L'AUTHENTIFICATION FORTE

1 Présentation de la solution client/serveur *Mobilegov Digital DNA ID BOX®*

1.1 Principe

Mobilegov ADN du Numérique permet la mise en œuvre facile d'authentification forte entre un prestataire de service et un utilisateur du service, connecté par un LAN, un WAN ou l'Internet, à travers un ordinateur ou un *smartphone* (Symbian, iPhone, Android, Windows Mobile…). Dans une première étape d'enrôlement, l'utilisateur décide quel composant matériel va lui servir de token d'authentification. La présence de ce token l'authentifiera par la suite.

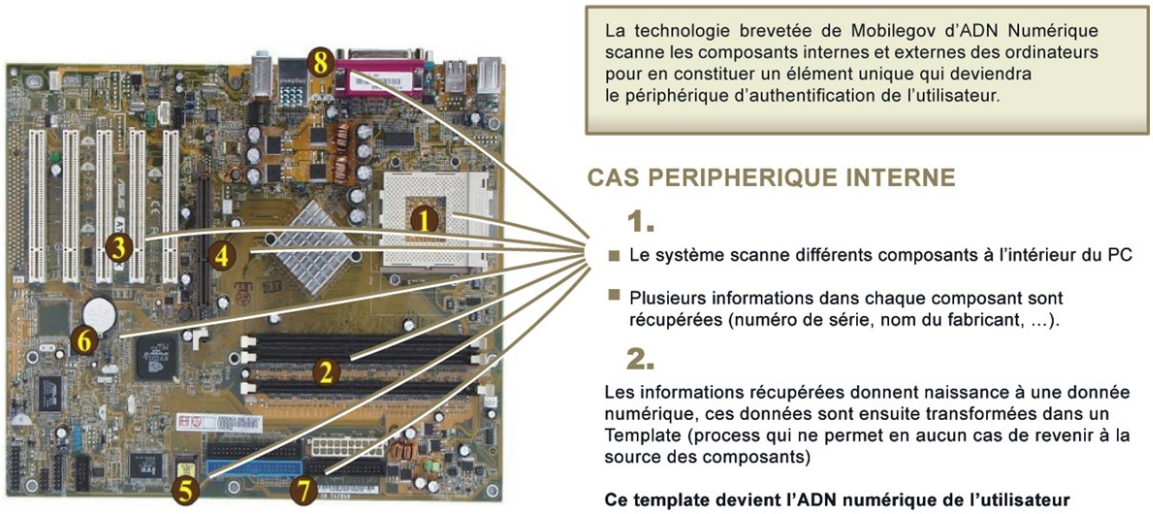

Figure 1: Principe de l'ADN du Numérique

1.1.1 Première étape : enrôlement

L'utilisateur choisit parmi son équipement soit son ordinateur, soit un périphérique connecté à l'ordinateur (une clé USB, un lecteur MP3…), soit un *smartphone*, le token qui servira à l'authentifier. Il peut combiner plusieurs éléments physiques comme son ordinateur ET une clef USB. Il pourra compléter le dispositif par un code PIN ce qui empêchera l'usurpation de son identité en cas de perte ou de vol de son token.

Un protocole défini entre le site web et l'utilisateur décrit comment se fait l'enrôlement : l'utilisateur peut effectuer toute l'opération en ligne ou bien la compléter par une voie supplémentaire comme un SMS. Dans ce cas, muni d'un mot de passe à usage unique reçu de façon traditionnelle (courriel, courrier ou SMS), l'utilisateur accède à la page web spécifique d'enrôlement et identifie en ligne le ou les tokens à utiliser pour son authentification (il lui suffit de cliquer sur les éléments qu'il souhaite parmi tous les éléments détectés).

En cas de perte de son token, l'utilisateur contacte le prestataire, comme il le fait aujourd'hui pour signaler la perte de sa carte de crédit. En retour (courriel, SMS, etc.), il reçoit son nouveau mot de passe à usage unique et il peut enrôler un nouveau token. La différence

réside bien entendu dans le fait qu'il n'a pas besoin d'attendre une logistique pour refaire sa carte bancaire.

