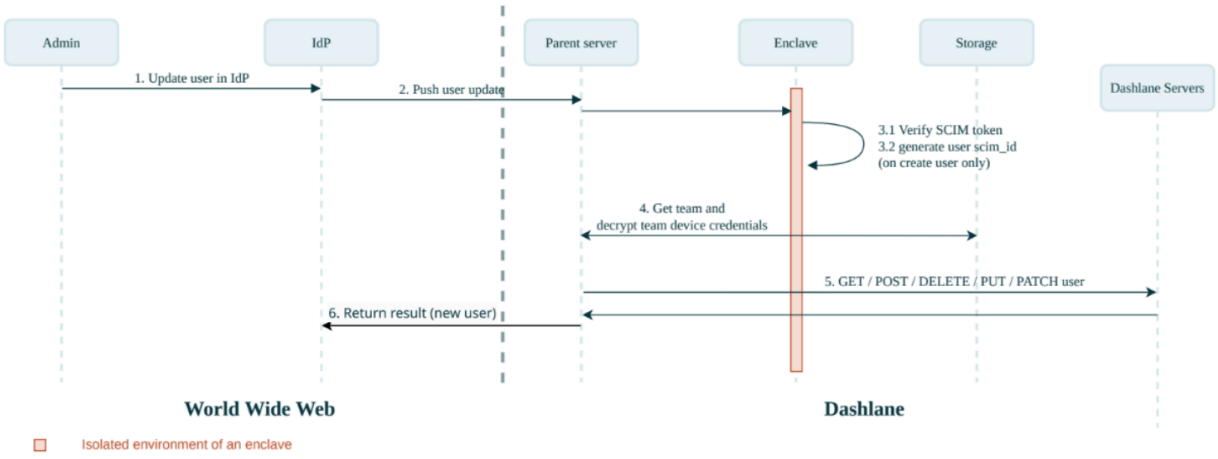


Figure: User provisioning

5.2.2 Group Provisioning

If enabled in the Admin Console, **group provisioning** occurs dynamically during user login through SAML assertions.

Group provisioning can give access to shared secrets, making it highly sensitive. SAML assertions are preferred because they transit through the secure tunnel created by the extension and are signed by the IdP. Therefore, user group memberships are updated on every user login.

Each SAML assertion includes the list of groups the user belongs to, which the enclave validates against the IdP's signature. The enclave then determines which groups to create, update, or revoke memberships for and instructs Dashlane's servers accordingly. The process is idempotent, ensuring that after each login, a user's group memberships in Dashlane exactly match those defined in the IdP.

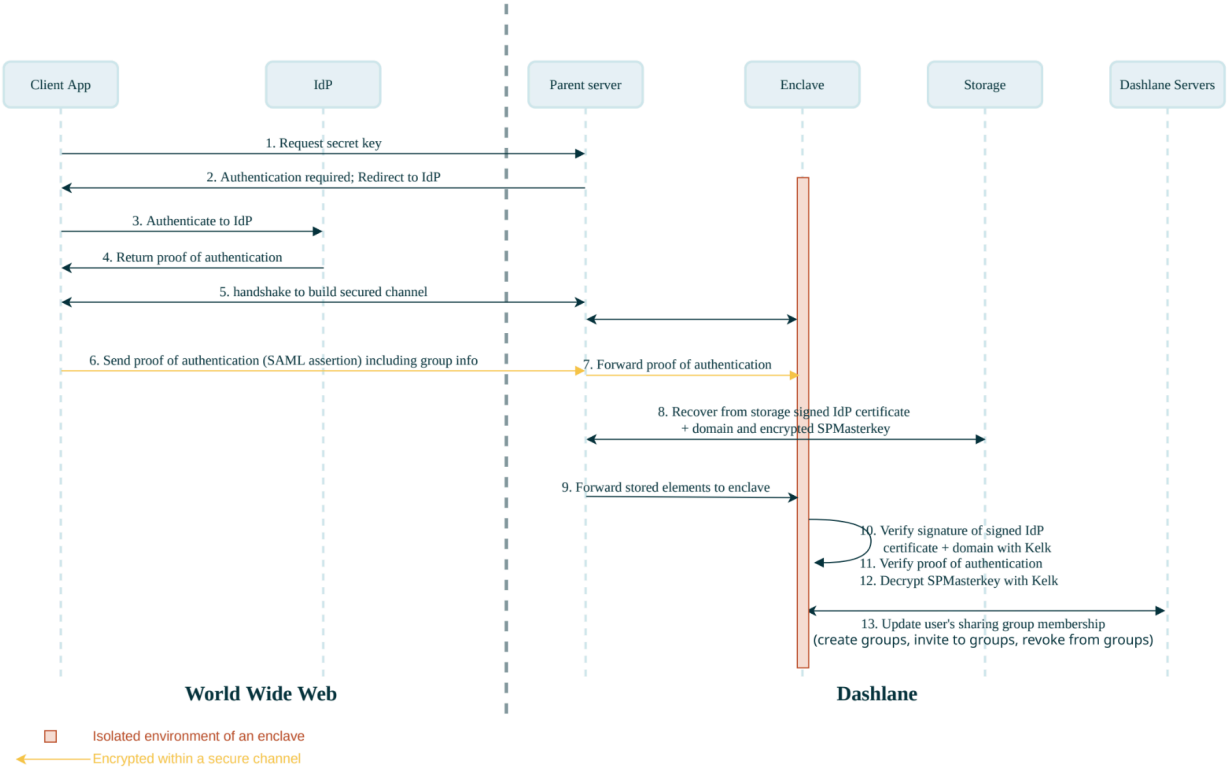


Figure: Group provisioning

This approach secures group management while supporting compliance and least-privilege principles:

- All provisioning logic and token handling occur inside the enclave.
- Group membership data is verified through signed SAML assertions.
- Operations are fully auditable and resistant to tampering.

By combining SCIM automation with secure, enclave-based processing, Dashlane ensures that enterprise user and group management remains both **frictionless and verifiably secure**.

5.3 Role-Based Access Control (RBAC)

Dashlane's **Role-based access control (RBAC)** model enables organizations to manage permissions and sharing responsibilities securely and at scale.

5.3.1 Role Hierarchy

Dashlane business accounts support multiple administrative roles, each with distinct privileges:

- **Admin:** Has full control over the organization's Dashlane workspace, including user provisioning, SSO/SCIM configuration, billing, and security tools and policies management. Admins can delegate group management rights and configure account recovery policies.
- **Group Manager:** Oversees sharing groups. Group Managers can manage both groups and membership within groups but have no access to other organizational data.
- **Scoped Group Manager:** Oversees only admin-assigned sharing groups. Scoped Group Managers can manage membership within assigned groups but have no access to other organizational data. It's also possible to assign only a specific set of groups for the user to manage.
- **Member/User:** Standard users who can create and manage personal or shared credentials within the policies defined by admins.

This hierarchy supports the **principle of least privilege**, ensuring each user has only the permissions necessary for their function.

5.3.2 Sharing Groups and Collections

Dashlane supports secure, structured credential sharing through **groups** and **Collections**, allowing admins to organize credentials based on teams, functions, or projects. Each shared item or collection is encrypted with a unique **GroupKey (AES-256)** that is then re-encrypted for each authorized member's public key.

- **Groups** simplify access control by linking permissions to organizational or project structures managed through SCIM or the Admin Console.

- **Collections** provide a flexible container for sets of shared items. Within a Collection, access is governed by two specific collection roles: Editor and Manager. Both roles are authorized to manage Collection contents and item-level permissions. The Collection Manager has the elevated privilege to control user and group access to the Collection.

Access modifications automatically trigger re-encryption of affected GroupKeys, ensuring that departing users lose access immediately while preserving zero-knowledge integrity.

5.3.3 Security Enforcement and Auditability

RBAC is enforced through cryptographic operations validated on each client device:

- Role assignments and sharing actions are recorded in the **Activity Log** for compliance and forensic purposes.
- Group and Collection changes propagate in real time via Dashlane's synchronization service, maintaining consistent policy enforcement across all platforms.

Through its cryptographically enforced RBAC, Dashlane ensures granular, auditable, and secure management of user permissions, supporting enterprise compliance and minimizing the risk of unauthorized access.

5.4 Activity Logging & Auditing

Dashlane provides enterprise administrators with **Activity Logging and auditing** capabilities to ensure visibility, accountability, and compliance across their organization. These logs capture critical events, such as user activity, admin actions, and security-related events, while preserving the platform's zero-knowledge design.

5.4.1 Overview

The **Activity Log** is accessible through the **Admin Console** and provides a timestamped record of user and administrative actions. Events are divided into two categories:

- **Standard Activity Logs:** Generated by Dashlane servers to record common user and admin events (e.g., user invitations, policy changes, SSO configurations).
- **Sensitive Activity Logs:** Additional logs produced by client applications, encrypted locally before being uploaded to Dashlane servers for secure aggregation. These contain user events related to their sharing and vault activity, though they will never

include actual login details. This set of logs can be enabled or disabled by administrators.

A complete and continuously updated list of event types is available in Dashlane's Activity Log documentation.

5.4.2 End-to-End Encrypted Logging

Dashlane's zero-knowledge architecture extends to encrypt all data, including Activity Logs. Our platform's Activity Log infrastructure addresses the challenge of securing the massive volume of sensitive data generated by end-user usage of Dashlane, ensuring these logs are fully encrypted while remaining queryable for Security Operations purposes.

The architecture allows for server-side processing and analysis of logs without Dashlane employees or any third party ever having access to the customer data. This unique combination of end-to-end encryption with necessary performance and usability sets it apart from other credential security or identity management platforms.

Dashlane's confidential computing infrastructure, powered by AWS Nitro, forms the foundation for our security architecture. Originally developed to support third-party integrations, such as Confidential single-sign on (SSO) and Confidential Provisioning with SCIM, this infrastructure was leveraged to isolate sensitive log data from external attackers and potential internal threats.

This architecture provides the following core security benefits:

1. **Complete, isolated log encryption:** All log data is encrypted with team-specific keys accessible only within the secure enclaves. Log decryption is performed exclusively inside the secure enclave.
2. **Secure communication for logs:** Server API endpoints responsible for storing and retrieving logs are moved into the Nitro enclaves. A secure tunnel capability establishes a private communication channel between client applications and the enclave, ensuring all log data remains encrypted in transit and is never decrypted on our standard server infrastructure.
3. **Secured Admin Access:** Secure log access for administrators is facilitated by the same secure tunnel. Logs are decrypted inside the enclave and sent back to the authorized admin via the secure tunnel, ensuring the decrypted data is never exposed on Dashlane's servers. All activity log endpoints are further protected with an Admin Access Token—a secret token shared only between administrators and our Nitro enclaves—to defend against external and internal threat actors.

4. **Device authentication for logged-out users:** Omnix generates critical activity logs even when employees are logged out of Dashlane. This flow is secured by provisioning unique encryption keys during team deployment, which cryptographically binds the logs to the correct organization. This mechanism prevents attackers from intercepting and redirecting log data between teams, even during unauthenticated sessions.

