yubico

JUILLET 2020

YubiHSM 2:

Une solution de stockage de clés sécurisée pour le secteur industriel

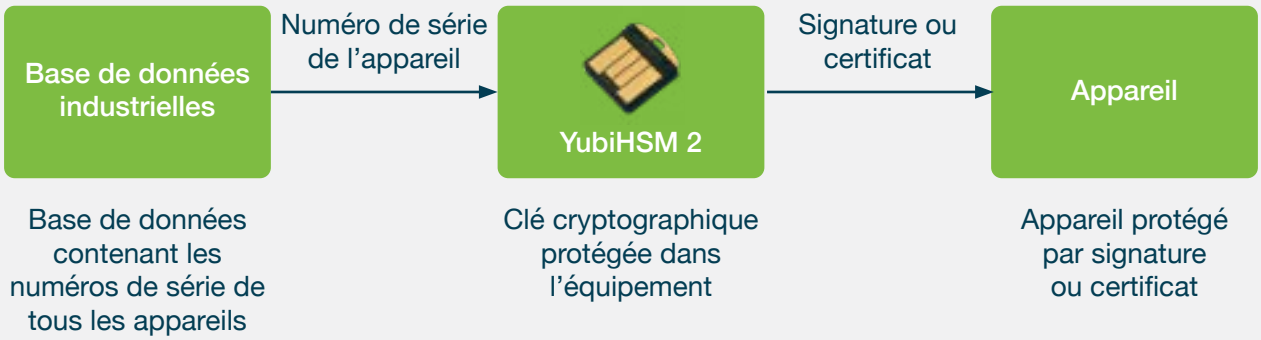
Extrait

Ayant travaillé aux côtés de plusieurs partenaires du secteur industriel responsable de la production à grande échelle de composants électroniques, Yubico a conçu une solution Hardware Security Module pour stocker et protéger les secrets industriels. Les industriels se tournent vers Yubico pour protéger leur chaîne d'approvisionnement et leur propriété intellectuelle, allant des capteurs et appareils médicaux, aux produits automobiles et autres produits contenant des composants électroniques, car l'intégrité de la chaîne d'approvisionnement devient de plus en plus primordiale dans un monde globalisé.

En toutes circonstances, YubiHSM 2 est placé au cœur de la solution, soit pour protéger le processus de fabrication en garantissant que seules des stations de programmation certifiées peuvent interagir avec les composants, soit pour implémenter la signature numérique sur des signatures numériques sur chaque composant afin d'en garantir l'authenticité. Dans les deux cas, la sécurité supplémentaire apportée par le YubiHSM 2 contribue à maintenir la réputation d'une entreprise et lui donne la certitude que ses produits fonctionneront comme prévu même après avoir quitté le site de fabrication.

Ce document présente deux approches courantes au sein du secteur industriel actuel : la première implique un groupe d'action conjoint sur les essais (JTAG pour Joint Test Action Group) et une unité de contrôle électronique (UCE) que l'on trouve couramment dans l'industrie automobile, et la seconde implique un certificat X.509 ou un certificat vérifiable par carte (CVC), que l'on trouve couramment dans la production de petits composants électroniques, de cartes à puce et de dispositifs à interface sans contact tels que l'IoT (Internet des Objets). Il apparaîtra plus tard que les sous-processus de gestion des clés, de signature et de vérification sont souvent comparables dans les deux approches, mais qu'ils divergent en ce qui concerne la production. La thèse de cette proposition est donc une tentative de généraliser simplement cette production, et d'arriver à une conception globale unique qui ne stipule pas de formats spécifiques. Il convient toutefois de souligner que cette approche générale peut être étendue à toute opération de fabrication en dehors des scénarios énumérés, et s'applique à tout processus où les secrets et l'authenticité des composants électroniques doivent être gérés.

Schéma 1 : Bref panorama



Le schéma 1 (ci-dessus) présente un bref aperçu de la solution proposée à mettre en œuvre, soit comme substitut, soit comme tout nouveau procédé dans les usines de fabrication et/ou les lignes de production, avec YubiHSM 2 au centre. Le point le plus important à souligner est que la clé cryptographique utilisée pour signer et/ou certifier les composants n'est jamais exposée en dehors du matériel YubiHSM 2, ce qui garantit un niveau de sécurité élevé.

Introduction

Contexte

Dans le secteur industriel, il est crucial de s'assurer que tous les composants impliqués dans un processus de bout en bout sont authentiques pour éviter les reproductions et les vols non sollicités, mais aussi pour garantir la qualité, puisqu'une chaîne de montage ne doit être constituée que de produits provenant réellement d'une source, l'ensemble étant toujours la somme de ses parties. Par conséquent, il doit toujours y avoir une solution en place pour protéger l'intégrité et la propriété intellectuelle de tous les composants, dans tous les contextes de fabrication, de la production et de l'assemblage, à la réparation et au remplacement.

Il est donc recommandé aux fabricants qui utilisent actuellement des solutions où des informations cryptographiques sensibles sont stockées dans des logiciels, d'envisager de migrer vers une solution YubiHSM 2 – solution centrée sur le fait que les clés maîtresses sont moins susceptibles d'être compromises. Ceci résulte directement du fait que les informations et les opérations cryptographiques sont effectuées directement dans la clé, ce qui protège de manière innée l'exportation d'informations privées et minimise ainsi la menace de compromission de clés privées.

Pour illustrer ce point, nous prendrons un exemple puisque les données confidentielles résidant dans le YubiHSM 2 ne sont jamais exposées. Si un attaquant distant était capable de compromettre le réseau ou l'ordinateur connecté à celui-ci, il n'y aurait toujours pas de vecteurs d'attaque évidents. D'autre part, si le même attaquant était capable d'obtenir un accès complet à un logiciel équivalent, ils pourraient effectuer une analyse de la mémoire ou des fichiers locaux pour détecter des failles ou des modèles potentiels.

Le même raisonnement peut s'appliquer également aux fabricants qui cherchent à mettre en œuvre une solution de toute pièce et évaluent les avantages d'une approche basée sur le logiciel par rapport à une approche basée sur le matériel. En outre, YubiHSM 2, qui possède une approche basée sur le matériel, offre des avantages supplémentaires en permettant une flexibilité dans la conception de la fabrication grâce à sa taille réduite et à son faible coût.

Objectif du document

L'objectif de ce document est de proposer une solution générique destinée aux fabricants et autres entreprises du secteur, centrée sur YubiHSM 2 pour traiter et gérer les informations cryptographiques dans le matériel.

Autres industries en dehors de la fabrication

Même si le public principal de ce livre blanc est censé être l'industrie de la fabrication, YubiHSM 2 n'est en aucun cas limité aux seuls cas d'utilisation industrielle. En fait, toute industrie ou entreprise cherchant à protéger sa propriété intellectuelle ou sa marque avec une cryptographie basée sur le matériel, est susceptible de bénéficier de la proposition contenue dans ce document, qu'elle soit du secteur public, des soins de santé, des services financiers ou d'autres secteurs. La conception elle-même est standard et peut facilement être généralisée et adaptée à un large éventail de circonstances, pour autant que l'objectif soit de protéger les données de l'entreprise.