1.1.2 Seconde étape : authentification forte

L'authentification repose sur la technologie d'ADN du Numérique® de Mobilegov. Le serveur web qui dispose de cette fonctionnalité contrôle la présence, sur le poste client, du token enrôlé au cours de l'étape précedente. Pour ce faire, il envoie au poste client lors de la première authentification un logiciel client (qui peut être une appliquette JAVA, un module ActiveX, un widget, un plugin du navigateur…). Ce logiciel client :

1. Exploite le certificat X509v3 pour vérifier l'identité du serveur et contrer une tentative de *phishing* ou hameçonnage.
2. Détermine le token d'authentification de l'utilisateur et permet à l'utilisateur de saisir son code PIN éventuel de façon robuste aux logiciels espions.
3. Retourne au serveur une information qui permet d'authentifier l'utilisateur, mais qui ne contient aucune information qui permettrait de déterminer les caractéristiques du token d'authentification. Cette information OTP User Token n'est utilisable qu'une seule fois, son interception ne permettrait pas d'usurper l'identité de l'utilisateur.

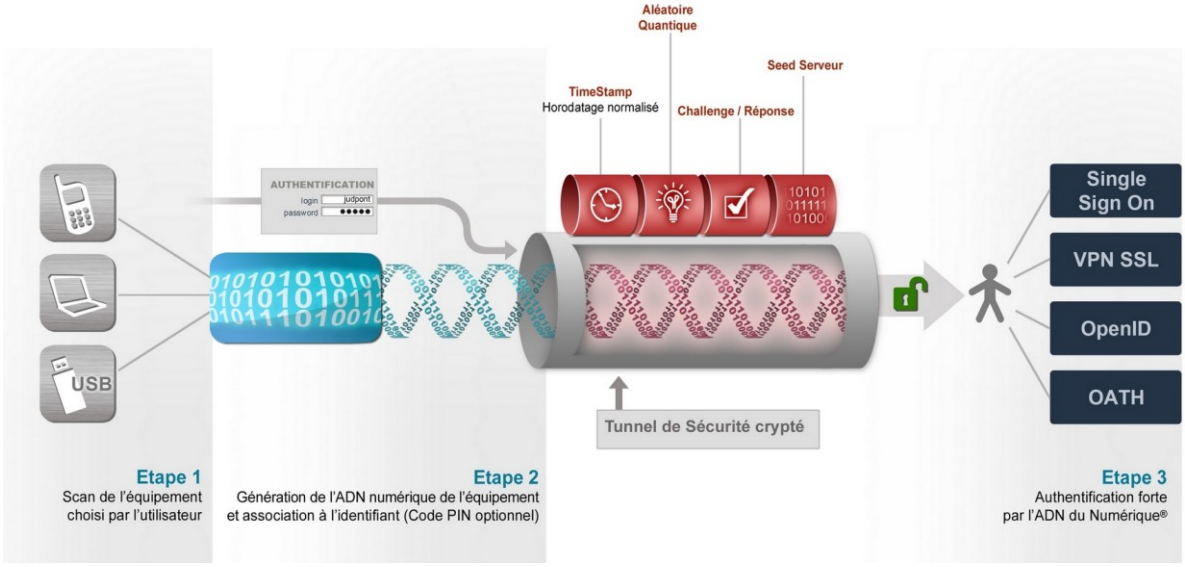

Figure 2: Schéma général de l'ADN du Numérique

1.2 Implémentation

Comme pour toute mise en place d'authentification forte, celle de l'ADN du Numérique nécessite une adaptation du contrôle d'accès du prestataire de service, site web ou site d'entreprise. Mais au contraire des autres solutions de mot de passe à usage unique, Mobilegov ADN du Numérique affranchit le prestataire de toute logistique liée à la gestion des tokens matériels.

Par son architecture, Mobilegov ADN du Numérique convient aussi bien à des configurations de quelques dizaines d'utilisateurs (cas d'un Extranet par exemple) qu'à des configurations de plusieurs millions d'utilisateurs (banque en ligne par exemple).