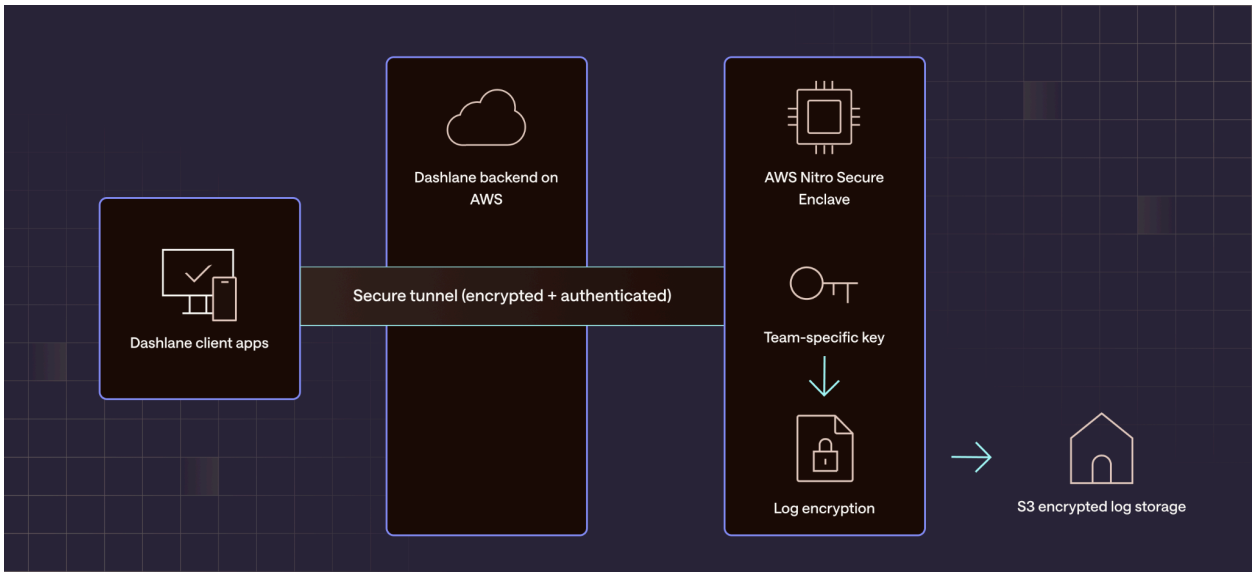


Figure: Activity Log encryption

5.5 Mass Deployment

Dashlane supports **mass deployment** capabilities to help IT and Security teams roll out the Dashlane browser extension across all managed endpoints efficiently and securely.

Administrators can distribute Dashlane via **enterprise management solutions** such as Microsoft Intune, Jamf, or Microsoft Group Policy Objects (GPO). Deployment scripts generated from Dashlane's **Admin Console** include configuration details like the organization's team key and extension settings.

During deployment, the Dashlane extension automatically associates installed instances with the organization through a **Mass Deployment Team Key**, allowing Credential Risk Detection and other policy enforcements to function even when end-users are logged out. All communications and configurations are handled securely, maintaining Dashlane's **zero-knowledge architecture**.

5.6 Account Recovery for Enterprise

Dashlane provides enterprise-grade account recovery mechanisms designed to restore user access without compromising its zero-knowledge architecture. The process is built to ensure that even during recovery, Dashlane cannot access a user's vault.

- **Account Recovery Key (ARK):** Provides a secure, self-service recovery option for employees.
- **Admin-assisted account recovery:** Allows Dashlane business users who log in with a Master Password to reset it securely while preserving data confidentiality. Dashlane's patented recovery process guarantees that the Master Password is never stored or transmitted.

These dual mechanisms, **admin-assisted recovery** and **ARK**, enable organizations to balance usability and compliance while maintaining Dashlane's zero-knowledge security model.

More details are available in 4.3 Account Recovery

5.7 Dashlane Omnix™ Platform Capabilities

Dashlane **Omnix** extends enterprise-grade visibility and proactive protection across the workforce through security services: **Credential Risk Detection**, **Nudges**, and **AI Phishing Alerts**. All components operate within Dashlane's zero-knowledge model and are designed to enhance organizational password hygiene and threat resilience.

5.7.1 Credential Risk Detection

Credential Risk Detection monitors the security posture of employee credentials, identifying weak, reused, or compromised passwords, for logged-in users, logged-out users, and employees without Dashlane accounts.

When deployed via endpoint management solutions (e.g., Intune, Jamf, or GPO), the Dashlane browser extension operates in a special mode authenticated with a **Mass Deployment Team Key**. After assessing password strength and exposure locally without accessing or transmitting vault data, this Mass Deployment Team Key allows the extension to securely log the results for exclusive visibility by the team admin.

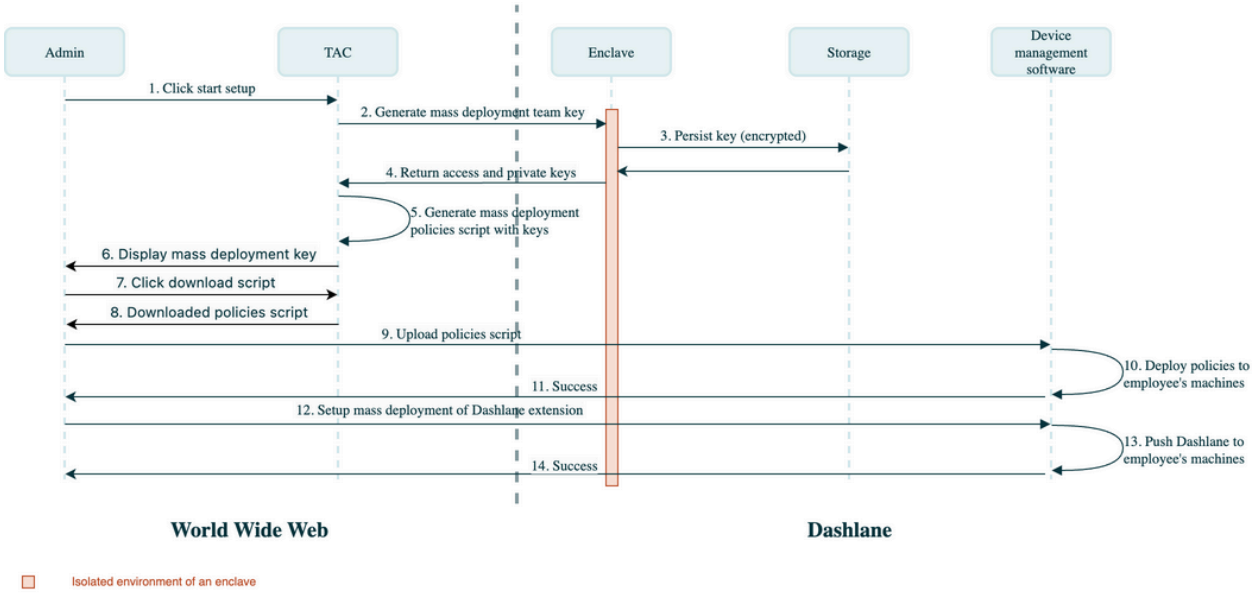


Figure: Credential Risk Detection mass deployment process

Credential Risk Detection performs the following checks:

1. **Weakness detection:** Evaluates passwords locally using the open-source `zxcvbn` library.
2. **Compromise detection:** Uses Dashlane's privacy-preserving leaked passwords database and Argon2 hashing to detect exposed credentials without uploading plaintext data. See 4.9.2 Dark Web Monitoring for Master Password section for more details.
3. **Activity logging:** Sends encrypted status reports (safe, weak, or compromised) to Dashlane's secure enclave for processing and admin visibility.

Evaluation of credentials occurs locally, and only the results of those operations are sent to a confidential computing environment. Administrators can view aggregated insights in the **Security Dashboard**, enabling them to assess organizational risk and prioritize remediation.

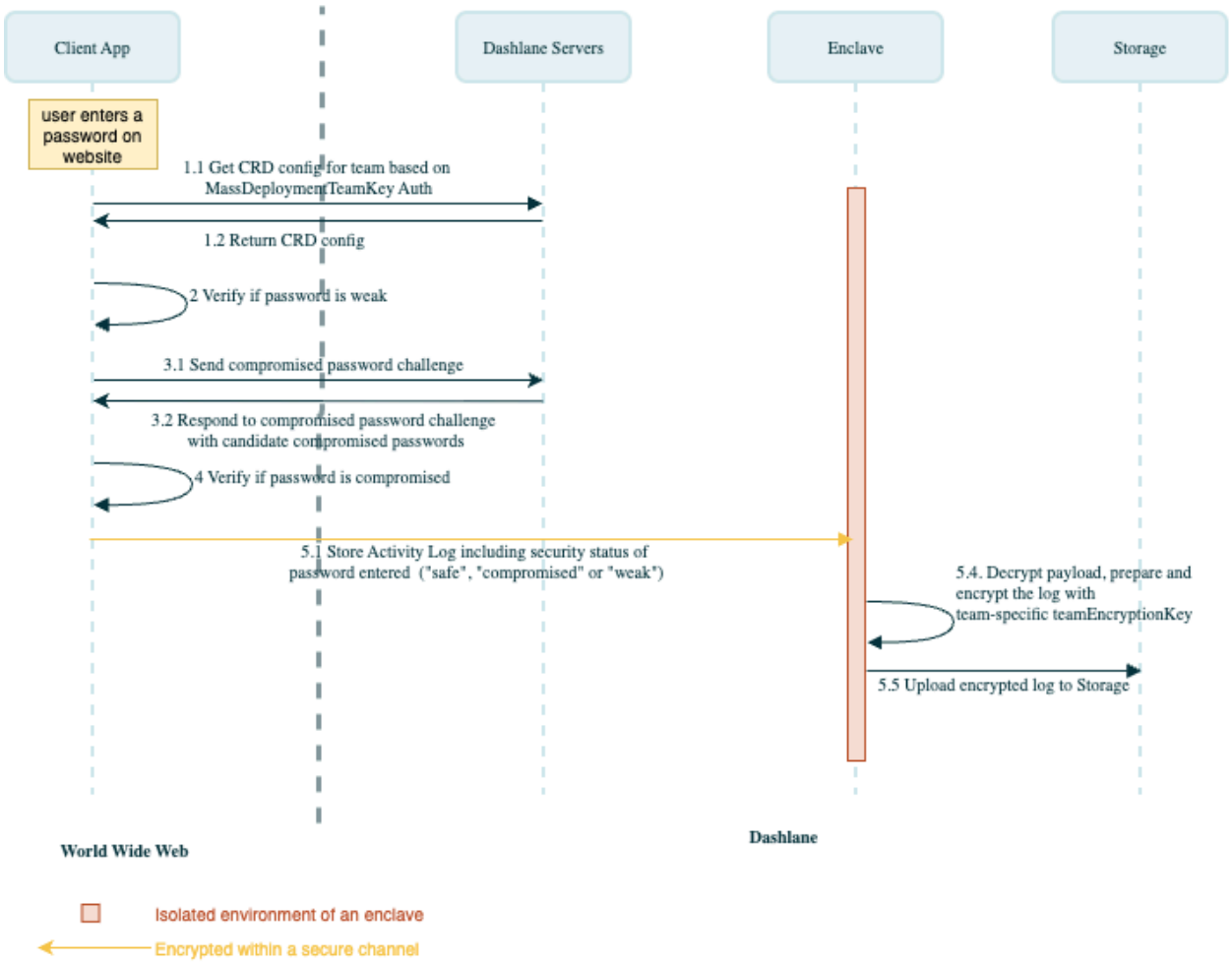


Figure: Credential Risk Detection activity log upload

5.7.2 Nudges

Nudges enable administrators to automate password hygiene campaigns and send scheduled and real-time prompts to employees about vulnerable credentials.

