La Sécurité chez Dashlane
Principes & Architecture

DASHLANE

29 janvier 2025

v2.7.0

Sommaire

Figures

Le gestionnaire de mots de passe Dashlane a été conçu en s'appuyant sur une architecure « zero-knowledge ». Toutes les données sont chiffrées localement sur l'appareil de l'utilisateur. Seul l'utilisateur peut accéder aux données en utilisant un mot de passe ou une autre forme d'authentification. Étant donné que Dashlane n'a pas accès au coffre-fort de l'utilisateur et ne stocke pas le mot de passe Maître de l'utilisateur, les acteurs malveillants ne peuvent pas voler les informations, même si les serveurs de Dashlane sont compromis.

1 Principes généraux de sécurité

Avant de stocker le coffre-fort de chaque individu sur ses serveurs, Dashlane le chiffre en utilisant le chiffrement avancé de la norme de chiffrement (AES) 256 bits. L'accès au coffre-fort nécessite soit un mot de passe Maître de l'utilisateur, qui n'est connu que par le titulaire du compte, soit, pour un utilisateur sans mot de passe, un mot de passe unique généré par machine. Dans les deux cas, ce mot de passe n'est pas stocké sur les serveurs de Dashlane et n'est pas accessible aux employés de Dashlane. Dashlane utilise une clé de l'appareil utilisateur séparée pour authentifier chaque personne sur ses serveurs. Lorsque quelqu'un crée un nouveau compte Dashlane ou active un appareil supplémentaire pour la synchronisation des données, Dashlane vérifie d'abord l'utilisateur autorisé en envoyant un jeton sur l'adresse e-mail enregistrée ou le numéro de téléphone mobile, puis génère automatiquement la clé de l'appareil utilisateur. Pour la connexion sans mot de passe, l'accès à l'appareil supplémentaire est subordonné à l'autorisation d'un appareil déjà enregistré, il n'est donc pas nécessaire d'envoyer le jeton par e-mail ou mobile.

Lorsqu'une personne saisit son mot de passe Maître dans l'application Dashlane, les données sont chargées dans la mémoire de l'appareil autorisé. Pour une sécurité supplémentaire, les individus qui se connectent avec leur mot de passe Maître peuvent lier leurs comptes Dashlane à une application de double authentification (2FA) telle que Google Authenticator. L'activation de l'option 2FA signifie que le mot de passe Maître et le code de l'authenticator sont nécessaires pour déchiffrer le coffre-fort. Toute communication entre l'application Dashlane sur l'appareil local et les serveurs de Dashlane se fait via le protocole cryptographique SSL / TLS. Et bien qu'une variété de processus de sécurité se produisent en arrière-plan lors de l'enregistrement et de l'authentification de l'utilisateur, l'expérience utilisateur est simple et rationalisée. Les administrateurs de compte Dashlane Business peuvent activer une fonctionnalité de récupération de compte optionnelle via leur console d'administration. Cette fonctionnalité permet aux employés de réinitialiser leur mot de passe Maître et de récupérer leurs données tout en préservant l'architecture « zero-knowledge » de Dashlane. Lorsqu'un employé initie la récupération de compte, l'administrateur agit comme le tiers approuvé pour vérifier l'identité de l'utilisateur et approuver la demande. De plus, une clé de récupération de compte est un mécanisme disponible pour tous les utilisateurs qui utilisent le mot de passe Maître et la connexion sans mot de passe pour récupérer l'accès à leur compte en utilisant une clé à usage unique.

1.1 Liste des secrets

Dashlane utilise de nombreux secrets pour sécuriser les données de l'utilisateur. Certains d'entre eux sont décrits comme suit :

Nom de clé	Symbole de clé	Description
Mot de passe Maître de l'utilisateur	$User_{MP}$	Mot de passe / phrase d'accès générée par l'utilisateur Dérive la clé pour chiffrer le coffre-fort de l'utilisateur. Le mot de passe Maître de l'utilisateur doit être aussi aléatoire que possible
Clé intermédiaire	$Intermediate_{Key}$	Clé aléatoire de 32 octets,générée localement sur les appareils Permet le chiffrement local
Clé de l'appareil utilisateur	$Device_{Key}$	Clé aléatoire de 32 octets, générée localement sur les appareils. Sert comme secret d'authenification avec les serveurs
Clé subsidiaire de l'utilisateur	$UserSecondary_{Key}$	Clé aléatoire de 32 octets, générée sur les serveurs lors de l'activation du la 2FA. Renvoyée par le serveur lors d'un validation d'une authentication 2FA
Clé de récupération de compte	$AccountRecovery_{Key}$	Chaine de 28 caractères générée par le générateur de mot de passe ($\simeq$ 145 bits d'entropie) Clé de pour le mécanisme de Recouvrement de Compte
Mot de passe Maître généré par machine	$MachineGenerated_{MP}$	Chaine de 40 caractères générée par le générateur de mot de passe ($\simeq$ 243 bits d'entropie) Sert de clé de chiffrement du Vault pour les comptes MPLess
Clé d'équipe de déploiement de masse	$MassDeploymentTeam_{Key}$	Clé aléatoire de 32 octets, générée par les serveurs de Dashlane, et transmise aux appareils Sert comme secret d'authenification avec les serveurs

Tab. 1 : Aperçu des secrets de Dashlane

1.2 Protection des données utilisateur dans Dashlane

La protection des données des utilisateurs dans Dashlane repose sur 3 secrets distincts :

- Le **mot de passe Maître de l'utilisateur** :
  - ▷ Dashlane utilises la librairie **zxcvbn**[1] pour calculer la robustesse du $User_{MP}$ fournis par l'utilisateur.

    De nombreuses vérifications sont menées sur $User_{MP}$, comme la comparaison avec une liste des mots de passe les plus utilisés ou un calcul de la complexité. Les analyses sont résumées en un score allant de 0 (facilement devinable) à 4 (pratiquement non devinable). Pour les scores les plus faibles, la librarie propose des conseils pour facilement créer des mots de passe plus forts.

    Le $User_{MP}$ fourni par l'utilisateur doit avoir un score minimal de 3 pour être accepté par les applications Dashlane (nombre de choix pour trouver le mot de passe estimé supérieur à 10^8).

  - ▷ Ce mot de passe n'est jamais stocké sur les serveurs de Dashlane et ses dérivés non plus (y compris les hachages).

  - ▷ Par défaut, le mot de passe Maître n'est pas stocké localement sur l'appareil de l'utilisateur. Il sert simplement à déchiffrer les fichiers locaux contenant les données de l'utilisateur.

  - ▷ Le mot de passe Maître est stocké localement à la demande de l'utilisateur

[1] https ://github.com/dropbox/zxcvbn

lorsque celui-ci active la fonctionnalité « Se souvenir de mon mot de passe Maître ».

  - ▷ De plus, nous nous assurons que le mot de passe Maître de l'utilisateur ne soit jamais transmis par Internet. [2]

[2] Le seul dérivé de celui-ci qui est envoyé sur Internet est le coffre-fort chiffré final. Les paragraphes suivants décrivent comment nous assurons sa résilience aux attaques.

- La **clé intermédiaire** : dans certains cas (stockage local), nous utilisons une $Intermediate_{Key}$ chiffrée avec un dérivé du $User_{MP}$.
- La **clé de l'appareil utilisateur**s : clé unique pour chaque appareil activé par un utilisateur :
  - ▷ Généré automatiquement pour chaque appareil.
  - ▷ Utilisée pour l'authentification.
- Le **mot de passe Maître généré par machine** (comme alternative au mot de passe Maître de l'utilisateur) :
  - ▷ Est une chaîne forte et unique générée par une machine de 40 caractères, générée avec le générateur de mots de passe.
  - ▷ Ce mot de passe n'est jamais stocké sur les serveurs de Dashlane et ses dérivés non plus (y compris les hachages).
  - ▷ Par défaut, le mot de passe Maître n'est pas stocké localement sur l'appareil de l'utilisateur. Il sert simplement à déchiffrer les fichiers locaux contenant les données de l'utilisateur.
  - ▷ Il est stocké localement lors de la connexion à l'extension Web de Dashlane.
  - ▷ De plus, nous veillons à ce que le $MachineGenerated_{MP}$ ne soit jamais transmis par Internet. [3]

[3] Le seul dérivé de celui-ci qui est envoyé sur Internet est le coffre-fort chiffré final. Les paragraphes suivants décrivent comment nous assurons sa résilience aux attaques.

1.3 Accès local aux données utilisateur

L'accès aux données de l'utilisateur nécessite l'utilisation du $User_{MP}$, qui n'est connu que par l'utilisateur. Ce mot de passe est utilisé pour générer la clé symétrique AES-256 bits (Advanced Encryption Standard) nécessaire pour le chiffrement et le déchiffrage des données personnelles sur l'appareil de l'utilisateur Dans le cas d'une utilisation sans mot de passe, le $MachineGenerated_{MP}$ n'est pas visible pour l'utilisateur, mais transporté de manière sécurisée entre les appareils lorsque l'utilisateur ajoute un nouvel appareil, puis utilisé exactement comme l'UserMP.

Nous utilisons l'API Web Crypto pour la majorité de la cryptographie basée sur les navigateurs et les bibliothèques natives sur iOS et Android. Nous utilisons la bibliothèque de référence Argon2 compilée en Web Assembly (Wasm) ou liée à l'application mobile.

1.4 Utilisation des données locales après le déchiffrement

Une fois que l'utilisateur a saisi son $User_{MP}$ localement dans Dashlane ou validé son $MachineGenerated_{MP}$ via un code PIN ou ses données biométriques et que ses données utilisateurs ont été déchiffrées, les données sont chargées en mémoire.

Le client Dashlane fonctionne avec des limites significatives pour utiliser les données utilisateur déchiffrées de façon efficace et sûre :

- Les processus de Dashlane accèdent aux mots de passe individuels pour pouvoir les remplir automatiquement sur les sites Web ou enregistrer des identifiants sans que l'utilisateur ait besoin de saisir son $User_{MP}$ ou son $MachineGenerated_{MP}$ à chaque fois.
- La dérivation Argon2d (ou PBKDF2) utilisée pour calculer les clés AES ajoute une certaine latence (le but est de se protéger des attaques par force brute).

  Voir le paragraphe Protection de la mémoire pour plus d'informations sur la gestion de la mémoire.

1.5 Utilisation des applications 2FA pour augmenter la sécurité des données utilisateur

À tout moment, un utilisateur peut lier son compte Dashlane à une application 2FA sur son appareil mobile (par exemple Google Authenticator). Toutes leurs données (les données stockées localement et les données envoyées aux serveurs de Dashlane à des fins de synchronisation) sont ensuite chiffrées avec une nouvelle clé, qui est générée par une combinaison de $User_{MP}$ et une clé générée aléatoirement $UserSecondary_{Key}$ stockée sur le serveur Dashlane, comme décrit dans les étapes suivantes :

- L'utilisateur relie son compte Dashlane à son application de 2FA.
- Les serveurs de Dashlane génèrent et stockent une $UserSecondary_{Key}$, qui est envoyée à l'application client de l'utilisateur.
- Toutes les données personnelles sont chiffrées avec une nouvelle clé AES-256 bit symétrique générée côté client à la fois depuis le $User_{MP}$ et $UserSecondary_{Key}$.
- $UserSecondary_{Key}$ n'est jamais stockée localement.
- La prochaine fois que l'utilisateur essaiera de se connecter à Dashlane, les serveurs de Dashlane lui demanderont de fournir un mot de passe unique généré par l'application de double authentification. Après avoir reçu et vérifié ce mot de passe unique, les serveurs de Dashlane enverront la $UserSecondary_{Key}$ à l'application client, ce qui permettra à l'utilisateur de déchiffrer ses données.

Les données de l'utilisateur ne peuvent être déchiffrées qu'en ayant à la fois le $User_{MP}$ et l'application de double authentification liés au compte de l'utilisateur.

1.6 Authentification

Certains services de Dashlane étant basés dans le cloud (synchronisation des données entre plusieurs appareils par exemple), l'utilisateur doit s'authentifier sur les serveurs de Dashlane.

L'authentification de l'utilisateur sur les serveurs de Dashlane est basée sur la $Device_{Key}$ et n'a **aucun lien avec le mot de passe Maître de l'utilisateur ou** $MachineGenerated_{MP}$.

Lorsqu'un utilisateur crée un compte ou ajoute un nouvel appareil pour synchroniser ses données, une nouvelle clé de l'appareil utilisateur est générée par les serveurs. La $Device_{Key}$ est composée de 40 octets aléatoires générés à l'aide de la fonction RAND_byte OpenSSD. Les 8 premiers octets sont la clé d'accès, et les 32 octets restants sont la clé secrète.

La $Device_{Key}$ est reçue par l'appareil de l'utilisateur et est stockée localement dans les données de l'utilisateur, chiffrées comme toutes les autres données de l'utilisateur, comme expliqué précédemment. Du côté du serveur, la partie clé secrète est chiffrée de sorte que les employés ne peuvent pas se faire passer pour un appareil utilisateur donné. Lorsqu'un utilisateur a obtenu l'accès à ses données en utilisant le $User_{MP}$ ou le $MachineGenerated_{MP}$, Dashlane peut accéder à la $Device_{Key}$ à ses données pour les authentifier sur nos serveurs sans aucune interaction de l'utilisateur.

De ce fait, Dashlane n'a pas à stocker le $User_{MP}$ ou le $MachineGenerated_{MP}$ pour effectuer une authentification.

1.7 Communication

Les communications entre l'application Dashlane et les serveurs de Dashlane utilisent le protocole TLS.

Le domaine dashlane.com utilise le mécanisme de liste HSTS préchargée pour forcer l'utilisation du TLS sur tout le domaine. De plus, Dashlane vérifie que les suites algorithmiques du protocole respectent les recommandations des organismes de sécurités.

Il est important de noter que la sécurité des communications entre applications et services de Dashlane ne repose pas uniquement sur la fiabilité du protocole HTTPS, et que la confidentialité des données n'est pas affectée si le protocole de transport est compromis.

1.8 Détails sur le flux d'authentification

L'inscription initiale pour un utilisateur suit le flux décrit dans Figure 1.

Comme illustré dans la Figure 1, $User_{MP}$ n'est jamais utilisé pour effectuer une authentification sur le serveur, et les seules clés stockées sur nos serveurs sont les clés d'appareil de l'utilisateur.

