|  |  |  |
| --- | --- | --- |
|  | swSSO  Documentation technique |  |

**Contenu du document**

[1. Stockage du mot de passe maître swSSO et sécurisation des mots de passe secondaires 3](#_Toc498939861)

[1.1. Cas d'application 3](#_Toc498939862)

[1.2. Principe de fonctionnement 3](#_Toc498939863)

[1.3. A quel moment sont déchiffrés les identifiants et mots de passe secondaires ? 4](#_Toc498939864)

[2. Fonctionnement de la synchronisation avec le mot de passe Windows 4](#_Toc498939865)

[2.1. Description 4](#_Toc498939866)

[2.2. Cas particuliers 5](#_Toc498939867)

[2.3. Protection du mot de passe Windows de l'utilisateur 5](#_Toc498939868)

[2.4. Cinématique détaillée de lancement de swSSO.exe 5](#_Toc498939869)

[3. Procédure de secours manuelle 6](#_Toc498939870)

[3.1. Cinématique 6](#_Toc498939871)

[3.2. Outillage du processus 7](#_Toc498939872)

[3.3. Sécurité 7](#_Toc498939873)

[3.4. Détail technique de la procédure 8](#_Toc498939874)

[4. Procédure de secours automatique (web service) 9](#_Toc498939875)

[4.1. Cinématique 9](#_Toc498939876)

[4.2. Sécurité 9](#_Toc498939877)

[5. Migration 9](#_Toc498939878)

[5.1. Principe 9](#_Toc498939879)

[5.2. Sécurité 10](#_Toc498939880)

# Stockage du mot de passe maître swSSO et sécurisation des mots de passe secondaires

## Cas d'application

La sécurisation des mots de passe secondaires présentée dans ce paragraphe s'applique dans les deux cas suivants :

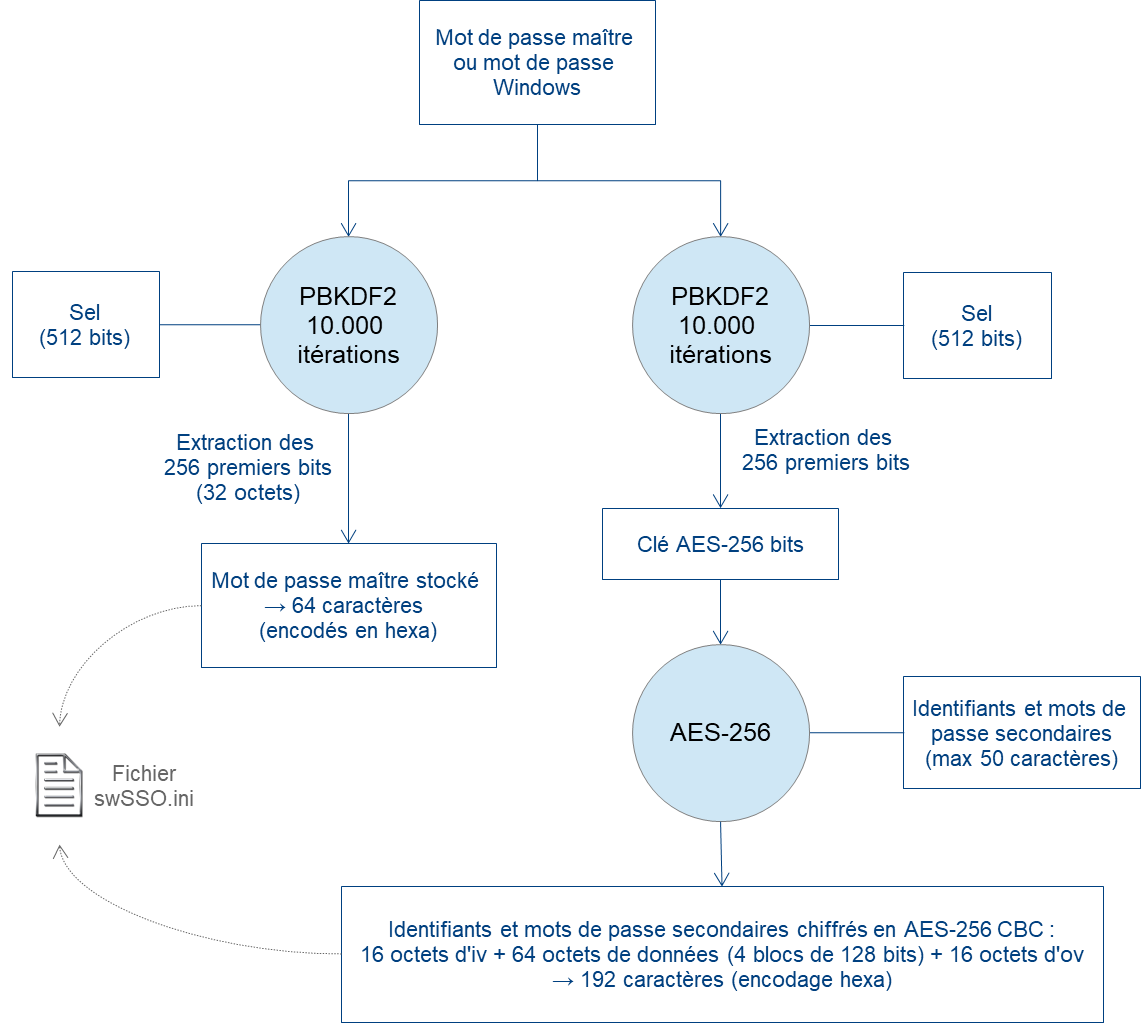
Un mot de passe maître swSSO est utilisé (mot de passe saisi au lancement de swSSO) ;