5.7.2.1 Slack Nudges

Admins can configure nudges in the **Admin Console** and integrate them directly with Slack.

- The admin installs the Dashlane Slack app and authorizes it through OAuth.

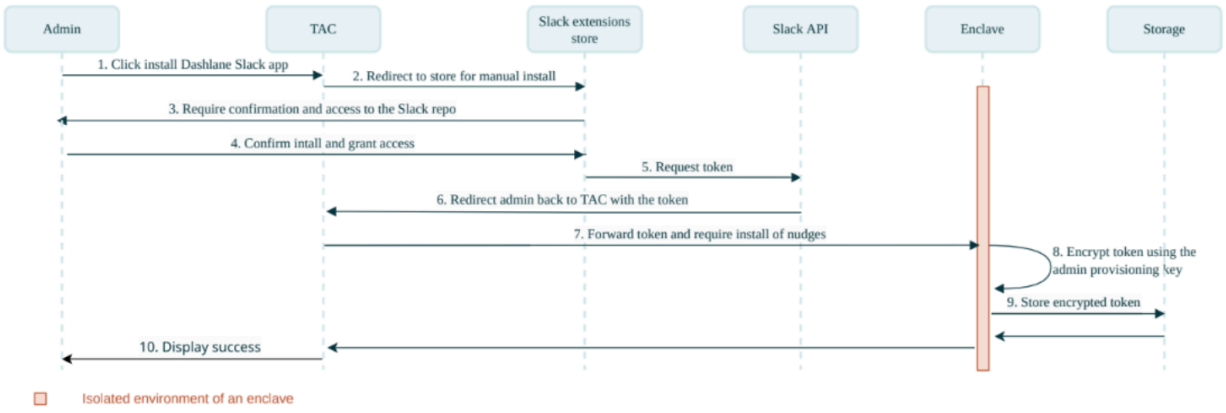


Figure: Dashlane Slack app installation

- A scheduled routine evaluates vault data (using encrypted health metrics) and sends context-aware notifications to targeted users.
- Results of each nudge execution are logged for compliance and transparency.

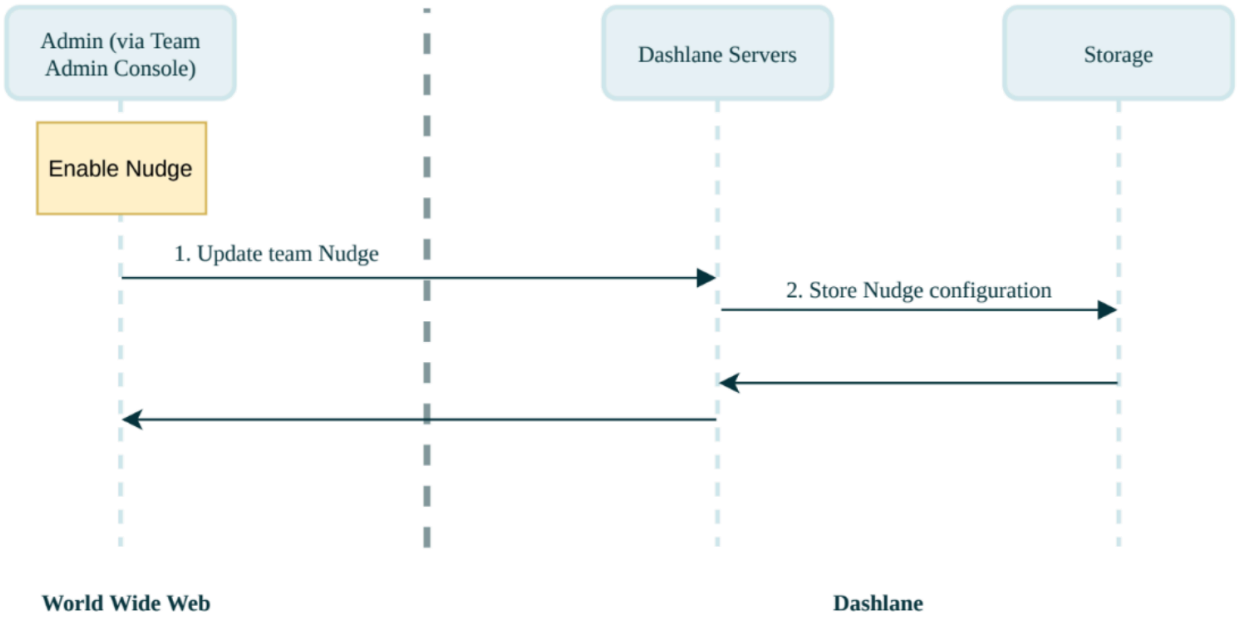


Figure: Nudge configuration

All Slack API tokens are encrypted and processed exclusively inside Dashlane's enclave, ensuring that no other system or internal employee can use these Slack tokens.

In addition, the nudge messages contain no specific information about the user's vulnerable credentials, only a deep link to the Password Health dashboard inside the Dashlane browser extension. This prevents Slack's systems or employees from seeing details of the user's vulnerable credentials.

Results of each nudge execution are logged for compliance and transparency, and they are stored encrypted by Dashlane's enclave so only the team's admins can access them (see 5.4 Activity Logging & Auditing).

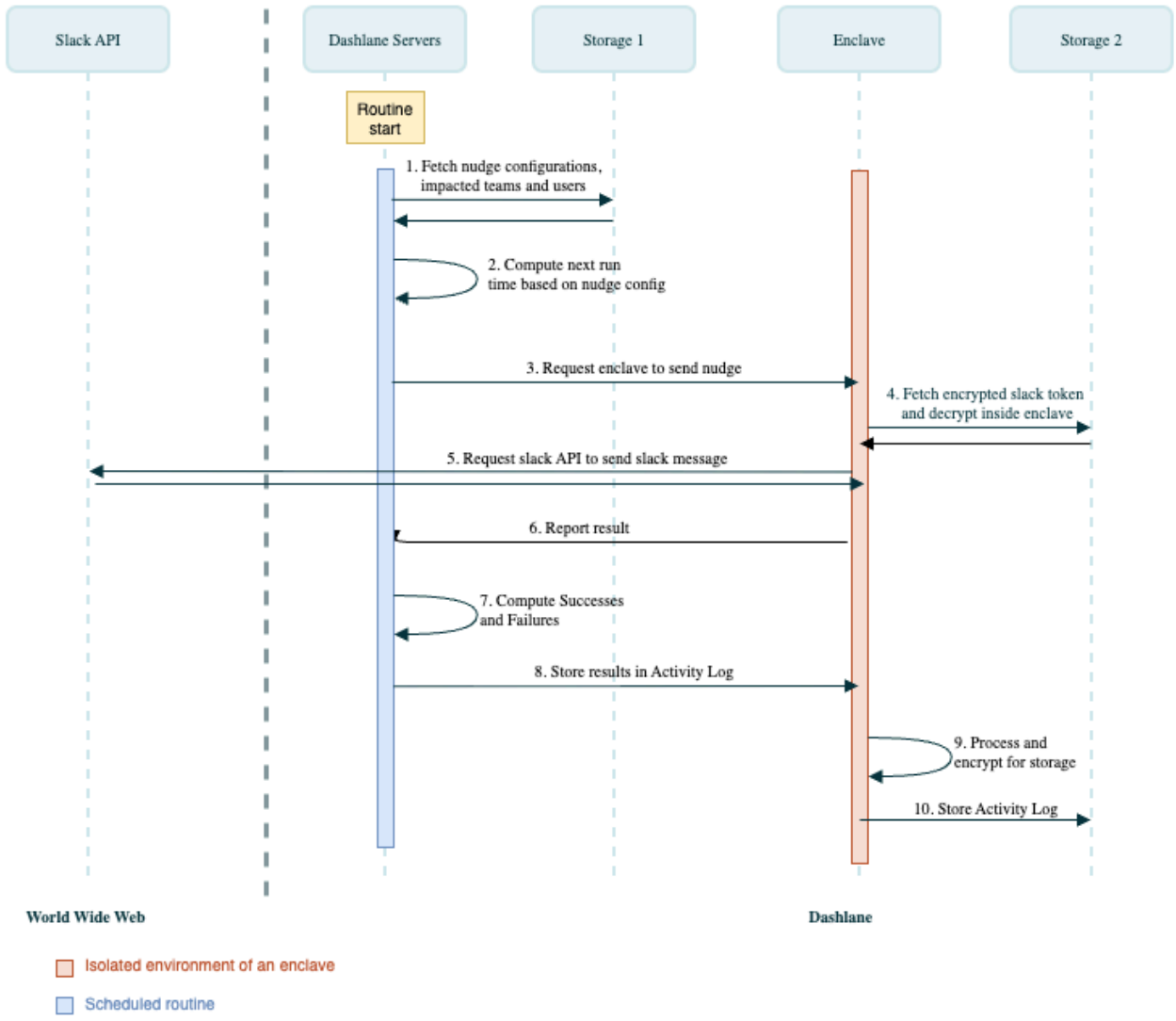


Figure: Nudge routine

5.7.2.2 In-Browser Nudges

Admins can configure in-browser nudges in the **Admin Console,** so the Dashlane browser extension will immediately inform users when they use vulnerable credentials.

This feature extends the Credential Risk Detection feature by displaying a notification to the user in addition to sending a securely encrypted activity log to the admin. See 5.7.1 Credential Risk Detection for details on this detection of vulnerable credentials.