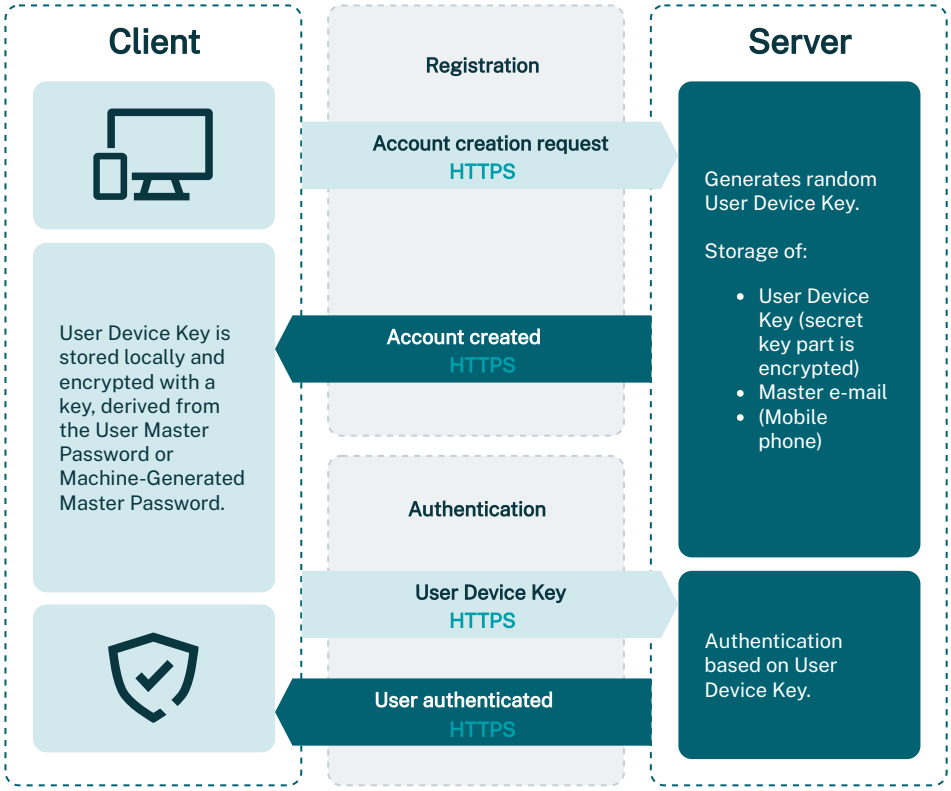


Fig. 1 : Processus d'authentification lors d'une inscription

1.8.1 Ajouter un nouvel appareil pour les utilisateurs qui font usage d'un mot de passe Maître

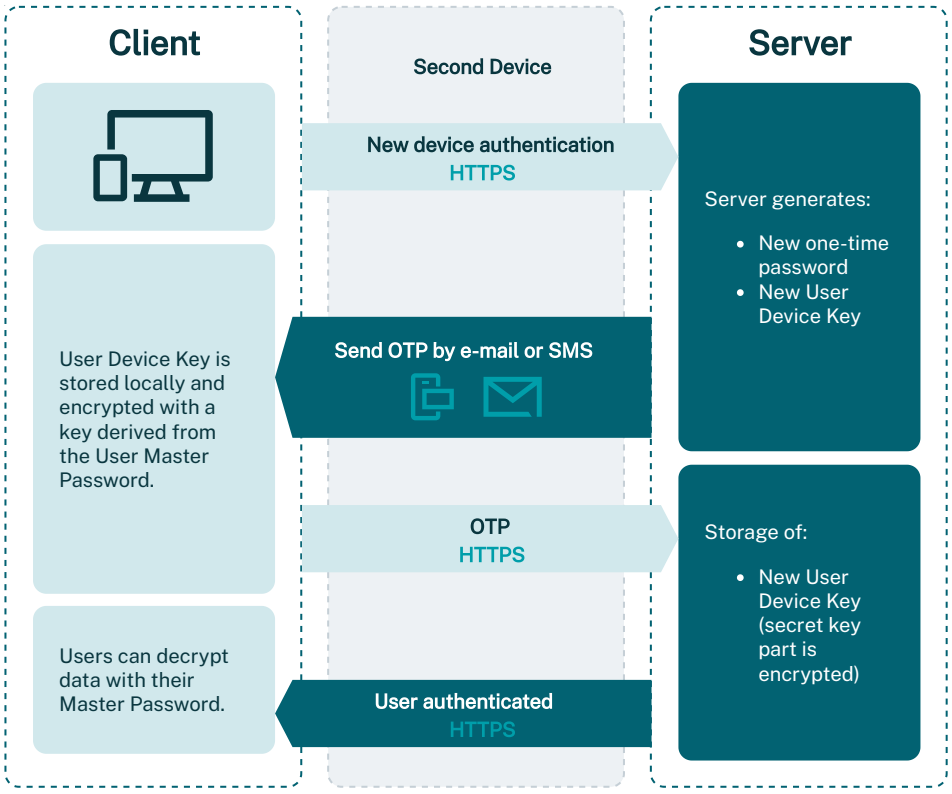


Fig. 2 : Authentification lors de l'ajout d'un nouvel appareil

Lorsqu'un utilisateur ajoute un appareil supplémentaire, Dashlane doit s'assurer que l'utilisateur qui ajoute cet appareil est bien le propriétaire légitime du compte. L'objectif ici est d'ajouter une couche de protection supplémentaire au cas où le $User_{MP}$ a été compromis et que le pirate, qui n'a pas accès à l'appareil déjà activé, essaie d'accéder au compte à partir d'un autre appareil.

Comme le montre le Figure 2, lorsqu'un utilisateur tente de se connecter à un compte Dashlane sur un appareil qui n'a pas encore été autorisé pour ce compte, Dashlane génère un mot de passe unique (un jeton) qui est envoyé à l'utilisateur soit à l'adresse e-mail utilisée pour créer le compte Dashlane initialement ou par SMS au téléphone mobile de l'utilisateur si l'utilisateur a choisi de fournir son numéro de téléphone mobile.

Pour activer le nouvel appareil, l'utilisateur doit saisir à la fois son $User_{MP}$ et son jeton. Ce n'est qu'une fois cette double authentification effectuée que les serveurs de Dashlane commenceront à synchroniser les données de l'utilisateur sur le nouvel appareil. Toute la communication est gérée par HTTPS, et les données de l'utilisateur ne voyagent que sous un formulaire chiffré AES-256. Veuillez noter que le $User_{MP}$ ne transmet jamais sur le Web.

1.8.2 Ajouter un nouvel appareil pour les utilisateurs sans mot de passe

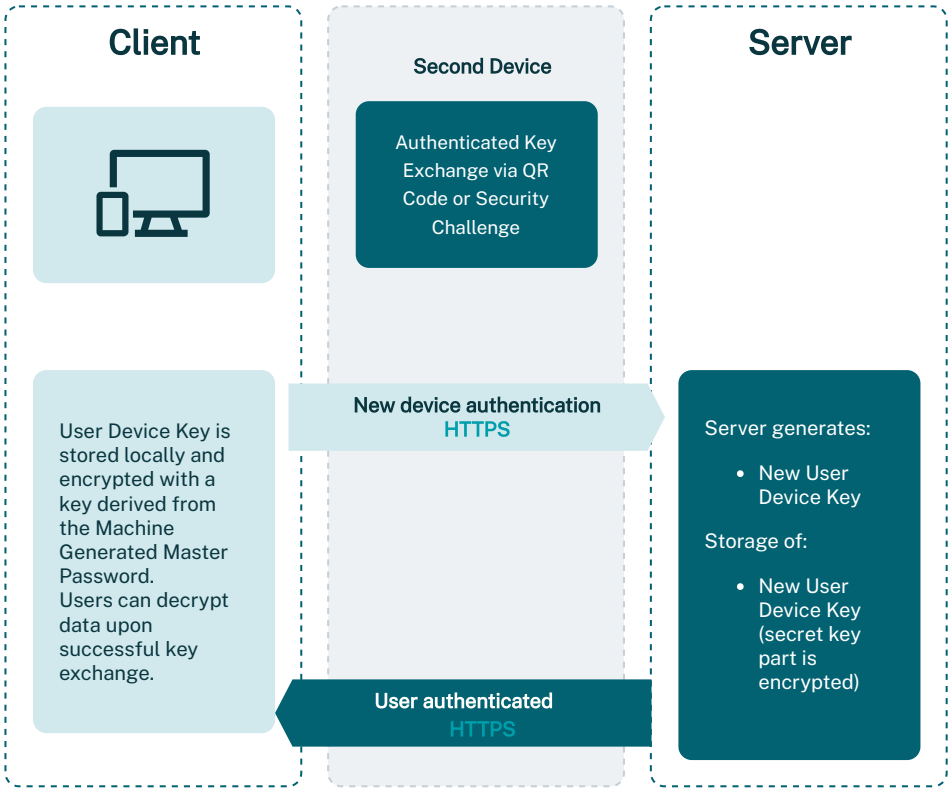


Fig. 3 : Authentification lors de l'ajout d'un nouvel appareil - Flux sans mot de passe

Lorsqu'un utilisateur sans mot de passe ajoute un nouvel appareil, il peut utiliser un appareil connecté existant pour terminer le processus de configuration. En fonction du type d'appareil connecté, l'utilisateur peut soit terminer la configuration de son nouvel appareil avec un balayage de code QR, soit terminer un défi de sécurité. L'objectif de l'échange est de transmettre de manière sécurisée le $MachineGenerated_{MP}$ à partir d'un appareil déjà approuvé vers un nouvel appareil.

Cet échange de clé est basé sur la cryptographie de courbe elliptique, en utilisant la Curve25519.

1.8.2.1 Transfert de proximité avec le balayage de code QR

Si un utilisateur sans mot de passe a un appareil mobile connecté, un balayage de code QR peut être utilisé pour ajouter un nouvel appareil. Lorsqu'un utilisateur saisit son adresse e-mail dans le nouvel appareil (non approuvé), une paire de clés X25519 est générée sur l'appareil et la clé publique est affichée sur l'écran comme un code QR. Ce code QR doit être scanné par un appareil connecté (approuvé). Après un échange de clé réussi, les deux appareils génèrent le même secret partagé, dérivé d'une clé cryptographique, qui sera utilisée pour chiffrer / déchiffrer le $MachineGenerated_{MP}$ transmis entre les appareils. Le coffre-fort peut ensuite être déchiffré localement sur le nouvel appareil.

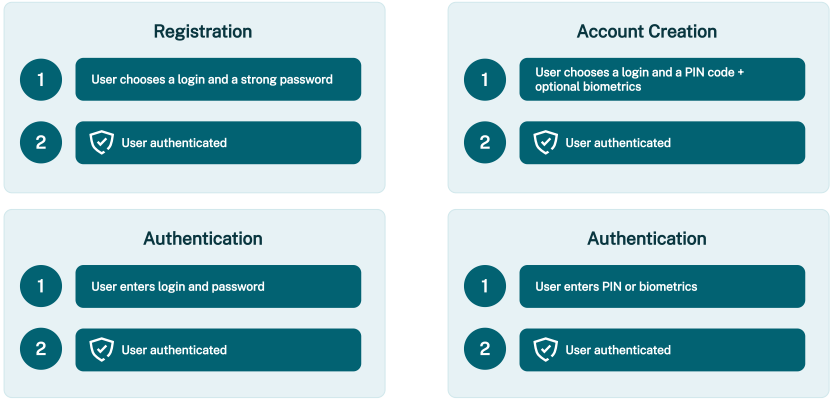
1.8.2.2 Échanger via le serveur avec une vérification visuelle

Si un utilisateur sans mot de passe ne dispose pas d'un appareil mobile connecté ou ne peut pas utiliser les fonctionnalités de l'appareil, un défi de sécurité peut être effectué. Sans la possibilité d'utiliser la proximité pour échanger le secret, les deux appareils doivent utiliser le serveur pour transporter les clés publiques. Dashlane garantit qu'un pirate ne peut pas altérer les clés lors de l'échange en authentifiant l'échange de clé avec la chaîne authentifiée courte :

- Du secret partagé (résultat de l'échange de clé), nous dérivons une clé vue comme une source d'entropie pour choisir cinq mots aléatoires dans une liste de mots.
- La liste de mots est https://www.eff.org/files/2016/07/18/eff_large_wordlist.txt.
- Si l'échange de clé n'a pas été altéré, les deux listes correspondront. Nous demandons à l'utilisateur de saisir un mot manquant (choisi aléatoirement) dans la liste des mots, pour l'inciter à vérifier que les deux listes correspondent. Cette confirmation se produit sur l'appareil approuvé (authentifié).
- Nous complétons ce mécanisme de sécurité par un Engagement de clé publique : l'appareil non approuvé envoie un hachage de sa clé publique X25519 au début de l'échange, et ne le libère à l'appareil non approuvé uniquement après avoir reçu sa clé publique. Ce mécanisme forcerait un Man in The Middle actif, qui espionne l'échange de clé, à fournir une clé publique à l'appareil approuvé avant de déterminer la chaîne courte authentifiée qu'il doit faire correspondre, ce qui réduirait considérablement la probabilité de réussir à détourner l'échange de clés.

Une fois le défi terminé, le $MachineGenerated$MP peut être transmis au nouvel appareil, et le coffre-fort est déchiffré localement sur le nouvel appareil de l'utilisateur.

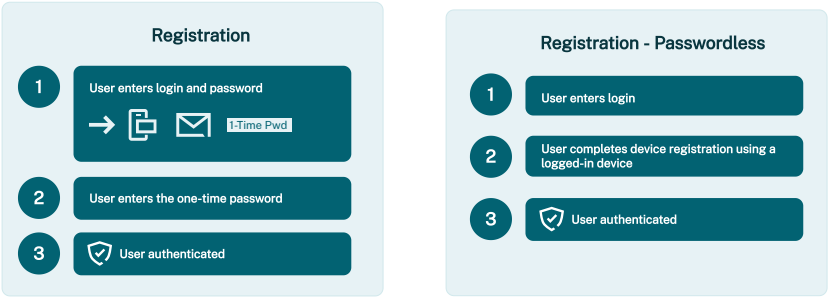
1.9 Une expérience utilisateur simplifiée



(a) Master Password (b) Passwordless

Fig. 4 : Registration and Authentication Steps

Depuis le début, notre objectif a été de conserver une expérience utilisateur simple et de cacher toute la complexité à l'utilisateur. La sécurité est de plus en plus importante pour les utilisateurs des services cloud, mais ils ne sont pas nécessairement prêts à sacrifier la commodité pour plus de sécurité.



(a) Master Password (b) Passwordless

Fig. 5 : Second Device Registration Steps

Même si ce qui se passe en arrière-plan lors des étapes d'inscription initiales est complexe (voir 4a), l'expérience utilisateur est très simple. Ils n'ont qu'à choisir entre créer un $User_{MP}$ (fort) ou passer à la connexion sans mot de passe, et toutes les autres clés sont générées par l'application sans l'intervention de l'utilisateur.

Lorsque l'utilisateur ajoute un nouvel appareil, le processus est tout aussi simple, mais reste sécurisé, grâce à la double authentification, comme illustré en 5a ou en utilisant un appareil existant déjà connecté.

1.10 Utilisation d'une application de 2FA pour sécuriser la connexion à un nouvel appareil

À tout moment, l'utilisateur peut associer son compte Dashlane à une application de double authentification (2FA) sur son appareil mobile. Lorsqu'il tente de se connecter à un nouvel appareil, au lieu de lui envoyer un mot de passe unique par e-mail, Dashlane demande à l'utilisateur de fournir un mot de passe unique généré par l'application de double authentification.

Après avoir reçu et vérifié le mot de passe unique fourni par l'utilisateur, les serveurs de Dashlane stockeront la $Device_{Key}$ générée par l'application cliente, comme décrit dans 5b.

1.11 Double authentification

Dashlane propose une double authentification (2FA) qui peut être activée dans les paramètres de sécurité de l'extension Web ou l'application mobile, pour exiger l'utilisation d'un deuxième facteur à chaque fois que l'utilisateur se connecte à Dashlane.

