C

TLS

Objectif

Cette annexe présente les notions de cryptage asymétrique, de certificats et de signatures électroniques, et décrit brièvement les protocoles SSL (*Secure Sockets Layer*) et TLS (*Transport Layer Security*). Ces derniers sont extrêmement utilisés pour établir des « tunnels » sécurisés *via* l'Internet.

C.1 CRYPTAGE SYMÉTRIQUE

Pour protéger un document confidentiel, et s'assurer qu'une personne non autorisée n'y aura pas accès, il existe principalement deux approches (complémentaires) :

- la « stéganographie », c'est-à-dire le fait de dissimuler le document[1] ;
- le « cryptage » (ou « chiffrement ») qui consiste à modifier le message pour le rendre incompréhensible, à part bien sûr pour ses destinataires légitimes.

Le premier type de cryptage à avoir été inventé, bien avant notre ère, est le cryptage « symétrique » : l'émetteur et le récepteur commencent par se mettre d'accord sur une méthode de cryptage. Par exemple, ils peuvent décider de remplacer telle lettre par telle autre : A par F, B par M, C par Z, etc. Ensuite, ils

1. Un exemple célèbre remonte à la seconde guerre mondiale : des documents confidentiels étaient parfois inscrits sur des microfilms de la taille d'un point. Ces points étaient ensuite collés à la place de quelques signes de ponctuation dans des lettres tout à fait anodines. Si l'ennemi parvenait à intercepter une lettre, il ne pouvait pas deviner que des informations confidentielles y étaient dissimulées !

doivent s'échanger la « clé » de cryptage. Si l'algorithme est une simple permutation de lettres, la clé est le tableau de correspondance entre les lettres non cryptées et les lettres cryptées. Au chapitre 7, nous avons vu un exemple de cryptage symétrique plus moderne avec le cryptage RC4.

On parle de cryptage « symétrique » car l'émetteur et le récepteur doivent connaître une même clé, servant à la fois aux opérations de cryptage et de décryptage. Le cryptage symétrique est généralement simple, rapide, et il peut être très sûr… pourvu qu'un espion ne puisse pas découvrir la clé secrète ! Or, avant de pouvoir communiquer ensemble, l'émetteur et le récepteur doivent se mettre d'accord sur cette clé secrète. Une solution consiste à se l'échanger en main propre, mais bien sûr ce n'est pas toujours possible. Cet échange initial de la clé de cryptage est le principal point faible du cryptage asymétrique.

Avant la fin du XX^{e} siècle, il était inconcevable qu'il existe un autre type de cryptage… jusqu'à ce que le cryptage asymétrique soit inventé.

C.2 CRYPTAGE ASYMÉTRIQUE

C.2.1 Clé privée, clé publique

Le cryptage asymétrique fait intervenir non pas une clé secrète, mais deux : l'une pour le cryptage, et l'autre pour le décryptage. Chaque personne possède un jeu de deux clés : il conserve secrètement la clé de décryptage, que l'on appelle donc la clé « privée », mais il divulgue librement la clé de cryptage, que l'on appelle la clé « publique ».

Si Alain veut envoyer un message à Bernard, il commence par lui demander sa clé publique. Grâce à cette clé publique, Alain crypte son message, et envoie le résultat à Bernard. Puisque Bernard est le seul à posséder la clé privée, il sera le seul à pouvoir décrypter le message. L'avantage par rapport au cryptage symétrique est que la clé de décryptage n'est jamais échangée.

La magie du cryptage asymétrique permet à deux personnes de communiquer ensemble de façon absolument confidentielle, même si un espion peut écouter toutes leurs communications, du premier au dernier message.

C.2.2 Mathématiques asymétriques

Le cryptage asymétrique repose sur des opérations mathématiques possédant elles-mêmes des propriétés asymétriques. L'opération la plus utilisée pour le cryptage asymétrique est le produit de deux nombres premiers[1]. Si l'on vous

1. Un nombre premier est un entier positif divisible uniquement par 1 et par lui-même. La série des nombres premiers est infinie et commence par 2, 3, 5, 7, 11, 13… (par convention, 1 n'est pas premier).

demande de calculer le produit de deux nombres premiers, par exemple `59 × 23 = ?`, cela vous prendra généralement un temps assez court, plus ou moins proportionnel au nombre de chiffres qui composent les nombres premiers (en l'occurrence, deux chiffres pour chaque nombre premier). En revanche, si l'on vous donne le produit de deux nombres premiers quelconques, par exemple `p × q = 4 819`, et que l'on vous demande de retrouver quels sont les deux nombres premiers en question (`p` et `q`), cela vous prendra un temps à peu près proportionnel aux nombres premiers eux-mêmes (et non aux nombres de chiffres qui les composent). Bref, ce sera beaucoup plus long. La différence sera d'autant plus importante que les nombres premiers seront grands. S'ils sont énormes, par exemple composés de 128 bits, c'est-à-dire de l'ordre d'une quarantaine de chiffres, la multiplication sera toujours assez rapide, mais la décomposition de leur produit sera pratiquement impossible à réaliser en un temps raisonnable : même avec les meilleurs ordinateurs, la durée de vie de l'univers serait largement insuffisante pour les retrouver !