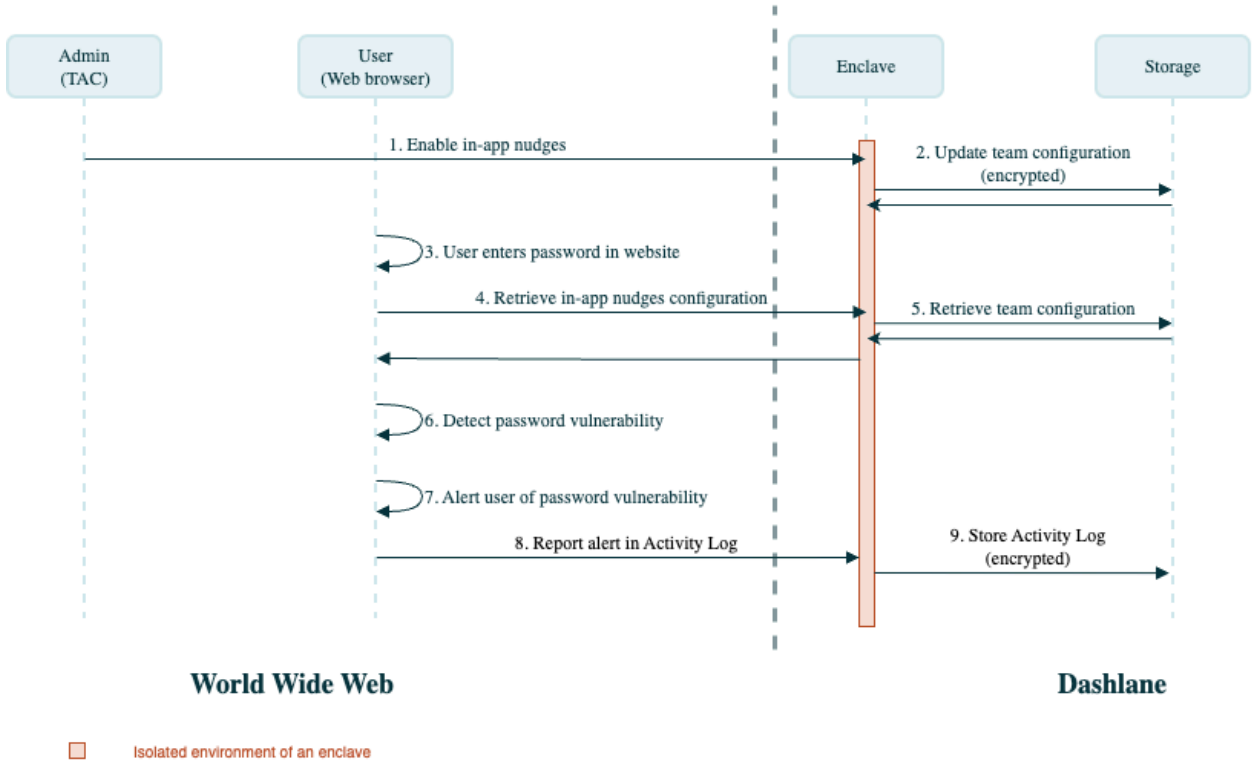


Figure: In-browser nudge processing

5.7.3 AI Phishing Alerts

Dashlane's **AI phishing alerts** use a combination of machine learning and contextual analysis to protect users from credential theft and fraudulent websites. Unlike traditional blocklist-based protections that rely solely on static lists of known domains, Dashlane's AI system proactively identifies emerging, never-before-seen phishing campaigns by analyzing the page's intrinsic characteristics in real time. It operates directly within Dashlane's browser extension, providing **on-device threat detection** that aligns with Dashlane's zero-knowledge model.

To ensure fast and private protection, the system identifies potential threats in real time through these components:

- **Optimized model architecture:** Models are developed using lightweight models (such as Random Forest and Gradient Boosting) via scikit-learn and converted to the Open Neural Network Exchange (ONNX) format to ensure cross-platform compatibility. We also strip the runtime of unnecessary components, keeping the average model size under 3MB to guarantee that the entire inference process runs locally on the user's device without ever sharing data to the cloud.
- **Multi-vector feature extraction:** The AI model analyzes the webpage DOM structure to extract critical features, including URL anomalies (such as typo-squatting), content indicators (suspicious text patterns or brand mimicry), and structural anomalies (hidden form fields, unusual redirect patterns, or iframe abuse).
- **Privacy-first training:** Our models are trained exclusively on data collected through internal crowdsourcing and publicly available threat intelligence feeds. No user personal data is ever used in the training process.

Administrators can configure custom rules to exclude phishing detection on internal websites or known safe domains. See 5.8.4 Anti-Phishing Custom Rules

This approach provides users and enterprises with **proactive, privacy-preserving defense** against phishing attempts, delivering protection that evolves as threats do, without compromising user confidentiality.

Note: AI phishing alerts are also available for personal customers, as AI-powered Scam Protection

5.8 Enterprise Policies and Settings: Security Governance and Enforcement

Dashlane enables administrators to enforce **enterprise-wide security policies** that align with organizational requirements, ensuring compliance and robust risk mitigation while maintaining a seamless user experience. These controls can be configured through the **Admin Console** and are rigorously applied across all plan members via cryptographic or client-side enforcement mechanisms. All policy changes are **enforced centrally and synchronized to all users' devices** in real time through Dashlane's secure synchronization service.

5.8.1 Identity and Access Management (IAM)

These controls govern user lifecycle management, provisioning, and the integrity of the authentication process required to decrypt the vault.

- **2-factor authentication (2FA) enforcement:** Mandates that all members must enable a second factor when authenticating with their Master Password.
- **New user provisioning**: Enables administrators to provision new user accounts using a dedicated, authenticated signup link or via automatic SCIM provisioning. User enrollment is secured by mandatory email verification to confirm identity before Master Password creation. See 5.2 Provisioning for details.
- **Cryptography Policy**: Enables administrators to select the cryptography that complies with the company's policy. Dashlane encrypts data using AES-256 and the Argon2d key derivation function (KDF) by default.
- **Restrict Non-verified Domains from Login into Dashlane**: Enables administrators to restrict employees from logging into Dashlane accounts with non verified work domains, such as personal accounts

5.8.2 Session Management and User Monitoring

These policies dictate session longevity, auditability, and the level of visibility afforded to administrators.

- **Enforced auto-logout on inactivity:** Administrators configure a mandatory session timeout (for example, 15 min, 30 min, 1 hr) after which the user's vault automatically locks if the system is not in a locked state.
- **Mandatory auto-lock on mobile app exit:** Ensures the Dashlane application on iOS and Android automatically locks immediately upon exiting the app.
- **Enhanced business activity logging:** Enables end-to-end logging of user sharing and vault activity by business users, while excluding all sensitive information such as login and password details.
- **Login list visibility:** Provides administrators with a list of the names or URLs of logins stored in an employee's business vault, while excluding all sensitive information such as login and password details.

5.8.3 Data Sharing Policy Controls

These policies govern the secure, cryptographically controlled flow of credential sharing between plan members and external entities.

- **Sharing restrictions**: Allows admins to enable or disable the zero-knowledge-based sharing function for logins, Secure Notes, and secrets
- **External sharing prohibition**: Allows compliance with data sharing policy and blocks all sharing attempts with users whose identity key is not associated with an account within the organization's plan
- **Link based sharing:** Link-based sharing requires the "Secure sharing for logins, Secure Notes and Secrets" policy to be enabled. In addition, Link-based sharing requires the "Allow sharing outside company" policy to be enabled.

5.8.4 AI Phishing Alerts Custom Rules

These policies allow the admin to **mute phishing alerts on specific domains** (typically internal domains). One rule can cover multiple domains (inclusive of sub-domains) and apply to AI phishing alerts, vault phishing alerts, or both.

The list of rules for each team is encrypted, authenticated, and processed exclusively within Dashlane's enclave, ensuring that no Dashlane employee or system can view the sensitive information it contains, including internal domains. The integrity of the data is also enforced with the use of authenticated encryption to prevent an adversary from tampering with the rule server-side.

As an additional precaution, cleartext domains are not sent to end users' devices; only a hash of each domain is sent. When the end user visits a domain, the Dashlane extension is able to hash it and compare it locally to the list of hashes for each rule without ever receiving the cleartext domains. Only the admin is able to retrieve the cleartext domains.

5.8.5 Other Security Controls

- **Turn off auto-login and autofill:** Prevents accidental credential exposure or autofill on specific high-risk, forbidden websites or IP addresses
- **Virtual private network (VPN):** Allows centralized control over the use of the VPN service for secure browsing on public networks

5.9 Dashlane Developer Tools

Dashlane provides developers and administrators with **command-line and API-based interfaces** to integrate Dashlane into enterprise automation, DevOps workflows, and security monitoring systems.

5.9.1 Command Line Interface (CLI)

The **Dashlane Command Line Interface (CLI)** enables secure, programmatic access to Dashlane vault data, Activity Logs, and other admin-level data without using the graphical application. It supports automation for credential retrieval, integration with CI/CD pipelines, and machine account access management.

- **Authentication:** The CLI uses the same zero-knowledge architecture as Dashlane's applications. Authentication can be completed using a Master Password or enterprise SSO, with local decryption of vault data on the device.
- **Encryption and key handling:** All secrets remain encrypted end-to-end; no plaintext credentials are ever transmitted or stored on Dashlane servers.
- **Integration capabilities:** The CLI integrates with tools such as Jenkins, GitHub Actions, and GitLab CI, enabling developers to securely inject credentials into build and deployment workflows.
- **Access control:** The CLI respects Dashlane's role-based access controls and sharing permissions, ensuring that automation processes can access only authorized credentials.
- **Admin capabilities:** Admins can see Activity Logs, access a list of team members, and view Dark Web Insights reports directly in the CLI. They can also automate tasks, use bots, and export important data for their organization's security information and event management (SIEM) tool.
- **Plan member capabilities:** Plan members can use the CLI to access their vault. They can also use the CLI to securely access secrets from the terminal and inject them into template files, environment variables, or the pastebin.

Comprehensive usage documentation, including installation and command references, is available in Dashlane's GitHub CLI repository and support article.

5.9.2 Public API

The **Dashlane Public API** provides a secure interface for enterprise integrations such as reporting automation and identity lifecycle management.

- **Authentication and authorization:** The API uses OAuth 2.0 with scoped permissions to control access and ensure compliance with enterprise least-privilege policies.
- **Supported operations:** Includes endpoints for retrieving organizational data such as company account status information, user information, password health metrics, and device information.

- **Data protection:** All API responses are encrypted in transit using TLS 1.3 and signed to ensure data integrity.

Dashlane's Public API documentation and support guide provide full reference schemas, sample code, and integration examples.

5.9.3 Dashlane MCP Server for Audit Logs

The Dashlane CLI includes a Model Context Protocol (MCP) server that exposes audit log data to local AI agents. This enables security and IT teams to natively interact with AI agents and query credential activity, investigate incidents, and run compliance checks using natural language, without exporting logs to external systems or building custom integrations.

Architecture Overview

The MCP server is not a standalone hosted service. It runs as a local process, launched by the Dashlane CLI (`dcli`) on the operator's machine, and communicates with AI agents over stdio transport. No audit log data is transmitted to a third-party AI provider. The AI agent queries the MCP server locally, and only the query results are passed to the model in context.

```
AI Agent (e.g. Claude Code)
      |
  stdio transport (local)
      |
  dcli team mcp  <-- MCP server process
      |
  Dashlane API  <-- authenticated with CLI keys, per-tenant scoped
```

The MCP server exposes audit logs as structured, read-only resources. It does not provide write access to any Dashlane data.

Authentication and Authorization

Access to the MCP server requires valid **Business CLI keys** (`DASHLANE_ENROLLED_TEAM_DEVICE_KEYS`). These keys are scoped to a specific team tenant and must be generated by an admin with CLI key generation permissions.

CLI keys can be passed at runtime or embedded in the MCP configuration. Dashlane enforces per-tenant authorization server-side: a CLI key for one team cannot access the data of another.

There is no unauthenticated access path to the MCP server. Any AI agent connecting to it must operate in an environment where valid CLI credentials are present.

Data Access Scope

The MCP server provides access to the same audit log data available through `dcli team logs`, subject to the same controls.