swSSO est couplé avec l'authentification Windows (pas de mot de passe saisi au lancement de swSSO). Dans ce cas, la génération de la clé de chiffrement des mots de passe secondaires utilise bien le mécanisme présenté ci-dessous, mais en dérivant la clé du mot de passe Windows. Le mot de passe Windows n'est pas stocké par swSSO dans le fichier .ini :la branche de gauche dans le schéma ci-après n'existe pas dans ce cas.

## Principe de fonctionnement

Branche de droite sur le schéma : la dérivation de la clé de chiffrement des mots de passe secondaire à partir du mot de passe maître (ou mot de passe Windows lorsque le couplage est activé) s'appuie sur une fonction PBKDF2 avec 10.000 itérations. Les 256 premiers bits (nommés KeyData dans la suite du document) permettent de générer une clé AES-256 qui sert à chiffrer / déchiffrer les mots de passe secondaires. Le chiffrement met en œuvre un IV (initial vector) aléatoire, l'algorithme AES en mode CBC et un padding PKCS#7.

Branche de gauche sur le schéma : le stockage du mot de passe maître (uniquement lorsque le couplage avec l'authentification Windows n'est pas activé) est protégé par PBKDF2, également avec 10.000 itérations. Dans le fichier swSSO.ini, il s'agit de la valeur « pwdValue » stockée dans la section [swSSO].