Abréviations utilisées dans ce document

API	Interface de programmation d'applications
CA	Autorité de Certifications
CMC	Gestion des certificats sur le SMC
CMP	Protocole de gestion des certificats
CMS	Syntaxe des messages cryptographiques
CNG	IPA cryptographique de nouvelle génération
CVC	Certificat vérifiable de carte
DER	Règles d'Encodage particulières
EAC	Contrôle d'accès étendu
ECDSA	Courbe elliptique Algorithme de signature numérique
ECU	Unité de Contrôle Électronique
HMAC	Code d'authentification de messages cachés
HSM	Module de sécurité matériel
HTTP	Protocole de transfert HTTP
IETF	Tâche d'ingénierie Internet
IoT	Internet des objets
ITU-T	Secteur de la normalisation des télécommunications de l'Union des télécommunications
JCA	Architecture de la cryptographie Java
JCE	Extensions de la cryptographie Java
JTAG	Groupe d'action conjoint sur les essais du JTAG
KSP	Fournisseur de stockage de clés
PBKDF2	Fonction de dérivation de clé basée sur le mot de passe 2
PCB	Circuit imprimé
PKCS #5	Norme de la cryptographie à clé publique #5 (Cryptographie par mot de passe)
PKCS #10	Norme de la cryptographie à clé publique #10 (Syntaxe de demande de certification)
PKCS #11	Norme de la cryptographie à clé publique #11 (Interface de jeton cryptographique)
PKI	Infrastructure à clé publique
RA	Autorité d'enregistrement
RFID	Identification à radiofréquence
RSA	Rivest Shamir Adleman
SCEP	Protocole d'inscription au certificat simple
SDK	Kit de développement logiciel
SHA	Algorithme de hachage sécurisé
SN	Numéro de série
TLS	Sécurité de la couche de transport
TLV	Tag-Length-Value
TPM	Module de plateforme de confiance
TRNG	Générateur de nombres aléatoires réels

Pourquoi choisir le YubiHSM 2 ?

Dans une époque où la demande de service télémétrique, de production distribuée ou externalisée et d'environnement de travail à distance augmente dans toutes les industries et dans tous les secteurs, vient également une demande encore plus grande de protection des actifs numériques de l'entreprise créée par ces opportunités. Les entreprises qui ne s'étaient pas préparées pour ce passage vers un mode de fonctionnement axé sur la sécurité des données (data security-centric), vont devoir lutter ou se dépasser pour pouvoir relever les défis de notre société actuelle.

Alors que la YubiKey a été créée avec l'intention de protéger l'identification et les données d'une personne par le moyen d'une clé de sécurité, le YubiHSM 2 a été créé spécifiquement pour les entreprises. L'héritage d'excellence de Yubico ainsi que sa réputation en tant qu'ambassadeur de haut niveau mais également sa sécurité basée sur le matériel informatique simple d'utilisation, se poursuit avec le YubiHSM 2, qui fournit un dispositif cryptographique basé sur le matériel pour rivaliser avec les modules de sécurité matériel (HSM) traditionnels. Comme indiqué ci-dessous, le YubiHSM 2 offre un nombre d'avantages surpassant toutes les solutions traditionnelles, et a été adopté par plusieurs des partenaires de Yubico dans l'espace industriel afin de résoudre une variété de problèmes de sécurité des données.

Des bénéfices uniques

Certains des principaux avantages de la mise en œuvre d'une solution centrée sur YubiHSM 2, et pas seulement sur n'importe quel HSM, peuvent être énumérés comme suit :

- YubiHSM 2 est le plus rentable HSM du marché, à une fraction du prix de ses concurrents
- YubiHSM 2 dispose d'une des plus petites tailles disponibles, à peu près la taille d'un ongle. De plus, il est assorti d'un port USB-A standard qui lui permet d'être interopérable avec une grande variété de services informatiques. Il convient de préciser que le format peut être adapté afin de ressembler à n'importe quel YubiKey si besoin, en fonction du volume de commande.
- La combinaison des deux facteurs mentionnés ci-dessus permet de déployer plus rapidement et facilement le YubiHSM2 sur de multiples sites (i.e. usines, lignes de productions) avec un coût généralement inférieur au coût unitaire équivalent d'un HSM d'un autre fabricant.
- Toutes les informations cryptographiques peuvent être générées et stockées avec le YubiHSM 2, et puisque les clés ne quittent jamais l'élément sécurisé en texte clair, elles ne sont jamais exposées au monde extérieur.
- Le mot de passe d'authentification YubiHSM 2 utilisé pour la gestion des clés et des appareils est distinct et discret par rapport à toute clé cryptographique sous-jacente. Cela implique que même si le mot de passe d'authentification est compromis, les valeurs de hachage dérivées des clés privées en plus des clés elles-mêmes resteront toujours sécurisées.
- Il est possible de dupliquer la clé matérielle sur YubiHSM 2 en utilisant la fonctionnalité "export Wrap" qui permet des déploiements identiques ou des sauvegardes sur plusieurs sites.
- YubiHSM 2 est assez flexible pour être adapté à de nombreux environnements et solutions existants, selon l'utilisation, car il supporte des standards ouverts et acceptés tels que PKCS #11 et Microsoft CNG.
- YubiHSM 2 prend en charge des schémas symétriques tels que AES et HMAC-SHA mais aussi schémas asymétriques tels que RSA et ECDSA. Cela le rend plus flexible et utile dans une variété de scénarios, comme il est décrit dans la partie suivante pour JTAG et les certifications, qui utilisent respectivement des schémas symétriques et asymétriques.

Utilisation n°1 : Protéger JTAG avec des schémas symétriques

Présentation du JTAG

JTAG est une interface de matériel physique classique qui fournit aux ordinateurs et aux appareils un moyen de communiquer directement avec les puces et les broches soudées à une carte de circuit imprimé (PCB). Il a été initialement créé par un consortium (à savoir : Joint Test Action Group) dans le milieu des années 80 afin de faire face à la difficulté croissante des essais au niveau des circuits intégrés, les PCB devenant plus compliqués. JTAG a été largement utilisé depuis, et même aujourd'hui, JTAG a élargi son champ d'action pour inclure le débuggage, la programmation et les tests sur pratiquement tous les appareils embarqués.

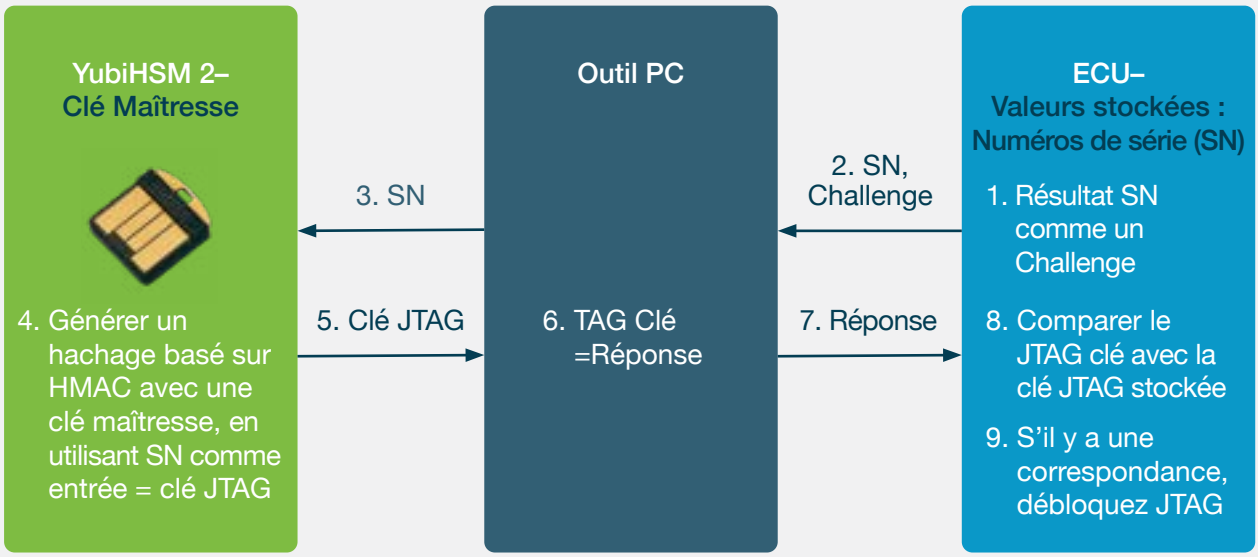
Il est important de noter que l'utilisation de JTAG pour "débugger" un PCB, dans les cercles de l'ingénierie électrique, est très différente du même terme lorsqu'il s'agit de logiciels ou programmes traditionnels. En ce qui concerne JTAG, il s'agit en fait de s'assurer que la broche A sur la puce A, par exemple, est physiquement connectée à la broche B sur la puce B, et en général toutes les broches fonctionnent correctement. Le terme « analyse des limites » est un autre terme courant utilisé pour décrire ce processus et est synonyme de JTAG.