Deployment Considerations

- The MCP server process runs on the same machine as the AI agent. Admins should evaluate the trust level of that environment before enabling sensitive log access.
- CLI keys should be treated as secrets and stored accordingly (e.g., in a secrets manager or environment variable store with restricted access).
- Because the MCP server fetches logs on demand from the Dashlane API, it does not persist any log data locally beyond what the AI agent processes in-session.

For setup instructions, see the AI Agents integration guide.

5.9.4 Automation and Lifecycle Management

Enterprises can leverage the CLI, MCP and API together to automate credential management and compliance reporting:

- Automatically rotate credentials used in service accounts.
- Audit password health across departments.
- Feed activity logs and credential risk logs directly into SIEM and governance systems.

5.10 Integrations

Dashlane offers a broad ecosystem of **enterprise integrations** designed to connect seamlessly with existing IT, identity, and security infrastructures. These integrations help

organizations streamline deployment, simplify user management, and enhance visibility across their security operations, all while preserving Dashlane's zero-knowledge guarantees.

5.10.1 Identity and Access Management (IAM)

Integrate Dashlane with your organization's **identity provider (IdP)** to enable SSO-based authentication and automated provisioning:

- **Supported SSO providers:** Microsoft Entra ID (formerly Azure AD), Okta, Ping Identity, Google Workspace, JumpCloud, Duo, Keycloak, and any IdP supporting SAML 2.0.
- **SCIM provisioning:** Automate user lifecycle management with SCIM integrations for Microsoft Entra ID, Okta, Ping Identity, JumpCloud, and Duo.

These integrations simplify onboarding and deprovisioning, ensuring that users gain or lose access automatically in line with corporate identity policies.

5.10.2 Multi-Factor Authentication (MFA)

Dashlane supports integration with popular MFA solutions to enforce strong authentication policies. Supported second-factor apps include:

- **TOTP authenticators:** Google Authenticator, Microsoft Authenticator
- **Enterprise MFA platforms:** Duo Security and other FIDO2-compliant devices

5.10.3 Endpoint and Browser Management

Dashlane integrates with leading **endpoint and device management solutions** to simplify organization-wide deployment:

- **Jamf** for macOS (Google Chrome and Mozilla Firefox)
- **Intune or Group Policy Objects (GPO)** for Windows (Edge, Chrome, Firefox)
- **Browser compatibility:** Dashlane Smart Extension supports all major browsers, Safari, Chrome, Edge, Firefox, Opera, Brave, and other Chromium-based browsers

5.10.4 Security Monitoring and SIEM

To support enterprise monitoring and compliance, Dashlane integrates with **Security Information and Event Management (SIEM)** platforms:

- **Pre-built connectors:** Splunk
- **Custom SIEM integration:** Available via the **Dashlane CLI**, allowing export of logs and telemetry for correlation with existing IT data sources

5.10.5 Productivity and Communication Tools

Dashlane integrates directly with collaboration tools to improve employee engagement and password hygiene:

- **Slack:** Dashlane integrates with Slack to send Nudges, secure, context-aware notifications to employees.
- **Credential import/export:** Seamlessly migrate existing credentials from Chrome, LastPass, or CSV files. On iOS, a secure, seamless protocol is available through the Credential Exchange protocol.

5.10.6 Developer and Custom Integrations

Dashlane's **Public API** and **CLI** allow organizations to create custom integrations for DevOps and security automation:

- Build internal tools to automate credential rotation and reporting.
- Integrate with CI/CD pipelines for secure secret injection.
- Extend security visibility by connecting Dashlane telemetry with SIEM tools or related monitoring tools, custom dashboards, and compliance systems.

5.10.7 Benefits of Integration

Dashlane integrations deliver:

- **Faster deployment:** Deploy organization-wide in minutes through SSO and endpoint management.
- **Reduced IT overhead:** Eliminate manual credential onboarding and offboarding.
- **Improved security posture:** Strengthen identity management and streamline threat monitoring via native integrations with IAM, SIEM, and MFA platforms.

By connecting Dashlane with existing enterprise systems, organizations achieve unified, scalable credential protection while maintaining seamless workflows for employees and administrators.

6. Phishing Resistance

Summary

Dashlane offers a comprehensive, phishing-resistant architecture that eliminates reliance on passwords.

- **Passkeys**: Built on FIDO2/WebAuthn standards and stored within AWS Nitro Enclaves, passkeys stored using Dashlane are portable, origin-bound, and protected using confidential computing.
- **Passwordless Dashlane sign-in**: A high-entropy, encryption key replaces the traditional user-selected Master Password.
- **Sign-in with FIDO2 security keys**: Provide hardware-based vault encryption using WebAuthn PRF extension, ensuring keys are derived within physical devices and never exportable.
- **Infrastructure resilience**: Anti-enumeration protections, domain-level blacklisting, and in-browser AI phishing detection defend Dashlane users and systems from impersonation and spoofing attempts.

Together, these controls deliver end-to-end phishing resistance, from user authentication to brand protection, within Dashlane's zero-knowledge, confidential-computing ecosystem.

6.1 Passkeys

Dashlane's **passkey implementation** brings phishing-resistant, passwordless authentication into its zero-knowledge architecture. Unlike traditional password managers that store passkeys alongside regular credentials, Dashlane uses a **confidential computing model** to ensure that cryptographic key material is protected from device compromise.

- **Phishing-resistant by design:** Passkey authentication is based on origin-bound credentials; traditional phishing attacks are ineffective.

- **Confidential computing protection:** AWS Nitro Enclaves isolate cryptographic passkey operations, and the private key never leaves the secure cloud environment during creation and usage.
- **Standard-based**: Passkeys are built on top of the W3C WebAuthn and FIDO CTAP standards and supported by the industry through the FIDO Alliance.

Each passkey consists of a **public/private key pair** generated using the WebAuthn standard. Dashlane does not store the private key on the user's device, it remains encrypted on the server.

Upon a request to generate a new passkey for a domain, the client sends a generation request to the AWS Nitro Enclave through a secure channel (as explained in 3.5.3 Secure Channels). The enclave generates both the passkey public/private key-pair and a symmetric encryption key. The private key is encrypted within the enclave using this symmetric key (AES-256-CBC + HMAC-SHA256). The symmetric key is then returned to the client to be stored in the user's locally encrypted vault, while the encrypted private key is stored in a classic datastore.

When a user attempts to log in, the client forwards the WebAuthn challenge, the clientDataJSON (containing the origin and challenge), and the symmetric encryption key to the enclave. The enclave uses the symmetric key to decrypt the passkey's private key. It then validates that the origin within the clientDataJSON matches the intended domain before signing the data. The resulting signature is forwarded back to the client, which completes the authentication as if the passkey were stored locally.

6.2 Passwordless Login

Dashlane's **passwordless login** eliminates the Master Password while maintaining the same cryptographic strength and zero-knowledge guarantees that define its security model. Instead of a user-defined password, each account is protected by a **Machine-Generated Master Password (MachineGeneratedMP)**, a high-entropy secret generated and stored securely on the user's device.

At account creation, Dashlane generates the MachineGeneratedMP locally and uses it to derive the user's AES-256 vault encryption key. The MachineGeneratedMP never leaves the user's device in plaintext and is never stored on Dashlane's servers. On mobile and desktop, this key is protected by the platform's **hardware security module (HSM)**, such as the Apple Secure Enclave or Android Keystore, and accessed only when the user authenticates with a biometric factor (Face ID, fingerprint) or PIN.

When a passwordless user adds a new device, they use an existing, logged-in device to complete the setup. This secured process is described in section 4.1.3.

6.3 FIDO2 Security Keys

Dashlane supports FIDO2-compliant security keys—such as YubiKeys and other hardware authenticators—to provide the highest level of phishing-resistant authentication for both enterprise and individual users. These hardware devices integrate with Dashlane's zero-knowledge architecture to deliver cryptographic, hardware-backed authentication as a primary factor for accessing the Dashlane application.

Passkeys stored on FIDO2 security keys extend the inherent phishing-resistant properties of FIDO2 by enabling the derivation of a credential-specific symmetric key through the WebAuthn PRF and CTAP hmac-secret extensions. Dashlane leverages these standards-based extensions to derive encryption keys that protect and encrypt the user's vault, ensuring that sensitive data remains accessible only to the authenticated user.

When a FIDO2 security key is used in this configuration, users are required to configure a local device PIN on the security key. This enforces a strong, multi-factor authentication model that combines something the user has (the hardware key) with something the user knows (the PIN), further safeguarding access to Dashlane credentials.

Through native support for FIDO2 hardware security keys, Dashlane delivers robust, standards-based, phishing-resistant authentication and encryption, strengthening the overall security posture of the Dashlane vault.

6.4 Preventing Phishing Attacks Against Dashlane

Dashlane employs multiple layers of defense to protect its users and infrastructure from phishing attacks, ensuring that neither Dashlane accounts nor its authentication workflows can be exploited.

6.4.1 Domain-Level and Application-Level Phishing Protection

Dashlane maintains continuous monitoring and **deny-list of malicious domains** that impersonate Dashlane or its services. This protection operates at multiple levels:

- **Application-level protection:** The Dashlane extension and mobile apps automatically prevent autofill on suspicious or impersonated domains.

- **DNS and infrastructure controls:** Dashlane enforces strict domain verification, HSTS preloading, and DMARC/SPF/DKIM policies to prevent spoofed emails or subdomain takeovers.
- **Threat intelligence and takedown:** Dashlane collaborates with external threat intelligence partners to identify and remove phishing domains targeting Dashlane users.

6.4.2 In-Browser Phishing Prevention

Within the Dashlane client, phishing protection mechanisms ensure users are never prompted for sensitive information outside trusted contexts. Autofill operates under strict **same-origin and content security policies**, meaning credentials can only be filled into verified sites that match their stored domains. In addition, the **AI-powered phishing detection system** continuously analyzes URLs and SSL certificates to prevent autofill on suspicious pages.

6.4.3 Continuous Monitoring and Response

Dashlane's Security team operates a dedicated monitoring and incident response process to detect phishing attempts against both users and Dashlane's brand. Alerts from domain monitoring, email authentication failures, and reported user incidents are triaged to ensure rapid mitigation.

7. Attack Scenarios and Threat Model against Dashlane

Summary

Dashlane's threat model assumes that motivated, well-resourced attackers may target our users, infrastructure, and supply chain.

We design for resilience even in the worst-case scenario, when devices, networks, or servers are compromised.

Our zero-knowledge architecture, separation of encryption and authentication keys, and use of confidential computing ensure that **no attacker, insider, or system breach can decrypt customer vaults**.

7.1 Overview of Dashlane's Threat Model

Dashlane's approach to threat modeling follows the principle of **assumed breach**:
We assume that any layer, client, network, or backend, could be compromised, and we architect the system to contain and neutralize those risks.