Les méthodes à deux facteurs prises en charge comprennent les applications de double authentification telles que Google Authenticator ou les appareils compatibles U2F tels que Yubikeys. U2F est un protocole ouvert de l'Alliance FIDO (https ://fidoalliance.org). Dashlane est un membre du conseil d'administration de l'Alliance FIDO.

1.12 Partage de données entre utilisateurs

Dashlane allows users to share credentials, Secure Notes, or secrets with other users, or with groups of users, in such a way that Dashlane never directly accesses a user's data at any point. In fact, Dashlane's servers never have access to the content of shared data.

Le partage de Dashlane repose sur le chiffrement asymétrique ; lors de la création d'un compte, une paire unique de clés RSA publiques et privées est créée par l'application Dashlane pour chaque utilisateur. La clé privée est stockée dans les données personnelles de l'utilisateur, et la clé publique est envoyée aux serveurs de Dashlane. Les clés publiques et privées RSA sont générées à l'aide de la fonction OpenSSL RSA_generate_key_ex, en utilisant une longueur de clé de 2048 bits, avec 3 comme exposant public.

Imaginons qu'une utilisatrice, Alice, souhaite partager un identifiant avec Bob. Voici le processus :

- Alice demande aux serveurs de Dashlane la clé publique de Bob.
- Alice génère une clé de chiffrement AES 256 bits en utilisant une fonction aléatoire sécurisée par cryptographie. Cette clé est unique pour chaque élément partagé. On l'appelle l'ObjectKey.
- Alice chiffre l'ObjectKey à l'aide de la clé publique de Bob, pour créer la BobEncryptedObjectKey.
- Alice envoie la BobEncryptedObjectKey aux serveurs de Dashlane.
- Alice chiffre ses identifiants avec l'ObjectKey, via AES-CBD et HMAC-SHA2 pour créer un EncryptedCredential.
- Alice envoie le EncryptedCredential aux serveurs de Dashlane.
- Lorsque Bob se connecte, les serveurs de Dashlane l'informent qu'Alice souhaite partager un identifiant avec lui. Bob doit accepter manuellement l'élément dans son application Dashlane et valider son acceptation en utilisant sa clé privée.
- Au moment de l'acceptation, les serveurs de Dashlane envoient à Bob la BobEncryptedObjectKey et le EncryptedCredential.
- Bob déchiffre la BobEncryptedObjectKey à l'aide de sa clé privée et obtient l'ObjectKey.
- Bob déchiffre le EncryptedCredential à l'aide de l'ObjectKey et ajoute l'identifiant d'Alice en texte clair à ses données personnelles.

Sharing an item with a group of users or sharing a collection of multiple items follows similar security principles :

- An AES key, the GroupKey is created for the group or collection and encrypted with each user's public key.

- An RSA public and private key pair is also created for the group.
- The private key is encryped with the GroupKey and used to sign the item or items in the group while the public key is used to encrypt the ObjectKeys within this group.

Users are then able to access the keys needed to decrypt individual items without Dashlane's servers being able to.

En résumé :

- Chaque utilisateur dispose d'une paire de clés RSA publique et privée en 2048 bits :
  - Les clés publiques sont utilisées pour chiffrer les informations que seul un utilisateur spécifique peut déchiffrer.
  - Les clés privées sont utilisées pour signer les actions que les utilisateurs effectuent.
- Pour chaque identifiant ou note sécurisée partagée, une clé intermédiaire AES 256 bits est créée et utilisée pour effectuer le chiffrement et le déchiffrement des données.

1.13 Récupération de compte

Dashlane dispose de deux méthodes de récupération disponibles pour les utilisateurs : la récupération de compte assistée par administrateur pour les utilisateurs professionnels qui se connectent avec un mot de passe Maître et la clé de récupération de compte, disponible pour tous les utilisateurs clients.

1.13.1 Récupération de compte assistée par administrateur

La récupération de compte assistée par administrateur permet aux utilisateurs de Dashlane Business de regagner l'accès à Dashlane en réinitialisant leur $User_{MP}$. Notre processus breveté préserve le « zero-knowledge ». Grâce à la récupération de compte, les mots de passe Maître ne sont jamais stockés sur les serveurs ni transmis sous quelque forme que ce soit.

Notre solution permet aux utilisateurs de réinitialiser leur $User_{MP}$ et de récupérer les données stockées sur un appareil autorisé. La récupération de compte est une fonctionnalité optionnelle que les administrateurs peuvent activer pour leur compte Dashlane Business dans la console d'administration.

Pour activer la récupération, la clé locale de l'utilisateur, elle-même chiffrée avec le $User_{MP}$, est également chiffrée à l'aide d'une clé de récupération de l'utilisateur unique, qui est générée et utilisée pour tous les appareils de l'utilisateur lorsqu'il opte pour la récupération de compte. Cette clé de récupération de l'utilisateur est ensuite chiffrée à l'aide d'une clé de récupération unique côté serveur, que seuls Dashlane et les appareils clients de l'utilisateur connaissent. Lorsqu'un administrateur

active la récupération de compte, sa clé publique est utilisée pour chiffrer la clé de récupération côté serveur, qui comme indiqué précédemment, était déjà utilisée pour chiffrer la clé de récupération de l'utilisateur. Un administrateur peut ensuite, via sa clé privée, accéder ultérieurement à la clé de récupération de l'utilisateur protégée par la clé de récupération côté serveur.

Lorsqu'un utilisateur demande la récupération de compte, il lui est demandé de vérifier son compte et de créer un nouveau $User_{MP}$. Une étape essentielle du processus de récupération est la vérification de l'identité de l'utilisateur. Il appartient à l'administrateur, agissant en tant que tiers approuvé, de s'assurer que l'utilisateur qui demande la récupération est bien le propriétaire du compte. Si un administrateur approuve la demande, la clé de récupération côté serveur, qui protège la clé de récupération de l'utilisateur, est échangée de manière sécurisée de l'administrateur à l'utilisateur via un système de clé publique / privée. Sur l'appareil de l'utilisateur, la clé de récupération de l'utilisateur est ensuite déchiffrée à l'aide de la clé de récupération côté serveur, fournie par Dashlane après la validation de l'identité et de la demande de l'utilisateur. La clé de récupération de l'utilisateur est ensuite utilisée pour déchiffrer la clé locale de l'utilisateur, qui à son tour est utilisée pour déchiffrer les données de l'utilisateur. Les données récupérées sont ensuite rechiffrées avec le $User_{MP}$ et re-synchronisées avec les serveurs de Dashlane.

Étant donné que ce processus implique une modification du mot de passe Maître, tous les appareils de l'utilisateur devront se connecter à nouveau à Dashlane pour que l'utilisateur puisse accéder à ses données chiffrées.

Remarque importante concernant la confidentialité : le processus de récupération de compte se base sur le fait que l'administrateur est un tiers de confiance. Dans le cas où l'administrateur Dashlane aurait accès à la fois à l'appareil de l'utilisateur et à son adresse e-mail utilisée pour le compte Dashlane, l'administrateur pourrait déclencher une récupération de compte à partir de l'appareil de l'utilisateur et accéder au coffre-fort et aux données personnelles de l'utilisateur.

1.13.2 Clé de récupération de compte

La clé de récupération de compte permet aux utilisateurs de configurer un mécanisme de récupération à usage unique afin de récupérer leurs données s'ils ne peuvent plus y accéder. La clé de récupération est une chaîne alphanumérique de 28 caractères qui doit être enregistrée et confirmée par l'utilisateur lors de la configuration. Elle est générée à partir des paramètres personnels de l'utilisateur à l'aide du générateur de mots de passe, et une clé dérivée de celui-ci avec les paramètres cryptographiques de l'utilisateur est utilisée pour chiffrer le $User_{MP}$ (le chiffrement AES-256). Une fois chiffré, il est envoyé et stocké sur le serveur.

Le mécanisme de la clé de récupération de compte peut être désactivé à tout moment à partir des paramètres de sécurité de l'utilisateur, ce qui invalide la clé de récupération de compte actuelle pour l'utilisateur.

Si un utilisateur a oublié son mot de passe Maître ou a perdu l'accès à tous ses appareils, l'utilisateur peut initier le mécanisme de récupération. Tout d'abord,

l'utilisateur doit terminer une étape de vérification de l'identité supplémentaire, soit un code de vérification par e-mail ou un jeton de double authentification, en fonction des paramètres de sécurité de l'utilisateur. Une fois la vérification de l'identité effectuée avec succès, l'utilisateur saisit le code de récupération, et le serveur libérera le $User_{MP}$ chiffré pour le client, qui essaiera de le déchiffrer avec la clé de récupération de compte. S'il y parvient, l'utilisateur sera invité à modifier son $User_{MP}$.

Une fois le processus terminé, la clé de récupération de compte actuelle n'est plus valide. Une nouvelle clé de récupération de compte doit être configurée à partir des paramètres de sécurité de l'utilisateur. La clé de récupération sera également désactivée après ces 2 événements : modification du mot de passe Maître et migration du mot de passe Maître vers la SSO.

1.14 Surveillance du dark Web pour le mot de passe Maître

Cette fonctionnalité permet aux utilisateurs de Dashlane d'être alertés si leur mot de passe Maître ou le mot de passe Maître d'un employé a été identifié lors d'une faille de données. Pour vérifier si le mot de passe Maître d'un utilisateur est compromis, nous allons vérifier s'il est présent dans les bases de données résultant des diverses fuites de données que nous collectons auprès de tiers. Nous collectons les données via des demandes d'API et transformons toutes les données en hachage à l'aide de la fonction Argon2 avant de les stocker sur nos serveurs. Lorsqu'un utilisateur saisit son mot de passe Maître sur son application mobile ou Web, nous commençons par le transformer à l'aide de la fonction Argon2 et d'un sel [4] présente dans l'application client, ce qui nous donne un hachage de 32 octets de long.

[4] Le sel que nous utilisons est spécifique à cette fonctionnalité et différent de celui utilisé pour compiler la clé de chiffrement de l'utilisateur

Algorithme	Itérations	Utilisation mem.	Parallélisme	Fils	Longueur de hachage
Argon 2d v1.3	3	32768	2	2	32

Tab. 2 : Configuration Argon2

Pour respecter notre architecture « zero-knowledge », nous utilisons un processus appelé "'K-anonymity" pour garantir que personne, pas même Dashlane ne peut accéder au mot de passe Maître. Pour cela, le hachage complet ne quitte jamais l'appareil de l'utilisateur, mais nous n'envoyons que les trois premiers octets de celui-ci à nos serveurs et comparons ces octets aux entrées que nous avons dans notre base de données. Si nous avons une ou plusieurs correspondances, nous envoyons la liste aux utilisateurs et finalement, l'application est capable de faire une comparaison complète entre le hachage local et ceux provenant des serveurs de Dashlane, et à la fin, avertir l'utilisateur si son mot de passe Maître a été trouvé dans une fuite de données.

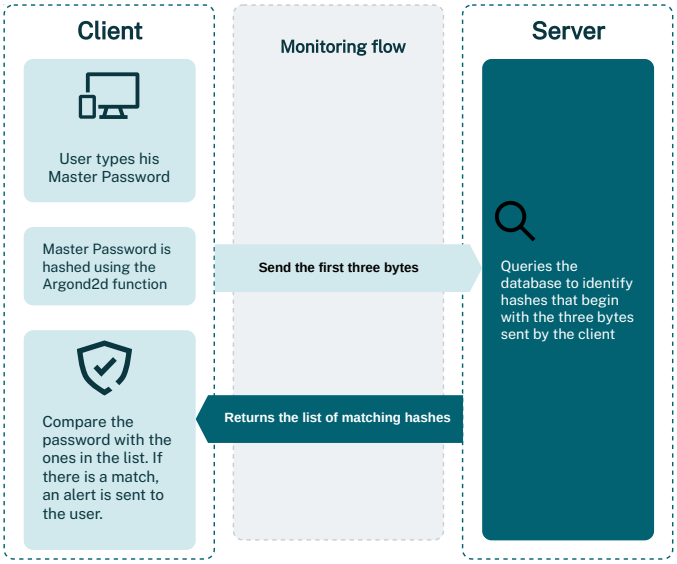


Fig. 6 : Surveillance du dark Web pour le flux de mots de passe Maître

1.15 Journaux d'activité

Dashlane fournit aux clients professionnels des journaux d'activité, un rapport horodaté disponible dans la console d'administration qui répertorie les actions entreprises par les administrateurs et les membres de l'équipe dans Dashlane. Cette fonctionnalité est importante pour les administrateurs, car elle leur permet d'obtenir des informations sur la posture de sécurité de votre organisation.

Pour produire ce rapport, Dashlane génère deux types d'événements :

- Les journaux d'activité : événements généraux de l'activité des membres. Ceux-ci sont générés par défaut sur le serveur.
- Les journaux d'activité sensibles : des événements supplémentaires générés par les applications clientes et envoyés à un point de terminaison pour être collectés côté serveur. Ces journaux ne sont pas activés par défaut et nécessitent l'activation des actions des administrateurs.

Les journaux d'activité sont générés à partir de diverses actions effectuées par les membres de l'équipe et les administrateurs, avec la liste complète des événements disponibles fournie dans l'annexe A.

Les journaux d'activité et les journaux d'activité sensibles sont d'abord stockés dans une base de données à des fins de mise en file d'attente. Un lot nettoie ensuite la file d'attente et transfère les événements à un stockage d'objet pour sa persistance. Le stockage d'objet est répliqué sur deux zones géographiques différentes (Irlande et Allemagne) pour obtenir un stockage fiable des journaux d'activité.

Les journaux d'activité peuvent être récupérés par les administrateurs. Cela peut se faire en deux étapes :

1 Une requête est envoyée au serveur ; le serveur répond avec un identifiant de requête.

2 Le serveur peut être demandé avec l'identifiant de requête pour obtenir l'état de la requête et éventuellement obtenir le résultat lorsque la requête a été finalisée.

1.16 Détection des risques liés aux identifiants

La détection des risques liés aux identifiants est une fonctionnalité Dashlane Business qui permet aux administrateurs de surveiller les mots de passe faibles et compromis utilisés par les utilisateurs de leur organisation qui n'utilisent pas activement Dashlane (c'est-à-dire que ces utilisateurs ne se sont pas authentifiés dans l'application). Cela est rendu possible par l'utilisation d'une solution de gestion unifiée des terminaux pour déployer l'extension web Dashlane sur les navigateurs des membres de l'équipe accompagnée d'une configuration (également appelée politique dans certains outils de gestion des terminaux). Dans ce document, le processus de distribution et d'installation de l'extension Dashlane sur un certain nombre d'appareils simultanément via une solution de gestion des terminaux sera parfois appelé "Déploiement de masse", "Déploiement en masse" ou "Mass Deployment". Cette configuration contient les informations qui seront incluses dans les journaux d'activité sensibles pour permettre à l'administrateur d'identifier l'utilisateur ainsi que la clé $MassDeploymentTeam_{Key}$ qui sera utilisée pour signer la requête HTTP et la lier à une équipe. Comme on peut le voir dans la Figure 7, l'administrateur déclenche la génération de la $MassDeploymentTeam_{Key}$ depuis la console d'administration de l'équipe (ou Team Admin Console abrégé en TAC). La clé est ensuite incluse dans les scripts générés automatiquement que l'administrateur exécutera pour déployer la configuration et déployer en masse l'extension Dashlane via son logiciel de gestion des appareils.