C.2.3 L'algorithme RSA

L'algorithme RSA, dont le nom est composé des initiales de ses inventeurs Ron Rivest, Adi Shamir et Len Adleman, a vu le jour en 1977. C'est le premier cryptage asymétrique à avoir été inventé[1]. Il repose sur l'asymétrie mathématique que nous venons de décrire : pour crypter un message, le produit de deux énormes nombres premiers est utilisé ; pour le décrypter, les nombres premiers eux-mêmes doivent être connus. La « clé publique » est le produit des nombres premiers et la « clé privée » est le couple des nombres premiers. Même si un espion parvient à obtenir la clé publique, il lui sera impossible d'en déduire la clé privée.

C.2.4 Les problèmes du cryptage asymétrique

Le cryptage asymétrique possède deux problèmes importants :

- Il est gourmand en puissance de calcul, et il est donc assez lent en comparaison au cryptage symétrique.
- Si un pirate parvient à s'insérer entre l'émetteur et le récepteur (c'est une attaque de type *Man in the Middle*, notée MiM), alors il peut tenter la supercherie suivante : lorsque l'émetteur demande sa clé publique au récepteur, le pirate renvoie sa propre clé publique. L'émetteur crypte donc son message avec la clé du pirate et envoie le résultat. Il suffit ensuite au

1. Un mathématicien anglais du nom de Clifford Cocks avait inventé un cryptage semblable quatre ans auparavant, mais il travaillait pour les services secrets, et il n'eut pas le droit de divulguer son invention.

pirate de décrypter le message grâce à sa clé privée ! S'il est malin, il peut même crypter ensuite le message (en le modifiant, s'il le souhaite) avec la clé publique du récepteur, et le lui envoyer. De cette façon, l'émetteur et le récepteur ne se rendent compte de rien, et le pirate peut allégrement décrypter et modifier tous les messages envoyés.

C.2.5 Les solutions

Puisque le cryptage asymétrique est assez lent, il est généralement utilisé uniquement au début d'une communication, dans le but d'échanger secrètement une clé de cryptage symétrique. Après ce premier échange, le reste de la communication repose sur ce cryptage symétrique (rapide).

Pour résoudre le deuxième problème, l'émetteur doit s'assurer que la clé publique qui lui a été fournie est bien celle du récepteur (et non d'un éventuel pirate). Pour cela, une solution consiste à se l'échanger en main propre : même si un espion assiste à l'échange, cela n'a pas d'importance, car l'essentiel est que l'émetteur soit assuré de la provenance de la clé publique. Toutefois, un échange en main propre n'est généralement pas réalisable. Une autre technique consiste à n'accepter une clé publique que si elle est signée électroniquement :

- soit par le récepteur, mais il faut alors connaître sa signature électronique, ce qui ne fait que repousser le problème (comment échanger la signature ?) ;
- soit par un tiers de confiance dont on connaît la signature électronique, et qui aura lui-même vérifié et signé la clé publique du récepteur.

C.3 SIGNATURES ÉLECTRONIQUES

Pour signer électroniquement un document, on utilise à nouveau le principe des clés asymétriques, mais la clé privée sert à crypter le document, et la clé publique sert à le décrypter. Si l'on possède la clé publique d'Alain, par exemple, et que l'on reçoit un message crypté, on peut vérifier si ce message peut être décrypté avec la clé publique d'Alain : si c'est le cas, on est assuré que le message a bien été crypté avec la clé privée d'Alain, et donc qu'il provient de lui : le message est donc « signé ».

Grâce à ce mécanisme, on peut vérifier si un document a bien été signé par une personne dont on connaît la clé publique (c'est « l'authenticité »), et que ce document n'a pas été modifié depuis (c'est « l'intégrité »). Toutefois, la signature n'assure par la confidentialité car n'importe qui peut décrypter le message avec la clé publique d'Alain.

Puisque les calculs peuvent être assez lourds, le document à signer est généralement d'abord « condensé », c'est-à-dire passé au travers d'une fonction de *hash* : il s'agit d'une fonction mathématique qui produit un nombre (généralement de 128 bits) à partir d'un document électronique. Deux documents identiques donnent toujours le même hash, mais deux documents différents (même d'un seul bit) donnent deux hash distincts, sans rapport entre eux. Il est presque impossible de trouver deux messages distincts qui produisent le même hash.

