**Cryptographie**

[](http://commons.wikimedia.org/wiki/File:Lorenz-SZ42-2.jpg)

La machine de Lorenz utilisée par les Allemands durant la Seconde Guerre mondiale pour chiffrer les communications militaires de haut niveau entre le quartier-général du Führer et les quartiers-généraux des groupes d'armées

La **cryptographie** est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de *secrets* ou *clés*. Elle se distingue de la stéganographie qui fait passer inaperçu un message dans un autre message alors que la cryptographie rend un message inintelligible à autre que qui-de-droit.

Elle est utilisée depuis l'Antiquité, mais certaines de ses méthodes les plus importantes, comme la cryptographie asymétrique, datent de la fin du XXe siècle.

**Étymologie et vocabulaire**

Le mot cryptographie vient des mots en grec ancien *kruptos* (« caché ») et *graphein* (« écrire »).

À cause de l'utilisation d'anglicismes puis de la création des chaînes de télévision dites « cryptées », une grande confusion règne concernant les différents termes de la cryptographie :

* chiffrement : transformation à l'aide d'une clé d'un message en clair (dit texte clair) en un message incompréhensible (dit texte chiffré) pour celui qui ne dispose pas de la clé de déchiffrement (en anglais *encryption key* ou *private key* pour la cryptographie asymétrique) ;
* chiffre : utilisation de la substitution au niveau des lettres pour coder1 ;
* code : utilisation de la substitution au niveau des mots ou des phrases pour coder1 ;
* coder : action réalisée sur un texte lorsqu'on remplace un mot ou une phrase par un autre mot, un nombre ou un symbole1 ;
* cryptogramme : message chiffré ;
* cryptosystème : algorithme de chiffrement;
* décrypter : retrouver le message clair correspondant à un message chiffré *sans posséder la clé de déchiffrement* (terme que ne possèdent pas les anglophones, qui eux « cassent » des codes secrets)2 ;
* cryptographie : étymologiquement « écriture secrète », devenue par extension l'étude de cet art (donc aujourd'hui la science visant à créer des cryptogrammes, c'est-à-dire à chiffrer) ;
* cryptanalyse : science analysant les cryptogrammes en vue de les décrypter ;
* cryptologie : science regroupant la cryptographie et la cryptanalyse.
* cryptolecte : jargon réservé à un groupe restreint de personnes désirant dissimuler leur communication.

Il apparaît donc que mis au regard du couple chiffrer/déchiffrer et du sens du mot « décrypter », le terme « crypter » n'a pas de raison d'être (l'Académie française précise que le mot est à bannir), en tout cas pas dans le sens où on le trouve en général utilisé[réf. nécessaire]. Dans sa dernière édition (entamée en 1992) le Dictionnaire de l'Académie française n'intègre pas « crypter » et « cryptage », mais ce dernier terme apparait dans le Grand Robert (qui date son apparition de 1980). L'Office québécois de la langue française intègre « crypter » au sens de « chiffrer », et « cryptage » au sens de déchiffrement dans son grand dictionnaire terminologique3.

**Histoire**

Utilisé depuis l'antiquité, l'une des utilisations les plus célèbres pour cette époque est le chiffre de César, nommé en référence à Jules César qui l'utilisait pour ses communications secrètes. Mais la cryptographie est bien antérieure à cela : le plus ancien document chiffré est une recette secrète de poterie qui date du XVIe siècle av. J.-C., qui a été découverte dans l'actuelle Irak.

Bien qu'éminemment stratégique, la cryptographie est restée pendant très longtemps un art, pour ne devenir une science qu'au XXIe siècle. Avec l'apparition de l'informatique, son utilisation se démocratise de plus en plus.

**Utilisations**

Les domaines d'utilisations de la cryptographie sont très vastes et vont du domaine militaire, au commercial, en passant par la protection de la vie privée.

**Algorithmes et protocoles**

**Algorithmes de chiffrement faibles (facilement cassables)**

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble. Ils consistaient notamment au remplacement de caractères par d'autres. La confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide.

Exemples d'algorithmes de chiffrement faibles :

* ROT13 (rotation de 13 caractères, sans clé) ;
* Chiffre de César (décalage de trois lettres dans l'alphabet sur la gauche).
* Chiffre de Vigenère (introduit la notion de clé)

**Algorithmes de cryptographie symétrique (à clé secrète)**

Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. L'un des problèmes de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sûre. La mise en œuvre peut s'avèrer difficile, surtout avec un grand nombre de correspondants car il faut autant de clés que de correspondants.