## A quel moment sont déchiffrés les identifiants et mots de passe secondaires ?

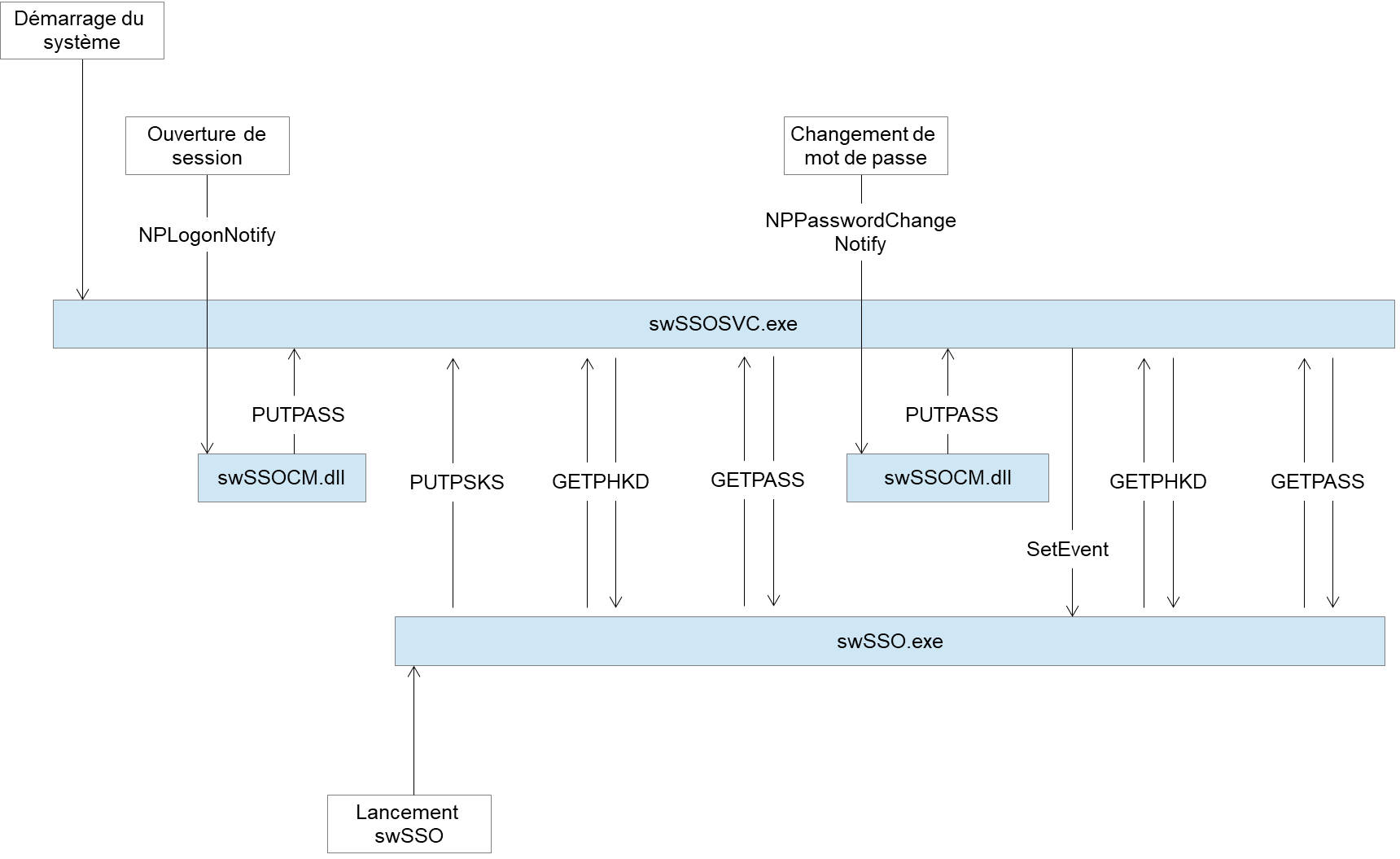
Dans le fichier swSSO.ini, les identifiants et les mots de passe sont chiffrés avec la méthode présentée précédemment :

Les identifiants sont déchiffrés dès le lancement de swSSO et conservés en clair en mémoire pendant tout le temps d'exécution de swSSO ;

Les mots de passe sont déchiffrés uniquement lorsque nécessaire, c'est-à-dire lorsqu'un SSO doit être réalisé ou lorsque l'utilisateur clique sur la loupe pour visualiser le mot de passe.

# Fonctionnement de la synchronisation avec le mot de passe Windows

## Description



Au démarrage du système, le service swSSOSVC.exe démarre et crée un pipe pour recevoir les demandes des autres modules ;

À l’ouverture de session, le système (Winlogon) appelle swSSOCM.dll (NPLogonNotify) et lui transmet le mot de passe Windows de l’utilisateur ;

swSSOCM.dll chiffre le mot de passe et l’envoie à swSSOSVC.exe via le pipe (PUTPASS) : swSSOSVC.exe authentifie l’appel en vérifiant que le processus appelant est bien mpnotify.exe dans le dossier system32.