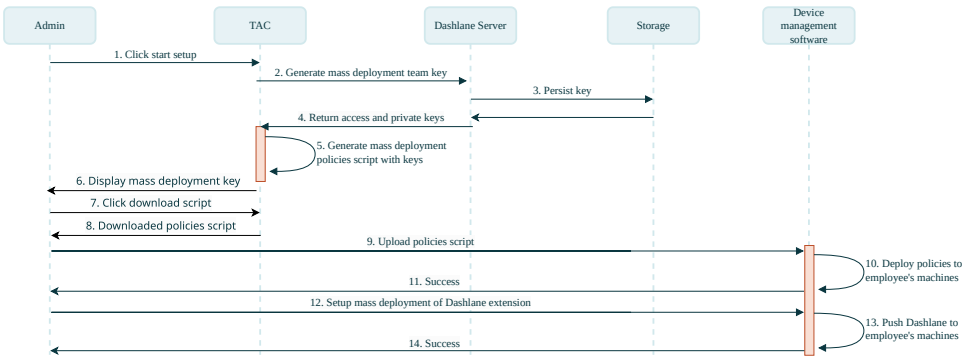


Fig. 7 : Procesus de déploiement de la détection des risques liés aux identifiants

Une fois que l'extension et les politiques ont été déployées sur l'appareil d'un membre de l'équipe, le processus de détection des risques commence. Comme montré dans la Figure 8, lorsque ce membre de l'équipe saisit un mot de passe sur internet sans être authentifié dans l'extension Dashlane, l'extension déployée en masse va exécuter les étapes suivantes :

1 Récupérer la configuration de Détection de Risque de Crédential depuis les serveurs pour vérifier que la fonctionnalité a été activée par l'administrateur

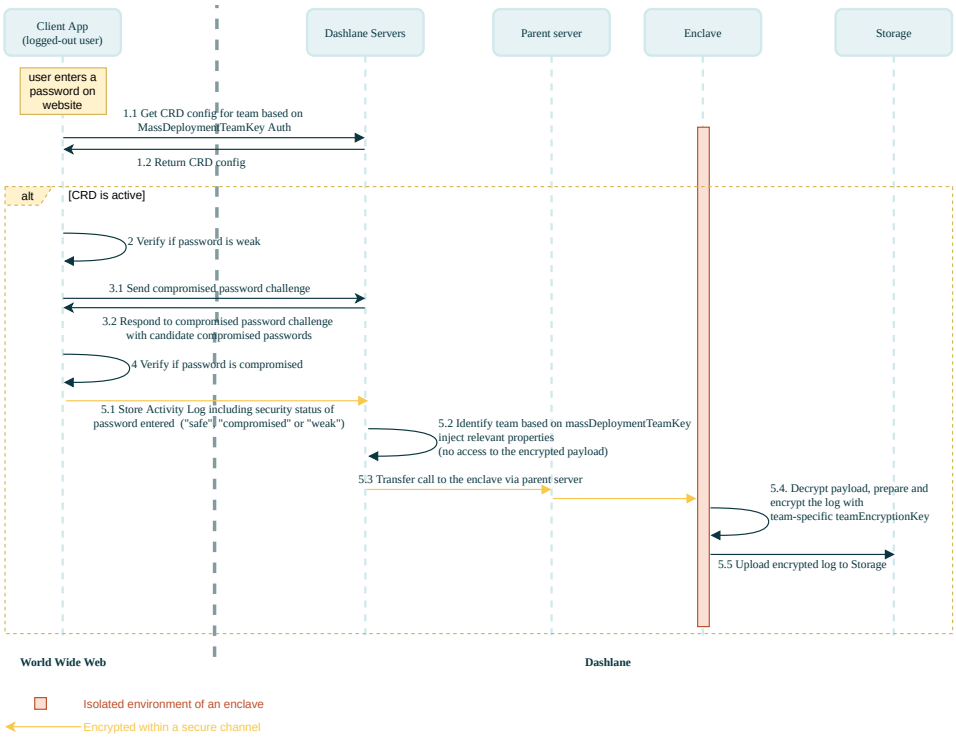


Fig. 8 : Chargement du journal d'activité de détection des risques liés aux identifiants

de l'équipe.

2 Vérifier si le mot de passe saisi est "faible" (cette vérification est basée sur un estimateur open-source de la force des mots de passe appelé **zxcvbn**[5] et est effectuée localement).

[5] https ://github.com/dropbox/zxcvbn

3 Vérifier si le mot de passe saisi est "compromis" (cette vérification nécessite des informations provenant du serveur, nous utilisons pour cela le même processus que celui qui est décrit dans la Figure 6 pour effectuer cette vérification sans que le mot de passe ne soit jamais envoyé sur nos serveurs).

4 En fonction des résultats des vérifications précédentes, envoyez un journal d'activité sensible (chiffré en transit via un tunnel sécurisé vers l'enclave et au repos par l'enclave) à l'enclave sécurisée. La charge utile chiffrée du journal contiendra le statut de sécurité du mot de passe saisi : "compromis","faible" ou "sûr" suivant cet ordre de priorité (c'est-à-dire que si un mot de passe est à la fois "compromis" et "faible", alors il est classé comme "compromis" car un mot de passe compromis représente un risque de sécurité plus important qu'un mot de passe faible).

1.17 Nudges

Les Nudges sont une fonctionnalité Dashlane Business grâce à laquelle les administrateurs peuvent définir un planning suivant lequel sont envoyés des rappels aux membres de l'équipe au sujet des problèmes de sécurité liés au contenu de leur coffre-fort.

Les utilisateurs concernés sont identifiés grâce rapport de santé des mots de passe. Les administrateurs peuvent ainsi définir un planning spécifique pour chaque type de vulnérabilité (mots de passe faibles, compromis et réutilisés). Les Nudges sont envoyés aux utilisateurs finaux via une intégration slack mais de nouveaux canaux de communication pourraient être ajoutés à l'avenir. Il y a 3 étapes majeures dans le cycle de vie des nudges :

1 L'administrateur installe la Slack-App Dashlane dans son espace de travail slack et charge le jeton d'identification slack sur les serveurs de Dashlane (voir Figure 9). Les permissions assignées au jeton slack sont détaillées dans le tableau 3.

2 L'administrateur configure / met à jour un nudge via la console d'administration de l'équipe (TAC) (voir Figure 10).

3 Une routine sur les serveurs de Dashlane s'exécute selon un planning et envoie des nudges aux utilisateurs finaux concernés (voir Figure 11). Le contenu des nudges est généré sur la base de modèles localisés fixes que l'équipe Dashlane maintient directement.

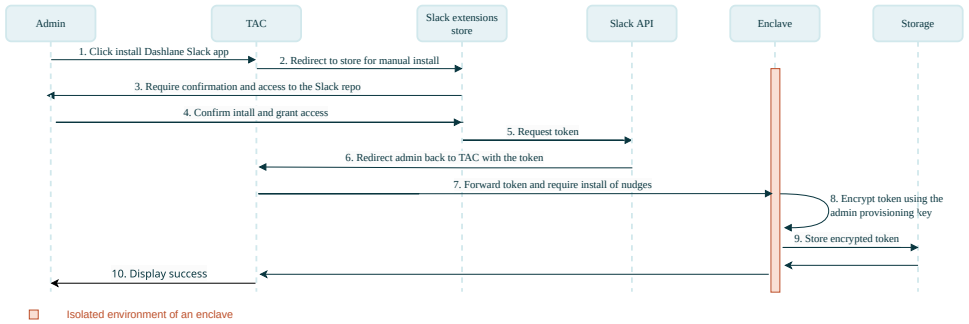


Fig. 9 : Installation de la Slack-App Dashlane

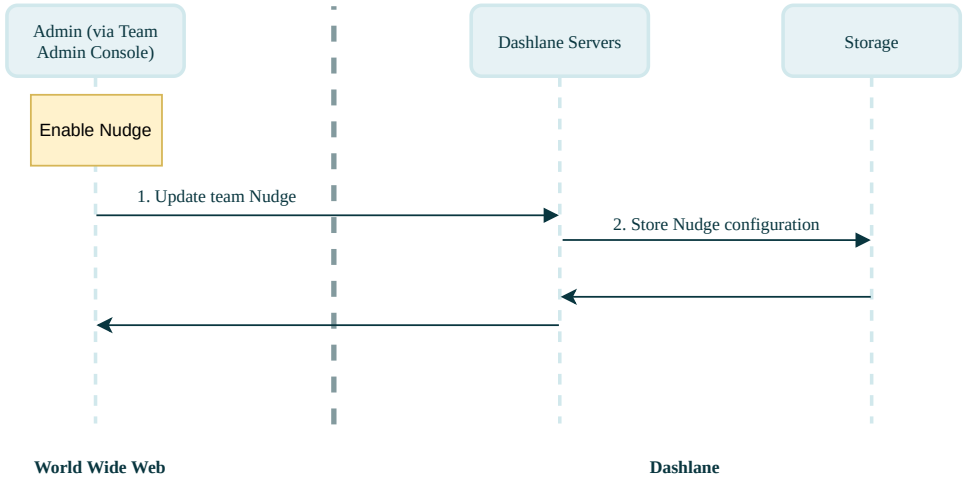


Fig. 10 : Configuration des Nudges

Le jeton slack étant une information sensible, il transite vers l'enclave nitro via un tunnel sécurisé établi entre l'extension web de l'administrateur et l'enclave. Il est

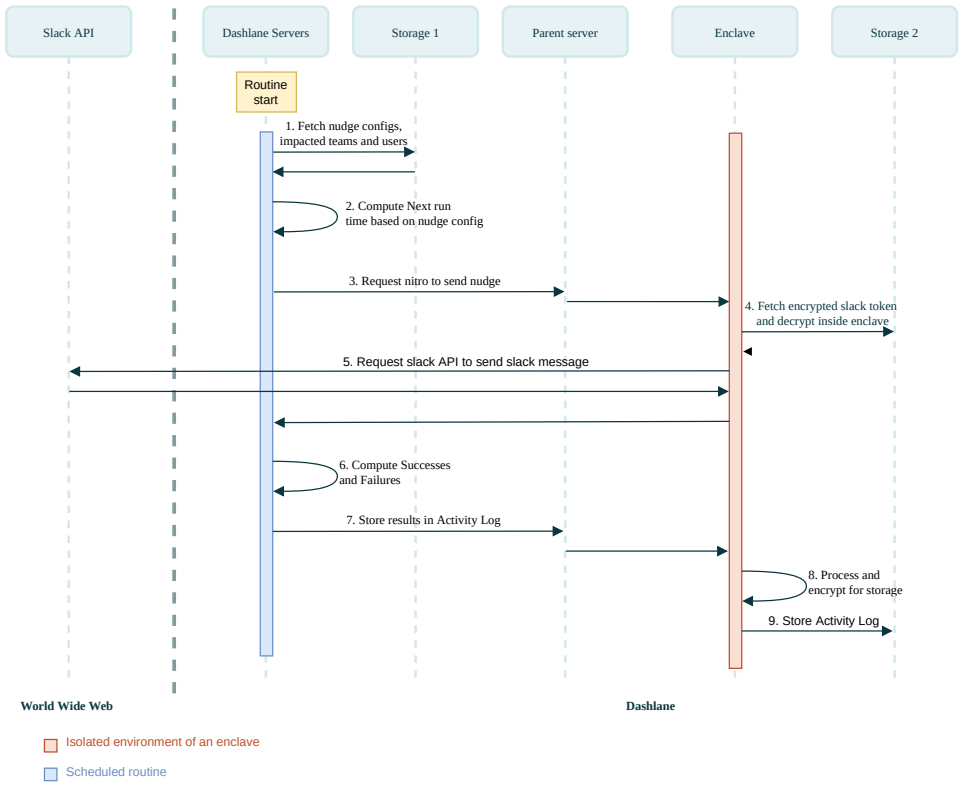


Fig. 11 : Routine Nudges

Permissions de lecture	Permissions d'envoi
"Voir les utilisateurs dans votre espace de travail"	"Envoyer des messages en tant que @dashlane"
"Voir les adresses email addresses des utilisateurs dans votre espace de travail"	"Envoyer des messages en tant que @dashlane avec un nom d'utilisateur et un profil personnalisé"

Tab. 3 : Permissions associées au jeton Slack pour Nudges

stocké chiffré avec une clé de chiffrement spécifique à l'équipe qui n'est disponible qu'à l'administrateur via son coffre-fort et à l'enclave, ce qui signifie que l'équipe Dashlane ne peut jamais accéder à ce jeton (voir Figure 9 11).

2 Authentification unique (SSO)

Dashlane s'intègre aux fournisseurs d'identité SSO (IdPs) qui utilisent le protocole d'authentification standard ouvert SAML 2.0, tels que Okta, Azure AD et ADFS. Cette intégration permet aux employés de déverrouiller leurs coffres de Dashlane avec leurs identifiants SSO plutôt que leur mot de passe Maître. Pour maintenir l'architecture « zero-knowledge » de Dashlane, l'intégration SSO nécessite un connecteur SSO pour stocker les clés de chiffrement des données de l'utilisateur et les livrer lors de l'authentification de l'utilisateur. Vous pouvez soit auto-héberger le connecteur SSO à l'intérieur de votre propre infrastructure ou choisir de le faire héberger par Dashlane dans une enclave sécurisée.

Si vous choisissez l'option auto-hébergée, le connecteur SSO agit comme le fournisseur de service dans le flux de travail SAML. Dashlane distribue le service, et vous l'hébergez

et le gérez comme un composant de serveur, soit sur site ou dans le cloud. Pour préserver le principe du zero-knowledge, le connecteur SSO stocke la première partie de la clé de chiffrement des données (64 octets aléatoires), et les serveurs cloud de Dashlane stockent l'autre moitié (64 octets aléatoires). Une fois l'authentification réussie et les deux parties de clé récupérées avec l'application Dashlane, elles sont comparées à l'aide de l'opération logique booléenne XOR, générant une autre clé de 64 octets qui déchiffre ou chiffre les données de l'utilisateur. Si Dashlane héberge et gère le connecteur SSO, le principe du zero-knowledge est activé par l'enclave sécurisée, un environnement qui isole les données et les processus de l'unité informatique du système d'exploitation et des autres processus sur la machine hôte. L'enclave sécurisée chiffre les données de stockage et dispose d'un mécanisme d'attestation pour s'assurer que seul le code autorisé peut traiter les données. Dashlane ne peut pas accéder aux clés de chiffrement de l'utilisateur ou à toutes les autres données des processus de connecteur SSO.

2.1 Introduction

Dashlane Business prend en charge la connexion avec l'authentification unique (SSO), en utilisant tout l'IdP activé par SAML 2.0.

Dans une configuration d'authentification unique, l'utilisateur n'a pas à saisir son $User_{MP}$. Au lieu de cela, une clé aléatoire est générée au moment de la création du compte. Cette clé (la clé de chiffrement des données) est livrée à l'application Dashlane après que l'utilisateur se connecte avec succès à l'IdP, et elle est utilisée comme clé de chiffrement symétrique pour chiffrer et déchiffrer les données de l'utilisateur.

Cette section détaille la méthode de stockage de la clé et sa livraison à l'utilisateur afin de respecter le principe « zero-knowledge ».

