Security Party

Marc SCHAEFER

22 octobre 2009

Résumé

L'utilisation d'outils cryptographiques est essentielle aujourd'hui : que ce soit pour chiffrer ou identifier l'émetteur d'un message électronique ou assurer la sécurité des transactions commerciales sur Internet.

Le but de cette présentation est de donner les bases minimales théoriques et pratiques pour la compréhension de ces outils et la création / signature de certificats électroniques, vous permettant d'utiliser ces technologies chez vous ou pour votre serveur Internet.

1 Cette présentation

introduction à la cryptographie, à la signature électronique et aux réseaux de confiance

Marc SCHAEFER

HE-Arc Ingénierie / ISIC

2 Plan

- la cryptographie en bref
- la signature électronique
- les réseaux de confiance (principalement cas SSL-TLS/CA et GPG/PGP)
- le fonctionnement dans le cas d'HTTPS (SSL/TLS) sur Internet
- Partie pratique

3 Principes de la cryptographie

- cacher (stéganographie) l'information
- chiffrer l'information (la rendre inintelligible)
- algorithmes mathématiques
  - publics, en règle générale
- types d'algorithmes et de clés
  - secrète (symétrique)
  - privée, publique (asymétrique)
- certificats (assurer l'identité)

3.1 Notes

Si le chiffrement n'est utilisé que pour les informations importantes, c'est comme illuminer les données intéressantes ! Il vaudrait mieux que tout le trafic soit chiffré, ce qui rendrait le tri des informations utiles plus difficile.

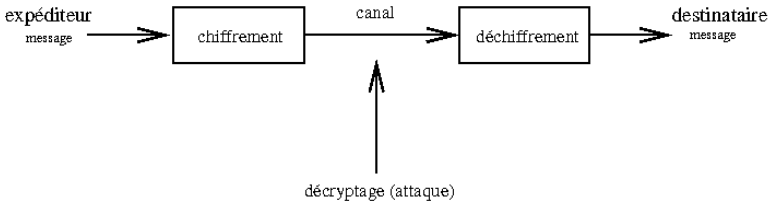
Notons aussi qu'en raison du fonctionnement d'Internet, il n'est normalement pas possible d'empêcher un attaquant éventuel de déterminer entre qui sont faits les échanges (p.ex. entre la HE-Arc et la Poste). En conséquence, une analyse différenciée du trafic permet d'en conclure des informations (p.ex. l'effet pizza au Pentagone– même si c'est une légende urbaine !)

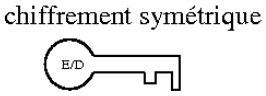
Des systèmes comme Torpermettent d'éviter ce problème, en rendant aléatoire les points du réseau par lesquels sortent les connexions effectives : mais mal utilisé, ce système peut être aussi utilisé pour une attaque.

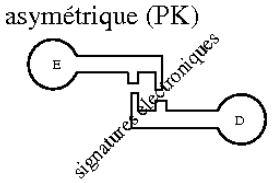
4 Systèmes à clés révélée (publiques) / Systèmes à clé secrète

clé secrète
– une clé unique (un mot de passe) connue des N participants
– on chiffre et déchiffre avec cette clé (symétrique)
clé publique
– un couple de clés (clé privée, clé publique)
– on chiffre avec la clé publique (connue, publiée, évt. certifiée)
– on déchiffre avec la clé privée
– support de la *signature électronique*

4.1 Notes







5 Cryptographie à clés secrètes (ou symétrique)

– algorithmes efficaces et sûrs même avec des tailles de clés faibles (p.ex. 128 bits)
  – AES, 3-DES, IDEA, etc
– une seule clé chiffre et déchiffre
– exemple : WiFi (802.11) WEP/WPA
– comment distribuer ces clés ?
  – N membres d'un réseau qui communiquent ensemble
    – au plus $N\frac{N-1}{2}$ communications bidirectionnelles
    – autant de clés à diffuser de manière sûre !

6 Cryptographie à clés publiques (révélées, ou asymétrique)

– algorithmes basés sur des fonctions difficilement inversibles en temps "payable"
  – RSA, EL-Gamal
– couple de clé (privée, publique)
  – lorsqu'une chiffre, l'autre permet de déchiffrer
  – on chiffre avec la publique, on déchiffre avec la privée
  – on signe avec la privée, on vérifie la signature avec la publique
    – signer == chiffrer un résumé (un hâchage, HMAC) du message avec la clé privée

6.1 Notes

Pour des raisons de performance, le chiffrement est souvent fait en symétrique, à partir d'une clé de session générée aléatoirement (!), elle-même chiffrée par l'algorithme asymétrique.

7 La signature électronique

– soit m le message à signer
– soit $h = HMAC(m)$ un hâchage cryptographique sûr du message (MD5, SHA, etc)
– soit la clé privée C_{priv}
– alors la signature électronique du message m est $s = f(h, C_{priv})$
– on vérifie alors en déchiffrant s avec la clé publique correspondante C_{pub} et en comparant avec le HMAC sur le message à signer.

le problème est surtout que l'on ne signe pas forcément ce que l'on croit.n'est-ce-pas.