Même si les JTAGs sont utiles aux fabricants de puces pendant les phases de conception, d'essai et même de production, ils représentent aussi un vecteur d'attaque potentielle énorme puisqu'ils fournissent un accès direct aux composants sous-jacents. Heureusement, les fabricants sont bien conscients de ce risque et prennent souvent des mesures pour empêcher l'accès aux interfaces de JTAG, comme brouiller les traces de JTAG sur la carte, les couper entièrement ou même souffler les fusibles dans le câblage de JTAG dans le cadre du processus de fabrication. Ces méthodes sont assez efficaces, bien qu'un attaquant déterminé et doté d'un fer à souder puisse presque toujours réparer les dommages. Une option moins invasive physiquement et sans doute plus saine est de sécuriser l'interface à l'aide de la cryptographie, comme avec un mécanisme Challenge-Réponse, mais même un simple mot de passe statique pourrait suffire dans de nombreux cas.

Mot de passe statique et Challenge-Réponse

Grâce à cette approche, le fabricant crypte ou hache le numéro de série de chaque carte de circuit imprimé pour générer une clé JTAG unique, puis la déploie sur les ECU correspondants dans le cadre du processus de production. La clé JTAG servirait uniquement à protéger l'interface JTAG associée, agissant comme une sorte d'accès, avant qu'un utilisateur puisse interagir et effectuer un débogage. Ainsi, lors d'une tentative d'interaction via le JTAG, l'utilisateur se voit présenter un "challenge" de l'ECU basé sur la clé JTAG, auquel il doit alors répondre par la "réponse" qui correspond exactement soit à la valeur de sortie statique attendue, soit à la valeur de sortie d'une opération algorithmique utilisant le "challenge" comme entrée pour cette opération. Les schémas 2 et 3 (ci-dessous) illustrent le déroulement pas à pas des mécanismes de mot de passe statique et de Challenge-Réponse respectivement, énumérées aux étapes 1 à 9.

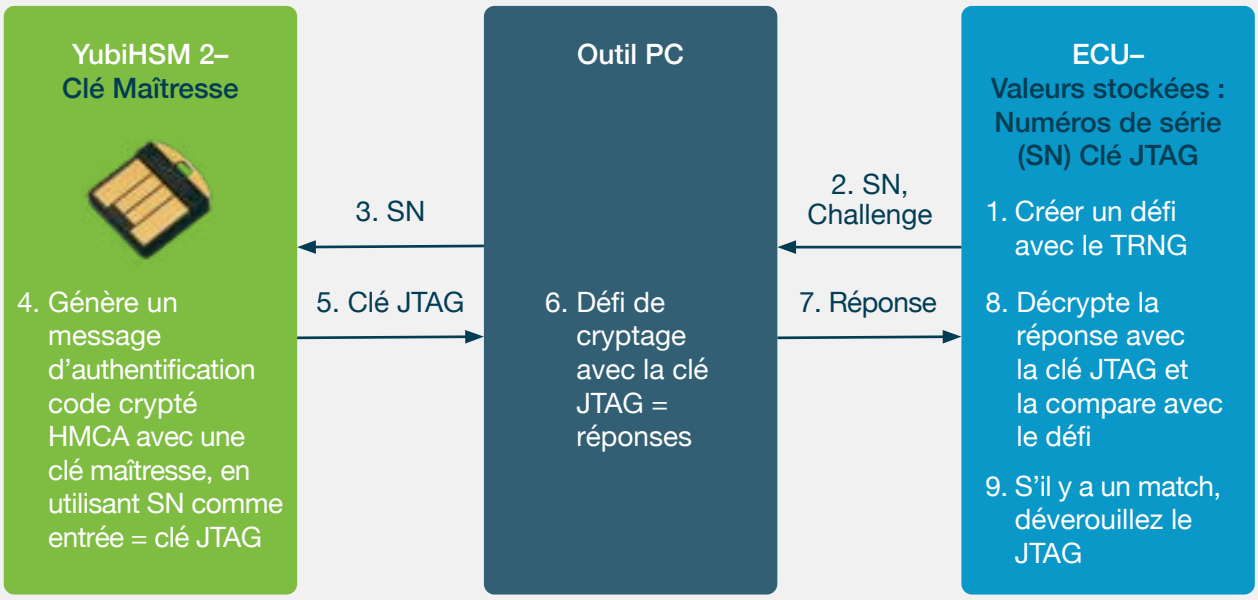
Schéma 2 – Mot de passe statique comme moyen d'autoriser ou de déverrouiller le JTAG



Challenge-Réponse plutôt que Statique

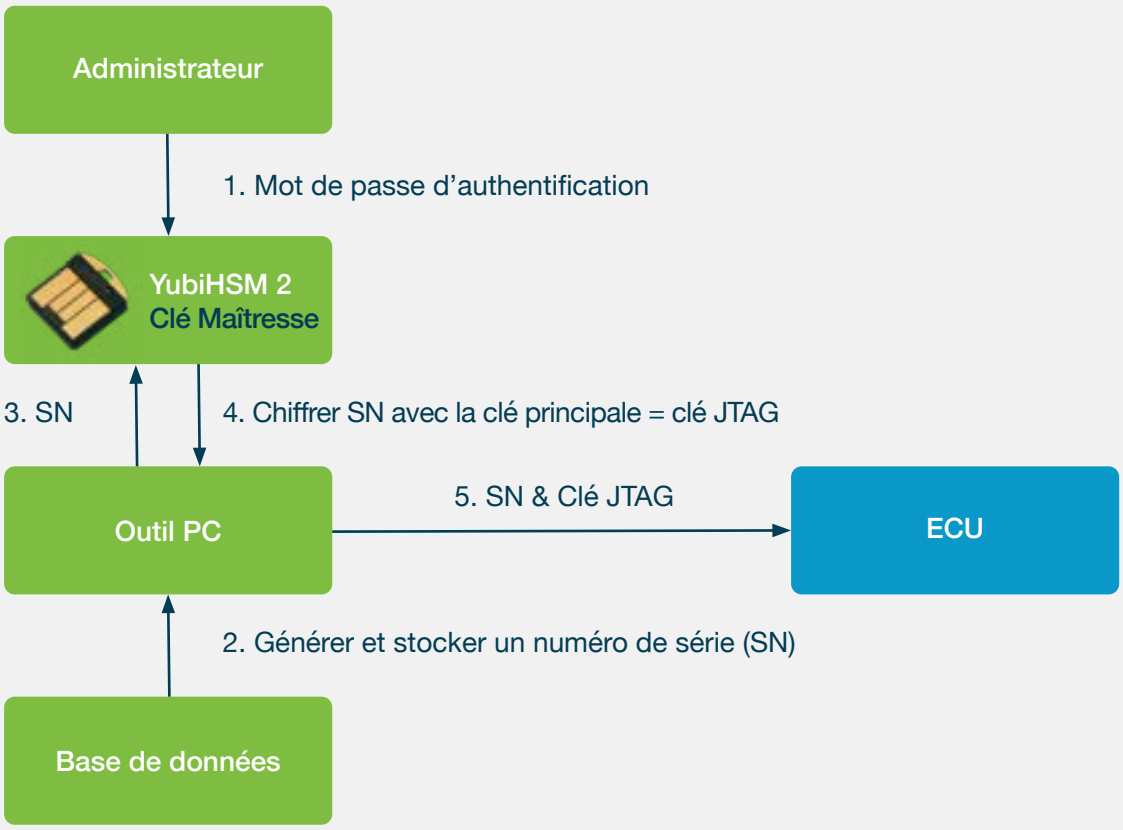
La raison pour laquelle un fabricant pourrait opter pour un mécanisme Challenge-Réponse plutôt que pour une implémentation de mot de passe statique peut être expliquée par le fait qu'il offre une plus grande sécurité, car les Challenge-Réponse sont tous deux générés dynamiquement à chaque nouvelle tentative, contrairement à un mot de passe statique qui est susceptible d'être attaqué à nouveau si le résultat est intercepté. L'inconvénient majeur, cependant, est qu'une Challenge-Réponse peut être plus difficile ou plus coûteuse à mettre en œuvre, selon les outils et les processus mis à disposition par le fabricant au moment de la mise en œuvre et de la réponse.