We categorize potential threats into five primary classes:

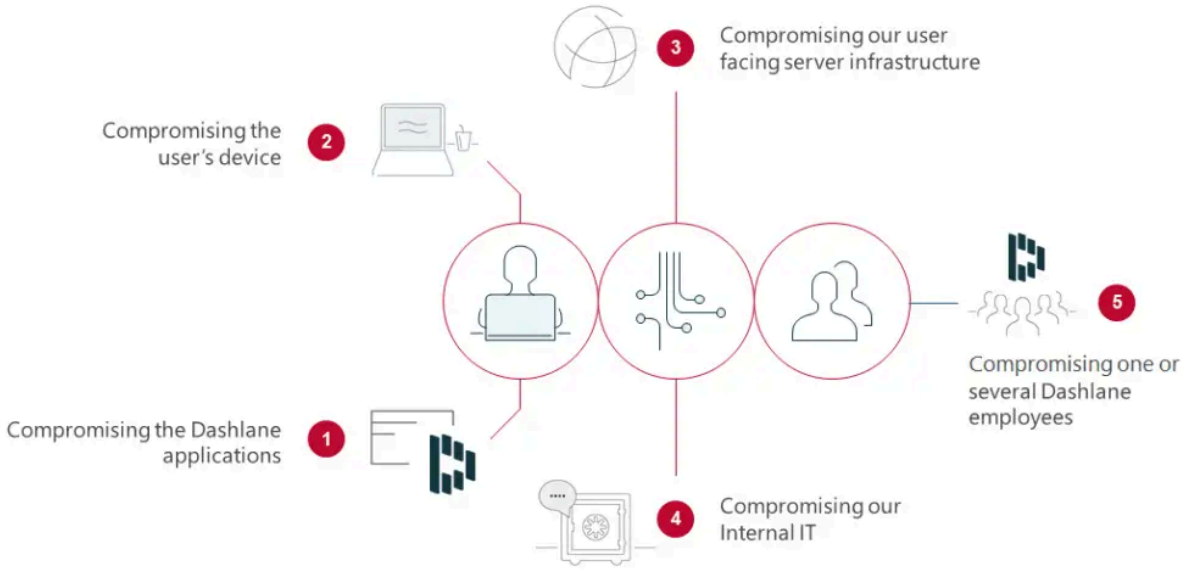


Figure: Dashlane Threat Model

1. **Compromise of Dashlane applications**
    - *Risk:* Exploiting vulnerabilities in our browser extension, mobile, or desktop apps
    - *Mitigations:* Secure SDLC, static and dynamic analysis, dependency scanning, formal threat modeling, penetration testing, and a public bug-bounty program
2. **User device compromise**
    - *Risk:* Malware, keyloggers, or OS-level exploits capturing user inputs or decrypted vault data
    - *Mitigations:* Local AES-256 encryption, Argon2 key derivation, memory obfuscation, and passkey-based phishing-resistant authentication; vaults are only decrypted in volatile memory, and sensitive data is wiped after use
3. **Server or infrastructure breach**
    - *Risk:* Attackers gaining access to backend systems or databases
    - *Mitigations:* Zero-knowledge design ensures encrypted vaults are useless without local keys; confidential computing enclaves (AWS Nitro Enclaves) isolate cryptographic operations so even an infrastructure compromise yields no plaintext data
4. **Insider or supply-chain attack**
    - *Risk:* Unauthorized access through compromised build pipelines or privileged insiders

- *Mitigations:* Segmented infrastructure, least-privilege access, MFA on all admin systems, code-signing verification, reproducible builds, and continuous integrity checks; no employee has the technical capability to decrypt a user vault

5. **Social engineering and human-factor exploitation**
    - *Risk:* Phishing, coercion, or insider manipulation
    - *Mitigations:* Dashlane employees cannot impersonate users or reset passwords; multiple engineers are required to ship an update to a Dashlane client application; Dashlane employees undergo regular security awareness training.

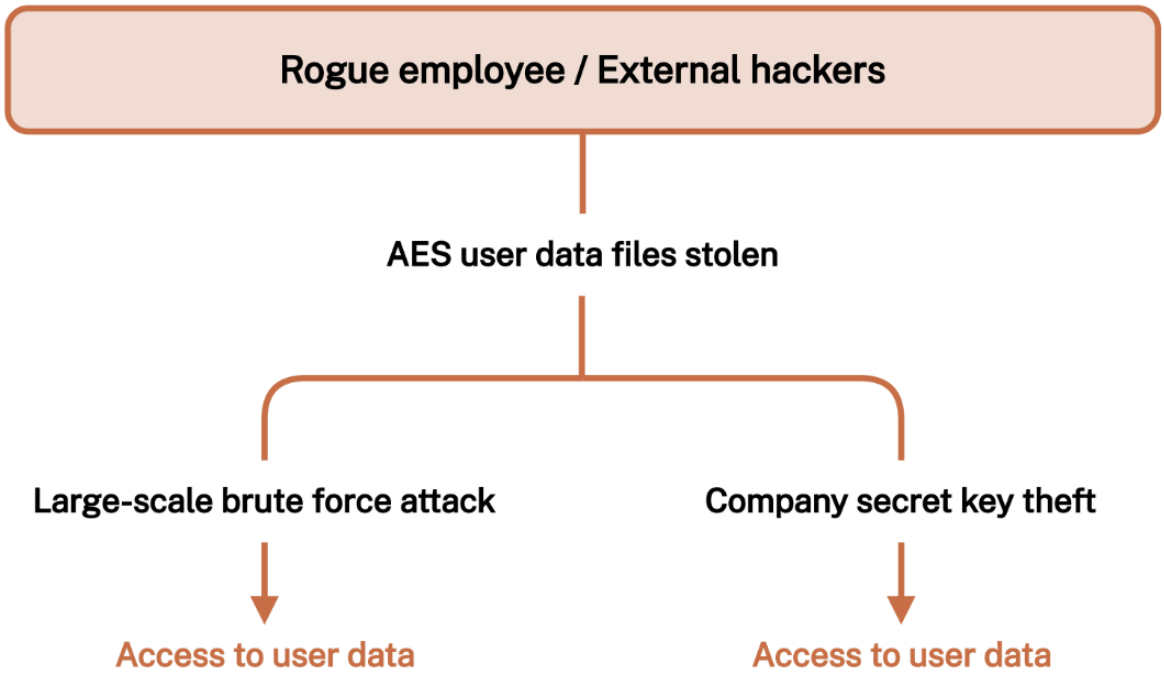
7.2 Comparative Security Architectures

The following sections illustrate how Dashlane's zero-knowledge architecture improves upon legacy and common password manager designs, mitigating the primary risks of centralized encryption and authentication coupling.

7.2.1 Minimal Security Architecture

In traditional cloud services, all user data is encrypted using a single, centrally managed key. This approach simplifies implementation but exposes catastrophic risk: If the key or the storage system is compromised, every user's data becomes accessible.

Attack vector: Theft of the central encryption key or data store
Impact: Complete compromise of all customer data
Example outcome: A single insider or attacker with access to that key can decrypt every stored credential



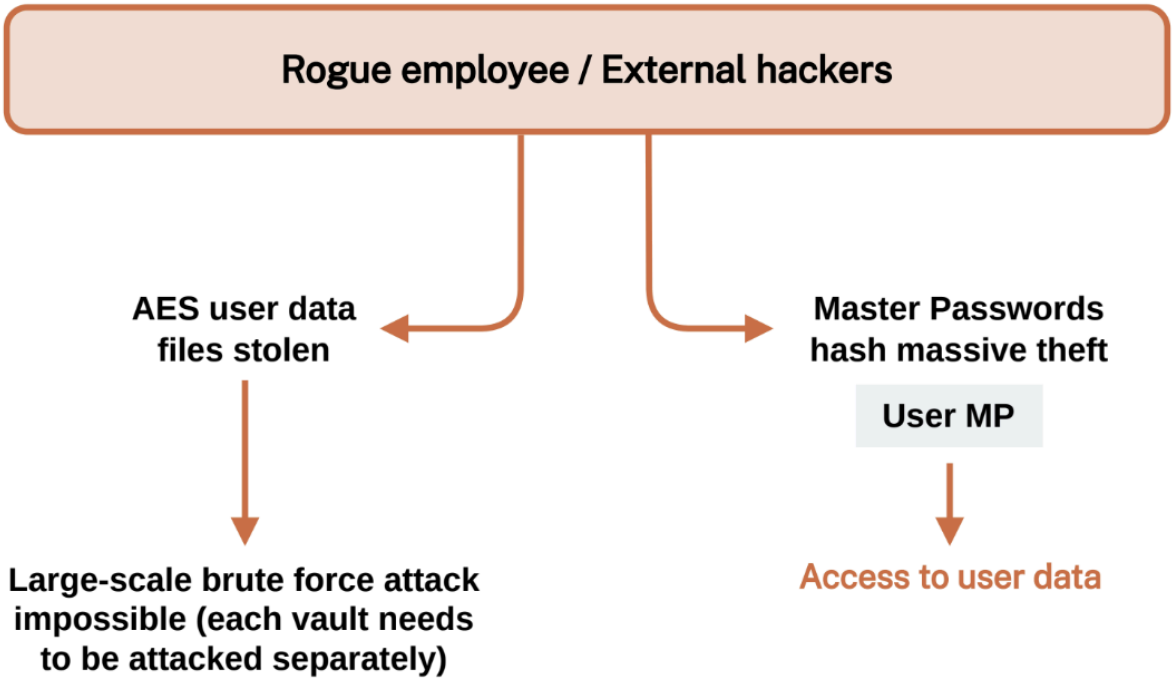
7.2.2 Common Cloud Architecture

A more secure but still vulnerable model is to derive encryption keys from each user's master password. This provides isolation between users, yet many password managers also use this same key to authenticate the user with their cloud services.

This coupling of encryption and authentication introduces several weaknesses:

- Exposure of password hashes or salts during server breaches
- Implementation errors such as missing salts or weak key derivation (e.g., PBKDF2 misconfigurations)
- Brute-force feasibility if user master passwords are weak or reused

Attack vector: Server compromise or credential reuse leads to vault decryption attempts
Impact: Partial data exposure; each user's vault must be attacked individually, but at a lower computational cost