Lorsqu’il démarre, swSSO.exe se connecte au pipe ouvert par swSSOSVC.exe : ce dernier authentifie l’appel en identifiant le processus appelant et en vérifiant que son hash SHA-256 correspond bien à celui enregistré en base de registre ;

swSSO.exe envoie les sels à swSSOSVC.exe (PUTPSKS) et lui demande le KeyData (GETPHKD) qui lui permet de construire la clé AES de (dé)chiffrement des mots de passe secondaires ;

Si swSSO a besoin du mot de passe Windows de l’utilisateur (utilisation de %ADPASSWORD% ou des options de synchronisation des mots de passe d’un groupe d’application avec le mot de passe Windows), il le demande à swSSOSVC.exe (GETPASS) qui lui retourne chiffré par la clé de chiffrement des mots de passe secondaires ;

À chaque changement de mot de passe Windows par l’utilisateur, le système (Winlogon) appelle swSSOCM.dll (NPLogonNotify et/ou NPPasswordChange) qui transmet le nouveau mot de passe de l’utilisateur à swSSOSVC.exe via le pipe, qui signale lui-même le changement de mot de passe par l’envoi d’un événement à swSSO.exe, qui transchiffre les mots de passe secondaires avec le nouveau mot de passe.

## Cas particuliers

Si swSSO n’est pas lancé au moment du changement de mot de passe, il n’arrivera pas à déchiffrer les mots de passe secondaires avec le nouveau mot de passe (ils sont toujours chiffrés avec l’ancien). Ce cas est géré, swSSO demande le KeyData correspondant à l’ancien mot de passe à swSSOSVC et peut ainsi faire le transchiffrement ;

Si le changement de mot de passe est fait sans que swSSOSVC puisse en avoir connaissance (réinitialisation du mot de passe dans l’AD, swSSOSVC arrêté, …), swSSO ne peut plus déchiffrer les mots de passe secondaires : la procédure de « récupération de mot de passe » est lancée par swSSO (procédure manuelle ou automatique via challenge / réponse).

## Protection du mot de passe Windows de l'utilisateur

Les précautions suivantes sont prises pour protéger le mot de passe Windows de l’utilisateur :

swSSOCM.dll : les mots de passe fournis par le système sont immédiatement chiffrés avec l'API CryptProtectMemory() pour Vista+ et avec l'API RtlEncryptMemory() pour XP. Le mode CROSS\_PROCESS est utilisé de sorte que swSSOSVC.exe pourra déchiffrer les données. Les données sources sont effacées de manière sécurisée avec SecureZeroMemory().

swSSOSVC.exe : à réception d'un nouveau mot de passe, swSSOSVC le déchiffre puis le rechiffre immédiatement en mode SAME\_PROCESS de manière à le conserver sans qu'un autre process ne puisse le déchiffrer.

Lorsque swSSO.exe a besoin de connaître le mot de passe de l'utilisateur (utilisation de %ADPASSWORD% ou des options de synchronisation des mots de passe d’un groupe d’application avec le mot de passe Windows), swSSOSVC lui fournit en le chiffrant par avec la clé KeyData, de manière à ce que seul l'utilisateur légitime ayant connaissance de cette clé puisse le déchiffrer.

## Cinématique détaillée de lancement de swSSO.exe

Trois cas de figure possibles :

**Cas n°1** - Le fichier swsso.ini n’existe pas : cas de l’utilisateur qui lance swSSO pour la 1ère fois) ;

**Cas n°2** - Le fichier swsso.ini existe et contient la valeur pwdValue (« hash » du mot de passe swSSO, calculé comme présenté dans le paragraphe « Dérivation de clé ») : cas de la personne utilisant déjà swSSO avec un mot de passe maître ;

**Cas n°3 -** Le fichier swsso.ini existe et ne contient pas la valeur pwdValue : cas de la personne utilisant swSSO en mode synchronisé avec Windows.

**Cas n°1 – Le fichier swsso.ini n’existe pas**

WinMain() appelle InitWindowsSSO() qui :

1. Génère les sels (PSKS = Password Salt Key Salt)
2. Envoie les sels à SVC (PUTPSKS)
3. Ecrit les sels dans .ini (pwdSalt et keySalt)
4. Demande le KeyData à SVC (GETPHKD :CUR = Password Hash Key Data)
5. Crée la clé AES de chiffrement des mdp secondaires
6. Génère la CheckSynchroValue et écrit dans le .ini (CheckSynchroValue). Cette valeur est un aléa chiffré avec la clé de chiffrement des mdp secondaires. La valeur en clair de l’aléa n’est pas conservée.

Puis WinMain() crée un header swsso.ini en appelant SaveConfigHeader().