Quelques algorithmes de chiffrement symétrique très utilisés :

* Chiffre de Vernam (le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message, qu'elle ne soit utilisée qu'une seule fois à chiffrer et qu'elle soit totalement aléatoire)
* DES
* 3DES
* AES
* RC4
* RC5
* MISTY1
* et d'autres (voir la liste plus exhaustive d'algorithmes de cryptographie symétrique).

**Algorithmes de cryptographie asymétrique (à clé publique et privée)**

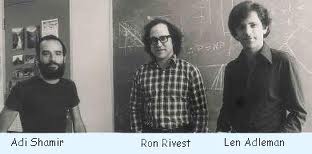
Pour résoudre le problème de l'échange de clés, la cryptographie asymétrique a été mise au point dans les années 1970. Elle se base sur le principe de deux clés :

* une publique, permettant le chiffrement ;
* une privée, permettant le déchiffrement.

Comme son nom l'indique, la clé publique est mise à la disposition de quiconque désire chiffrer un message. Ce dernier ne pourra être déchiffré qu'avec la clé privée, qui doit rester confidentielle.

Quelques algorithmes de cryptographie asymétrique très utilisés :

* RSA (chiffrement et signature);
* DSA (signature);
* Protocole d'échange de clés Diffie-Hellman (échange de clé);
* et d'autres ; voir cette liste plus complète d'algorithmes de cryptographie asymétrique.



Les concepteurs du RSA : *Rivest*, *Shamir* et *Adleman*

Le principal inconvénient de RSA et des autres algorithmes à clés publiques est leur grande lenteur par rapport aux algorithmes à clés secrètes. RSA est par exemple 1000 fois plus lent que DES. En pratique, dans le cadre de la confidentialité, on s'en sert pour chiffrer un nombre aléatoire qui sert ensuite de clé secrète pour un algorithme de chiffrement symétrique. C'est le principe qu'utilisent des logiciels comme PGP par exemple.

La cryptographie asymétrique est également utilisée pour assurer l'authenticité d'un message. L'empreinte du message est chiffrée à l'aide de la clé privée et est jointe au message. Les destinataires déchiffrent ensuite le cryptogramme à l'aide de la clé publique et retrouvent normalement l'empreinte. Cela leur assure que l'émetteur est bien l'auteur du message. On parle alors de signature ou encore de scellement.

**Fonctions de hachage**

Une fonction de hachage est une fonction qui convertit un grand ensemble en un plus petit ensemble, l'empreinte. Il est impossible de la déchiffrer pour revenir à l'ensemble d'origine, ce n'est donc pas une technique de chiffrement.

Quelques fonctions de hachage très utilisées :

* MD5 ;
* SHA-1 ;
* SHA-256 ;
* et d'autres ; voir cette liste plus complète d'algorithmes de hachage.

L'empreinte d'un message ne dépasse généralement pas 256 bits (maximum 512 bits pour SHA-512) et permet de vérifier son intégrité.

**Communauté**

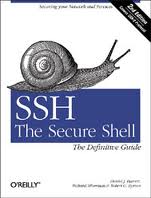
* Projet NESSIE
* Advanced Encryption Standard process
* Les cryptologues sont des experts en cryptologie : ils conçoivent, analysent et cassent les algorithmes (voir cette liste de cryptologues).

**Implémentations du cryptage et du chiffrement**

**Secure SHell**

***Secure Shell*** (**SSH**) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un *sniffer* pour voir ce que fait l'utilisateur.

Le protocole **SSH** a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet, rcp, ftp et rsh.



SSH – Secure Shell – O’ Reilly

## Le protocole

Le protocole **SSH** existe en deux versions majeures : la version 1.0 et la version 2.0. La première version permet de se connecter à distance à un ordinateur afin d'obtenir un shell ou ligne de commande. Cette version souffrait néanmoins de problèmes de sécurité dans la vérification de l'intégrité des données envoyées ou reçues, la rendant vulnérable à des attaques actives. En outre, cette version implémentait un système sommaire de transmission de fichiers, et du *port tunneling*.

La version 2 qui était à l'état de *draft* jusqu'en janvier 2006 est déjà largement utilisée à travers le monde. Cette version est beaucoup plus sûre au niveau cryptographique, et possède en plus un protocole de transfert de fichiers complet, le SSH file transfer protocol.

Habituellement le protocole SSH utilise le port TCP 22. Il est particulièrement utilisé pour ouvrir un shell sur un ordinateur distant. Peu utilisé sur les stations Windows (quoiqu'on puisse l'utiliser avec PuTTY, mRemote, cygwin ou encore OpenSSH), SSH fait référence pour l'accès distant sur les stations Linux et Unix.

