La Technologie
Carte à Puce EAP-TLS
v2.0

Une sécurité forte,
pour les services basés sur des infrastructures PKI,
tels que *applications WEB*, *VPNs*, *Accès Réseaux*

Pascal Urien
Avril 2009

Architectures à clés publiques et protocole SSL

Les infrastructures à clés publiques constituent les fondations de la confiance pour les services offerts par la toile d'araignée mondiale. Le principe d'une PKI (Infrastructure à clé publique) est simple, une autorité de certification (CA) qui s'est auto-délivrée un certificat (le certificat racine ou *Root Certificate*) attribue des certificats X509 aux sites WEB ou à des serveurs d'authentification; optionnellement le client est également équipé d'un certificat qui permet son authentification forte à l'aide de sa clé privée RSA.

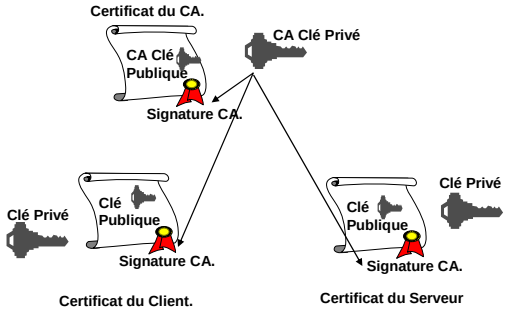

Figure 1. Principe d'une infrastructure PKI

Conçu en 1996 par la société Netscape, le protocole SSL (standardisé par l'IETF sous le sigle TLS) repose essentiellement sur une infrastructure PKI, il garantit la sécurité des informations échangées sur Internet, et en particulier la sécurité du commerce électronique grâce au chiffrement des numéros de cartes bancaire.

Le déroulement typique d'une session SSL comporte quatre étapes.

Le client délivre un nombre aléatoire (*client-random*) et propose une suite d'algorithmes cryptographiques de chiffrement et d'intégrité. Le serveur sélectionne un couple de fonctions cryptographiques (le *Cipher-Suite*) et produit un nombre aléatoire (*server-random*); il transmet au client son certificat. Le client authentifie le serveur en vérifiant la signature de son certificat; il génère une donnée secrète (le *pre-master-secret*), puis la transmet chiffrée avec la clé publique extraite du certificat serveur. De manière optionnelle le client prouve son identité en divulguant son certificat et en produisant une signature avec sa clé RSA privée. A partir des deux nombres aléatoires (*client-random* et *server-random*) et du *pre-master-secret*, une clé maître, le *master-secret* est calculée. Cette dernière valeur, associée aux deux nombres aléatoires permet d'éditer quatre clés éphémères de chiffrement et d'intégrité (le *Keys-Block*) mises en œuvre pour le transport sécurisé des informations telles que une requête HTTP et la réponse correspondante. La connaissance de ces quatre clés est prouvée dans les 3^{ième} et 4^{ième} messages établissant également l'intégrité du dialogue entre client et serveur.

Cet échange à quatre passes, nommé *full mode*, est dans la pratique peu fréquemment utilisé en raison du coût de traitement qu'il implique (charge importante de calculs RSA côté serveur). Un mode allégé, ou *resume mode*, tire profit d'un *master-secret* précédemment établi; ce protocole à trois passes échange deux nombres aléatoires (*client-random* et *server-random*), négocie les algorithmes de chiffrement et d'intégrité (*Cipher-Suite*), et génère un

nouveau bloc de clés (*Keys-Block*) à partir des paramètres *master-secret*, *client-random*, et *serveur-random*.

Les attaques du protocole SSL

De multiples travaux basés sur des techniques mathématiques formelles ont démontré la solidité théorique du protocole SSL. Cependant son déploiement dans des systèmes d'information réels, ouvre la porte à plusieurs types d'attaques. En voici quelques illustrations.