Pour ce faire, Mobilegov ADN du Numérique a été conçu de manière modulaire et réplicable (technologie JEE qui permet de répartir la charge sur plusieurs machines physiques). Mobilegov ADN du Numérique s'interface avec les principales bases de données SQL et

avec les annuaires d'entreprises (Active Directory, eDirectory…). Il est compatible avec la norme Radius AAA[1], répond aux critères SOX[2] et HIPAA[3] et utilise les standards de la sécurité (chiffrement SSL, AES, protocole CHAP…)

Le serveur de Mobilegov ADN du Numérique est fourni sous forme de boitier *appliance dénommée « Digital DNA ID-BOX »* pour une intégration simple, rapide et efficace. Chaque *appliance* intègre un générateur de nombre aléatoires basé sur la mécanique quantique (qui a passé avec succès les tests NIST et Diehard – *voir annexe 1 Random quantique*) : la sécurité est bien meilleure que les solutions basées sur une génération de nombre pseudo-aléatoires tels que celle utilisée par les systèmes OTP. En effet, la sécurité des OTP repose essentiellement sur l'aspect non prévisible du hasard. Ainsi, avec l'utilisation d'un algorithme quantique le « hasard » est moins prévisible qu'avec un calcul logiciel… L'OTP ADN du numérique est donc de bien meilleure sécurité qu'un OTP « classique ».

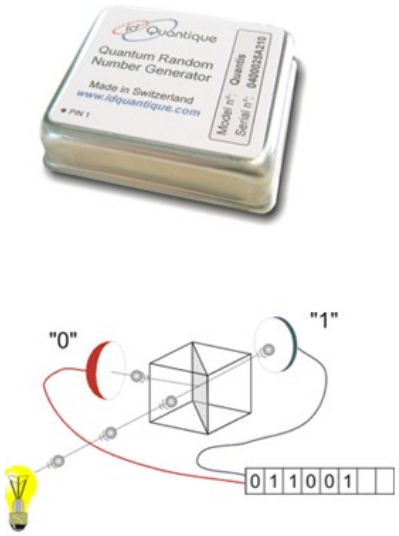

Figure 3: Principe d'un générateur quantique de nombres aléatoires

Mobilegov ADN du Numérique peut être ainsi associé à des fonctions d'horodatage et d'archivage à valeur probante pour fournir une trace légale des transactions.

Mobilegov ADN du Numérique peut être géré localement par le prestataire de service, sur un ou plusieurs serveurs connectés sur son LAN, ou peut être délocalisé dans un datacenter (mode hébergé chez un tiers de confiance ou prestataire de service Internet de type *cloud computing*).

Mobilegov ADN du Numérique authentifie des composants matériels connectés à des clients fonctionnant sous la plupart des systèmes d'exploitation connus (MS Windows, MacOS, Linux, Symbian OS, iPhone, Android …). Le principe intrinsèque de Mobilegov ADN du Numérique le rend adaptable à toute plateforme sur laquelle des composants numériques sont connectés.

---

[1] http://en.wikipedia.org/wiki/AAA_protocol

[2] http://en.wikipedia.org/wiki/Sarbanes-Oxley_Act

[3] http://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act

Comme les autres solutions qui reposent sur des tokens connectés, Mobilegov ADN du Numérique résiste aux attaques de type *keylogger* et *screen logger*. Mobilegov ADN du Numérique permet également de détecter l'usurpation d'interface du serveur et protège contre les attaques de type phishing (le logiciel client est signé par un certificat X509, ce certificat peut également être protégé par un code PIN saisi sur un clavier virtuel pour contrer les *keyloggers*).

Le contrôle de l'ADN du numérique peut être réactivé à tout moment par le serveur de manière transparente et donc sans déranger l'utilisateur. Ainsi, la détection de l'absence du token bloquera la transaction en cours.

Mobilegov ADN du Numérique est de plus engagé dans des procédures de qualification CESG (CCMT), CSPN (Certification de Sécurité de Premier Niveau) et Critères Communs[4] (niveau EAL3+) de façon à prouver la sécurité de son dispositif.

2 Composants logiciels

Mobilegov ADN du Numérique comporte deux composants logiciels : un serveur d'authentification et un logiciel client qui communique avec ledit serveur.