Voici comment un message est signé électroniquement : l'émetteur commence par calculer le hash de son document. Ensuite, il utilise sa clé privée pour crypter (c'est-à-dire signer) ce hash. Enfin, il envoie le message, accompagné du hash crypté. Le récepteur peut calculer le hash du document qu'il a reçu, et utiliser la clé publique de l'émetteur pour décrypter le hash crypté. Si les deux hash diffèrent, il sait que le document a été modifié pendant son transport ou que la signature n'est pas celle de l'émetteur : le message doit donc être rejeté.

Bien sûr, un document signé, c'est-à-dire un document accompagné de son hash crypté, peut lui-même être crypté avec la clé publique du récepteur (ou à l'aide d'un cryptage symétrique) afin d'en assurer la confidentialité.

C.4 CERTIFICATS ÉLECTRONIQUES

C.4.1 Des informations signées

Un certificat électronique est un document signé électroniquement, contenant des informations au sujet de quelqu'un ou de quelque chose (comme une société ou encore une machine) : par exemple, son nom, son adresse, sa fonction, ses coordonnées, et surtout… sa clé publique (et parfois la clé privée, protégée par un mot de passe).

Lorsque quelqu'un vous fournit son certificat électronique, celui-ci peut être signé par n'importe qui. S'il est signé par quelqu'un que vous ne connaissez pas, ou en qui vous n'avez pas confiance, le certificat n'a aucune valeur pour vous. En revanche, s'il est signé par quelqu'un en qui vous avez décidé de faire confiance, vous pouvez désormais admettre que la personne dont le nom est précisé dans le certificat possède bien la clé publique qui y est également indiquée. Il vous est alors possible de lui envoyer un message crypté, en toute sécurité.

C.4.2 Le tiers de confiance

Certaines sociétés, par exemple Verisign ou Thawte, sont spécialisées dans la confiance électronique et jouent le rôle d'autorité de certification (*Certification Authority*, CA) : leur rôle est de signer des certificats électroniques, après avoir

validé l'identité du demandeur. Admettons par exemple qu'une banque souhaite mettre en place un site Web sécurisé, afin que ses clients puissent consulter leurs comptes sur Internet. La banque commence par générer un certificat (pas encore signé) indiquant le nom de domaine du site Web à protéger, par exemple « `www.banque.fr` ». Ensuite, elle envoie ce certificat à une autorité de certification, par exemple Verisign. Ce dernier vérifie alors l'identité de la banque, afin de s'assurer qu'elle existe bien, qu'elle se situe à l'adresse indiquée dans le certificat, et qu'elle possède bien le nom de domaine. Enfin (après paiement), Verisign signe le certificat, avec sa propre signature, et le donne à la banque. Par la suite, lorsqu'un client se connecte au site Web de la banque, le certificat signé lui est envoyé : puisque la plupart des navigateurs Internet connaissent la signature des principales autorités de certification, la vérification de la signature peut être réalisée automatiquement, sans intervention du client.

C.5 LES IGC

Il arrive fréquemment qu'une société souhaite signer ses propres certificats pour sa sécurité interne (généralement pas pour les serveurs Web publics). Le but est ne pas avoir à dépendre d'une autorité de certification externe et à payer ses services. Dans ce cas, la société doit jouer elle-même le rôle d'autorité de certification : elle doit créer un certificat « racine », dont l'unique but est de signer d'autres certificats. Ensuite, elle doit déployer le certificat racine sur le poste de chaque employé, dans la catégorie des « autorités de certification » : ainsi, tous les certificats signés par le certificat racine seront automatiquement acceptés par les employés.

Les certificats ainsi signés peuvent servir à sécuriser l'accès à des serveurs internes à l'entreprise : par exemple, un serveur Web donnant accès à l'intranet, ou encore un serveur RADIUS (voir le chapitre 10), s'il met en œuvre une méthode d'authentification EAP/TLS, TTLS ou encore PEAP (voir le chapitre 8). Mais on peut aller plus loin : chaque employé peut avoir son propre certificat. Dans ce cas, les employés peuvent s'échanger des messages signés, ou même cryptés, et réaliser toutes sortes d'opérations sécurisées. Pour faciliter la maintenance de toutes ces paires de clés publiques/privées et des certificats associés, il est généralement nécessaire d'utiliser des logiciels prévus à cet effet. Ces logiciels permettent de facilement créer de nouveaux certificats, renouveler des certificats expirés, révoquer des certificats qui ont été compromis (si la clé privée a été volée ou perdue), déployer les certificats sur les postes des utilisateurs, etc. On parle alors d'Infrastructure à Gestion de Clé (IGC) ou *Public Key Infrastructure* (PKI).