Lors d'une authentification par PKI il y a deux opérations critiques: la vérification du certificat du serveur, et la génération optionnelle d'une signature par le client à l'aide de sa clé privée. Il est important de souligner que ces opérations sont réalisées par des systèmes d'exploitation dont les vulnérabilités aux chevaux de Troie et autres logiciels malveillants sont bien connues. En d'autres termes le risque de vol d'identité (usage illicite d'une clé privé) est important.

Le mode full, basé sur la cryptographie asymétrique est rarement mise en œuvre, par exemple une fois toute les dix minutes. Cet intervalle de temps est l'un des multiples paramètres de configuration d'un serveur WEB. La confiance dans le mode *resume* repose sur la capacité du terminal à protéger efficacement le master-secret.

Le terminal doit être configuré avec le certificat racine (CA) et optionnellement le certificat client. De fait le terminal vérifie l'identité du serveur et authentifie le client. Si l'internaute dispose de plusieurs terminaux il faudra installer des certificats sur chacun d'entre eux. La mobilité de l'utilisateur n'est pas gérée avec souplesse.

Les certificats clients sont généralement émis au format PKCS12, pour lequel la clé privée est protégée à l'aide d'un mot de passe. La connaissance de cette information permet d'extraire cette clé, qui est la pierre angulaire de la confiance dans une infrastructure PKI. Parfois le propriétaire du terminal installe lui même son certificat, il existe donc un risque de duplication de la clé privée, involontaire ou maligne. Dans ce mode de transport du certificat, une autre menace provient du fait que la clé secrète peut être dérobée à l'insu de son propriétaire.

Phising et mots de passe

Le ***phising*** est une technique d'attaque qui consiste à attirer l'internaute vers un leurre, qui reproduit précisément l'apparence d'un site WEB connu, offrant des services tels que la gestion de compte bancaire en ligne. Le pirate espère grâce à cette réplique fidèle capturer des informations critiques par exemple login et mot de passe. Le protocole SSL ne protège pas l'internaute de ces attaques, car fréquemment les terminaux ne sont pas équipés avec les certificats des CA, ce qui oblige l'internaute à accepter des connexions avec des sites dont l'identité est incertaine. La racine du mal est l'usage quasi imposé de mot de passe notifié à un site WEB dont il est difficile de prouver l'authenticité.

Comment supprimer les mots de passe

Le protocole SSL, le «s» de l'acronyme HTTPS, associé fréquemment à une icône représentant un cadenas, possède une option d'authentification forte du client à l'aide d'un certificat et d'une clé RSA privée. L'authentification est mutuelle, le client analyse la signature du certificat serveur, et en cas de succès transmet son certificat et prouve son identité grâce à l'usage de sa clé privée.
Le mode *resume* permet de rafraichir à chaque session HTTPS les clés cryptographiques assurant le chiffrement des messages HTTP, le client est authentifié à chaque nouvelle session en prouvant sa connaissance du *master-secret*. Ce paradigme est parfaitement illustré par les variables d'environnements présentes dans les serveurs WEB qui attestent de manière indifférenciée l'existence d'une mutuelle authentification aussi bien en mode «full» que «resume».

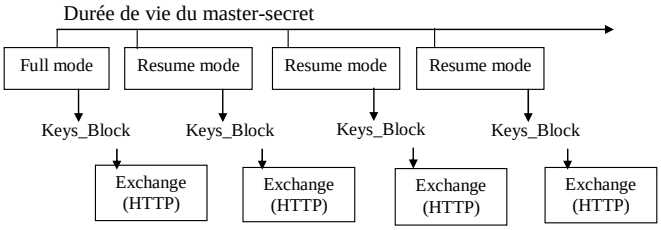

Figure 2. Mode Full et Resume dans le protocole SSL.

Au sujet des cartes à puce

La carte à puce est une pastille de silicium, d'environ 25mm^2 qui comporte un CPU de 8, 16 ou 32 bits, de la mémoire non volatile de 0,1 à 1 Mo, et de la mémoire RAM (quelques kilo-octets). Ce système informatique est qualifié de «tamper resistant», c'est-à-dire qu'il est protégé contre les menaces d'intrusion à l'aide de contremesures physiques et logiques. En particulier la lecture de données secrètes est impossible et les bibliothèques cryptographiques détectent les attaques connues.