Pour chacun de ces deux logiciels, différentes mesures de sécurité sont appliquées. Le code objet est *obfusqué* [4] ce qui rend difficile le reverse engineering. Toutes les parties natives sont développées en langage de bas niveau (C/C++), sont sous forme de DLL ou d'exécutables et sont signées (signature des *assembly*) par certificat officiel.

```
#include <stdio.h>
main(t,_,a)char *a;{return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2,_+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\
;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l \
q#'+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n')(){}#}w'){){nl]'/+#n';d}rw' i;# \
){nl]!/n{n#'; r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \
;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n' ')# \
}'+}##(!!/")
:t<-50?_==*a?putchar(31[a]):main(-65,_,a+1):main((*a=='/')+t,_,a+1)
  :0<t?main(2,2,"%s"):*a=='/'||main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);}
```

Figure 4: Code obfusqué (Source: Wikipedia)

2.1 Serveur d'authentification

Le serveur d'authentification fonctionne sur un serveur unique, partagé ou répliqué (suivant les besoin de réponse aux charges). Il effectue les opérations d'authentification. L'architecture en trois tiers permet de gérer la montée en charge, assure la fiabilité et la disponibilité.

Le serveur d'authentification permet de gérer les comptes à base d'ADN du numérique et d'authentifier les utilisateurs. Une même *appliance* peut gérer plusieurs groupes de comptes donc plusieurs sites internet par exemple. Une interface graphique conviviale

[4] « *Obfsucation* » http://fr.wikipedia.org/wiki/Evaluation_Assurance_Level & http://en.wikipedia.org/wiki/Obfuscated_code

d'administration est disponible et permet aux administrateurs de se connecter via une clé USB sécurisée (authentification forte à base d'ADN du Numérique).

L'*appliance* intègre en standard un processeur *Intel Core*, un disque dur non mécanique de dernière génération SD, ainsi qu'une carte quantique qui permet de générer des aléas « purs ».

Le serveur d'authentification Digital DNA ID-BOX est une application n-tiers Java Enterprise Edition (JEE) tournant à ce jour (v2.15) sur un serveur d'application, sous Linux Debian (protégé contre les accès non autorisés par les outils traditionnels (gestion des droits, filtrage réseau, …)) et utilise une base de données chiffrée SQL[5]. L'*appliance* peut se synchroniser avec les annuaires d'entreprise (LDAP tels qu'Active Directory de Microsoft). Elle est compatible également avec la norme Radius (AAA). La génération de ses OTP est basé sur un aléa quantique, les logs sont conformes aux normes HIPAA et SOX. (Notamment dans la traçabilité séparée des activités d'administration et d'utilisation)

L'*appliance* Digital DNA ID-BOX a été pensée pour s'intégrer facilement à toute infrastructure informatique. Elle est également disponible en mode *SaaS* (accès à distance) en « *cloud computing »*.

2.2 Logiciel client

Le logiciel client fonctionne sur un ordinateur ou sur un *smartphone,* avec ou sans installation préalable :

1. Sur ordinateur sous MS Windows, MacOS ou Linux :

   Basé sur un UCK « Universal Connector Kit » (librairie développée en langage proche du système C/C++, *obfusqué* avec dll signées), le logiciel client peut prendre la forme suivant les besoins d'une appliquette Java ou ActiveX, d'un plugin (Internet Explorer, Firefox…), d'une web application exécutée via le navigateur web (MS Internet Explorer, Firefox, Safari, Opera…etc…). L'UCK permet également d'intégrer une solution d'ADN du Numérique dans des applications desktop.

   Le logiciel client est signé numériquement par un certificat X509v3 qui authentifie son origine (Mobilegov ou une autre autorité de confiance comme la banque émettrice ou un site web de confiance). L'utilisateur peut vérifier cette signature avant d'autoriser l'exécution. Cette étape de vérification peut être automatisée, de façon à décharger l'utilisateur, ce qui nécessitera alors l'installation d'un logiciel spécifique sur le poste client. Le logiciel client est en charge de collecter l'information relative au matériel présent de manière à générer l'ADN du Numérique du client, de saisir le code PIN de l'utilisateur (cette saisie se fait via un clavier virtuel pour éviter les *keyloggers*) puis d'envoyer un hash de l'ADN du Numérique sur l'ID BOX via les protocoles standards (CHAP, https…).

2. Sur *smartphone* sous Symbian OS, iPhone ou Android, Windows mobile, le traitement est identique, mise à part l'utilisation de Java qui est remplacée par l'utilisation du langage natif du *smartphone*.

L'intelligence côté client est protégée par chiffrement et *obfuscation* quelle que soit la plateforme d'exécution.