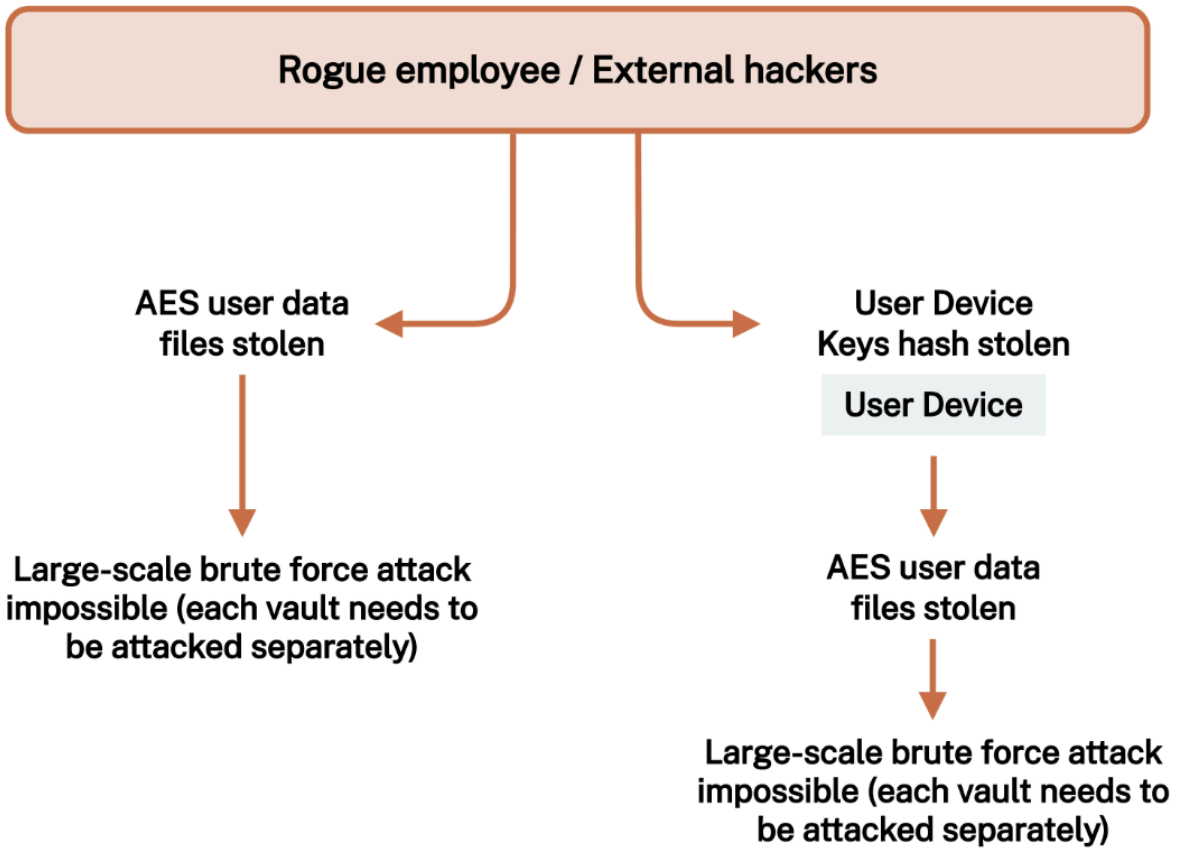
7.2.3 Dashlane's Zero-Knowledge Architecture

Dashlane eliminates these systemic weaknesses by fully separating encryption from authentication:

- **Vault encryption:** Performed locally on the device with a key derived from the user's Master Password or Machine-Generated Master Password.
- **Server authentication:** Managed through a distinct, per-device Device Key that is never shared between devices or transmitted to Dashlane servers in plaintext.

This dual-key design ensures that even if Dashlane's infrastructure were breached and all encrypted vaults and Device Key hashes stolen, attackers could not decrypt vault data:

- Each vault would require an independent brute-force attack protected by Argon2 key stretching and per-user salts
- No centralized secret or shared key exists to compromise multiple users
- Device-based authorization prevents vault decryption without control of a previously trusted device
- Cryptographic operations in the cloud run inside isolated AWS Nitro Enclaves, protected by confidential computing attestation



Security properties:

- No Dashlane employee or system can access customer vaults, by design
- Master Passwords and derived keys never leave the device or transit in plaintext
- Separation of duties ensures server breaches do not endanger encryption keys
- Even under a complete infrastructure compromise, user vaults remain cryptographically opaque

7.3 Clickjacking & Cross-Site Attacks

Dashlane integrates with browsers but enforces strict controls to prevent websites from triggering unauthorized actions.

Defenses:

- **Same-origin policy:** Credentials saved for one domain cannot autofill on another
- **HTTPS-only autofill:** Dashlane never autofills credentials on non-secure (HTTP) sites

- **Anti-clickjacking:** Pop-ups and autofill prompts cannot be controlled by page JavaScript, ensuring users must perform explicit actions
- **Content-Security Policy (CSP):** Enforces browser-level boundaries for web extension components

7.4 Memory Attacks

If an attacker gains full control of a device, they might attempt to read vault contents from memory.

While no application can fully defend against a rooted or jailbroken environment, Dashlane minimizes risk through:

- **Volatile decryption:** Vault data exists only in RAM during active sessions
- **Automatic memory wiping:** Sensitive buffers are zeroized on logout or app lock
- **OS-level isolation:** Mobile operating systems prevent inter-process memory reads
- **Hardware protections:** Use of secure enclaves (e.g., iOS Secure Enclave, Android TEE) for cryptographic operations
- **Confidential computing:** Extends these protections to the cloud for cryptographic workloads

Even under extreme conditions, a rooted device, a compromised OS, or a server breach, Dashlane's layered defenses maintain data confidentiality

7.5 Protection of public keys

When a user shares an item in Dashlane, the item's symmetric encryption key is encrypted with the recipient's public key. Only the recipient's corresponding private key can decrypt the item key and access the shared data.

This ensures:

- End-to-end encryption of shared content.
- Dashlane cannot decrypt shared items.
- Access is cryptographically restricted to intended recipients.

To encrypt for a recipient, the sender's client retrieves the recipient's public key from Dashlane's service.

As with most server-mediated end-to-end encrypted systems, this creates a structural dependency on the authenticity of the public key directory. If a highly privileged attacker were able to compromise the service and substitute a recipient's public key, the sender could

unknowingly encrypt data to an attacker-controlled key. This is commonly referred to as a key substitution attack.

Strong, user-verifiable public key authentication at global scale remains a difficult industry-wide problem. There is no broadly adopted solution that delivers strong cryptographic guarantees without introducing usability, operational, or additional trust trade-offs.

Dashlane mitigates this risk through hardened infrastructure, strict access controls, and continuous monitoring.

Dashlane does not currently provide user-facing public key fingerprint verification in standard sharing flows. We do not claim independent, user-verifiable authentication of recipient public keys.

Dashlane is actively evaluating advanced approaches such as Key Transparency and related mechanisms. These techniques aim to make public key directories auditable and tamper-evident, reducing reliance on implicit server trust while preserving usability.

Any adoption will be carefully assessed against our requirements for security, privacy, performance, and user experience.

Security is an ongoing process. We are transparent about the boundaries of our current design and committed to strengthening public key authenticity over time.

7.6 Transaction Replay and Vault Integrity

Dashlane's synchronization model relies on encrypted transactions stored and processed by the server. These transactions allow clients to reconstruct the current state of a user's vault and enable recovery in case of device loss or local failure.

Security research has noted that because transaction keys are not uniquely bound to each individual transaction, a malicious server could theoretically replay, reorder, or drop encrypted transactions without immediate client-side detection.

This class of issue represents an integrity risk, not a confidentiality breach.

- It does not allow the server to decrypt vault data.
- It does not expose secrets or cryptographic keys.
- The potential impact is limited to inconsistent state or loss of items.

In the most extreme scenario, a compromised server could cause a vault to reflect an earlier or altered state by manipulating the transaction history.

Dashlane explicitly accepts this risk as part of a broader architectural trade-off.

The transaction-based model enables:

- Reliable multi-device synchronization.
- Recovery of historical data.
- Restoration of customer vaults in real-world corruption or failure scenarios.

The same properties that allow replay at the protocol level also allow us to preserve and reconstruct vault history for user protection and operational resilience.

8. Security Operations

Summary

Dashlane's security operations focus on **resilience, transparency, and ongoing improvement**, ensuring enterprise trust through proactive defense and clear communication.

- **Vulnerability management:** Continuous scanning, patching, and dependency checks minimize exposure. Secure SDLC practices and code reviews catch issues early.
- **Incident response:** 24/7 monitoring, a tested IR plan, and a **Code Red + Crisis Communication Plan** enable fast containment and transparent updates, rehearsed annually
- **Bug bounty:** Dashlane's decade-long **HackerOne** program engages global researchers, integrates findings directly into engineering, and accelerates remediation
- **Penetration testing:** Independent annual tests cover all apps and cloud infrastructure. Results are remediated and shared via the **Trust Center**.
- **Continuous improvement:** Dashlane's **Risk Committee** drives regular updates and hardening
- **Code transparency:** Portions of Dashlane's code (iOS, Android, and web extension) are publicly available, enabling independent verification of our zero-knowledge model

Dashlane's defenses evolve continuously, combining prevention, fast response, and openness to keep enterprises secure.

8.1 Secure by Design

Dashlane's approach to security is rooted in the **Secure by Design** philosophy, embedding protection and resilience into every stage of development and operation. Dashlane has formally signed the U.S. CISA Secure by Design pledge, reinforcing its commitment to proactive risk management, transparency, and continuous improvement.

This chapter covers how Dashlane approaches security operations to uphold the highest security standards and ensure that Dashlane remains **resilient by design**, aligning with industry best practices while maintaining the trust of enterprise customers.

8.2 Vulnerability Management

Dashlane applies a structured vulnerability management program designed to identify, assess, and remediate risks before they can be exploited.

Key practices:

- **Continuous scanning and patching:** Automated tools monitor infrastructure and applications for vulnerabilities. All findings are triaged and remediated according to severity and impact.
- **Secure SDLC:** Security is integrated throughout the Software Development Life Cycle (SDLC). Developers follow secure coding guidelines and mandatory code reviews, complemented by automated static and dynamic analysis tools.
- **Third-party library monitoring:** All dependencies are tracked for vulnerabilities using industry-standard tooling. Critical updates are prioritized to maintain a minimal exposure window.
- **Dependency verification:** Package integrity and signature validation ensure that only trusted components are used within Dashlane's codebase.

8.3 Incident Response

Dashlane maintains a formal, tested Incident Response (IR) plan to ensure preparedness for potential security incidents.

Core elements:

- **Continuous security monitoring:** Dashlane's systems are continuously monitored through SIEM integrations, log analytics, and alerting pipelines.
- **Clear roles and escalation paths:** Dedicated incident response and security engineering teams coordinate detection, containment, eradication, and recovery phases.
- **Code Red Plan:** In the event of a critical security incident or breach, Dashlane activates a dedicated Code Red Plan that defines executive-level coordination, immediate containment measures, and external communications protocols.
- **Communication Crisis Plan:** A companion Crisis Communication Plan ensures clear, timely, and transparent communication with customers, partners, and regulators.

Both the Code Red and Crisis Communication Plans are reviewed annually and rehearsed through a company-wide tabletop simulation.

- **Post-incident transparency:** Following any significant event, Dashlane conducts root-cause analyses and shares relevant information with affected customers in line with our commitment to transparency.

8.4 Bug Bounty & Responsible Disclosure

Dashlane fosters an open and responsible security ecosystem through a public **bug bounty program** hosted on HackerOne.

Program highlights:

- Open to all ethical researchers, focusing on vulnerabilities in Dashlane's applications and APIs
- Rewards are based on impact and severity, encouraging high-quality submissions
- Dashlane commits to **cooperative disclosure**, maintaining clear timelines and communication with security researchers
- Our Responsible Disclosure Policy outlines the principles of safe testing and reporting

After more than a decade of continuous operation, Dashlane's bug bounty program has become a cornerstone of our security culture. As described in our 10-year retrospective, we have received thousands of reports and collaborated with hundreds of security researchers worldwide. The program has evolved with structured triage workflows, higher reward tiers for impactful vulnerabilities, and direct integration with our engineering backlog to ensure rapid remediation. This long-term engagement has helped shape our secure development lifecycle and strengthened the transparency and trust we maintain with the global security community.