SSH peut également être utilisé pour transférer des ports TCP d'une machine vers une autre, créant ainsi un tunnel. Cette méthode est couramment utilisée afin de sécuriser une connexion qui ne l'est pas (par exemple le protocole de récupérations de courrier électronique POP3) en la faisant transférer par le biais du tunnel chiffré SSH.

Il est également possible de faire plusieurs sauts entre consoles SSH, c'est-à-dire ouvrir une console sur un serveur, puis, de là, en ouvrir une autre sur un autre serveur.

## Historique

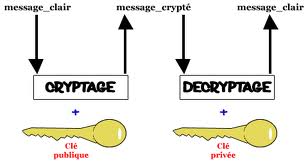
La première version de SSH (SSH-1) a été conçue par Tatu Ylönen, à Espoo, en Finlande en 1995. Il a créé le premier programme utilisant ce protocole et a ensuite créé une entreprise, SSH Communications Security pour exploiter cette innovation. Cette première version utilisait certains logiciels libres comme la bibliothèque Gnu libgmp, mais au fil du temps ces logiciels ont été remplacés par des logiciels propriétaires. SSH Communications Security a vendu sa licence SSH à F-Secure (anciennement connue sous le nom de Data Fellows).

La version suivante a été nommée SSH-2. Le groupe de recherche de l'IETF « secsh » a défini en janvier 2006 le standard Internet SSH-2, que l'on retrouve actuellement dans la plupart des implémentations. Cette version permet une compatibilité ascendante avec les implémentations du *draft* de SSH-2 qui étaient en version 1.99.

## SSH avec authentification par clés

Avec SSH, l'authentification peut se faire sans l'utilisation de mot de passe ou de phrase secrète en utilisant la cryptographie asymétrique. La clé publique est distribuée sur les systèmes sur lesquels on souhaite se connecter. La clé privée, qu'on prendra le soin de protéger par un mot de passe, reste uniquement sur le poste à partir duquel on se connecte. L'utilisation d'un « agent ssh » permet de stocker le mot de passe de la clé privée pendant la durée de la session utilisateur.

Cette configuration profite aussi à SCP et à SFTP qui se connectent au même serveur **SSH**.



Le cryptage par clés

## Recommandations

SSH V2 est définie par plusieurs recommandations :

* RFC 4251 : Architecture générale du protocole
* RFC 4252 : Protocole d'authentification (PKI, par mot de passe ou par machine)
* RFC 4253 : Protocole de transport sécurisé (Chiffrement, Signature numérique, intégrité)
* RFC 4254 : Protocole de connexion (Port tunneling, Shell)

## Implémentations logicielles

* *OpenSSH*, le projet libre d'outils SSH. OpenSSH est l'implémentation ssh la plus utilisée, y compris par les distributions GNU/Linux.
* *Portable OpenSSH*, une implémentation OpenSSH multiplate-forme.
* *lsh*1, une implémentation distribuée par le projet GNU selon les termes de la licence GNU GPL.
* *MacSSH*2, une implémentation *lsh* pour Mac OS classic 68k et PPC.
* *FRESH*3, une implémentation ssh en environnement JBoss.
* *SSHWindows*4, une implémentation pour Windows non maintenue.
* *Dropbear*5, une implémentation libre ayant pour but de remplacer OpenSSH sur les systèmes Unix ayant peu de ressources (processeur, mémoire, etc.) comme par exemple les systèmes embarqués.

**TLS**

**Transport Layer Security** (**TLS**), et son prédécesseur **Secure Sockets Layer** (**SSL**), sont des protocoles de sécurisation des échanges sur Internet, développés à l'origine par Netscape (SSL version 2 et SSL version 3). Il a été renommé en Transport Layer Security (TLS) par l'IETF à la suite du rachat du brevet de Netscape par l'IETF en 2001. Le groupe de travail correspondant à l'IETF a permis la création des RFC 2246 pour le TLS et RFC 4347 pour son équivalent en mode datagramme, le DTLS. Depuis son rapatriement par l'IETF, le protocole TLS a vécu deux révisions subséquentes : TLSv1.1 décrite dans la RFC 4346 et publiée en 2006 et TLSv1.2, décrite par la RFC 5246 et publiée en 2008.

Il y a très peu de différences entre SSL version 3 et TLS version 1 (qui correspond à la version 3.1 du mécanisme SSL) rendant les deux protocoles non interopérables. TLS a tout de même mis en place un mécanisme de compatibilité ascendante avec SSL. En outre, TLS diffère de SSL pour la génération des clés symétriques. Cette génération est plus sécurisée dans TLS que dans SSLv3 dans la mesure où aucune étape de l'algorithme ne repose uniquement sur MD5 pour lequel sont apparues des faiblesses en cryptanalyse.