Schéma 3 – Challenge-Réponse comme moyen d'autoriser ou débloquer le JTAG



Maintenant, avant même qu'un mécanisme symétrique puisse être utilisé, les fabricants doivent générer les numéros de série et les clés JTAG, enregistrer les numéros de série émis et enfin, déployer les clés JTAG sur les ECU concernées. Le schéma 4 (ci-dessous) résume le processus par lequel les fabricants pourraient généralement y parvenir, en utilisant un algorithme cryptographique basé sur HMAC ou un YubiHSM 2, selon le cas.

Schéma 4 – Le processus de génération d'une clé JTAG et son écriture dans une ECU



Pour développer les éléments énumérés dans le diagramme :

1. Le mot de passe d'authentification de l'administrateur est utilisé pour accéder à l'algorithme cryptographique, et est différent de la clé maîtresse HMAC sous-jacente (pour éviter les fuites de clé).
2. Chaque numéro de série (SN) généré avec succès doit être enregistré dans une base de données (sécurisée) à des fins d'audit et de suivi ultérieur.
3. L'utilisation d'un outil intermédiaire fonctionnant sur un PC connecté est probablement nécessaire, afin d'orchestrer l'envoi et la réception des SN et des clés JTAG.
4. Pour chaque ECU, le SN est utilisé comme « sel » de l'algorithme cryptographique, et la clé maîtresse est utilisée pour le cryptage ; la sortie sera toujours une clé JTAG unique.
5. La clé JTAG est alors écrite dans l'ECU, avec le SN correspondant.

Dans les cas où ce processus est déjà en place, bien que toute forme de logiciel soit capable d'exécuter l'algorithme cryptographique, il devrait être possible de le remplacer par un YubiHSM 2, car les entrées et les sorties sont généralement standardisées (voir cette section à venir pour plus d'informations). Dans l'ensemble, la proposition est non seulement simple, mais elle est également sûre, comme en témoignent les preuves de succès auprès des clients de Yubico.

Utilisation n°2: Création de certificats pour faire valoir l'authenticité

Infrastructure à clé publique (PKI)

Avant de développer sur la manière dont YubiHSM 2 pourrait soutenir le déploiement d'un certificat de clé publique, il est important de comprendre le concept d'autorité racine et son rôle dans un cadre de confiance, autrement connu sous le nom d'infrastructure à clé publique (PKI). Une autorité racine (ou parfois appelée autorité de certification racine), est le premier ou le principal nœud à être établi dans un éventuel réseau de nœuds, et est considérée comme la source de vérité au sein d'une PKI. Au début, seule l'autorité racine elle-même est en mesure d'ajouter au réseau en "faisant confiance" explicitement à de nouveaux nœuds (ou, en termes techniques, en signant elle-même chaque nouveau certificat), mais à mesure que le réseau s'étend, des nœuds supplémentaires peuvent également faire confiance à des nœuds qui ont déjà été approuvés par l'autorité racine. Un nœud situé dans la première couche de nœuds supplémentaires est parfois appelé autorité de certification (CA) au sein de la PKI, et ces nœuds, à leur tour, sont capables d'augmenter la portée du réseau de manière exponentielle en tant que mandataire de l'autorité racine. Il doit être clair que chaque développement de la confiance vers l'extérieur se traduit par une "chaîne de confiance" qui peut finalement (et toujours) remonter jusqu'à l'autorité racine. À titre d'exemple, si l'autorité racine A fait confiance au nœud B, et que le nœud B fait ensuite confiance au nœud C, puis à une entité extérieure observant ces trois nœuds, il existe une chaîne de confiance de A à B à C, et on peut conclure que les deux nœuds B et C sont dignes de confiance si A est une autorité racine de confiance.

Le concept de PKI est issu des principes de la cryptographie asymétrique, selon lesquels l'autorité de base, les CA et tous les autres nœuds sont uniquement en possession de leur clé privée et ne communiquent avec les autres nœuds que par des messages qui peuvent être décryptés par leur clé publique. Pour élaborer, dans une PKI sécurisée, les clés privées ne peuvent pas être devinées, reproduites ou forcées de manière brutale en raison de leur longueur et de leur complexité, et prouvent ainsi, sans l'ombre d'un doute, que le porteur est bien celui qu'il prétend être. En raison des propriétés mathématiques de la cryptographie asymétrique (qui ne seront pas expliquées, car elles dépassent le cadre du présent document), tout certificat signé par la clé privée d'une CA racine ne peut être vérifié que par la clé publique correspondante et vice versa. En d'autres termes, si un message crypté peut être décrypté par une Clé Publique, la seule conclusion logique est que seul le porteur de la Clé Privée correspondante doit avoir écrit ou construit le message original.

L'objectif de la CA est de délivrer des certificats de clé publique, qui sont essentiellement des documents publics structurés et prédéfinis contenant des détails sur le porteur et, bien sûr, sur la clé publique elle-même. Deux types de certificats seront décrits dans ce livre blanc : les certificats X.509 et les certificats vérifiables par carte (CVC).

Enfin, l'autorité d'enregistrement (RA) est un nœud de la PKI qui vérifie et transmet les demandes de signature de certificats à une CA appropriée dans le réseau, mais est également responsable d'autres fonctions de gestion du cycle de vie des certificats elles-mêmes, telles que la révocation.

X.509 certificates

Un certificat X.509 contient des informations sur l'identité du porteur de ce certificat, la clé publique appartenant au porteur, et enfin, la CA qui l'a émit. Les premiers certificats X.509 ont été émis dans le cadre du secteur de la normalisation des télécommunications de l'Union internationale des télécommunications (ITU-T) et de la norme des services d'annuaire X.500. Plus tard, l'IETF a créé un profil Internet du certificat X.509 dans le groupe de travail PKIX.

D’un point de vue technique, le certificat X.509 est encodé selon les Règles d’Encodage Distinguées (DER). Il existe plusieurs formats et protocoles supplémentaires au sein du groupe de travail PKIX qui s’occupe de l’émission, de la révocation et de la gestion du cycle de vie des certificats X.509, mais ils ne seront pas abordés ici, car ils sortent du cadre du présent document. Des milliards de certificats X.509 émis sont utilisés aujourd’hui, principalement dans le cadre de la Sécurité de la couche transport (TLS) sur Internet, des signatures électroniques, du cryptage et de plusieurs autres protocoles de sécurité.

Certificats vérifiables par carte (CVC)

Les CVC sont des certificats numériques conçus pour être traités par des appareils à la puissance de calcul limitée. On les trouve souvent intégrés dans des appareils ou des objets tels que les cartes de crédit, les dispositifs intelligents IoT et les passeports électroniques, entre autres. Les informations vitales sont inscrites dans chaque CVC sous forme de “champs”, tels qu’un identifiant unique (ou numéro de série), des détails spécifiques au produit, des informations sur le fabricant et, comme nous le verrons en détail prochainement, des informations de sécurité ou de certification.