En 2009 3,5 milliards de cartes à puce ont été fabriquées, dont plus d'un milliard de Java Card, c'est-à-dire des composants munis d'une machine virtuelle java (JVM) et en conséquence capables d'exécuter des programmes écrits en langage JAVA.

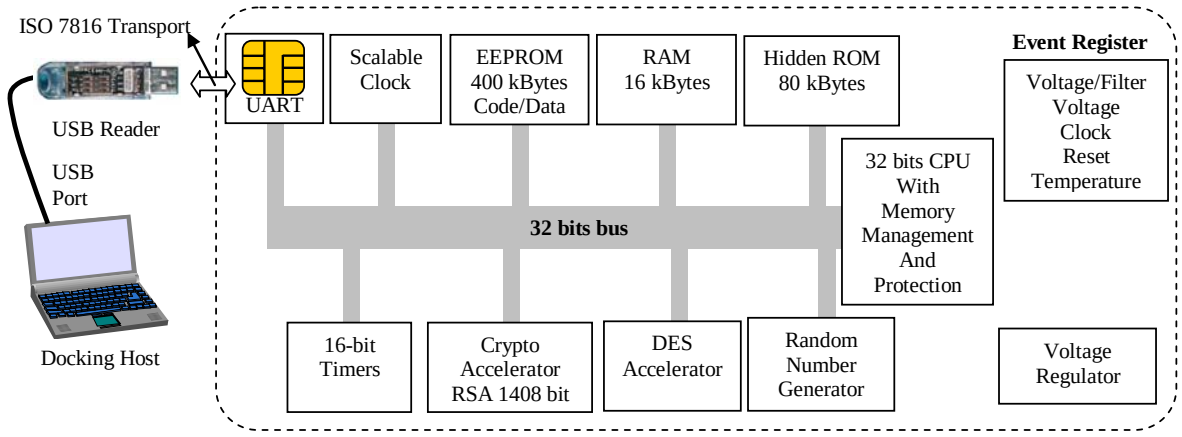

Figure 3. Exemple d'architecture d'une carte à puce

La carte EAP-TLS

Le protocole EAP-TLS est un standard ouvert, proposé en 1999 par la société Microsoft, et normalisé par *l'Internet Engineering Task Force* (IETF). EAP (Extensible Authentification Protocol) est une architecture d'authentification destinée aux contrôles d'accès réseaux mais également aux authentifications fortes pour les VPNs (*Virtual Private Network*). L'intérêt majeur de l'approche EAP-TLS est l'encapsulation transparente de messages TLS dans des paquets EAP; traditionnellement SSL est transporté par TCP, ce qui implique un décor logiciel reposant sur des sockets TCP/IP. Dans EAP-TLS les paquets SSL sont échangés grâce à des datagrammes, possédant un entête de seulement quelques octets. Ce mode de fonctionnement est parfaitement compatible aves les environnements logiciels des cartes à puce.

Une proposition de standard IETF (*draft-urien-eap-smartcard*) décrit précisément le codage binaire de l'interface d'une carte à puce EAP-TLS, conformément au standard ISO7816.

Le concept *carte à puce EAP-TLS* est simple, robuste, et très sure. Les paquets SSL ne sont pas traités par le terminal du client, ce dernier se comporte comme un relais de communication passif entre serveur SSL et carte à puce. La carte traite intégralement les messages SSL, et stocke toutes les informations nécessaires à cette tâche telles que certificat racine (CA), certificat client et clé privée.

Une implémentation ouverte de carte EAP, nommée *OpenEapSmartcard* dédiée à des environnements Java Card, a été présentée lors de l'évènement JAVAONE 2007 à San Francisco.