---

[5] *Il est à noter ces éléments peuvent être remplacés avec d'autres standards, la solution peut donc fonctionner sur Microsoft Windows Server 2003 ou 2008 (32 ou 64 bits) ou sur d'autres distributions Linux (Redhat, Ubuntu…). Cette modularité est assurée par une implémentation métier en Java (avec code obfusqué).*

3 L'ADN du Numérique

3.1 L'ADN du Numérique

Le logiciel client a comme principale fonction la génération de l'ADN du Numérique (ADNN). Tout composant physique a des caractéristiques qui permettent de l'identifier de manière unique.

Prenons l'exemple d'une simple clé USB. Jusqu'à 16 zones de caractéristiques sont définies (la marque, le modèle, le numéro de série, la taille mémoire…etc…) dans l'interface avec le système d'exploitation. Ces caractéristiques sont visibles sous Microsoft Windows via le gestionnaire de périphériques (raccourci touche WINDOWS + touche "PAUSE" suivi d'un click droit sur un périphérique puis propriétés/Détails affiche la liste des caractéristiques prises en compte dans la génération de l'ADN du Numérique) :

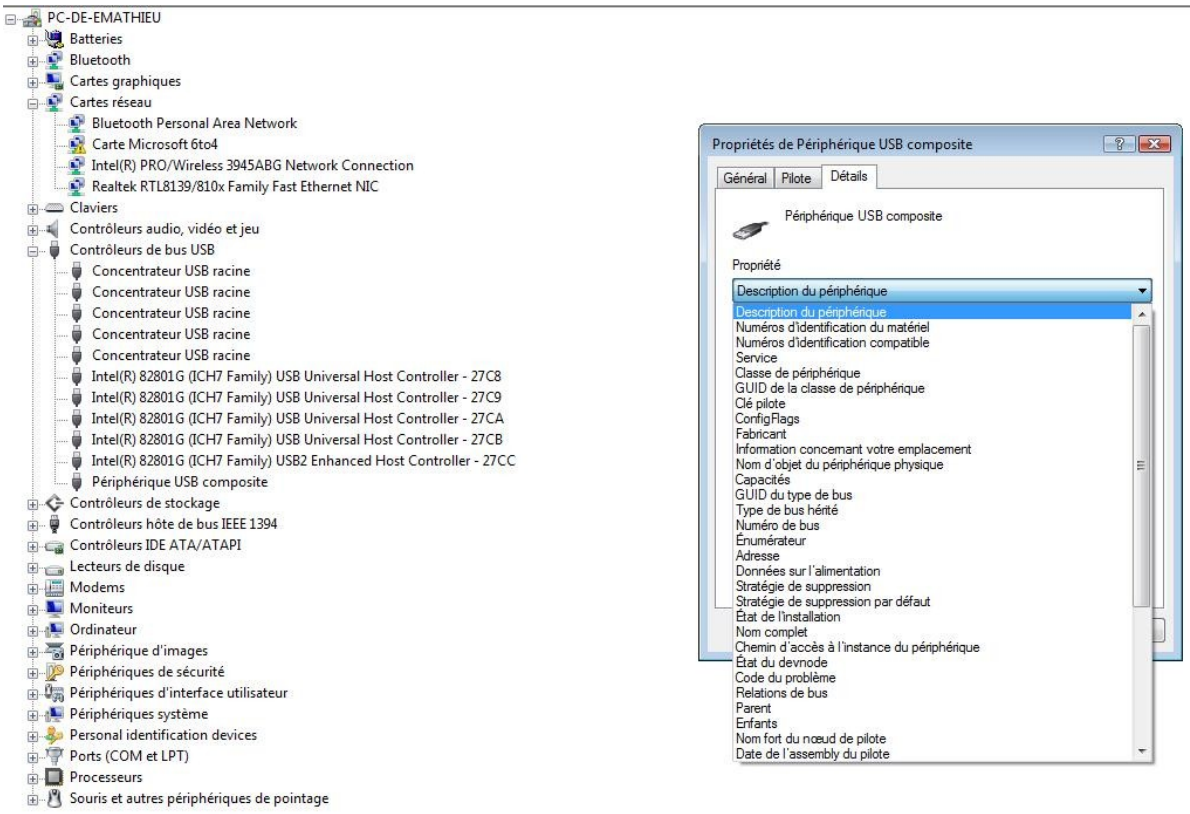

Figure 5: Ce que l'OS connaît des composants matériels

Tout comme une empreinte digitale est représentée par une matrice de points caractéristiques, l'ADN du Numérique est la concaténation de toutes les caractéristiques disponibles pour un matériel (tel qu'une clé USB) ou une liste de matériels (tel que les composants d'un ordinateur). Tout comme l'empreinte digitale est plus déterministe que la taille ou le poids d'un individu, L'ADN du Numérique est nettement plus déterministe qu'une simple adresse IP ou qu'une adresse MAC ou qu'un numéro de série : il est composé de tout le potentiel disponible par toutes les caractéristiques des éléments matériels disponibles.