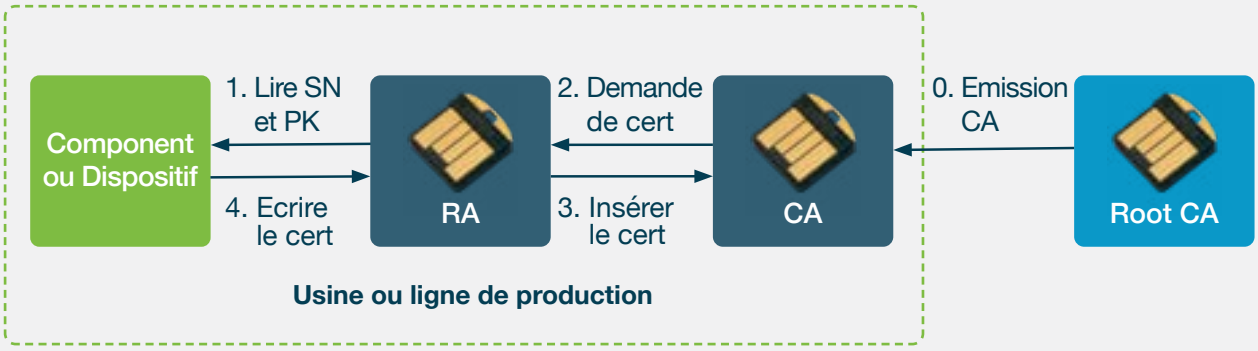
D’un point de vue technique, les CVC utilisent un concept connu sous le nom de codage Tag-Length-Value (TLV), ce qui signifie que chaque champ du certificat sous-jacent est de longueur fixe (les valeurs sont complétées pour atteindre la longueur maximale si elles sont insuffisantes) et que chaque champ vient dans un ordre prédéfini. Cela rend l’analyse des valeurs respectives extrêmement simple, contrairement à d’autres formes de saisie de variables qui peuvent nécessiter une plus grande puissance de traitement ou des registres de mémoire pour stocker temporairement les valeurs des champs avant d’être consommées.

YubiHSM 2, la PKI et le processus de délivrance

YubiHSM 2 peut être déployé pour protéger une PKI et son réseau de clés cryptographiques, à savoir celles qui appartiennent à la CA racine, et tous les CA et AR.

Le processus d’émission de certificats utilisant une PKI basée sur YubiHSM 2, en ce qui concerne une usine, une chaîne de production ou un autre site de déploiement, est illustré dans le schéma 5 (ci-dessous).

Schéma 5 – Aperçu de l’ICP (infrastructure de clé publique) et du processus de délivrance d’un certificat



Selon le modèle illustré, la CA racine utilise YubiHSM 2 pour héberger sa paire de clés (c'est-à-dire ses clés privées et publiques) en plus de son certificat. En raison de la taille réduite du YubiHSM 2, le serveur de la CA racine peut pratiquement être situé n'importe où, bien que traditionnellement, la CA racine soit hébergée dans un centre de données hors site pour des raisons de sécurité, mais aussi en raison de la taille physique énorme des fermes de serveurs et d'un HSM typique.

À l'usine ou à la chaîne de production, une CA peut être déployée et configurée avec un certificat signé par la CA racine. Ce processus est désigné comme l'étape 0 dans le schéma 5, et est généralement effectué hors ligne et une seule fois, lorsque l'usine est mise en ligne pour la première fois. La paire de clés et le certificat de la CA sont également protégés par YubiHSM 2, tout comme les références de la CA en aval.

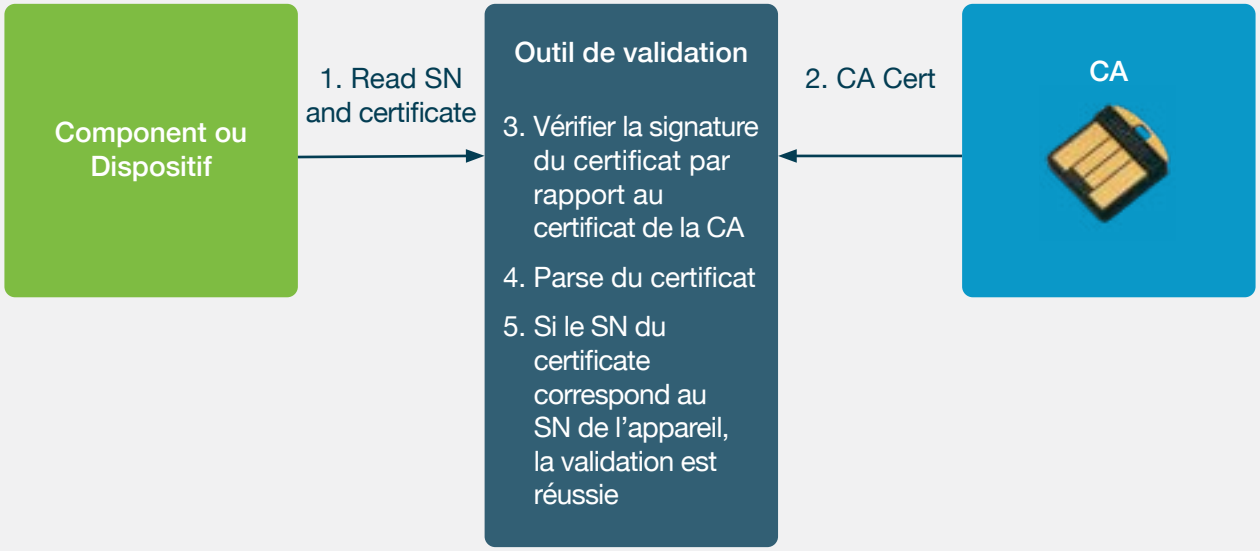
Le processus de délivrance des certificats suit le processus ci-dessous :

1. Le RA lit le numéro de série (SN) ou un identifiant unique équivalent du composant ou du dispositif en cours de fabrication. Si l'appareil comprend un TPM capable de générer une paire de clés asymétriques, la clé publique est également lue.
2. Le RA crée une demande de certificat (par exemple au format PKCS #10), qui comprend le numéro de série et la clé publique du composant ou dispositif en question. Notez que si aucune clé publique n'a jamais été lue auparavant, une clé publique est générée au nom du composant ou du dispositif. L'autorité de certification utilise également YubiHSM 2 pour signer la demande de certificat avec sa clé privée, et transmet le certificat à l'autorité de certification (en utilisant un protocole tel que CMC, CMP ou SCEP).
3. La CA signe le certificat avec sa clé privée. Le certificat délivré est renvoyé à la RA.
4. La RA écrit le certificat sur l'appareil, où il peut également être associé à la clé privée résidant dans le TPM, le cas échéant. Si l'appareil n'a pas de TPM, le certificat et la clé privée sont écrits sur le composant ou l'appareil.

Processus de validation d'un certificat pour en déterminer l'authenticité

Afin de vérifier l'authenticité d'un appareil fabriqué, le lecteur doit vérifier la validité du certificat d'un composant ou d'un appareil. Le principal processus de validation est illustré dans le schéma 6 (ci-dessous).

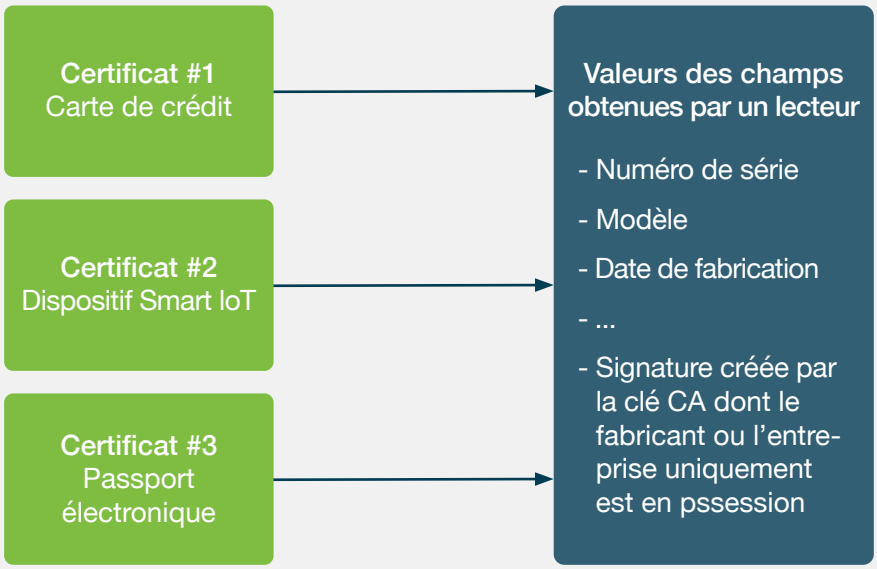
Schéma 6 – Processus de validation du certificat