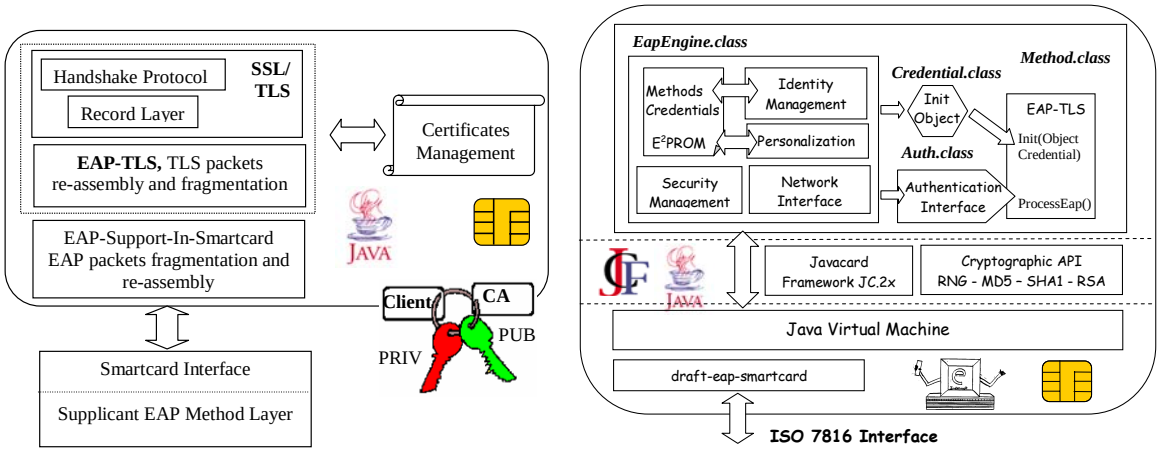

Figure 4. Carte EAP-TLS (à gauche) et implémentation ouverte JAVACARD (à droite)

Les avantages de la technologie carte EAP-TLS.

On peut souligner trois bénéfices majeurs induits par l'usage de carte EAP-TLS : la protection contre le vol d'identité, la mobilité facilitée du client, et le confinement de la sécurité.

La protection contre le vol d'identité. La clé privée RSA clé est stockée dans la carte à puce, sa valeur n'est pas connue de son utilisateur, son clonage est impossible. Son usage n'est pas libre, car la génération d'une signature implique l'authentification préalable du serveur. La carte est protégée par un code PIN, qui garantit sa protection en cas de perte ou vol, à l'aide d'un mécanisme de blocage après trois essais infructueux.

La mobilité du client. Le terminal n'est pas configuré avec des certificats relatifs à son utilisateur. Il dispose d'un composant logiciel qui relaye de manière passive les messages SSL depuis/vers la carte à puce. C'est le client d'un service qui est identifié et non l'ordinateur personnel qu'il utilise, la mobilité est donc facilitée.

Le confinement de la sécurité. Le protocole SSL (mode *full* et *resume*) est intégralement exécuté dans la carte à puce, toute modification des messages émis ou reçus est détectée par cette dernière (et le serveur d'authentification) et provoque l'échec de la procédure d'authentification. On obtient ainsi une protection efficace contre les attaques des chevaux de Troie et autres logiciels malveillants.

La technologie TANDEM

La technologie TANDEM ou pile duale SSL est une extension du concept de carte EAP-TLS. Un ensemble de mesures conduites sur un parc hétérogène de cartes à puce a montré que le chiffrement et les calculs d'intégrité réalisés par ces dispositifs limitent les performances de traitement de données à quelques kilo-octets par seconde, en raison des contre mesures déployées par les systèmes d'exploitation, et les capacités modestes des processeurs embarqués.

Bien que l'initialisation des sessions SSL en mode *full* ou *resume* soit compatible avec les ressources de calculs des cartes actuelles (avec des temps de réponse observés inférieurs à 10 secondes) le transfert de fichiers de grandes tailles (multimédia en particulier) n'est pas envisageable en l'état actuel de l'art.