4 Processus d'authentification

4.1 Schéma d'authentification

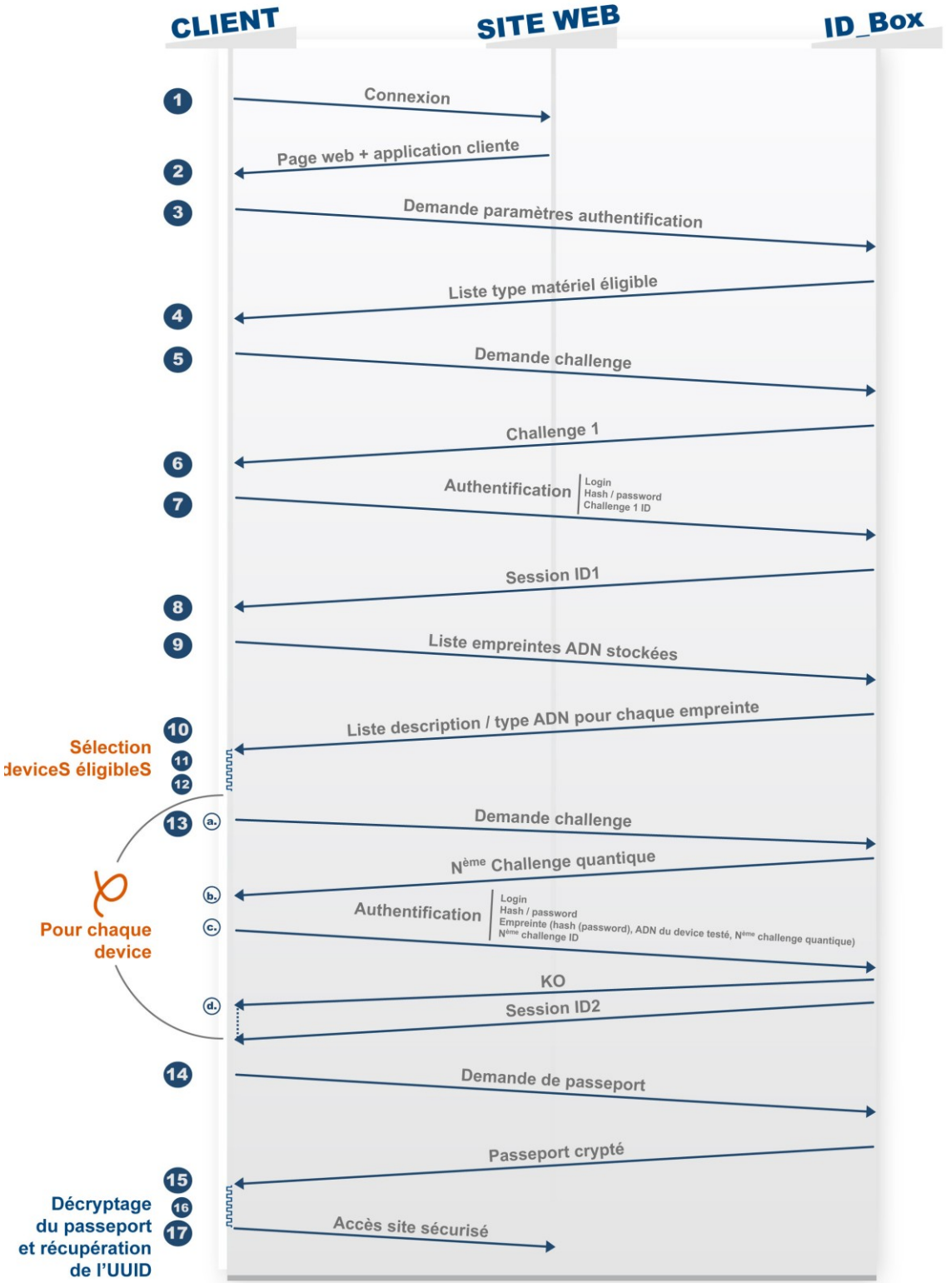

Figure 6: Schéma d'authentification

4.2 Explications du schema

L'authentification fait entrer en jeu le client, le serveur d'authentification (ID BOX) ainsi que le site web. Le processus d'authentification se déroule suivant les étapes ci-dessous. Tous les échanges se font via des canaux https. Voir plus loin le Glossaire pour une explication des termes en **gras**.

1 Le client accède à la page de login du site web

2 Le site web retourne la page web. Si le logiciel client n'est pas déjà disponible sur le poste client, alors il est aussi automatiquement téléchargé depuis le site web.

3 Le logiciel client demande au serveur d'authentification les **paramètres d'authentification.**

4 Le serveur retourne les **paramètres d'authentification**.

5 Le logiciel client demande au serveur un **challenge**

6 Le serveur répond avec un *challenge1***.**

7 Le logiciel client envoie les éléments d'authentification primaire à savoir : le login, un **hash du code PIN** utilisateur et le *challenge1*.

8 Le serveur vérifie les éléments d'authentification primaire puis envoie sa réponse (positive ou négative), avec un ***sessionID1*** en cas de succès de l'authentification primaire.

9 Le logiciel client demande en utilisant le *sessionID1* la description et le type de chacune des **empreintes ADNN** conservées sur le serveur.

10 Le serveur répond avec la description et le type de toutes les **empreintes ADNN** de l'utilisateur.

11 Le logiciel client utilise les types de matériel éligibles au calcul de l'ADNN récupérés en (4) et crée une première sélection de matériels trouvés sur le poste client.

12 Le logiciel client utilise ensuite la liste récupérée en (10) et filtre la précédente sélection.

13 Le logiciel client tente alors une authentification secondaire avec chacun des matériels restant dans la sélection jusqu'à trouver le bon.