Pour plus de clarté, les cinq étapes sont expliquées ci-dessous :

1. L'outil de validation lit le numéro de série (SN) et le certificat de l'appareil à valider
2. Sur la base des informations de la CA dans le certificat, le certificat de la CA correspondante est téléchargé directement de la CA émettrice. Notez que le certificat de la CA peut également, à son tour, être validé en utilisant la clé publique de la CA racine pour garantir l'authenticité de la CA. Dans certains cas, le certificat de la CA validée peut être mis en cache par l'outil de validation afin d'accélérer les futures validations.
3. La signature du certificat est validée en le décryptant avec la clé publique de la CA, et comparée à l'empreinte stockée dans le certificat
4. Le numéro de série (ou un identifiant unique similaire) est analysé à partir du certificat. Ce processus d'analyse s'applique à tous les différents appareils, comme l'illustre le schéma 7 (ci-dessous)
5. Si le numéro de série extrait du certificat valide correspond au numéro de série du composant ou de l'appareil, il doit donc être considéré comme authentique

Schéma 7 – Lecture des valeurs des champs des certificats et obtention de la preuve d'authenticité



Protocole de validation des Challenge-Réponse dynamiques

Si le composant ou le dispositif prend également en charge une TPM qui inclut la clé privée du dispositif, il est possible de concevoir un protocole dynamique de Challenge-Réponse similaire à l'exemple couvert dans le cas d'utilisation du JTAG. Dans ce cas précis, l'outil de validation génère un défi aléatoire signé par la clé privée de l'appareil. Le défi signé peut ensuite être vérifié par l'outil de validation à l'aide de la clé publique du certificat, avant que le reste du processus de validation du certificat décrit ci-dessus ne soit lancé.

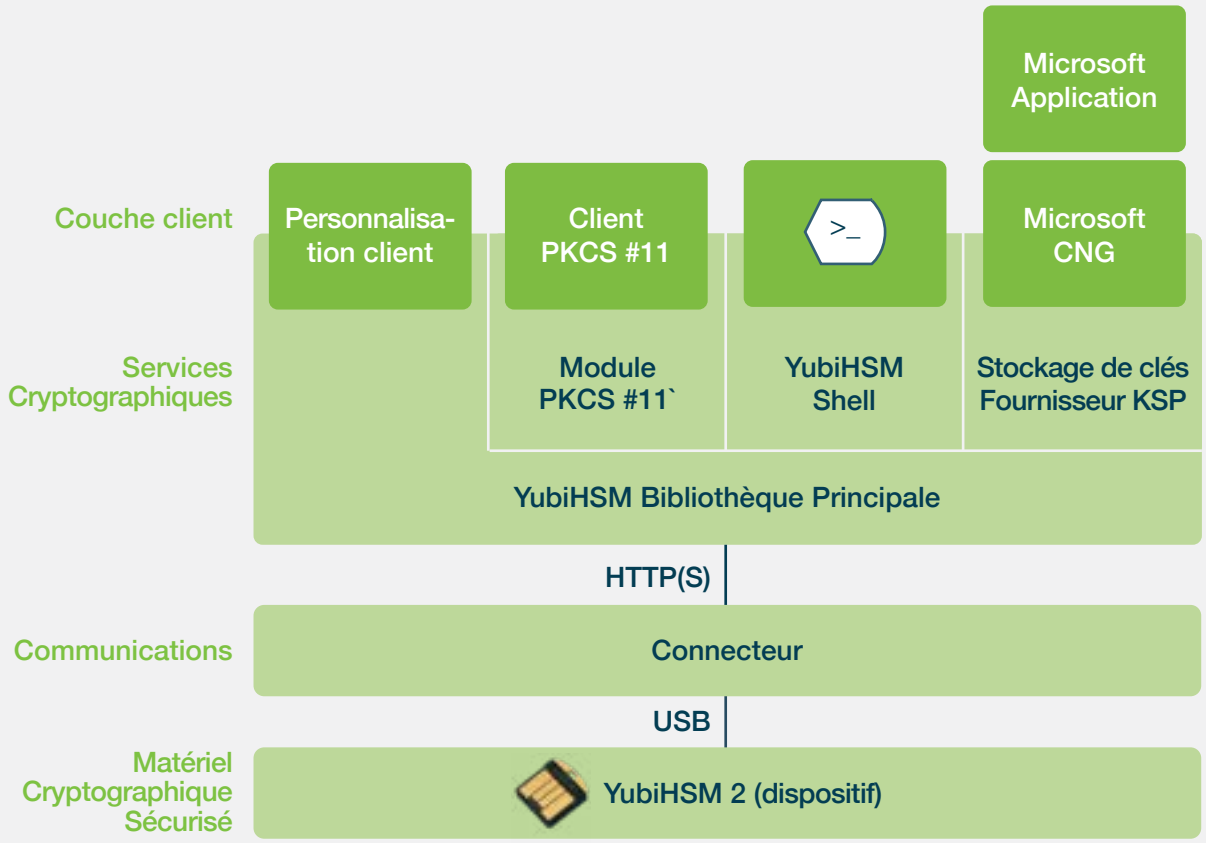
Pile de communication du matériel au logiciel

La pile de communication du YubiHSM 2 est représentée sur la schéma 8 (ci-dessous), et illustre comment la couche client est finalement connectée au dispositif physique. En bref (de bas en haut) :

1. Un dispositif YubiHSM 2 est physiquement inséré dans un port USB-A, qui à son tour est connecté à un ordinateur et/ou à un réseau
2. Le système d'exploitation résidant sur l'ordinateur et/ou le réseau connecté communique avec le périphérique par HTTP, via un binaire déployé fourni par Yubico appelé Connecteur (ou connecteur yubihsm), qui se comporte de la même manière qu'un pilote
3. La couche des services cryptographiques, en particulier la bibliothèque centrale YubiHSM (ou libyubihsm), fournit une API qui "traduit" les messages entre la couche client située au-dessus et le connecteur situé directement en dessous sur la pile

4. Enfin, la couche client comprend une variété de méthodes telles que le PKCS #11 (une norme cryptographique largement utilisée), une interface de ligne de commande fournie par Yubico connue sous le nom de YubiHSM Shell (ou yubihsm- shell), Microsoft Cryptographic API Next Generation (CNG) pour les applications spécifiques à Microsoft, plus un nombre quelconque de mises en œuvre personnalisées, pour contrôler et gérer la fonctionnalité cryptographique