Par abus de langage, on parle de SSL pour désigner indifféremment SSL ou TLS.

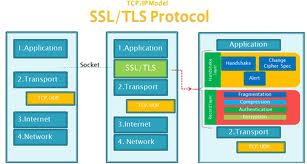


Schéma de fonctionnement – Sécurisation d’une communication

TLS fonctionne suivant un mode client-serveur. Il fournit les objectifs de sécurité suivants :

* l'authentification du serveur ;
* la confidentialité des données échangées (ou session chiffrée) ;
* l'intégrité des données échangées ;
* de manière optionnelle, l'authentification ou l'authentification forte du client avec l'utilisation d'un certificat numérique ;
* la spontanéité, c'est-à-dire qu'un client peut se connecter de façon transparente à un serveur auquel il se connecte pour la première fois ;
* la transparence, qui a contribué certainement à sa popularité : les protocoles de la couche d'application n'ont pas à être modifiés pour utiliser une connexion sécurisée par TLS. Par exemple, le protocole HTTP est identique, que l'on se connecte à un schème http ou https.

## Présentation

Avec le développement d'Internet, de nombreuses sociétés commerciales commencent à proposer des achats en ligne pour les particuliers. L'offre se met à croître régulièrement, mais le chiffre d'affaires dégagé par le commerce électronique restera encore modeste, tant que les clients n'auront pas une confiance totale dans le paiement par carte bancaire. Une des façons de sécuriser ce paiement est d'utiliser des protocoles d'authentification et de chiffrement tels que TLS.

La première version de SSL parue, la SSL 2.0, possédait un certain nombre de défauts de sécurité, parmi lesquels la possibilité de forcer l'utilisation d'algorithmes de chiffrement plus faibles, ou bien une absence de protection pour la prise de contact et la possibilité pour un attaquant d'exécuter des attaques par troncature1. Les protocoles PCT 1.0, puis SSL 3.0, furent développés pour résoudre la majeure partie de ces problèmes, le second devenant rapidement le protocole le plus populaire pour sécuriser les échanges sur Internet.

Avec un système SSL/TLS, la sécurité a été sensiblement améliorée, et les risques pour le client minimisés. Le degré de confiance à placer dans cette technologie, doit être comparé aux solutions de paiement par chèque ou par mandat. Dans celles-ci, d'après différentes études, le risque est près de 20 fois supérieur au risque lié à la toute première version de SSL. De plus, les distributeurs bancaires courants utilisent presque tous cette même technologie pour faire transiter les informations vers la banque, et utilisent parfois même une version plus ancienne de SSL que les sites de commerce électronique actuels, ce qui tend à prouver la fiabilité de cette solution.

La session chiffrée est utilisée généralement lors de l'envoi du numéro de carte bancaire, mais elle peut l'être dans d'autres cas. Le chiffrement est réalisé par à la fois un chiffrement asymétrique (qui va permettre une authentification), comme l'algorithme RSA, et à la fois par un chiffrement symétrique (qui est plus léger à réaliser qu'un chiffrement asymétrique) et qui va assurer la transmission des informations (comme le AES). On y adjoint une fonction de hachage, comme le SHA-1, pour s'assurer que les données sont transmises sans être corrompues.

En 2009, TLS est utilisé par la plupart des navigateurs Web. On reconnaît qu'une transaction est chiffrée lorsqu'une clé ou un cadenas fermé s'affiche dans un coin inférieur de l'écran ainsi qu'à gauche dans la barre d'adresse2, l'adresse commence par https://...

## Fonctionnement

|  |  |
| --- | --- |
| **modèle OSI              pile de protocoles** | |
| 7 - couche application | HTTP, SMTP, FTP, SSH, IRC, SNMP, SIP ... |
| 6 - couche de présentation |  |
| 5 - couche de session | TLS, SSL,SSH-user, NetBIOS |
| 4 - couche de transport | TLS, SSL,TCP, UDP, SCTP, RTP, DCCP ... |
| 3 - couche réseau | IPv4, IPv6, ARP, IPX ... |
| 2 - couche de liaison | Ethernet, 802.11 WiFi, Token ring, FDDI, ... |
| 1 - couche physique | Câble, fibre optique, ondes radio... |

Dans la pile de protocole TCP/IP, SSL se situe entre la couche application (comme HTTP, FTP, SMTP, *etc.*) et la couche transport TCP.  
Son utilisation la plus commune reste cependant en dessous de HTTP. Le protocole SSL est implémenté par la couche session de la pile, ce qui a deux conséquences :