2.2 Principe général

L'intégration de la SSO avec l'application Dashlane nécessite une entité qui stocke les clés de chiffrement des utilisateurs et les livre lors de l'authentification. Cette entité a connaissance de la clé de chaque utilisateur, elle est donc très sensible. De plus, Dashlane ne peut pas héberger une telle entité sans s'en inquiéter davantage, car cela mettrait fin à notre principe d'architecture « zero-knowledge » en nous fournissant l'accès aux clés de chiffrement de nos utilisateurs.

L'entité précédente en charge des clés de chiffrement des utilisateurs est appelée le service de chiffrement et il pourrait être hébergé de deux façons différentes afin de respecter notre règle du « zero-knowledge » :

- **Auto-hébergé :** le service de chiffrement est un serveur déployé à l'intérieur de l'infrastructure du client Dashlane Business.
- **Hébergé dans une enclave sécurisée par Dashlane :** le service de chiffrement est un service s'exécutant dans l'infrastructure de Dashlane, dans une enclave sécurisée pour respecter notre principe du zero-knowledge.

2.3 SSO avec le Connecteur Auto-Hébergé

2.3.1 Aperçu

Pour éviter de stocker toutes les clés au même endroit, la clé de chiffrement des données se compose de deux parties :

- 64 octets aléatoires détenus par le service de chiffrement.
- 64 octets aléatoires détenus par les serveurs de Dashlane dans le cloud.

Le service de chiffrement est un composant de serveur que le client utilise (soit dans le cloud ou sur site). Il sert de fournisseur de service dans le flux SAML 2.0. Après une authentification réussie au service de chiffrement à l'aide de SAML, la première partie de la clé est livrée à l'application client Dashlane avec un jeton qui lui permet d'obtenir la deuxième partie de la clé depuis le serveur de Dashlane.

Une fois les deux parties des clés récupérées par l'application client, elles sont chiffrées en mode XOR, et les 64 octets résultants sont utilisés comme clé symétrique pour chiffrer et déchiffrer les données de l'utilisateur.

Ce système garantit le principe « zero-knowledge », puisque la première partie de la clé n'est connue que du service de chiffrement et de l'application cliente, qui sont tous deux gérés par le client.

Cela permet également d'éviter qu'un service de chiffrement compromis ne puisse récupérer les clés des utilisateurs sans laisser de trace sur les serveurs de Dashlane (un appel API vers le serveur de Dashlane est requis pour récupérer la deuxième partie de la clé).

2.3.2 Services

Serveur / API (API) de Dashlane Les serveurs exploités par Dashlane dans le cloud, où les données de l'utilisateur sont stockées chiffrées.

Service de chiffrement (SP) Un service qui sert de fournisseur de service dans le flux SAML 2.0. Le service est distribué par Dashlane, mais il est hébergé et géré par le client sur place ou dans le cloud.

Fournisseur d'identité (IdP) Le fournisseur d'identité SAML 2.0 (par exemple ADFS, Azure AD, Okta) du client. Ce service n'est pas fourni par Dashlane. Il est géré par le client ou par un tiers.

2.3.3 Clés, clés secrètes et certificats

Clé et certificat IdP (IdP_{Key} / IdP_{Cert}) Clés publiques et privées de l'IdP. La clé privée est détenue par l'IdP, tandis que le certificat doit être fourni au SP dans le fichier de configuration. Il est utilisé par l'IdP pour signer et par le SP pour vérifier les assertions SAML.

Clé SP Maître / clé de service de chiffrement ($Master\_SP_{Key}$) Une clé secrète à

64 octets, générée aléatoirement par la console d'administration de l'équipe (côté client). Elle est stockée dans le fichier de configuration du SP, et n'est connue que par l'administrateur de l'équipe. Elle est utilisée par le SP pour chiffrer / déchiffrer les $User\_SP_{key}$ avant de les stocker dans l'API.

Clé SP de l'utilisateur ($User\_SP_{Key}$) Une clé secrète de 64 octets, générée aléatoirement par le SP. Elle est stockée et chiffrée dans l'API.

Clé de serveur de l'utilisateur ($Server_{Key}$) Une clé secrète de 64 octets, générée aléatoirement par le client. Elle est stockée dans un état non chiffré dans l'API.

Clé de serveur de l'utilisateur ($Vault_{Key}$) $User\_SP_{Key} \oplus Server_{Key}$. Elle est utilisée par le client pour chiffrer / déchiffrer les données des utilisateurs avant de les stocker dans l'API.

2.3.4 Workflow

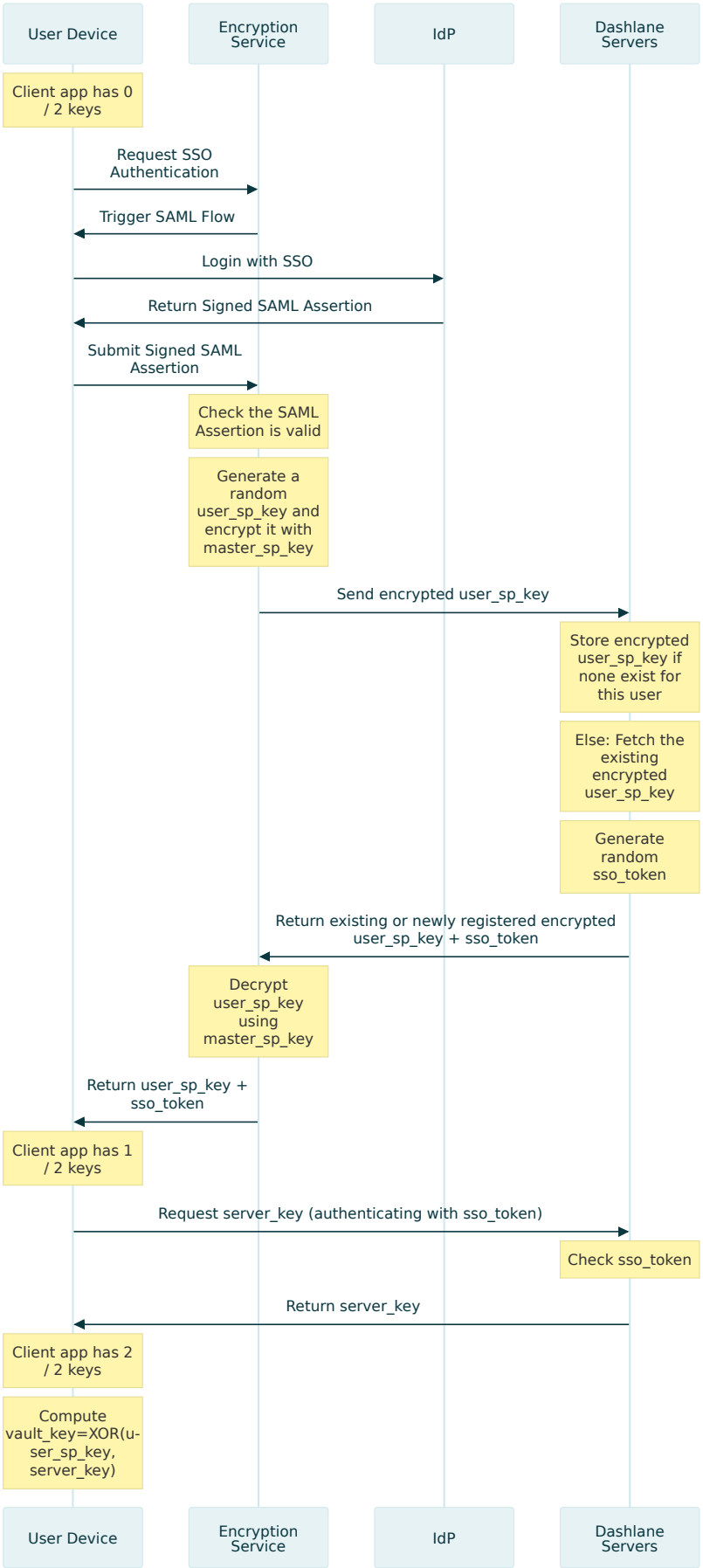


Fig. 12 : Flux de travail SSO auto-hébergé

2.4 SSO avec le Connecteur Hébergé par Dashlane

2.4.1 Aperçu

Dans la configuration de l'adaptateur hébergé par Dashlane, le service de chiffrement est hébergé et géré par Dashlane. Pour empêcher Dashlane d'accéder à la clé de chiffrement des utilisateurs, en brisant le principe du zero-knowledge, le service de chiffrement s'exécute dans une enclave dite sécurisée.

Une enclave sécurisée est un terme venant du domaine de l'informatique de confiance. Il s'agit du nom donné à une unité informatique isolée ou à un environnement d'exécution approuvé (TEE). Cette technologie fournit un moyen de traiter les données à l'intérieur d'un environnement qui n'est pas lisible par tout autre processus de la machine d'hébergement en dehors du processus s'exécutant à l'intérieur de l'enclave. De plus, les enclaves sécurisées peuvent générer une attestation avec l'empreinte digitale du code qu'elles exécutent. De cette façon, les clients qui communiquent avec une enclave peuvent obtenir des assurances du code avec lequel ils communiquent et décider s'ils se connectent à ce code pour traiter leurs données.

Les enclaves sécurisées ne sont que des unités informatiques avec le CPU et les ressources de mémoire volatile. Elles ne sont pas fournies avec le stockage persistant. Pour résoudre ce problème, un service de gestion de clé (KMS), qui peut authentifier le fait que les demandes proviennent d'enclaves approuvées, est nécessaire pour chiffrer le stockage des enclaves sécurisées.

Dashlane tire parti de la technologie d'enclave sécurisée pour exécuter un service de chiffrement sans être en mesure d'accéder aux clés de chiffrement des utilisateurs traitées par le service de chiffrement.

2.4.2 Matériaux cryptographiques

Les flux de travail SSO confidentiels de Dashlane nécessitent beaucoup de clés cryptographiques et de certificats définis dans le tableau 4. Toutes les clés définies sont de 32 octets.

2.4.3 Flux de travail

2.4.3.1 Étape d'initialisation d'une enclave

La première étape est de générer une clé Maître d'enclave dans le KMS et de bâtir des politiques d'accès à cette clé Maître d'enclave afin que l'accès soit accordé uniquement à l'enclave. Cela se fait en basant les politiques sur les informations fournies par l'attestation de l'enclave : lorsque le KMS reçoit une demande de clé Maître d'enclave, il correspond à l'attestation fournie avec les politiques pour accorder ou refuser la demande.

Ensuite, l'enclave est déployée et demande au KMS de générer une clé locale d'enclave et de renvoyer de manière sécurisée à l'enclave deux versions de la clé locale d'enclave : l'une chiffrée par la clé Maître d'enclave et l'autre chiffrée avec une clé publique éphémère fournie par l'attestation. L'enclave demande le stockage

Nom de clé	Symbole de clé	Description
Clé Maître d'enclave	EM_{Key}	Clé générée et stockée dans le KMS afin de chiffrer / déchiffrer EL_{Key}
Clé locale d'enclave	EL_{Key}	Clé générée dans le KMS lors de la première amorce de l'enclave et envoyée à cette enclave pour en dériver EE_{Key}
Clé de suppression d'enclave	EU_{Key}	Clé générée par le processus de déploiement lors de la première amorce et envoyée à l'enclave, afin de dériver EE_{Key}
Clé de chiffrement d'enclave	EE_{Key}	Clé dérivée de $EL_{Key} \oplus EU_{Key}$ afin de chiffrer la $SPMaster_{cl}$
Clé principale du fournisseur de services	$SPMaster_{Key}$	Clé générée dans l'enclave sur une nouvelle inscription d'équipe afin de chiffrer / déchiffrer $UserSP_{Key}$
Clé de fournisseur de services utilisateur	$UserSP_{Key}$	Clé générée dans l'enclave lorsque l'utilisateur est provisionné pour l'authentification SSO, afin de chiffrer $Remote_{Key}$
Clé de serveur SSO	$SSOServer_{Key}$	Clé générée par le serveur lors de la création du compte, afin de chiffrer $Remote_{Key}$
Clé distante	$Remote_{Key}$	Clé générée par le client lors de la création du compte, afin de chiffrer le coffre-fort de l'utilisateur
Certificat de fournisseur d'identité	IdP_{Cert}	Certificat de clé publique de l'IdP pour vérifier l'assertion SAML

Tab. 4 : Clés et certificats cryptographiques implicites dans les flux de travail hébergés Dashlane

de la clé locale d'enclave chiffrée et garde la clé locale d'enclave en texte clair dans cette mémoire volatile. De cette façon, si l'enclave redémarre ou une nouvelle instance est déployée, l'instance demandera ensuite au stockage la clé locale d'enclave chiffrée, puis le KMS la déchiffrera avec la clé Maître d'enclave. De cette façon, l'enclave est fournie avec la clé locale d'enclave pour chiffrer les données, et la clé locale d'enclave n'est jamais en texte clair en dehors d'un environnement sécurisé ; l'enclave ou le KMS.

La figure 14 décrit le flux de travail pour fournir des enclaves sécurisées avec EL_{Key}.

Ensuite, le processus de déploiement peut monter un canal sécurisé (basé sur l'attestation de l'enclave) pour envoyer la EU_{Key} à l'enclave. De cette façon, l'enclave sécurisée peut dériver la clé de chiffrement d'enclave comme suit :

$$EE_{Key} = EL_{Key} \oplus EU_{Key}$$

2.4.3.2 Stockage de l'enclave sécurisée

Une enclave sécurisée est un environnement d'exécution sans stockage persistant. Les données doivent être chiffrées avant d'être transférées via le serveur parent vers la banque de données.

Les données au sein de l'enclave sécurisée nécessitant un stockage persistant sont les suivantes :

- La clé locale d'enclave EL_{Key}, chiffrée par la clé Maître d'enclave EM_{Key}.
- Les clés principales$SPMaster_{Key}$ de chaque équipe du fournisseur de services, chiffrées par la clé de chiffrement d'enclave EE_{Key}.
- Clés de fournisseur de service utilisateur $UserSP_{Key}$ de chaque utilisateur, chiffrées par la $SPMaster_{Key}$ de leur équipe.