**Cas n°2 – Le fichier swsso.ini existe et contient la valeur pwdValue**

WinMain() appelle AskPwd() qui demande à l’utilisateur de saisir son mot de passe swSSO et crée la clé de chiffrement des mdp secondaires dans ghKey1.

Une fois les configurations chargées et déchiffrées avec ghKey1, WinMain() appelle MigrationWindowSSO() qui :

1. Lit les sels dans le .ini (pwdSalt et keySalt)
2. Envoie les sels à SVC (PUTPSKS)
3. Demande le KeyData à SVC (GETPHKD:CUR)
4. Crée la clé de chiffrement des mdp secondaires dans ghKey2
5. Appelle swTranscrypt() pour transchiffrer le fichier de ghKey1 vers ghKey2
6. Détruit ghKey2 et met la clé de chiffrement des mots de passe secondaires dans ghKey1
7. Supprime la clé pwdValue du fichier swsso.ini
8. Génère la CheckSynchroValue et écrit dans le .ini (CheckSynchroValue).

**Cas n°3 – Le fichier swsso.ini existe et ne contient pas la valeur pwdValue**

WinMain() appelle CheckWindowsPwd() qui :

1. Lit les sels dans le .ini (pwdSalt et keySalt)
2. Envoie les sels à SVC (PUTPSKS)
3. Demande le KeyData à SVC (GETPHKD:CUR)
4. Crée la clé de chiffrement des mdp secondaires
5. Vérifie la CheckSynchroValue (remarque : on se base sur l’erreur de dépadding pour considérer que la CheckSynchroValue est fausse, puisqu’on n’a pas gardé l’aléa en clair pour le vérifier).
6. Si la vérification échoue, demande à SVC le KeyData correspondant à un éventuel ancien mot de passe de l’utilisateur (GETPHKD:OLD) et revérifie la CheckSynchroValue. Si c’est OK,  demande transchiffrement qui sera réalisé une fois les configurations chargées.

# Procédure de secours manuelle

## Cinématique

L’utilisateur clique sur le bouton « mot de passe oublié » dans la fenêtre de login swSSO (cas de l'utilisation d'un mot de passe maître) ou swSSO détecte une désynchronisation du mot de passe Windows.

swSSO génère une séquence de caractères, dénommée challenge par la suite, et l'affiche à l'utilisateur. L'utilisateur copie le challenge dans le presse-papier ou l'enregistre dans un fichier texte.

L'utilisateur envoie ce challenge par ses propres moyens au support (par mail s'il a encore accès à sa messagerie ou en utilisant la messagerie d'un collègue par exemple).

L’opérateur du support colle le challenge (ou importe le fichier texte) reçu par mail dans l’outil de réinitialisation de mot de passe.

Le nom du compte Windows de l'utilisateur ayant émis le challenge est affiché à l'écran. L'opérateur du support doit impérativement s'assurer que nom de compte Windows affiché correspond bien à l'utilisateur qui a fait la demande de secours.

Une réponse est générée : l'opérateur du support copie la réponse dans le presse-papier, l'enregistre dans un fichier texte ou clique sur le bouton « Envoyer par mail » (dans ce cas, un mail est automatiquement généré avec la réponse, l'opérateur du support n'a qu'à renseigner une adresse mail consultable par l'utilisateur et à envoyer le mail).

L'utilisateur relance swSSO et copie la réponse (ou importe le fichier texte) reçu par mail dans la fenêtre qui lui est présentée.

Dans le cas où un mot de passe maître est utilisé, l'utilisateur doit alors en définir un nouveau. Sinon, le mot de passe Windows est automatiquement resynchronisé et l'utilisateur peut continuer à utiliser swSSO.

## Outillage du processus

L'outil en ligne de commande swSSOGenKey.exe permet de générer des couples clés publiques / clés privées :

La clé publique est stockée dans un fichier .reg (ex : swSSO-PublicKey-0001.reg) ; les 4 chiffres 0001 correspondent à l'identifiant de la clé, permettant d'identifier la clé à utiliser en cas de changement ;

La clé privée est stockée dans un fichier .txt (ex :swSSO-PrivateKey-0001.txt). La clé privée est protégée par un mot de passe fort.