   a. Le logiciel client demande au serveur un **challenge**
   b. Le serveur répond avec un $n^{ième}$ ***challenge***.
   c. Le logiciel client calcule une **empreinte ADNN** et la retourne au serveur
   d. Le serveur compare l'empreinte ADNN avec les empreintes associées à l'utilisateur. S'il ne trouve pas d'empreinte identique, le serveur retourne une réponse negative et le logiciel client essaie l'élément matériel suivant. Sinon, l'authentification secondaire est réussie et le serveur retourne *sessionID2*.

14 Le logiciel client demande le **passeport** de l'utilisateur en utilisant le *sessionID2*

15 Le serveur répond avec le **passeport** chiffré

16 Le logiciel client déchiffre le **passeport** pour récupérer la strongkey de l'utilisateur. Il utilise pour ce faire le code PIN de l'utilisateur et l'ADNN du matériel qui l'a authentifié.

17 Le client accède au site via la strongkey.

4.3 Glossaire des termes du schéma

4.3.1 "Paramètres d'authentification"

Les paramètres d'authentification fournissent la liste des types de tokens éligibles pour cette application de l'ADN du Numérique : CPU, carte mère, périphériques USB, smartphone, etc. Les types de tokens éligibles sont définis par l'administrateur du serveur. La liste est un string codé en base 64.

4.3.2 "Challenge"

Le serveur renvoie un objet ayant, entre autres, un champ "challenge" de type string. Son contenu est généré à l'aide de la carte quantique intégrée à l'appliance.

4.3.3 "Empreinte ADNN"

Au moment de l'enregistrement d'un ADNN le résultat de PKCS_PBKDF2_HMAC<SHA256> (sha256(pin code user) + hash(ADNN) + magic number + sel) est calculé par le logiciel client puis stocké sur le serveur, c'est l'empreinte d'ADNN.

4.3.4 "SessionID"

Le serveur renvoie un objet dont on considère un des champs - le champ "token" - comme "sessionID". Ce champ token est un string généré aléatoirement.

4.3.5 "Hash du code PIN"

On utilise le sha256 de la librairie open source cryptopp.