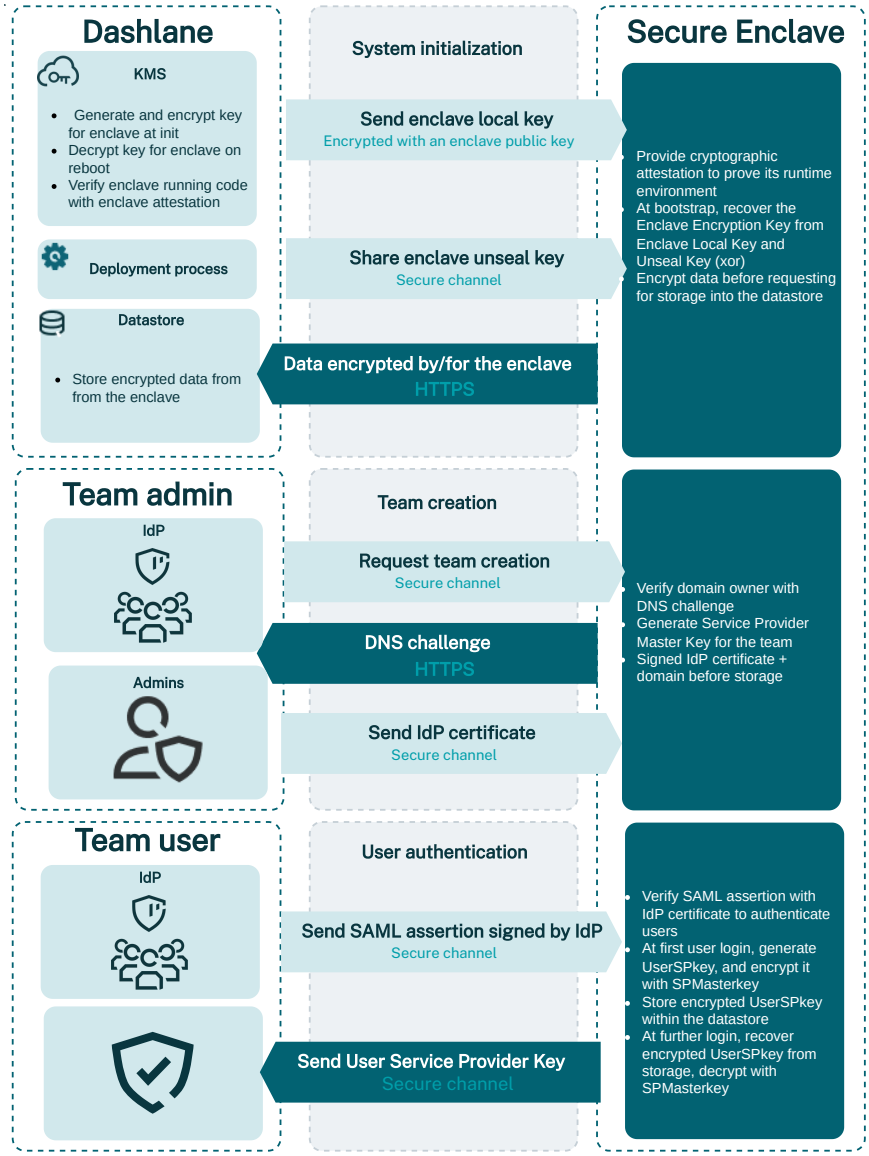


Fig. 13 : Flux de travail SSO hébergés par Dashlane

2.4.3.3 Création d'une équipe

L'étape de création d'une équipe est la configuration de la SSO pour une organisation : l'enclave est fournie avec le certificat IdP pour vérifier les assertions SAML pour authentifier les utilisateurs d'un domaine (par exemple les utilisateurs avec un e-mail d'un domaine donné). L'enclave doit toujours vérifier que l'administrateur effectuant l'opération est le propriétaire du domaine demandé : de cette façon, cela empêche toute personne de fournir un certificat IdP frauduleux pour un domaine qu'elle ne possède pas. En effet, la SSO est basée sur le domaine de l'e-mail de l'utilisateur. Par exemple, si un utilisateur demande à se connecter avec le nom d'utilisateur user@example.com, et que le domaine example.com est lié à un IdP, l'utilisateur passera par le flux d'authentification avec cet IdP. De cette façon, l'enregistrement d'un IdP pour un domaine est une opération sensible, nécessitant l'enclave sécurisée pour effectuer la vérification du domaine.

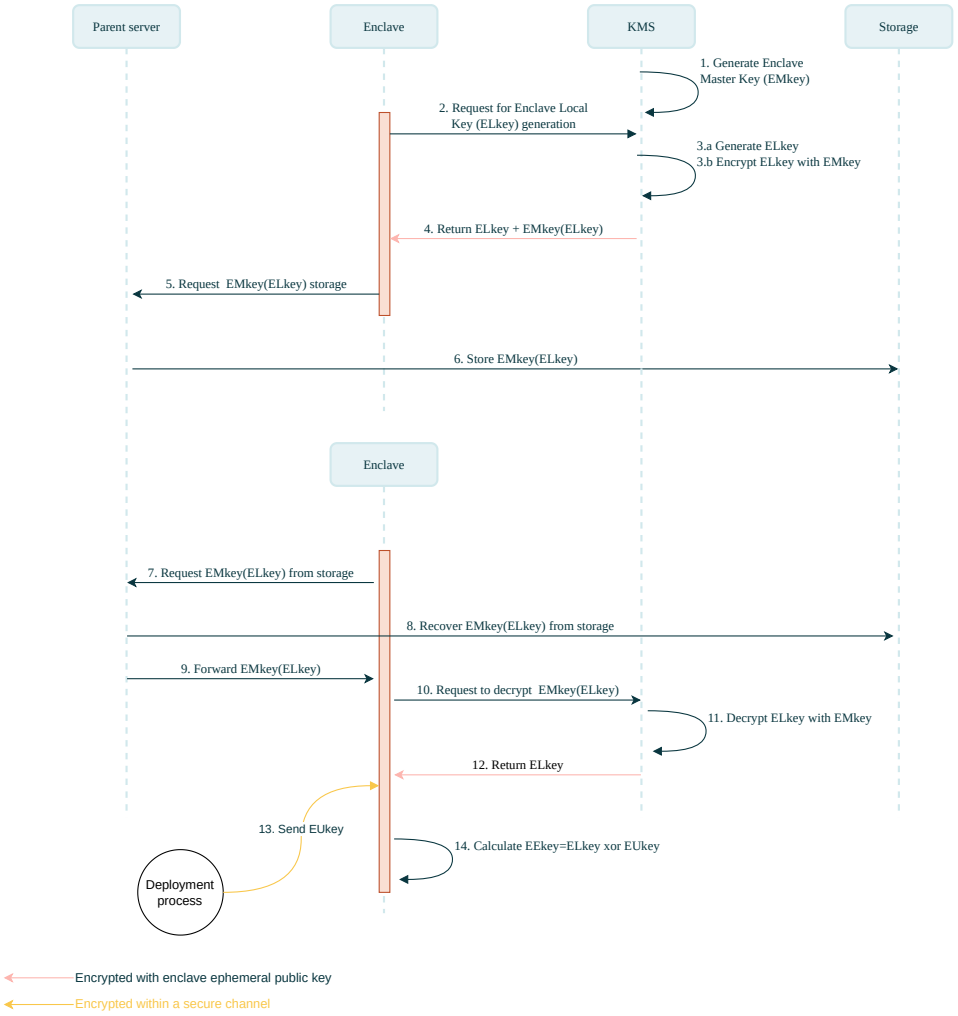


Fig. 14 : Initialisation de la Dashlane Confidential SSO

Le flux de création d'une équipe est décrit dans la figure 15

1 L'administrateur informatique de l'organisation configurera l'IdP et obtiendra l'URL du point de terminaison IdP et le certificat IdP de la clé, qui signera la preuve d'authentification des autres utilisateurs ; l'administrateur informatique démarre le flux de configuration de la SSO dans l'application d'administrateur.

2 L'application d'administrateur effectue une poignée de main avec l'enclave pour créer un canal sécurisé.

3 Par le canal sécurisé, l'application client envoie le certificat IdP et le domaine ; cela se fait dans le canal sécurisé pour protéger l'intégrité du certificat IdP (pour empêcher le certificat d'être remplacé pendant le transit par un certificat frauduleux).

4 L'enclave renvoie une valeur aléatoire pour initier la vérification du domaine.

5 L'administrateur informatique et l'enclave effectuent le défi DNS : l'objectif est de laisser l'enclave vérifier qu'elle parle avec un propriétaire du domaine

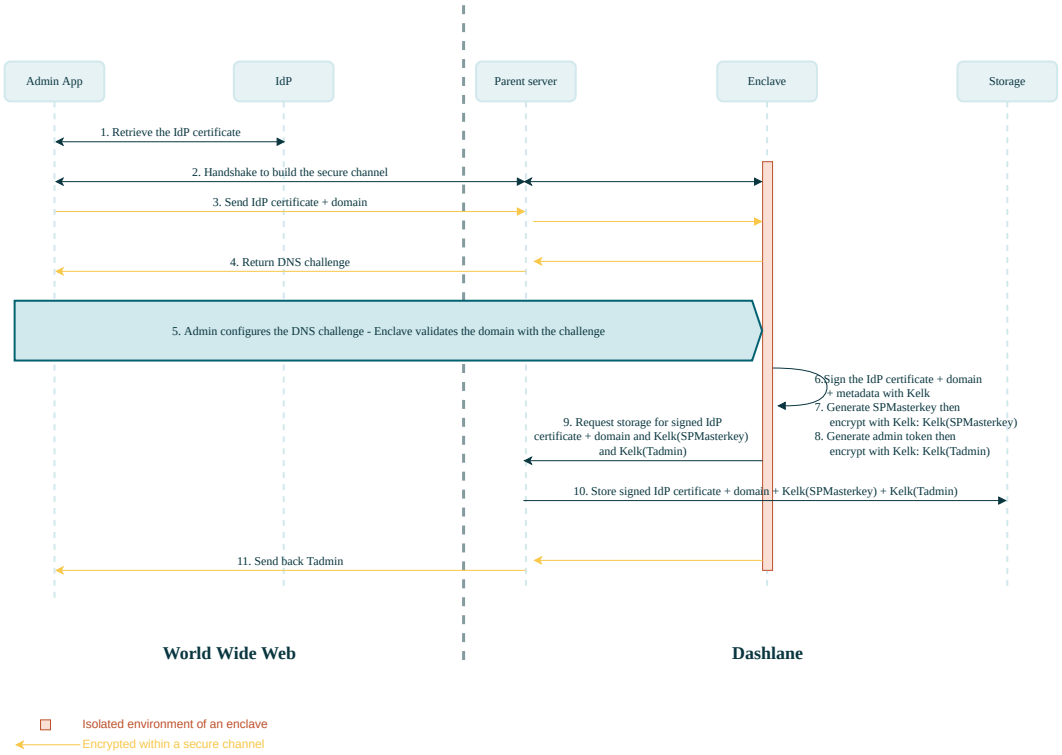


Fig. 15 : Flux de création d'une équipe avec Dashlane Confidential SSO

réclamé ; pour cela, l'administrateur informatique doit placer la valeur aléatoire à la racine du domaine, puis l'enclave peut vérifier cette valeur avec le DNS (mieux avec une version sécurisée du protocol) ; de cette façon, l'enclave valide le fait que l'administrateur informatique est le propriétaire du domaine réclamé.

6 L'enclave génère un code d'authentification de message (MAC) pour le certificat IdP + domaine + métadonnées de EE_{Key}.

7 L'enclave génère la clé principale du fournisseur de services pour le domaine $SPMaster_{Key}$, puis la chiffre avec EE_{Key}.

8 L'enclave génère un jeton pour authentifier les administrateurs du domaine (le jeton sera partagé entre les comptes d'administrateur du domaine), puis le chiffre avec EE_{Key}.

9 L'enclave demande à l'instance parent de stocker le certificat IdP + domaine signé, la $SPMaster_{Key}$ chiffrée et l'administrateur du jeton chiffré.

10 L'instance parent stocke le certificat IdP + domaine signé, la $SPMaster_{Key}$ chiffrée, et l'administrateur du jeton chiffré.

11 L'instance renvoie l'administrateur du jeton via le canal sécurisé.

2.4.3.4 Connexion SSO de l'utilisateur

Après la création de l'équipe, un utilisateur peut s'attendre à ouvrir son coffre-

fort avec le flux SSO. Atteindre la page de connexion de leur application client qui les redirige vers la page de connexion de leur IdP. Une fois que l'IdP authentifie l'utilisateur, il redirige l'utilisateur vers l'application client avec une assertion SAML prouvant son identité. Ensuite, l'application client peut envoyer l'assertion au service de chiffrement pour recevoir en retour la $UserSP_{Key}$, qui déchiffre le coffre-fort de l'utilisateur.

Jusqu'à ce que la preuve d'authentification soit envoyée, le flux est le même pour les utilisateurs qui effectuent leur première connexion et les utilisateurs qui ont déjà activé leur compte.

Le début du flux est décrit par la figure 16

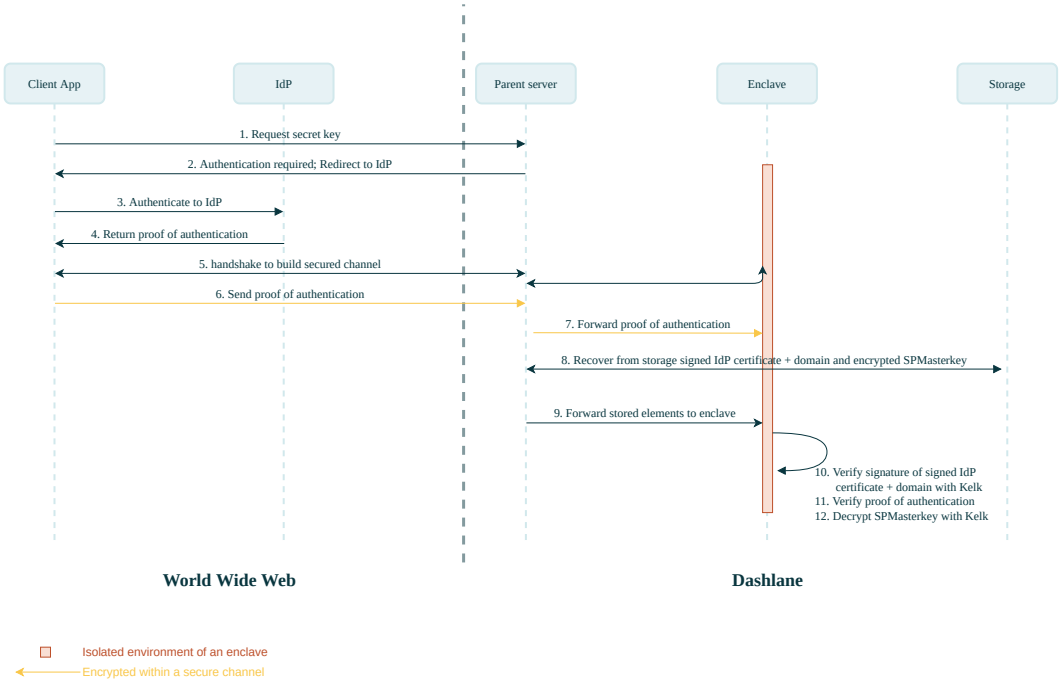


Fig. 16 : Confidential SSO de Dashlane - Flux de connexion de l'utilisateur

À partir de ce point, l'utilisateur est authentifié dans l'enclave. Le flux diverge entre le compte de la première connexion et le compte déjà activé.