La technologie Tandem divise une session SSL en deux phases. La première (*Phase I*) réalise le protocole SSL dans la carte en quatre passes (mode *full*) ou trois passes (mode *resume*). A la fin de cette phase les algorithmes cryptographiques sélectionnés (pointés par le paramètre Cipher-Suite) et les quatre clés associées (Keys-Block) sont transférés au terminal qui gère par la suite la *Phase II*, c'est-à-dire les opérations de chiffrement, de déchiffrement et l'intégrité de l'information échangée. Par exemple dans le cas du protocole HTTPS cela signifie que l'ouverture de la session SSL est assurée par la carte EAP-TLS, mais le chiffrement/déchiffrement des paquets HTTP est réalisé par le terminal.

Ce paradigme permet d'insérer les cartes EAP-TLS dans de nombreuses applications WEB, la carte joue le rôle d'une ***pile SSL amovible***, le navigateur télécharge des fichiers au terme de la phase d'authentification forte.

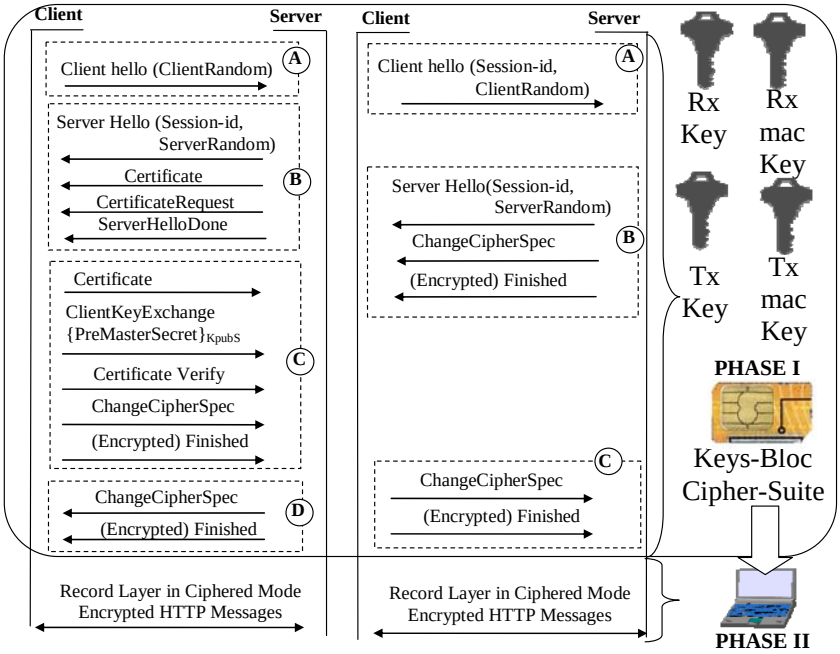

Figure 4. La technologie TANDEM ou pile duale SSL

Les Applications des cartes EAP-TLS

On peut diviser les applications des cartes EAP-TLS en deux grandes catégories les applications de ***contrôle d'accès*** et les ***applications WEB***.

Les applications de contrôle d'accès adressent les réseaux câblés et sans fil ainsi que l'authentification forte pour les technologies VPN. On peut citer à titre d'exemple

- Des clés d'accès aux réseaux compatibles IEEE 802.1x
- Des clés d'accès Wi-Fi
- Des clés d'accès WiMAX
- Des clés d'authentification pour les VPNs tels que IPSEC, PPTP, L2TP
- Des clés pour serveur d'authentification RADIUS

Les applications WEB comprennent l'authentification forte, le téléchargement sécurisé, et le Single Sign On (SSO).

- L'authentification par la technologie TANDEM supprime les mots de passe en les remplaçant par une session SSL avec mutuelle authentification.
- Le téléchargement sécurisé réalise le transfert de fichiers à l'aide de clés produites par une session SSL avec mutuelle authentification.
- Dans les infrastructures SSO, telles que OpenID, la carte EAP-TLS authentifie le client à l'aide d'une session SSL avec mutuelle authentification et assure également le transfert de données sécurisées entre l'internaute et son OpenID Provider (OP).