Le fichier swSSO-PublicKey-0001.reg doit être importé dans le base de registre de tous les postes de travail : il contient la clé publique qui sera utilisée pour sécuriser le processus de secours.

Le fichier swSSO-PrivateKey-0001.txt doit être importé dans l'outil swSSORecover : cet outil possède son propre magasin de clés, nommé swSSO-Keystore.txt, protégé par un mot de passe par l'opérateur du support. Ce mot de passe devra être saisi à chaque lancement de l'outil et protège toutes les clés importées. Ainsi, le mot de passe de chaque clé privée peut être rangé au coffre, il n'est nécessaire que lors de l'importation de la clé dans le keystore.

## Sécurité

Notation pour la suite du paragraphe :

Kpub/Kpriv : clé publique / clé privée RSA 2048 bits ;

KpubID : identifiant du couple clé publique / clé privée ;

KeyData : données permettant de générer la clé AES-256 de chiffrement des mots de passe secondaires ;

Ks : clé symétrique AES-256 générée aléatoirement au démarrage de la procédure de secours côté client swSSO ;

UserID : nom du compte Windows de l'utilisateur ;

(données)Kpub : données chiffrés par la clé publique ;

(données)Ks : données chiffrées par la clé Ks

## Détail technique de la procédure

**Au 1er lancement**, swSSO.exe récupère Kpub en base de registre et stocke dans swSSO.ini au format :

KpubID:(KeyData+UserID)Kpub

**Au démarrage de la procédure de secours**, swSSO.exe génère la clé aléatoirement la clé Ks, la stocke dans swSSO.ini et génère le challenge :

(KeyData+UserID)Kpub + (Ks)Kpub

Le challenge est encodé en hexadécimal et précédé et suivi des balises :

---swSSO CHALLENGE---

swSSORecover.exe utilise la clé privée pour déchiffrer les 2 blocs composant le challenge :

KeyData+UserID

Ks

swSSORecover.exe construit la réponse :

(KeyData)Ks

La réponse est encodée en hexadécimal et précédée et suivi des balises :

---swSSO RESPONSE---

swSSO.exe utilise la clé Ks stockée dans swSSO.ini pour déchiffrer la réponse et obtient :

KeyData

swSSO.exe peut alors transchiffrer toutes les données : déchiffrement avec la KeyData reçue dans la réponse et rechiffrement, soit avec le nouveau mot de passe maître défini par l'utilisateur, soit avec la KeyData issu de l'authentification Windows fournie par swSSOSVC.exe.

# Procédure de secours automatique (web service)

## Cinématique

Lorsque le client swSSO détecte une désynchronisation de mot de passe, il invoque le web service de resynchronisation en lui fournissant un challenge. En retour, il reçoit une réponse qui lui permet de réinitialiser le mot de passe maître swSSO avec le mot de passe Windows de l’utilisateur et de transchiffrer les mots de passe secondaires (déchiffrement avec la clé correspondant à l’ancien mot de passe Windows et rechiffrement avec la clé correspondant au nouveau mot de passe Windows).

Techniquement, c'est simplement une automatisation de la procédure de secours, dans laquelle un web service remplace l'opérateur du support et l'application swSSORecover.exe.