Schéma 8 – Vue d'ensemble de la pile de communication YubiHSM 2



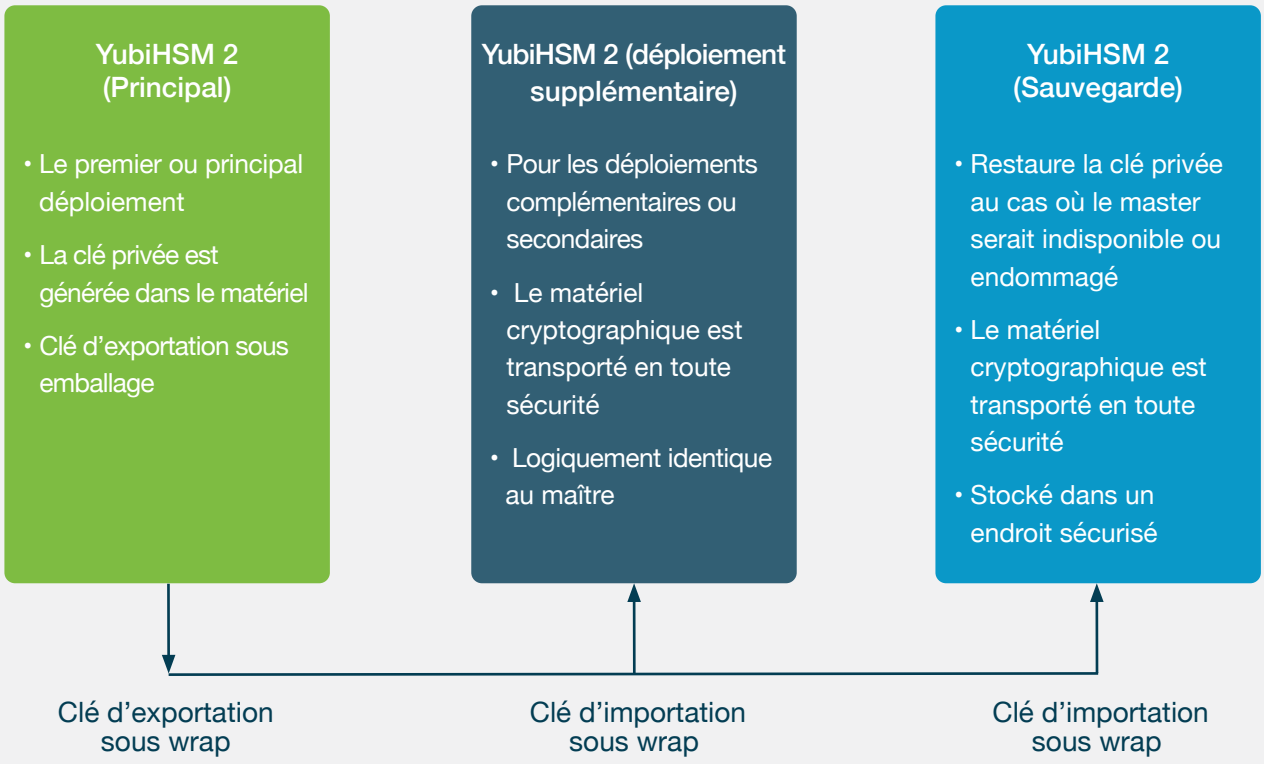
De plus amples informations sur YubiHSM Shell, le PKCS #11 et le Microsoft CNG sont disponibles dans la section à venir, intitulée “Intégration avec YubiHSM 2”.

Reproduire le contenu du YubiHSM 2

Toutes les clés privées YubiHSM 2 peuvent être dupliquées sur d'autres appareils YubiHSM 2 en exportant d'abord le matériel Wrap de l'appareil source, puis en l'important sur les appareils de destination. Sans trop entrer dans les détails techniques, le Key Wrapping est le processus de cryptage d'une clé à l'aide d'une autre clé, afin de la stocker de manière sécurisée ou de la transmettre sur un canal non fiable. Le canal non fiable dans le contexte du YubiHSM 2, pour être clair, est tout endroit en dehors de l'élément sécurisé.

De cette manière, un YubiHSM 2 contenant une clé privée peut être reproduit pour créer une clé d'authentification logiquement identique en toute sécurité, ce qui permet d'obtenir une solution rapide et évolutive sur plusieurs sites sans fuite de matériel. Le schéma 9 (ci-dessous) en donne un aperçu et résume les avantages que procure la réplication. Il convient de noter que le mot de passe d'authentification de chaque appareil ne doit pas nécessairement être le même, et peut (doit) être mis à jour en fonction des préférences de déploiement spécifiques.

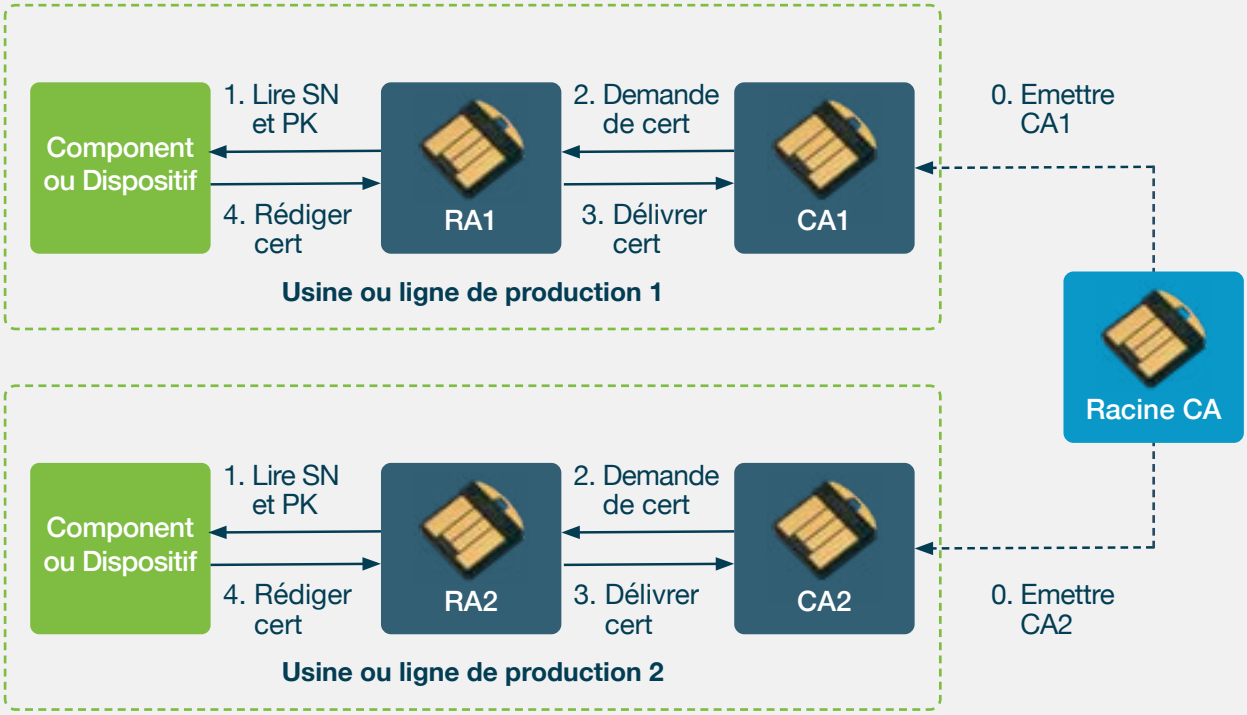
Schéma 9 – Vue d'ensemble du déploiement des appareils sur plusieurs sites



En reliant le concept de réplication au cas d'utilisation du JTAG couvert précédemment dans ce document, un dispositif YubiHSM 2 pourrait être déployé directement sur une chaîne de production pour écrire des numéros de série cryptés dans les ECU, tandis qu'un autre dispositif pourrait être livré à un client de confiance pour débloquer les JTAG correspondants pour les tests. Un autre dispositif pourrait encore être stocké dans un coffre-fort en guise de sauvegarde, au cas où le dispositif principal serait perdu, volé ou endommagé.

En faisant de même, mais en se concentrant sur le cas d'utilisation du certificat, plusieurs usines pourraient être mises en ligne en utilisant une réplique de la CA à la RA pour la configuration des composants/appareils. Le schéma 10 (ci-dessous) prolonge la conception originale de le schéma 5, en ajoutant des sites supplémentaires. Bien que seulement deux soient illustrés, il n'y a pas de plafond au nombre réel dans la pratique.