* pour toute application existante utilisant TCP, il peut exister une application utilisant SSL. Par exemple, l'application HTTPS correspond à HTTP au-dessus de SSL ;
* une application SSL se voit attribuer un nouveau numéro de port par l'IANA. Par exemple HTTPS est associé au port 443.
* dans certains cas, le même port est utilisé avec et sans SSL. Dans ce cas, la connexion est initiée en mode non chiffré. Le tunnel est ensuite mis en place au moyen du mécanisme StartTLS. C'est le cas, par exemple des protocoles de mails IMAP et SMTP ou LDAP

## Navigateurs web

La plupart des navigateurs web gèrent parfaitement SSL 2.0, SSL 3.0 et TLS 1.0 (ce dernier n'étant pas activé par défaut sous Internet Explorer 6). Cependant, SSL 2.0 est désactivé par défaut dans Firefox 2+ et Internet Explorer 73, ce qui obligerait la plupart des vendeurs en ligne à modifier la configuration de leurs serveurs (modification somme toute minime), comme le fait remarquer cet (fr) article de ZDNet.fr. Une extension pour Firefox est maintenue conjointement par le projet Tor et l'EFF depuis 20104.

* Internet Explorer 9 supporte : SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1 et TLS 1.2
* Firefox 16 et 17 supportent : SSL 3.0 et TLS 1.0

## Authentification du client SSL par certificat numérique

Dans la majorité des cas, l'utilisateur authentifie le serveur TLS sur lequel il se connecte. Cette authentification est réalisée par l'utilisation d'un certificat numérique X.509 délivré par une autorité de certification (AC). Mais de plus en plus d'applications web utilisent maintenant l'authentification du poste client en exploitant TLS. Il est alors possible d'offrir une authentification mutuelle entre le client et le serveur. Le certificat client peut être stocké au format logiciel sur le poste client ou au format matériel (carte à puce, token USB) pour augmenter la sécurité du lien TLS. Cette solution permet d'offrir des mécanismes d'authentification forte.

## Attaques

En 2001, Serge Vaudenay découvre une attaque par canal auxiliaire contre SSL. Cette attaque profite d'une mauvaise implémentation du remplissage qui est utilisé lorsque les entrées ont une taille variable. Le mode de chiffrement CBC (cipher block chaining) consiste à diviser les données en plusieurs blocs de même taille et à les chiffrer de manière chaînée (le résultat précédent est utilisé lors du chiffrement suivant). L'attaque de Vaudenay utilise les temps de réponse des serveurs en cas d'erreurs lors du remplissage. Avec un peu de chance, il est possible de découvrir les dernières données qui ont été envoyées et de les récupérer. L'attaque est toutefois inopérante avec un chiffrement de type RC4 et n'est valable que sous certaines conditions. Elle a malgré tout été utilisée avec succès contre certains « webmails » qui envoient plusieurs fois les mêmes données. À la suite de cette attaque, le standard a été mis à jour.

Cette attaque est maintenant totalement dépassée et ne peut plus du tout être utilisée.

## Sécurisation du protocole

Le protocole TLS permet de créer un tunnel entre un ordinateur et un serveur. Ce tunnel sécurisé permet un échange d'informations en contournant les dispositifs de sécurité installés pour un serveur ou un ordinateur. Passant outre les systèmes de protection il est alors possible que des actions malveillantes soient menées au travers du point d'entrée du tunnel. Afin de limiter les risques, il est techniquement possible de filtrer les contenus d'un tunnel TLS par la mise en place d'un dispositif qui authentifie le client et le serveur. Deux tunnels sont alors mis en place, un depuis le client vers le dispositif d'authentification et le second du dispositif vers le serveur. Ce système permet alors une analyse et une sécurisation transparente des contenus transférés par le tunnel TLS5.

## Textes de référence

Le premier texte formel définissant le protocole TLS est la RFC 2246 publiée par l'IETF en 1999. D'autres textes ont suivi sur ce protocole :

* RFC 27126 : addition de la suite Kerberos à TLS ;
* RFC 28177 : Passage à TLS lors d'une session HTTP 1.1 ;
* RFC 28188 : HTTP sur TLS ;
* RFC 32689 : Utilisation du système de chiffrement AES pour TLS.
* RFC 524610 : TLS version 1.2 (publication août 2008)
* RFC 609111 : Utilisation de certificats OpenPGP pour l'authentification TLS.
* RFC 610112 : SSL version 3.0 (spécification tardive du protocole ayant inspiré TLS).



Logo de l’IETF