## Sécurité

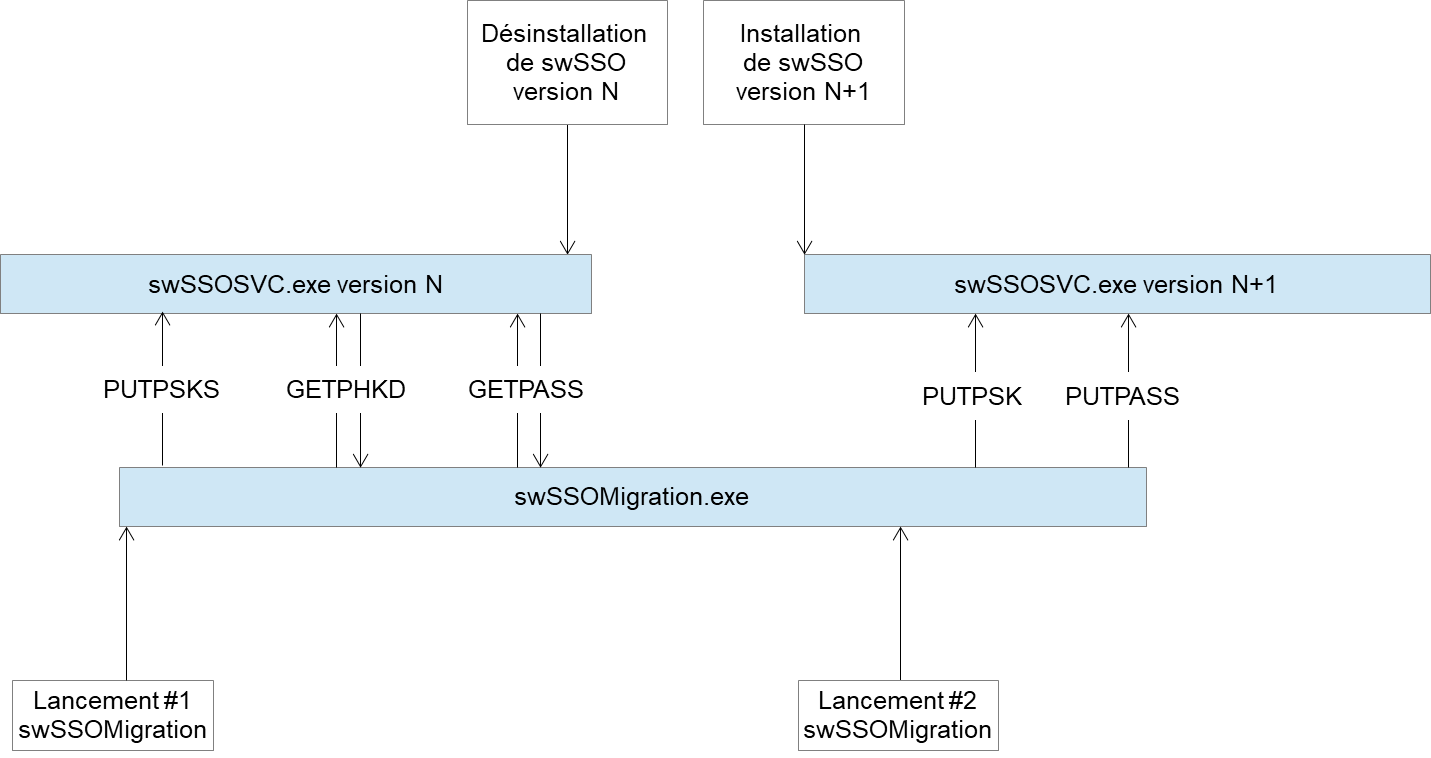
Les principes de sécurité sont les mêmes que dans la procédure manuelle. La seule différence est que l’utilisateur n’est pas authentifié par l’opérateur du support, mais par le serveur IIS configuré en authentification intégrée Windows. La sécurité de la procédure et l’authentification de l’utilisateur repose donc sur sa session Windows.

# Migration

## Principe

Pour permettre de migrer d’une version N à N+1 sans imposer une fermeture / réouverture de session à l’utilisateur qui était nécessaire jusqu’à la version 1.10 pour que swSSOCM.dll fournisse le mot de passe Windows de l’utilisateur à swSSOSVC.exe qui est arrêté le temps de la migration, il est possible d’utiliser l’outil swSSOMigration.exe qui récupère le mot de passe de l’utilisateur auprès de swSSOSVC.exe version N, le conserve en mémoire et le refournit à swSSOSVC.exe version N+1.

Le schéma ci-dessous décrit les échanges entre swSSOMigration.exe et swSSOSVC.exe :



## Sécurité

swSSOSVC.exe authentifie que les appels GETPHKD et GETPASS proviennent bien de swSSOMigration.exe sur la base d’un hash SHA-256 défini en base de registre.

Comme dans le cas de swSSO, le mot de passe Windows retourné par GETPASS est chiffré par la clé de (dé)chiffrement des mots de passe secondaires (KeyData). swSSOMigration déchiffre le mot de passe et le rechiffre en mode CROSS\_PROCESS pour le refournir à swSSOSVC, qui le traite comme décrit dans la phase de connexion avec swSSOCM.dll.