4.3.6 "Passeport"

Le **"passeport"** est crée, chiffré et stocké à la création de l'utilisateur (avec ou sans ADNN) de la façon suivante :

Le logiciel client concatène dans un string :

- un UUID demandé au server
- une strongKey créée en générant un string aléatoire hashée en sha256 avec du sel
- une privateKey (réservée à un usage ultérieur)

Ce passeport est ensuite chiffré par le logiciel client en AES en utilisant comme clé de chiffrement le résultat de

PKCS_PBKDF2_HMAC<SHA256> (sha256(pin code user) + hash(ADNN) + magic number + sel)

Le passeport est stocké sur le serveur pour les authentifications de l'utilisateur via sa strongkey.

4.4 Principaux avantages du schéma d'authentification utilisé

Le processus d'authentification présente deux avantages majeurs :

1. Ni l'utilisateur ni le serveur ne connaissent l'ADNN car ce dernier est généré et il ne quitte jamais le poste client (seule l'empreinte est envoyée). Côté serveur, ceci protège la vie privée de l'utilisateur. Côté client, ceci évite le vol de mot de passe.
2. Mobilegov ADN du Numérique s'appuie sur des schémas standards de sécurité (https, Challenge Handshake Authentication Protocol (CHAP), SHA1, AES…). La

génération des challenges est basée sur un aléa quantique qui répond aux normes les plus strictes (voir Annexe 1).

5 Résistance de L'ADN du numérique® au piratage

5.1 La sécurité mise en œuvre

Le secret partagé entre client et serveur se résume à une information relative à certains composants matériels du client et enfouie dans ces matériels, couplée à un challenge de type OTP. Nous appelons cette information l'ADN du Numérique des composants : ADNN. Cette information est par nature assez proche par exemple de l'information biométrique d'un humain dans le cas de la sécurité biométrique. Aussi, nous mettons en œuvre les mêmes solutions de sécurité dans la collecte, l'extraction et la transmission de ces informations entre client et serveur à travers l'application de la norme X9.84:

- Transfert sécurisé des données d'enrôlement et d'authentification vers le serveur : les données matérielles choisies par l'utilisateur pour être enrôlées sont collectées par l'application puis transmises vers le serveur au travers d'un protocole sécurisé (https). Ce protocole s'appuie sur un échanges de clé initié par le serveur au début de la session et par un chiffrement AES 256 bits des données avant transmission. Il y a également un ajout de *seed* « grain de sel » et de timestamp pour assurer la sécurité et la validité des messages.
- Protection des données rémanentes : toutes les données rémanentes sensibles (clés AES, données matérielles, paramètres) sont effacées des mémoires du PC hôte à la fin de la session.
- Intégration du challenge-réponse de la norme X9.84 pour interdire le rejeu : le protocole d'échanges des données entre le serveur et le client intègre une protection contre le rejeu des trames d'authentification, basé sur l'échange de challenges et sur une fenêtre temporelle à respecter pour réaliser l'échange. L'architecture de Mobilegov SID (Challenge) repose sur l'enveloppe X9.84. L'algorithme de chiffrement utilisé est l'AES 256 en mode CBC. Pour le challenge/réponse, l'algorithme utilisé est le HMAC-SHA1.
- Encapsulation de l'enveloppe PKCS#11 dans l'enveloppe X9.84 contenant l'ADN Numérique : pour être conforme à l'interopérabilité des solutions d'authentification existantes, notre système fournit deux bibliothèques pour la gestion des codes PIN / PUK :

  Dans cette gestion, on retrouve notamment la durée de vie du PIN, le nombre d'essai du PIN, l'activation / désactivation du code PUK.

  - Bibliothèque CSP : permettant pour les applications Microsoft (IE, Outlook, Windows) une interopérabilité avec l'ADN numérique liée à un 2ème facteur qu'est le code PIN.
- Logiciel client signé (personnalisée par code PIN) numériquement par un certificat X509v3 ce qui aide à détecter les tentatives de *phishing.*
- Verrouillage du code: Le code exécutable de l'ADN du Numérique® est protégé par *obfuscation* contre le désassemblage. Des mesures de sécurité sont également implémentées afin de renforcer la sécurité lors de l'exploitation (fermeture de la session forcée après une période d'inactivité ou une erreur par exemple). Enfin, les assembly (DLL et exécutables) sont signées.