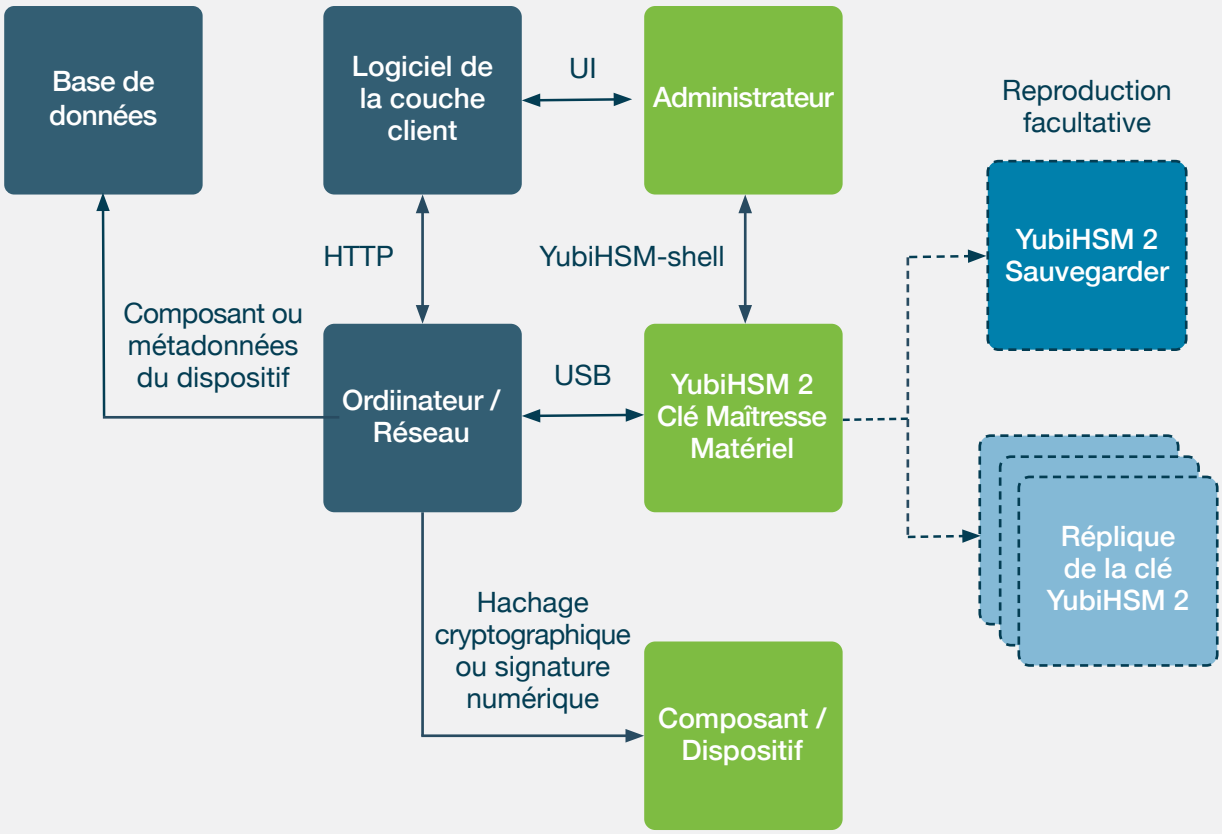
Schéma 10 – Extension de la procédure initiale de délivrance du certificat YubiHSM2



La proposition de design harmonisé YubiHSM2

En rassemblant toutes les informations jusqu'à ce point, depuis les cas d'utilisation, la pile de communication et l'option de réplication, il est possible de généraliser la manière dont YubiHSM 2 peut être installé (ou adapté) dans le processus de production d'un fabricant. Le schéma 11 (ci-dessous) illustre l'amalgame de tous les facteurs en une proposition de conception cohérente et unifiée.

Schéma 11 – Conception de la solution unifiée proposée



Bien que chaque plan de déploiement et chaque circonstance de fabrication soit unique et ne nécessite pas nécessairement tous les composants décrits dans le schéma 10, l'objectif global devrait toujours être de conserver toutes les clés privées en toute sécurité à l'intérieur de YubiHSM 2 afin de minimiser les menaces pour le sous-processus cryptographique. Les avantages de l'intégration d'une conception autour de YubiHSM 2 ont déjà été énoncés, et la conception peut facilement être adaptée en augmentant le nombre de dispositifs répliqués pour convenir aux fabricants, qu'ils soient grands ou petits, locaux ou dispersés.

Intégration avec YubiHSM 2

Il existe plusieurs API et outils différents disponibles pour l'intégration avec YubiHSM 2 :

- YubiHSM Shell
- PKCS #11
- Microsoft CNG
- Java libraries
- Python

YubiHSM Shell

YubiHSM Shell est un outil en ligne de commande qui peut être utilisé pour configurer et pour interagir avec les fonctions cryptographiques sous-jacentes. Le cas d'utilisation typique du YubiHSM Shell est la configuration initiale de YubiHSM 2 avec les clés maîtresses, mais aussi la réplication de ces clés vers plusieurs appareils YubiHSM 2.

YubiHSM Shell peut également être utilisé pour créer des clés JTAG, mais ces opérations nécessitent une interaction manuelle de la part d'un administrateur, et ne sont pas recommandées pour les déploiements à grande échelle. Pour une intégration programmatique avec YubiHSM 2, les API PKCS #11, Microsoft CNG, Python, ou les bibliothèques Java sont recommandées.

PKCS #11

Une solution plus simple à utiliser et plus avant-gardiste (par rapport au fonctionnement en ligne de commande de YubiHSM Shell), elle consiste à intégrer des applications au YubiHSM 2 en utilisant la couche client YubiHSM 2 PKCS #11 Library API. De telles applications ou outils peuvent alors présenter aux utilisateurs un menu de navigation ou une interface graphique qui peut effectuer des opérations YubiHSM 2. Une intégration programmatique avec l'API PKCS #11 permettra également un processus automatisé de création de clés ou de certificats JTAG. La bibliothèque PKCS #11 fournit des fonctions cryptographiques de bas niveau, qui sont utiles lors de l'énumération et en sélectionnant les clés qui ont été générées dans le YubiHSM 2.

Microsoft CNG

Si l'environnement en question fonctionne sous Microsoft Windows, l'API de cryptographie Microsoft de nouvelle génération (CNG) est une option viable. YubiHSM 2 Key Storage Provider (KSP) peut être branché sur la couche GNC et accessible depuis n'importe quelle application développée sur le GNC de Microsoft. Le GNC Microsoft offre plus de fonctionnalités en termes de codage des messages et des certificats cryptographiques, et pourrait être une alternative pour l'émission de certificats. Le PKCS #11 est cependant le choix recommandé pour la signature HMAC des clés JTAG.

Java libraries

Pour un environnement Java, il y a deux options lors de l'intégration avec YubiHSM 2 : soit le fournisseur natif JCE PKCS #11, soit la bibliothèque Java YubiHSM 2.

Le fournisseur JCE PKCS #11 est mis en œuvre et soutenu par Oracle (anciennement Sun) pour l'architecture cryptographique Java (JCA). Le fournisseur JCE charge la bibliothèque PKCS #11 via une interface native Java, et est connecté au cadre de la JCA. Grâce à cette architecture, les développeurs peuvent mettre en œuvre des applications Java en plus de la JCA, et avoir accès aux clés et fonctions cryptographiques de YubiHSM 2.

Comme alternative à l'architecture JCE/JCA, Yubico fournit la bibliothèque Java YubiHSM 2, qui se connecte via HTTP(S) au connecteur YubiHSM. La bibliothèque Java YubiHSM 2 est disponible en tant que projet GitHub, et n'est pas encore officiellement publiée en tant que SDK Yubico.

Python

La dernière alternative est de s'intégrer avec YubiHSM 2 en utilisant la bibliothèque Python YubiHSM. La bibliothèque Python YubiHSM permet aux développeurs de se connecter avec YubiHSM 2 via le service Connector en utilisant le langage de programmation Python. Naturellement, la bibliothèque Python YubiHSM est une option viable uniquement pour le développement dans Python.

yubico

À propos de Yubico

Yubico établit de nouvelles normes mondiales pour un accès simple et sécurisé aux ordinateurs, serveurs, et comptes Internet.

Son invention principale, la YubiKey, offre une protection hardware puissante. D'un simple toucher, il est possible de s'authentifier de façon sécurisée sur un nombre illimité de systèmes informatiques et de services en ligne. Le YubiHSM, le module de sécurité matérielle de Yubico, protège les données sensibles des serveurs.

Yubico est l'un des principaux contributeurs des normes d'authentification ouvertes FIDO2, WebAuthn, et FIDO Universal 2nd Factor. Les YubiKeys sont déployées et plébiscitées par 9 des 10 principales sociétés technologiques et des millions d'utilisateurs dans plus de 160 pays.

Fondée en 2007, Yubico est une société privée, avec des bureaux en Australie, en Allemagne, à Singapour, en Suède, au Royaume-Uni et aux États-Unis.