Utilisation de la cryptographie

Laurent Signac – CC-BY-SA – 30-08-20 2226 c2fd2f4852aa9c95f5c8

1 Documents

- Référentiel général de Sécurité[1] de l'ANSSI

2 Définitions

- confidentialité : protection des données
- authentification : vérification de l'identité
- intégrité : garantie que les données n'ont pas été modifiées
- non répudiation : impossibilité pour l'auteur de nier son action

De nombreux problèmes de sécurité sont liés à la cryptographie.

3 Authentification

On distingue trois groupes :

- ce que je sais : couple login / mot de passe, questionnaire, authentification par couple clé privée/publique…
- ce que je possède : un téléphone, une carte à puce, un badge quelconque…
- ce que je suis : mes empreintes digitales, la texture de mon iris, la vascularisation de ma rétine, le son de ma voix…

L'authentification multi-facteur utilise des méthodes issues de groupes différents, par exemple : mot de passe + code reçu par SMS.

La méthode des mots de passe est très largement utilisée (sur un simple ordinateur de bureau, sur des pages Web etc..). Généralement, les mots de passe stockés le sont sous forme de condensés, de telle sorte que la connaissance de ces condensés (si le fichier les contenant est dérobé) ne permet pas de retrouver les mots de passe en clair. Il existe néanmoins des logiciels spécialisés qui calculent les condensés de très nombreux mots de passe et les compare avec ceux utilisés par le système. Ces logiciels sont basés sur l'usage de dictionnaires, d'où la nécessité de choisir des mots de passe longs et/ou difficiles à retenir (l'objectif est qu'ils soient difficiles à dériver de mots du dictionnaire). Les services qui stockent les mots de passe utilisent généralement un *sel* (chaîne de caractère fixe préfixée au mot de passe avant hachage).

Les fuites de bases de données contenant des mots de passe sont très courantes (voir haveibeenpwned[2]).

L'authentification par clé privée/publique est de plus en plus utilisée. Un système disposant de la clé publique d'Alice peut authentifier Alice en vérifiant ce qu'elle est capable de chiffrer avec sa clé privée ou en lui demandant de déchiffrer ce qui a été chiffré avec sa clé publique (utilisé par SSH, l'accès à des dépôts Git…).

4 Signature numérique

L'authentification de documents papiers repose généralement sur la signature. Il existe un équivalent numérique, basé sur la cryptographie asymétrique.

Le but de la signature, lors de l'envoi d'un message quelconque est que :

- le destinataire puisse vérifier l'identité de l'expéditeur ;
- l'expéditeur ne puisse pas répudier son message ;

[1] https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/liste-des-documents-constitutifs-du-rgs-v-2-0/
[2] https://haveibeenpwned.com/

- le destinataire ne puisse pas fabriquer un tel message.

Si l'algorithme de chiffrement asymétrique a la propriété suivante : appliquer l'algorithme de chiffrement (basé sur la clé publique) sur l'algorithme de déchiffrement (basé sur la clé privée) redonne le message en clair, il peut être utilisé pour de la signature.

Supposons qu'Alice veuille signer un message qu'elle envoie à Bob. Nous noterons respectivement E_A, D_A, E_B et D_B les fonctions de chiffrement et déchiffrement d'Alice et Bob (qui nécessitent la connaissance des clés appropriées). Pour signer et chiffrer son message M, Alice envoie à Bob $C = E_B(D_A(M))$ ce qui est possible puisque Alice connaît sa propre clé privée ainsi que la clé publique de Bob. À la réception, Bob obtient le message en calculant $M = E_A(D_B(C))$, ce qui est possible puisque Bob connaît sa propre clé privée et la clé publique d'Alice. Il a l'assurance qu'Alice est bien l'expéditeur puisque seule Alice peut produire le message $D_A(M)$, à condition qu'Alice garde effectivement sa clé privée secrète (c'est un prérequis) De même, Alice ne pourra pas nier avoir envoyé le message puisqu'elle seule peut produire $D_A(M)$.

Si elle souhaite simplement signer son message (sans le chiffrer, ce qui est souvent le cas des documents papier), Alice peut calculer $D_A(c_M)$ où c_M est un condensé de son message. Bob pourra le déchiffrer en calculant $E_A(D_A(c_M))$ et vérifier que le condensé qu'il obtient est bien celui du message. Ainsi, pour simplement signer ses messages, nul besoin de connaître à l'avance l'identité du destinataire (propriété qu'on retrouve dans la signature papier).

Les algorithmes les plus utilisés pour la signature numérique sont RSA, DSA (problème du logarithme discret), et ECDSA (variante de DSA utilisant les courbes elliptiques).

5 Pretty Good Privacy, OpenPGP

Le système PGP (Pretty Good Privacy) a été inventé par Philip Zimmermann. Il est destiné à l'authentification et au chiffrement du courrier électronique. De son logiciel PGP est née une spécification : OpenPGP qui détaille les fonctionnalités de sécurisation des emails. Le logiciel GnuPG est une implantation libre de la spécification OpenPGP. Ce qui est valable pour PGP est donc *a priori* valable pour GnuPG.