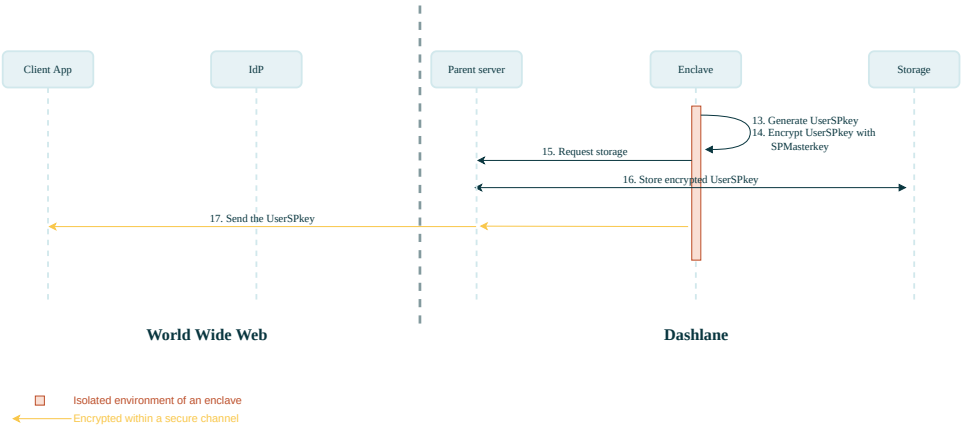
Une fois les utilisateurs sont authentifiés (la signature de leur preuve d'authentification est vérifiée) pour la première connexion, le flux de travail est comme décrit dans la figure 17a :

13 L'enclave génère la $UserSP_{Key}$.

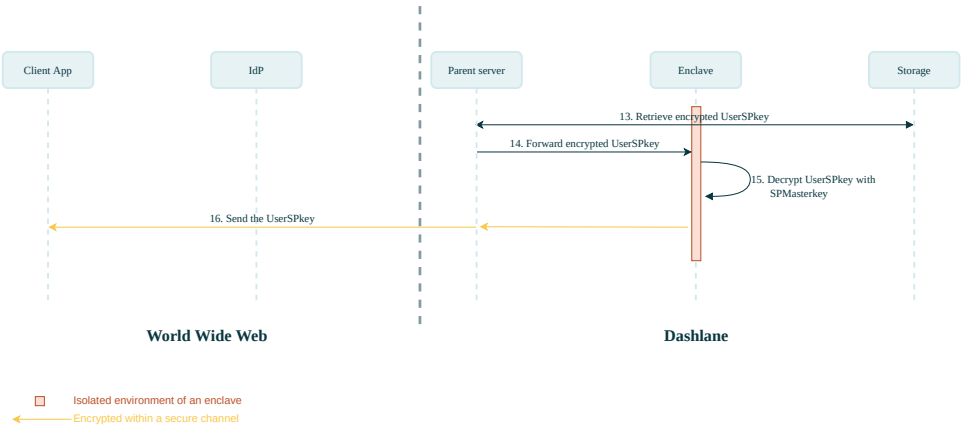
14 L'enclave chiffre la $UserSP_{Key}$ avec la $SPMaster_{Key}$.

15 L'enclave demande au stockage du serveur parent $SPMaster_{Key}(UserSP_{Key})$.

16 Après la confirmation que la $SPMaster_{Key}(UserSP_{Key})$ est stockée, l'enclave renvoie à l'application client la $UserSP_{Key}$, chiffrée via le canal sécurisé.



(a) Première connexion



(b) Connexion standard

Fig. 17 : Confidential SSO de Dashlane - Flux de connexion de l'utilisateur (partie 2)

Une fois les utilisateurs authentifiés (la signature de leur preuve d'authentification est vérifiée) pour un compte déjà autorisé, le flux de travail est comme décrit dans la figure 17b :

13 Le serveur parent récupère la $SPMaster_{Key}(UserSP_{Key})$ depuis le stockage.

14 Le serveur parent transfère les éléments stockés à l'enclave.

15 L'enclave déchiffre la $SPMaster_{Key}(UserSP_{Key})$.

16 L'enclave renvoie à l'application client la $UserSP_{Key}$, chiffrée dans le canal sécurisé.

2.4.3.5 SCIM User provisioning

L'administrateur peut configurer le provisionnement confidentiel d'utilisateurs dans la console d'administration. Pour terminer la configuration, l'extension Dashlane génère un "bearer token" en utilisant uuidv4.Ce token est ensuite transmis via un tunnel sécurisé à l'enclave qui l'encrypte avant de le stocker.

En parallèle, l'administrateur ajoute ce token dans son fournisseur d'identité ainsi que l'adresse URL spécifique à l'équipe à laquelle le fournisseur d'identité devra envoyer les synchronisations SCIM (cette adresse est fournie à l'administrateur via la console d'administration).

Une fois ces étapes de configurations terminées, les synchronisations peuvent commencer à être transmises par le fournisseur d'identité via des requêtes HTTPS. Comme visible sur la figure 18, l'enclave vérifie le token SCIM avant de transmettre les opérations à effectuer aux serveurs de Dashlane pour synchroniser les utilisateurs en conséquence.À la création d'un utilisateur l'enclave générera Uuid (scimId) et renverra cet identifiant au fournisseur d'identité pour que le fournisseur d'identité puissent à l'avenir partager un identifiant unique pour cet utilisateur SCIM.

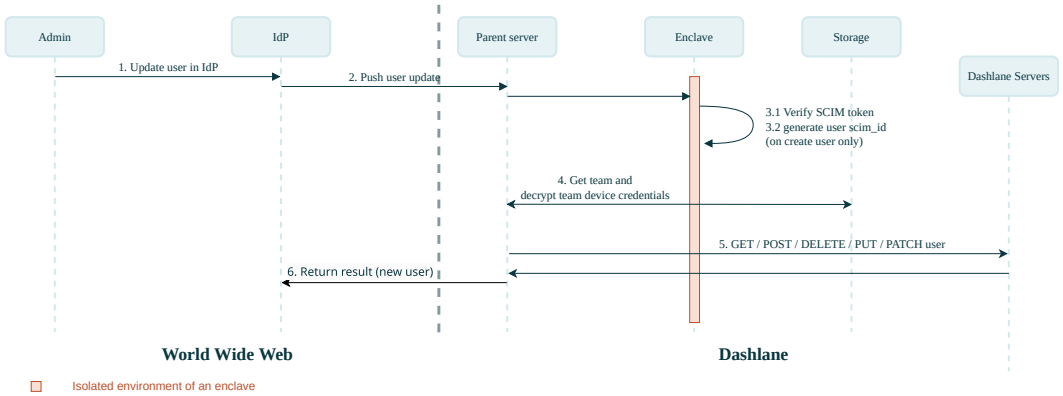


Fig. 18 : Provisionnement "confidentiel" d'utilisateurs de Dashlane

2.4.3.6 Provisionnement de Groupes

Si l'administrateur l'a activée via la console d'administration, la synchronisation automatique des groupes a lieu lors de la connexion de chaque utilisateur.

Cette synchronisation se base sur l'assertion SAML qui est transmise pendant le processus de connexion SSO, comme décrit sur la figure 19.

Quand un utilisateur se connecte dans Dashlane par SSO l'enclave reçoit de la part de l'extension une assertion SAML qui inclut les noms des groupes dont l'utilisateur fait partie. L'enclave récupère ensuite auprès des serveurs de Dashlane l'ensemble des groupes associés à l'équipe de l'utilisateur. Elle détermine ensuite :

- les nouveaux groupes à créer,
- les groupes existants dans lesquels inviter l'utilisateur,
- les groupes existants dont l'utilisateur doit être exclu.

Les serveurs de Dashlane sont ensuite appelés pour exécuter ces actions. Ce processus de synchronisation des groupes est idempotent : on reçoit la liste des groupes dont l'utilisateur devrait faire partie et, à la fin du processus, l'utilisateur est membre de chacun de ces groupes et uniquement ceux-là.

L'assertion SAML qui contient la liste des groupes est signée par le fournisseur d'identité et l'enclave vérifie cette signature. Cette vérification garantit que la liste des groupes n'a pas été modifiée par un tiers.

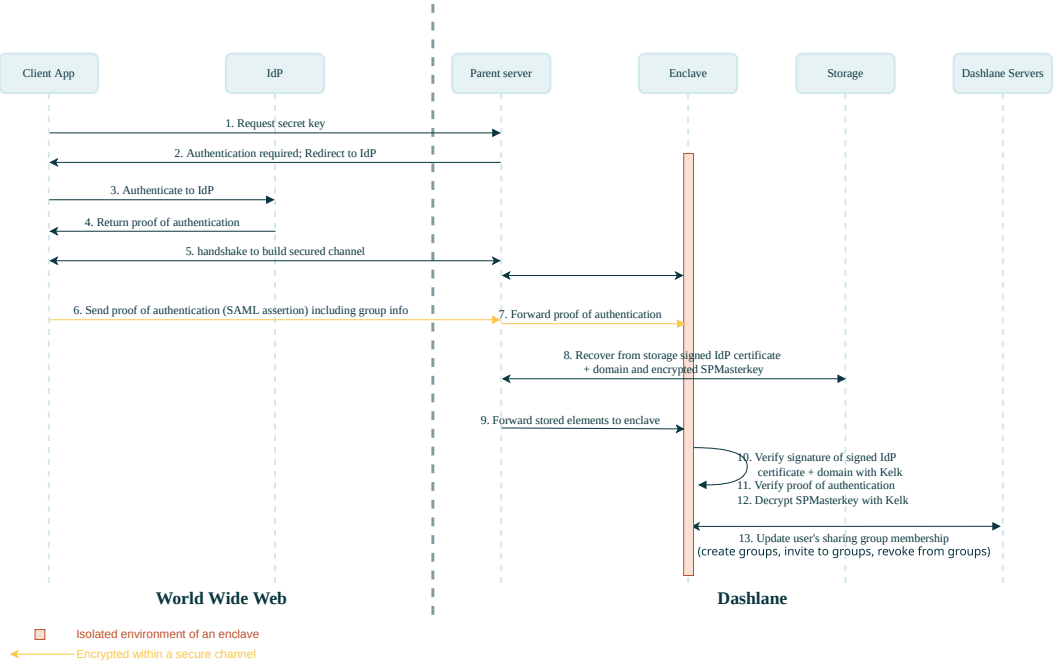


Fig. 19 : Provisionnement "confidentiel" de groupes de Dashlane

Modèle de Sécurité du Provisionnement des Groupes

Le Provisionnement de groupes pouvant donner accès à des secrets partagés, l'opération est considérée comme sensible.
Le protocol SCIM ne permet pas d'authentifier l'enclave distante depuis l'IdP, posant un risque au modèle de sécurité pour le provisionnement de groupes. En revanche, les assertions SAML étant transmises à l'enclave par l'extension via le tunnel securisé et étant signée par l'IdP, le provisionnement de groupe ne peut pas falsifié.

3 Impact sur les scénarios d'attaque potentiels

Dashlane a intégré une variété de protocoles de sécurité à l'architecture pour empêcher la compromission des données de l'utilisateur due à une attaque d'acteurs malveillants externes ou internes. Certains exemples de ces protocoles comprennent :

- La séparation de la clé pour le chiffrement des données de l'utilisateur et de la clé d'authentification de l'utilisateur sur le serveur de Dashlane, ce qui garantit que les clés de chiffrement des données de l'utilisateur ne sont pas stockées n'importe où et ne peuvent pas être accessibles aux employés de Dashlane ou aux attaquants si les serveurs de Dashlane sont compromis.

- Les mesures de protection Web, y compris les dispositions anti-clickjacking

qui empêchent les sites Web frauduleux de déclencher un clic malveillant et d'extraire des données de l'application Dashlane ; et la politique de même origine, qui ne saisit automatiquement un mot de passe enregistré que sur les sous-domaines d'URL exacts.

- L'utilisation de la fonction Argon2, qui protège les données de l'utilisateur chiffrées des attaques par force brute ou par dictionnaire.

3.1 Architecture de sécurité minimale

Les services cloud peuvent utiliser un **seul secret privé**, qu'ils contrôlent généralement, **pour chiffrer toutes les données de l'utilisateur**. Il s'agit évidemment d'un choix plus simple d'un point de vue de la mise en œuvre, de plus il offre l'avantage de faciliter la **déduplication** des données, ce qui peut fournir des avantages économiques importants lorsque le volume de données de l'utilisateur est élevé. De toute évidence, ce n'est pas un scénario optimal d'un point de vue de la sécurité, car si la clé est compromise (attaque d'un pirate informatique ou employé frauduleux), toutes les données de l'utilisateur sont exposées.

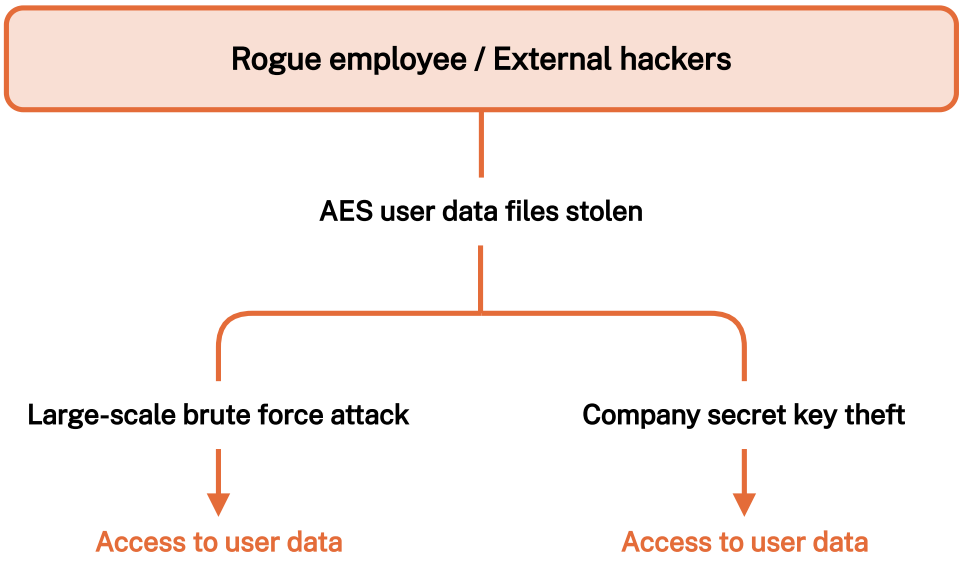


Fig. 20 : Scénarios d'attaques avec une sécurité minimale

3.2 Architecture de sécurité la plus courante

Il est préférable d'utiliser une clé différente pour chaque utilisateur. La pratique la plus courante est de demander à l'utilisateur de fournir un $User_{MP}$ (fort) et de dériver la clé de chiffrement pour chaque utilisateur depuis leurs $User_{MP}$. Cependant, pour simplifier les choses pour l'utilisateur, de nombreux services ou applications ont tendance à utiliser la comme clé d'authentification pour la connexion à leurs services. Cela implique qu'un attaquant pourrait accéder au coffre-fort d'un utilisateur en connaissant simplement le mot de passe Maître. Cela pourrait également facilement entraîner des erreurs de mise en œuvre (attaques de tables de sel / arc-en-ciel manquantes, hachage inadéquat / faible, etc.).