C.6 LE PROTOCOLE SSL

C.6.1 Le commerce électronique

Le protocole *Secure Socket Layer* (SSL) a été conçu par la société Netscape Communications Corporation pour son navigateur Internet : il a pour but de permettre l'établissement d'un « tunnel » sécurisé entre l'utilisateur et le serveur Web afin de sécuriser leurs échanges. Ce protocole est aujourd'hui le plus utilisé sur Internet, car le protocole HTTPS (le « s » de « HTTPS » signifie « sécurisé ») repose sur lui : lorsque vous naviguez sur un site Web sécurisé, vous utilisez SSL sans le savoir. SSL a sans doute été l'une des étapes les plus importantes dans le développement du commerce électronique : grâce à lui, les internautes ont commencé à pouvoir faire des transactions sur Internet, en toute confiance.

C.6.2 Un exemple : une banque en ligne

Le principe de fonctionnement du protocole SSL est assez simple : dans une première phase, un cryptage asymétrique permet à l'utilisateur et au serveur de s'échanger une clé de cryptage symétrique. Par la suite, tous leurs échanges sont cryptés avec cette clé symétrique : le tunnel sécurisé est en place.

Prenons un exemple concret : vous souhaitez consulter votre compte bancaire sur Internet. Vous allez commencer par démarrer votre navigateur Internet et vous rendre à l'adresse `https://www.votrebanque.fr`. Au moment où votre navigateur contacte le serveur Web de votre banque, il commence par lui demander son certificat électronique. Le serveur s'exécute et renvoie son certificat. Votre navigateur vérifie alors deux choses : que le nom qui est indiqué dans le certificat correspond bien au nom du site Web (`www.votrebanque.fr`), et que le certificat est bien signé par quelqu'un en qui vous avez confiance (par exemple, Verisign).

Ensuite, votre navigateur génère aléatoirement une clé de cryptage symétrique. La clé publique du serveur Web (contenue dans son certificat) est utilisée par votre navigateur pour crypter la clé de cryptage symétrique, et le résultat est envoyé au serveur Web. Celui-ci est le seul à posséder la clé privée qui permet de décrypter la clé symétrique : ainsi, une clé symétrique a pu être échangée de façon complètement confidentielle.

Cette première phase s'appelle le *handshake*, c'est-à-dire la « poignée de main » ou encore la « négociation ». Une fois terminée, l'utilisateur et le serveur Web peuvent s'échanger des informations tout à fait confidentielles (comme des numéros de cartes bancaires), car elles sont dorénavant cryptées grâce au cryptage symétrique.

C.6.3 Autres aspects de SSL

Le protocole SSL est plus souple que le simple exemple précédent ne laisse paraître :

- Lors de la négociation, l'utilisateur peut également envoyer son certificat, s'il en possède un. Sur certains sites Web, ceci permet au serveur de s'assurer de l'identité de l'utilisateur (sinon, seul l'utilisateur s'assure de l'identité du serveur).
- Le protocole SSL peut être utilisé pour sécuriser n'importe quel protocole qui repose habituellement sur TCP (voir l'annexe A) : SMTP, POP, LDAP… En effet, après la négociation initiale, un tunnel SSL est semblable, du point de vue des couches réseaux supérieures, à un socket TCP[1].
- Lors de la négociation, tout type d'algorithme de cryptage symétrique peut être négocié entre l'utilisateur et le serveur.
- Un algorithme de compression peut également être utilisé pour réduire la taille des données passant par le tunnel.

C.7 LE PROTOCOLE TLS

Le protocole SSL a été défini par la société Netscape. Lorsque sa version 3.0 a été publiée en 1996, l'IETF (voir le chapitre 8) a décidé de s'en inspirer pour créer son propre standard. C'est ainsi que le protocole TLS version 1.0 a vu le jour, au sein de la RFC 2246 : TLS est en quelque sorte, la nouvelle version de SSL.

Toutes les nouveautés sont maintenant apportées au protocole TLS, et non au protocole SSL : par exemple, le cryptage AES (voir le chapitre 9, paragraphe 9.4) peut maintenant être utilisé comme algorithme symétrique après la négociation TLS. Pour l'essentiel, les protocoles SSL et TLS sont identiques, et il n'est donc pas nécessaire de rentrer dans le détail de TLS.

1. Une présentation rapide des protocoles UDP et TCP se trouve au paragraphe 10.2.1 du chapitre 10.