En ce qui concerne le chiffrement, PGP réalise les actions suivantes :

- compression du texte du courrier ;
- génération d'une clé secrète aléatoire ;
- chiffrement du texte compressé avec la cré secrète (avec un algorithme adapté, par exemple AES) ;
- chiffrement de la clé secrète avec la clé publique du destinataire.

En plus de la fonction de chiffrement, PGP permet de signer numériquement les messages. Pour cela, PGP calcule un condensé du message à envoyer, puis le chiffre avec la clé privée de l'expéditeur.

6 Transport Layer Security

La plupart des protocoles applicatifs d'internet ne sont pas chiffrés (POP, SMTP, HTTP…). Les protocoles de plus bas niveau ne le sont pas non plus IP, TCP… Par défaut, rien n'est chiffré, et l'espionnage du réseau est très facile. Depuis quelques années, les algorithmes de chiffrement asymétriques sont utilisés pour sécuriser les communication entre deux machines, au niveau application (de bout en bout) ou des protocoles de niveau inférieur.

La norme actuelle est TLS (*Transport Layer Security*), qui fait suite à SSL. Le protocole «s'intercale» entre un protocole application (HTTP par exemple) et TCP. TLS propose le chiffrement des informations et l'authentification du serveur (et éventuellement du client). L'authentification se fait par un échange de certificats. En ce qui concerne le chiffrement, il est basé sur des algorithmes de chiffrement symétriques. Le chiffrement utilisé est négocié entre le client et le serveur et l'échange des clés peut être fait selon le protocole de Diffie-Hellman-Merkle, ou par chiffrement asymétrique.

Les protocoles utilisant TLS fonctionnent sur un port dédié. Voici un récapitulatif des plus utilisés :

Service	Port standard	Port TLS
POP	110	995
HTTP	80	443
IMAP	143	993
SMTP	25	465
FTP	21	115
MQTT	1883	8883

7 Dans les objets connectés

Les objets connectés pullulent (que ce soient les terminaux (téléphone, etc..)) ou les capteurs).

Dans le cas de *petits* objets connectés, peu chers, la vérification des protocoles cryptographiques, voire leur simple utilisation est loin d'être acquise. La non prise en compte des problèmes de sécurité dès l'étape de conception des produits a déjà occasionné des dégâts (ex : Mirai en 2016, bien que la faille était plus simple à exploiter qu'une faille cryptographique).

8 Ces dernières années...

8.1 Factorisation

Lien vers tous les résultats sur Wikipédia[3]

Problème	Nb chiffres (10)	Prix ($)	Date
RSA-100	100		1991
RSA-576	174	10,000	Déc 2003
RSA-200	200		Mai 2005
RSA-640	193	20,000	Nov 2005
RSA-768	230	50,000	Déc 2009
RSA-240	240		Déc 2019
RSA 250	250		Fév 2020
RSA-2048	617	200,000	

```
RSA-768 = 123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3
```

```
RSA-768 = 33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489
        * 36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917
```

Factoriser RSA-768 représente plus de deux ans de calcul, et l'utilisation de plusieurs centaines de machines (réf[4]) (équivalent à 2000 ans de calcul sur un processeur standard cadencé à 2.2 Ghz)

```
RSA-250 = 21403246502407449612644230728393335630086147151447550177977549208814180234471401366433455190958046796109928518724709145876873962619215573630474547705208051190564931066876915900197594056934574522305893259766974716817380693648946998715784949759374979 37
```

[3] https://en.m.wikipedia.org/wiki/RSA_Factoring_Challenge
[4] https://eprint.iacr.org/2010/006.pdf