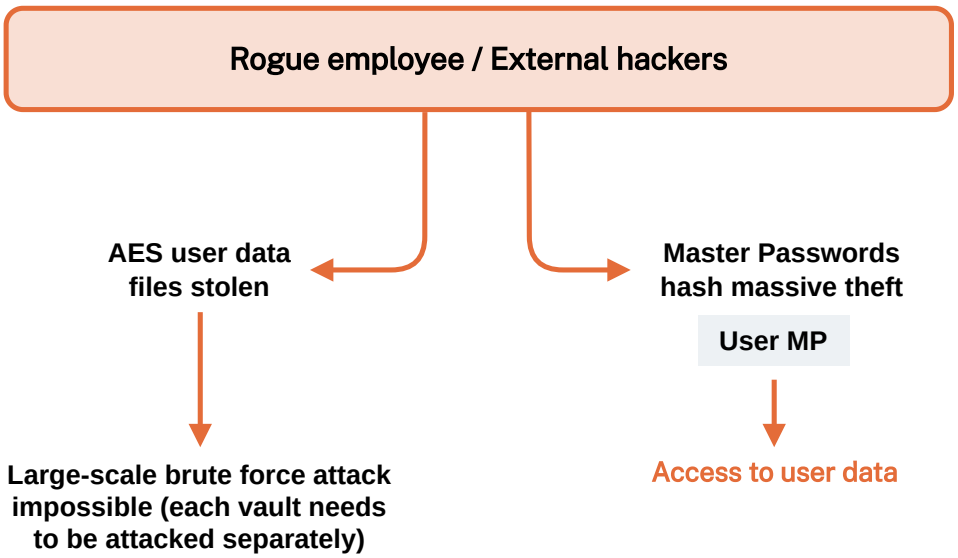


Fig. 21 : Scénarios d'attaque avec la plupart des architectures cloud

3.3 Architecture de sécurité de Dashlane

Pour rendre ce scénario d'attaque impossible, nous avons pris la décision de séparer la clé utilisée pour le chiffrement des données de l'utilisateur et la clé utilisée pour l'authentification basée sur le serveur (voir Figure 22). Les données de l'utilisateur sont chiffrées avec une clé, qui est un dérivé du $User_{MP}$ ou du $MachineGenerated_{MP}$. Une $Device_{Key}$ séparée (unique à chaque couple appareil-utilisateur) est utilisée pour effectuer l'authentification sur les serveurs de Dashlane. La $Device_{Key}$ est générée automatiquement par Dashlane. Par conséquent :

- Les clés de chiffrement pour les données utilisateur ne sont stockées nulle part.
- Aucun employé de Dashlane ne peut avoir accès aux données des utilisateurs.
- Les données utilisateur sont protégées par $User_{MP}$ ou $MachineGenerated_{MP}$ même dans l'éventualité où les serveurs de Dashlane seraient compromis.

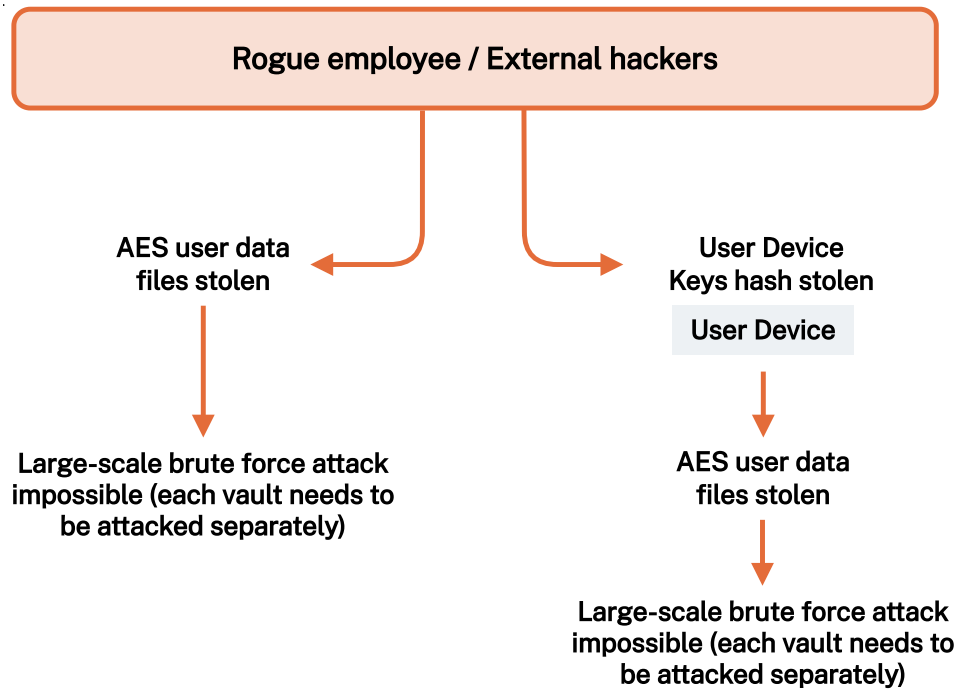


Fig. 22 : Scénarios d'attaque avec l'architecture de sécurité de Dashlane

Même si ce scénario se produit, un employé frauduleux ou un pirate externe aurait beaucoup de mal à exécuter une attaque par force brute ou par dictionnaire sur les fichiers de données de l'utilisateur AES, car nous utilisons l'algorithme Argon2d (ou PBKDF2-SHA2). Comme les données de l'utilisateur sont chiffrées à l'aide d'une clé salée, qui est un dérivé du $User_{MP}$ ou du $MachineGenerated_{MP}$, aucune attaque préconçue ne devrait être possible.

3.4 Mesures anti-clickjacking

Afin de protéger les utilisateurs de Dashlane contre les sites Web frauduleux qui tenteraient d'utiliser des techniques de détournement de clics (clickjacking) ou d'autres attaques basées sur JavaScript, afin d'extraire des données de l'application Dashlane, nous avons fait en sorte qu'aucune des interactions basées sur des pages Web, qui concernent des données utilisateur non liées au site en question n'utilisent JavaScript.

Les pop-ups utilisés pour déclencher la saisie automatique dans les formulaires sur une page Web, utilisent une variété d'API de sécurité du navigateur, pour empêcher le contrôle du code JavaScript de la page consultée. De ce fait, un site Web frauduleux ne peut pas déclencher un clic qui ferait croire à Dashlane que l'utilisateur a effectivement cliqué et ne peut donc pas extraire des informations à moins que l'utilisateur ne clique explicitement dans le champ.

3.5 Politique de même origine

Dashlane connecte automatiquement les utilisateurs aux sites Web. Pour éviter de fournir les informations des utilisateurs aux sites Web frauduleux, la politique de

même origine est toujours respectée.

Tout d'abord, un identifiant enregistré par Dashlane lorsqu'il a été utilisé sur un site Web avec l'URL de mysubdomain.mydomain.com ne sera pas automatiquement saisi sur un autre site Web avec l'URL de myothersubdomain.mydomain.com. Cela empêche l'identifiant d'un site Web spécifique d'être fourni à un autre site Web qui partage le même nom de domaine de premier niveau.

De plus, un identifiant enregistré par Dashlane, quand il a été utilisé sur un site dont l'URL commence par https, ne sera pas automatiquement saisi sur un autre site Web dont l'URL commence par http.

3.6 Protection de la mémoire

Un problème peut survenir si un pirate réussit à contrôler l'appareil client de l'utilisateur. Dans ce scénario, l'attaquant pourrait récupérer les données de l'utilisateur déchiffrées à partir de la mémoire.

C'est un scénario extrême, car dans ce cas, le pirate peut contrôler de nombreux aspects, par exemple, en ajoutant un keylogger pour récupérer le $User_{MP}$ ou le code PIN.

- Les systèmes d'exploitation mobiles (Android, iOS) s'assurent qu'aucun processus ne puisse jamais accéder à la mémoire d'un autre processus et ne sont pas directement affectés.

Enfin, nous pensons que l'intégrité du système et la sécurité entre les processus constituent une fonction système et que Dashlane ne peut pas (et ne doit pas) tout réinventer ni complexifier les choses inutilement, ce qui pourrait créer d'autres vulnérabilités et avoir des effets secondaires.

Pour plus d'informations sur la façon dont Dashlane peut vous aider à améliorer la sécurité des mots de passe, veuillez nous contacter et nous demander.

Appendices

A Journal d'Activité - Liste des événements

A.1 Les journaux d'activité par défaut

Nom de l'événement	Message d'événement
master_password_reset_accepted	A accepté la demande de récupération de compte effectuée par %(email)s
master_password_reset_refused	A refusé la demande de récupération de compte effectuée par %(email)s
user_device_added	A ajouté l'appareil %(name)s
user_device_removed	A supprimé l'appareil %(name)s
requested_account_recovery	Demande de récupération de compte
completed_account_recovery	Accès au compte recouvré grâce à la récupération de compte
dwm_email_added	Adresse %(email)s ajoutée à la surveillance du dark Web
dwm_email_removed	Adresse %(email)s supprimée de la surveillance du dark Web
user_group_created	A créé un groupe nommé %(groupName)s
user_group_renamed	A renommé le groupe %(oldGroupName)s en %(newGroupName)s
user_group_deleted	A supprimé le groupe %(groupName)s
user_joined_user_group	A rejoint le groupe %(groupName)s
user_invited_to_user_group	A invité %(email)s dans le groupe %(groupName)s
user_declined_invite_to_user_group	A refusé de rejoindre le groupe %(groupName)s
user_removed_from_user_group	A supprimé %(email)s du groupe %(groupName)s
team_name_changed	A changé le nom de votre entreprise en « %(name)s »
new_billing_period_created	A repoussé la date d'expiration de votre compte au %(date)s
seats_added	A ajouté %(count)s licences à votre compte
domain_requested	A ajouté %(domain)s comme domaine non vérifié
domain_validated	A vérifié le domaine %(domain)s
collect_sensitive_data_audit_logs_enabled	(utilisateur) a activé les journaux du coffre-fort non chiffrés
collect_sensitive_data_audit_logs_disabled	(utilisateur) a désactivé les journaux du coffre-fort non chiffrés
sso_idp_metadata_set	A mis à jour les métadonnées du fournisseur d'identité SSO
sso_service_provider_url_set	A configuré l'URL du fournisseur SSO
sso_enabled	A activé la SSO
sso_disabled	A désactivé la SSO
contact_email_changed	A changé son adresse e-mail de contact pour %(email)s
master_password_mobile_reset_enabled	A activé la récupération biométrique pour %(deviceName)s
two_factor_authentication_login_method_added	A activé une méthode de double authentification
two_factor_authentication_login_method_removed	A supprimé une méthode de double authentification
user_invited	A invité %(email)s à rejoindre votre compte
user_removed	A révoqué l'accès de %(email)s à votre compte
team_captain_added	A modifié %(email)s pour accorder des droits administrateurs
team_captain_removed	A modifié %(email)s pour donner des droits de membres
group_manager_added	A modifié %(email)s pour donner des droits d'administrateur de groupe
group_manager_removed	A modifié %(email)s pour donner des droits de membres
user_reinvited	A renvoyé une invitation à %(email)s
billing_admin_added	A désigné %(name)s en tant que responsable de la facturation
billing_admin_removed	A révoqué le statut de %(name)s en tant que responsable de la facturation
nitro_user_provisioning_activated	Dashlane Confidential SSO - User Login Flow
nitro_user_provisioning_deactivated	Provisionnement confidentiel des utilisateurs désactivé
nitro_group_provisioning_activated	Provisionnement confidentiel des groupes activé
nitro_group_provisioning_deactivated	Provisionnement confidentiel des groupes désactivé
nitro_siem_activated	Export des logs d'activité vers un SIEM activé
nitro_siem_edited	Configuration de l'intégration SIEM modifiée
nitro_siem_deactivated	Export des logs d'activité vers un SIEM désactivé
nitro_integration_app_installed	%(integration$_{a}pps)sinstalle$
nitro_integration_app_uninstalled	%(integration$_{a}pps)sdsinstalle$
nudge_configured	%(nudge_name)s activé/désactivé
nudge_executed	%(successes)s nudges envoyés pour %(nudge_name)s
user_received_nudge	%(nudge_received)s nudges reçus
mass_deployment_configuration_updated	Définir la détection des risques de de masse sur height

Tab. 5 : Les journaux d'activité de Dashlane

A.2 Les journaux d'activité sensibles supplémentaires

Nom de l'événement	Message d'événement
collect_sensitive_data_audit_logs_enabled	(utilisateur) a activé les journaux d'activité supplémentaires (non chiffrés)
collect_sensitive_data_audit_logs_disabled	(utilisateur) a désactivé les journaux d'activité supplémentaires (non chiffrés)
user_shared_credential_with_group	(utilisateur) a partagé des droits %(rights [limited/full]) de la %(domain)s connexion avec %(group)s
user_shared_credential_with_email	(utilisateur) a partagé des droits %(rights [limited/full]) de la %(domain)s connexion avec %(email)s
user_shared_credential_with_external	(utilisateur) a partagé des droits %(rights [limited/full]) de l'identifiant %(domain)s avec l'utilisateur externe %(email)s
user_accepted_sharing_invite_credential	(utilisateur) a accepté une invitation de partage pour l'identifiant %(domain)s
user_rejected_sharing_invite_credential	(utilisateur) a rejeté une invitation de partage pour l'identifiant %(domain)s
user_revoked_shared_credential_group	(utilisateur) a révoqué l'accès à l'identifiant de %(domain)s pour %(group)s
user_revoked_shared_credential_external	(utilisateur) a révoqué l'accès à l'identifiant %(domain)s de l'utilisateur externe %(email)s
user_revoked_shared_credential_email	(utilisateur) a révoqué l'accès à l'identifiant de %(domain)s pour %(email)s
user_created_credential	(utilisateur) a créé un identifiant pour %(domain)s
user_modified_credential	(utilisateur) a modifié l'identifiant pour %(domain)s
user_deleted_credential	(utilisateur) a supprimé l'identifiant pour %(domain)s
user_created_collection	(user) created a Collection %(name)s
user_imported_collection	(user) imported [] logins into the Collection %(name)s
user_added_credential_to_collection	(user) added the login for %(domain)s to the Collection %(name)s
user_removed_credential_from_collection	(user) removed the login for %(domain)s from the Collection %(name)s
user_renamed_collection	(user) modified the name for the Collection %(name)s
user_shared_collection_with_user	(user) shared Collection %(name)s with %(roles)s role with %(email)s
user_shared_collection_with_usergroup	(user) shared Collection %(name)s with %(roles) role with %(group)s
user_accepted_collection_invite	(user) accepted the sharing invitation to the Collection %(name)s
user_rejected_collection_invite	(user) rejected the sharing invitation to the Collection %(name)s
user_added_credential_to_shared_collection	(user) added the %(domain)s login with %(rights) to to the Collection %(name)s
user_updated_collection_usergroup	(user) updated %(group)s from %(roles) role to %(roles) role for the Collection %(name)s
user_updated_collection_user	(user) updated %(email)s from %(roles) role to %(roles) role for the Collection %(name)s
user_revoked_collection_usergroup	(user) revoked access to the Collection %(name)s for %(group)s
user_revoked_collection_user	(user) revoked access to the Collection %(name)s for %(email)s
user_typed_password	Mot de passe %(security$_{s}tatus[faible/compromis])ssaisisur$

Tab. 6 : Les journaux d'activité sensibles de Dashlane

B Change History

v2.2.0 (2024-01-08)

fix : resize and rename Table 1
fix : change page numbering in the table of contents
feat : adding new collection activity logs
feat : add change history section

v2.3.0 (2024-02-02)

fix : remove some format misconfigurations
feat : add info about entropy in tab.1
fix : remove specific antivirus mention in section 3.6
fix : simplify section 1.7

v2.4.0 (2024-03-18)

feat : add paragraph about zxcvbn for Master Password

v2.5.0 (2024-05-29)

feat : add sections about confidential user & group provisioning

v2.6.0 (2024-06-10)

chore : remove Dashlane Authenticator mentions

v2.7.0 (2025-01-29)

feat : add sections about CRD and Nudges
fix : update figure 4 and figure 5 caption and labels
chore : add Nudges activity logs to the appendix