This ongoing collaboration with the security community enhances Dashlane's defenses and complements internal security testing.

8.5 Penetration Testing

Dashlane engages **independent, accredited third parties** to perform annual penetration tests.

All findings are reviewed, prioritized, and tracked through remediation until closure. A summary of results is made available to enterprise customers under NDA through the Dashlane **Trust Center**.

8.6 Continuous Improvement

Dashlane views security as an evolving process, not a fixed state. Continuous improvement is embedded in the company's governance and culture.

Governance and iteration mechanisms:

- **Risk Committee:** Cross-functional body that meets monthly to review risks and provide executive sponsorship to initiatives aiming to reduce risk. This forum is led by Dashlane's CEO and includes representatives from across the organization, such as Dashlane's CTO, CISO, General Counsel, CMO, and CPO.
- **Lessons learned loop:** Findings from incidents, penetration tests, and bug bounty reports directly inform architectural updates and operational defenses.
- **Employee education:** All staff participate in ongoing security training, phishing simulations, and secure coding refreshers.

8.7 Code Auditability and Transparency

Dashlane's commitment to transparency extends to its engineering practices:

- **Publicly available codebases:** Large portions of Dashlane's client code are open source and available on GitHub:
  - iOS
  - Android
  - Web extension
- **Verifiable integrity:** Customers and researchers can independently review the code to validate our design choices and zero-knowledge implementation

8.8 End of Life and continuous deprecation

Dashlane manages the lifecycle of its products, features, and technical components with the same security discipline applied to active systems. Outdated client versions, deprecated APIs, and unmaintained code represent active security risk through unpatched vulnerabilities, widened attack surface, and inconsistent security behavior across platforms.

Deprecation at Dashlane is continuous, not episodic. This applies to both planned End-Of-Life (EOL) milestones for our client applications, deprecation of features, such as the sunset of the

standalone web app or our password changer feature, and reactive deprecations driven by external constraints, including OS version minimums enforced by Apple and Google, browser policy changes, or third-party ecosystem shifts.

Key practices:

- **Attack surface reduction:** Components that can no longer be maintained to Dashlane's security standards are retired on a defined schedule, before they become exploitable.
- **Bounded backward compatibility:** Dashlane maintains backward compatibility during transition periods, but support windows are time-limited and communicated in advance.
- **Ecosystem alignment:** Where platform vendors impose constraints, Dashlane aligns its deprecation schedule accordingly.

8.8.1 End of Life of client applications

Dashlane enforces a 12-month support lifecycle for all client application versions. Dashlane provides progressive in-app notifications starting at 3 months without an update, escalating in urgency through the 9-12 month window. Once a version exceeds that threshold, the client application is automatically deactivated server-side: affected devices are logged out and prevented from reconnecting until the user updates. The same policy applies to deprecated OS versions, when Dashlane drops support for an older OS release (following Apple's annual OS cycle, for example), users who cannot upgrade their OS will lose access to app updates and are subject to the same 12-month end-of-life clock.

The primary driver behind this policy is security. A password manager handles highly sensitive credential data, and outdated app versions cannot receive security patches. Unpatched clients represent an active attack surface regardless of official support status. Attackers do not distinguish between supported and unsupported versions. Maintaining live connections from old versions also forces the engineering team to preserve backward compatibility in server-side logic, increasing the risk of unintended exposure through legacy code paths.

8.8.2 Deprecating features and code

Keeping a large codebase healthy requires deliberate, ongoing deprecation of outdated code, APIs, frameworks, and dependencies. At Dashlane, this covers a range of scenarios: retiring legacy API endpoints in favor of improved versions, removing feature flags for experiments that have concluded, upgrading core frameworks and third-party libraries, and eliminating dead code paths that accumulate over time.

The security dimension is central here. Outdated dependencies are one of the most common vectors for supply chain attacks. A library that is no longer maintained stops receiving CVE patches, and any vulnerability it carries becomes a permanent liability. Dashlane treats dependency hygiene as a core security practice.

8.8.3 Data retention

Dashlane applies a data retention policy to limit the attack surface associated with dormant user data. For personal accounts, user vaults that remain inactive for more than 13 months are subject to automatic deletion. Before any account is removed, users receive advance notification at 30 days and again at 10 days, with a 30-day grace period after the deletion date during which access can still be restored.

Deleting obsolete data reduces the overall data footprint Dashlane must protect and limits the blast radius of any potential exposure. This is following a principle of data minimization.

9. Compliance & Certifications

Summary

Dashlane aligns with globally recognized security and privacy standards to ensure enterprise trust and transparency. Our compliance framework is regularly validated through independent third-party audits and certifications, reflecting our ongoing commitment to robust security governance.

For up-to-date details, visit the Dashlane Trust Center.

9.1 Certifications

SOC 2 Type II
Dashlane undergoes annual SOC 2 Type II audits conducted by independent assessors. These reports validate the design and operational effectiveness of our security, availability, and confidentiality controls across infrastructure, development, and operations.

ISO/IEC 27001
Dashlane maintains an ISO/IEC 27001 certification for its Information Security Management System (ISMS), which governs risk management, control implementation, and continuous improvement across our services and internal processes.

GDPR & CCPA
Dashlane is fully compliant with the EU General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). Our data collection and processing practices are grounded in strong privacy principles, including data minimization, purpose limitation, and user control.

9.2 Data Residency & Privacy

Dashlane is built on a zero-knowledge foundation, ensuring that encrypted user data remains private even within multi-region deployments.

- **End-to-end encryption:** User data is encrypted on the device before synchronization and remains encrypted at rest, in transit, and in use.

- **Secure enclaves for separation of duties:** Cloud secure enclaves (AWS Nitro Enclaves) isolate cryptographic operations, ensuring no Dashlane employee or system can access decrypted data.
- **Data Processing Agreement (DPA):** A DPA is available to enterprise customers, formalizing our commitment to data protection and regulatory compliance.

9.3 Supporting Enterprise Compliance

Dashlane supports organizations in meeting their own compliance objectives through transparent documentation, auditable controls, and integration with enterprise security frameworks.

- **Detailed documentation:** The Dashlane Trust Center provides customers and prospects with self-service access to security information, including technical documentation, audit summaries, encryption architecture documents, and compliance reports for due diligence and vendor assessments.
- **Exportable logs:** Admins can export detailed activity and audit logs for alignment with frameworks such as ISO 27001, SOC 2, NIST 800-53, or CIS.
- **SCIM-based provisioning:** Automated user lifecycle management enforces least privilege and ensures compliance hygiene during onboarding and deprovisioning.
- **Role-based access and SSO:** Granular access controls, combined with SAML-based authentication, strengthen identity assurance and compliance posture.
- **Code auditability:** Dashlane's client applications are source available, allowing external experts and customers to review and validate security practices. Source code is available on GitHub for iOS, Android, and the web extension.

9.4 Patents

Dashlane's security innovations are protected by multiple granted patents and supported by active applications. The list below excludes abandoned filings and reflects our current portfolio.

9.4.1 Granted Patents

Patent Title	Patent Number (Grant Date)	Summary

Cloud-based data backup and sync with secure local storage of access keys	US 9,330,245 (May 3, 2016)	Secure vault synchronization with local key storage in a zero-knowledge model
Master password reset in a zero-knowledge architecture	US 10,432,397 (Oct 1, 2019)	Secure recovery while preserving zero-knowledge guarantees
Methods and systems for user authentication	US 10,574,648 (Feb 25, 2020)	Passwordless authentication via challenge/response
Zero-knowledge architecture between multiple systems	US 10,848,312 (Nov 24, 2020)	Inter-system vault synchronization without exposing keys
Resume user session in a zero-knowledge architecture on an insecure platform	US 10,904,004 (Jan 26, 2021)	Session resumption while maintaining zero-knowledge
Crowdsourced learning engine for semantic analysis of webpages	US 11,080,597 (Aug 3, 2021)	Semantic analysis to drive secure and accurate autofill
Multiple relying parties in a single-sign-on environment	US 12,052,232 (Jul 30, 2024)	SSO architecture supporting multiple relying parties
Integration of Identity Access Management Infrastructure with Zero-Knowledge Services	US 18/124,326 (Oct 21, 2025)	IAM integration with zero-knowledge services in cloud secure enclaves

9.4.2 Active Applications

Application Title	**Application No. (Publication/Status)**	**Summary**
Authentication with Cloud-Based Secure Enclave	**US 18/417,228** , **US-2024-0283664-A1** (published Aug 22, 2024)	Attested cryptographic operations in cloud secure enclaves
Systems and Methods for Analysis of Hypertext Markup Language (HTML Embeddings)	**US 18/735,711** (pending)	Robust analysis of web markup to improve security automation
Device-to-Device Secret Transfer , Systems and Methods to Transfer High-Entropy Keys	**US 18/906,749** (pending)	Secure proximity/remote secret exchange between devices
Systems and Methods for Enhanced Security Using Low-Entropy Secrets on Insecure Environments (PIN on Web)	**US 18/984,084** (pending)	PIN-based protections for web contexts
Zero-Knowledge, Secure and Private Monitoring Channel of Web Activities for Unauthenticated Users	**US 19/267,168** (pending)	Private monitoring channel to enable risk telemetry without vault access
Multi-Factor Anti-Phishing Systems and Methods	**US 63/749,081** (pending)	Multi-signal anti-phishing with ML and crowdsourcing

9.5 Publications

Dashlane promotes transparency and industry collaboration through active knowledge sharing:

- Engineering blog posts: Our Engineering team regularly publishes articles on security, architectural design, and Dashlane innovation and practices on the Dashlane blog.
- Conference participation: Dashlane's Security and Engineering leaders frequently speak at international cybersecurity and technology events to share insights on credential security, zero-knowledge architectures, and digital identity.
- Community engagement: Dashlane is a board member of the FIDO Alliance and contributes to open standards bodies, including the W3C Web Extension Community Group. We participate in secure software design initiatives, including the CISA Secure by Design pledge, reflecting our ongoing commitment to a safer internet ecosystem.

Change History

Date	Description
Sep 2016	Initial public release of Dashlane Security White Paper
Jun 2017	Added secure sharing details
Jan 2018	Updated password hashing and user device key explanations
Nov 2018	Added SCIM provisioning and SSO flow diagrams
Feb 2019	Clarified secure enclave architecture for SSO
Sep 2019	Major update: included passwordless authentication flows
May 2020	Added details on confidential computing and zero-knowledge enforcement
Nov 2020	Added dark web monitoring feature details
Mar 2021	Expanded device linking explanations
Oct 2021	Added credential health and password hygiene controls

Apr 2022	Updated SSO provisioning flow with Nitro Enclave details
Aug 2023	Added admin-assisted account recovery workflows
Jan 29, 2025	Updated for passkey support, advanced phishing protections, credential risk detection
Feb 6, 2026	Enterprise-focused revamp: reorganized for CISOs, expanded Omnix features, enhanced compliance and architecture detail
Apr 13, 2026	Link Based Sharing. MCP Server for Audit Logs. End of Lyfe process. Various minor updates across the document.

Copyright © 2026 Dashlane. All rights reserved.