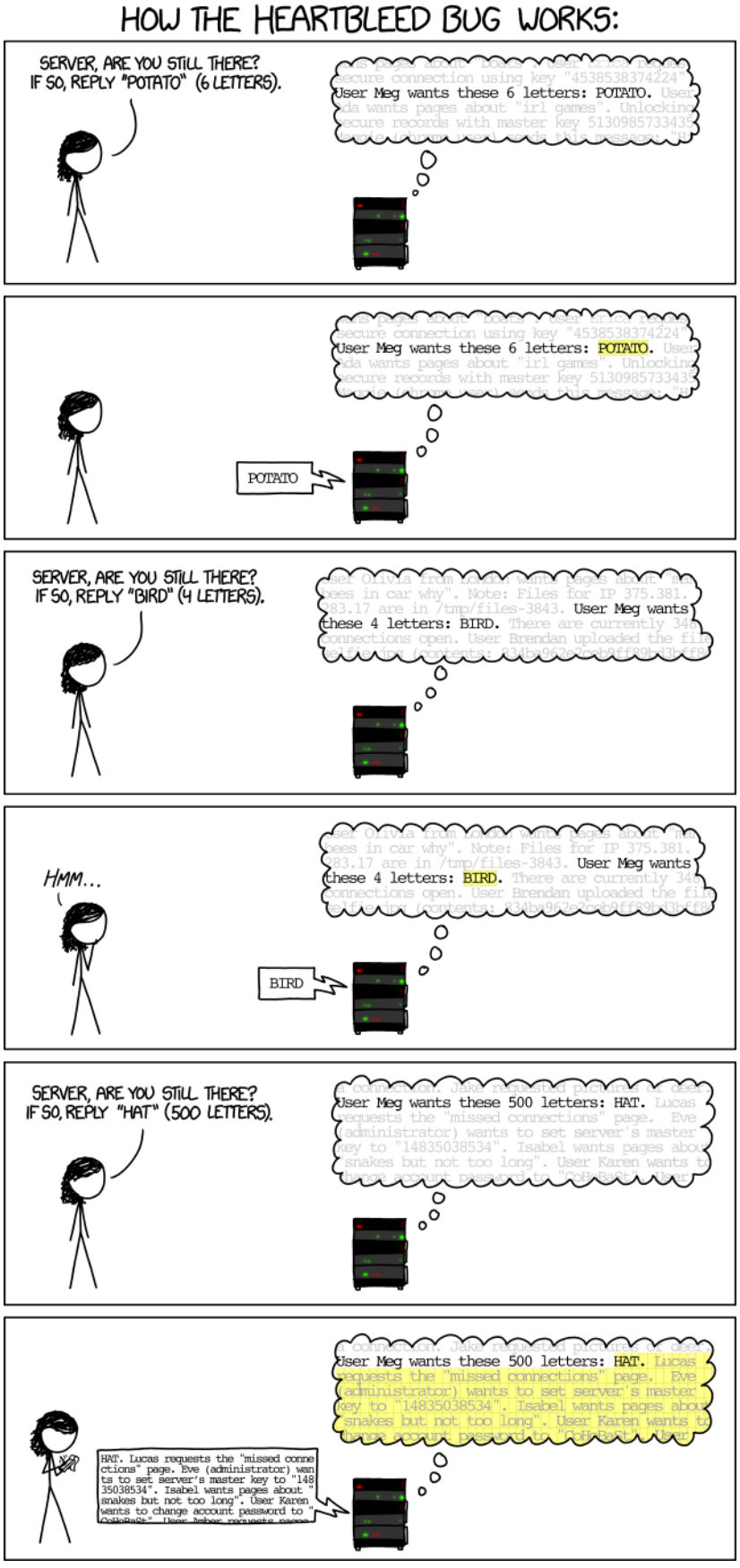


Figure 1: Bug Heartbleed Xkcd : http://xkcd.com

```
RSA-250 = 64135289477071580278790190170577389084825014742943447208116859632024532344630238623598752668347708737661925585694639798853367
        * 33372027594978156556226010605355114227940760344767554666784520987023841729210037080257448673296881877565718986258036932062711
```

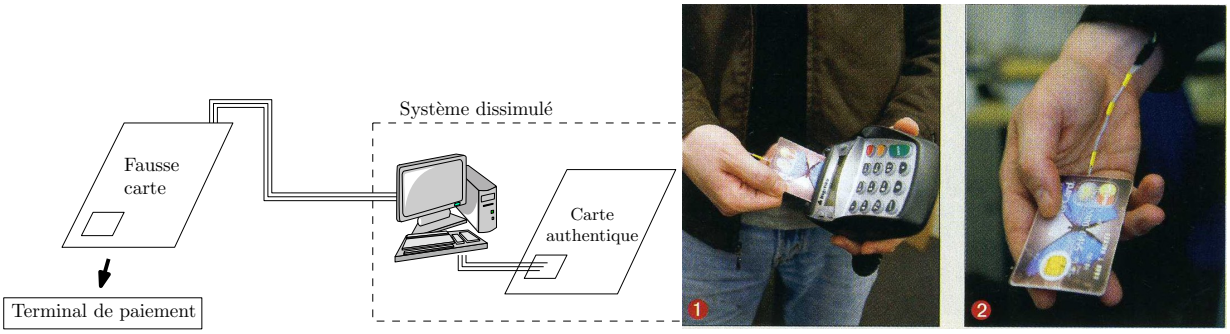
8.2 Attaques par canal auxiliaire

- analyse du traffic (présence/absence)
- analyse électromagnétique ou de puissance (carte à puce)
- tempest attack (pose d'écrans électromagnétiques soumis à autorisation gouvernementale…)
- virus + chevaux de Troie (obtention des informations avant chiffrement)

Adi Shamir et son équipe a pu obtenir des informations sur un chiffrement RSA à partir de la signature acoustique du processeur pendant le calcul….

8.3 Faiblesse des protocoles

Si l'algorithme cryptographique est sophistiqué… le problème peut se cacher dans le protocole :



8.4 Cryptographie quantique

Basée sur l'impossibilité de mesurer la polarisation d'un photo sans la modifier, la cryptographie quantique permet un échange de clés sûr, sur un canal non sécurisé. En cas d'espionnage, les interlocuteurs s'en rendront nécessairement compte.

8.5 Informatique quantique

L'apparition de futurs coprocesseurs quantiques pourra rendre caduques des méthodes largement utilisées aujourd'hui (RSA par exemple).

En 2017, le NIST a annoncé une compétition visant à établir une standardisation des algorithmes cryptographiques post-quantiques. La procédure est actuellement en cours.