7.1 Notes

En droit suisse, la signature électronique est assimilable, sous certaines conditions, à la signature manuscrite. En plus de ces conditions – qui concernent surtout les éditeurs de logiciels et les mainteneurs d'infrastructures –, quelques suppositions lourdes de sens sont faites :
– les algorithmes utilisés sont sans failles théoriques
– vous savez ce que vous signez (le logiciel est sans bug ou malveillance)
– votre clé privée est conservée précieusement
– d'éventuels espions sont mis hors d'état de nuire
Au sujet des failles théoriques
– notez qu'on peut exhiber des collisions avec un hâchage MD5
– utiliser plusieurs algorithmes simultanément !
Au sujet du problème de "quel message signe-t-on vraiment" ?
– on fait confiance soit au logiciel, soit à un équipement (terminal de signature)
– on peut distinguer plusieurs niveaux de sécurité (cas PC)

1. la clé privée est stockée sur l'ordinateur : elle peut donc être volée par un programme malveillant, dès qu'elle est débloquée (passphrase)
2. la clé privée est stockée dans une carte électronique, mais est transférée dès que le PIN est tapé (en pratique : aussi sûr que 1.)
3. la clé privée est stockée dans une carte électronique chiffrante. La carte chiffre elle-même le message dès que le PIN est tapé (impossible de voler la clé, mais dès que le PIN est tapé, d'autres messages peuvent être signés par un programme malveillant tant que la carte est présente)
4. idem, mais en plus le PIN doit être tapé à chaque signature directement sur la carte chiffrante (ou, éventuellement, sur un terminal chiffrant à qui on fait confiance, sans que le PIN-code soit accessible par le logiciel sur le PC). Dans ce cas, le seul risque est de signer le mauvais message.
5. idem, mais en plus un résumé du message apparaît sur un LCD sur la carte, voire sur le terminal chiffrant. Dans ce cas, les risques résiduels sont liés à des mauvaises implémentations embarquées (sur la carte ou le terminal chiffrant), y compris des mauvaises résistances à des observations extérieures (p.ex. rayonnement, consommation, etc)

8 Ingrédients d'une bonne infrastructure à clé publique (PKI)

mathématiques
– bon hâchage cryptographique
– bon algorithme symétrique
– bon algorithme asymétrique
techniques
– bon *générateur* aléatoire
– source et fenêtre de temps fiable (éviter les rejeux – replay attack)
– bonnes clés
humain
– bons systèmes de distribution / de confiance
– utilisateurs formés
– utiliser le chiffrement par défaut !

9 Le problème de la confiance

- la société est basée sur la confiance
  - p.ex. confiance en la monnaie papier (plus de garantie or !)
- la confiance est souvent basée sur des critères subjectifs
  - p.ex. : compagnie aérienne a un accident d'avion : faut-il avoir plus ou moins confiance dans sa gestion des prochains vols ?
- en général, plus on trouve de problèmes, moins on a confiance
  - dans les logiciels, c'est surtout le fait de ne pas (pouvoir) trouver de problèmes qui est un problème !
  - tout logiciel a des bugs, l'essentiel est de les trouver.

utiliser des logiciels de cryptographie, ou sa carte bancaire c'est faire confiance tout d'abord à des logiciels qui agissent pour vous !

10 L'attaque par "l'homme du milieu"

Man in the middle
- A veut contacter B de manière sécurisée
- A a donc besoin de la clé publique de B
- mais comment être sûr que c'est bien la clé publique de B ?

L'attaque Man in the middle
- faire croire à A que B a la clé publique C
- C peut alors déchiffrer le message, puis le rechiffrer avec la clé publique de B

Solutions
- si B a la vraie clé publique de A, il peut vérifier que le message chiffré vient bien de A
- si A a la vraie clé publique de B, le problème ne se pose pas.

Dans les deux cas, il faut un système de distribution de clés (et/ou de signatures), une infrastructure à clé publique (PKI).

11 Les modèles de confiance

- classique (hiérarchique, chaîne de confiance)
  - p.ex. le modèle de confiance des certificats SSL/TLS tel qu'implémenté dans les navigateurs
  - on peut vérifier que la clé publique est valide en vérifiant avec une clé publique administrative (certificat administratif, certificate authority (CA))
  - un certificat racine est livré avec le logiciel, il signe les certificats administratifs (chaîne de signature possible)
  - les clé privées administratives et surtout racines doivent être très bien protégés !
  - les erreurs de la part du CA sont fatales.
  - il faut payer cher, voire passer des audits compliqués, pour avoir son propre CA dans les navigateurs (installation manuelle possible !)
- mutuel (réseau de confiance, mesh)
  - chacun assure que d'autres sont dignes de confiance
  - s'il existe un chemin de confiance de A à B, la communication est sécurisable ; s'il en existe plusieurs, c'est encore mieux !
  - p.ex. réseau de confiance GPG/PGP
  - chacun fait partie du réseau de confiance : mais la défaillance d'un maillon n'est un problème que si l'on n'exige pas plusieurs chemins
- manuel
  - vérifier l'empreinte (fingerprint, thumbprint) au travers du téléphone

11.1 Notes

Exemple : carte de visite avec empreinte de la clé publique GPG

12 Le certificat X.509

- garantit le lien entre une entité physique ou morale et une entité numérique

- contient
  - une identité (qui est certifié ? p.ex. *.alphanet.ch, Marc SCHAEFER, etc)
  - une signature numérique
  - des informations (qui signe, quand et pour quelle durée de validité)
- délivré par une autorité de certification (CA)
  - elle-même connue d'une manière ou d'une autre

13 L'autorité de certification (CA)

- tiers de confiance
- rôle
  - émettre, délivrer et révoquer les certificats
  - assigner une période de validité
  - maintenir une liste de certificats révoqués
- CA de l'autorité non installé
  - CA local à une organisation
  - CAcert
- certificats auto-signés
  - usage interne
  - pas de tiers de confiance

13.1 Notes

Démonstrations
- la chaîne de confiance du certificat SSL de https ://www.yellownet.ch/et de https ://wiki.alphanet.ch/(y compris alarmes)
- l'installation d'un certificat administratif (CA) dans le navigateur Firefox et ses conséquences
- exemple de la liste de révocation (CRL) de CAcert.
- la clé GPG de Jacques CHIRACet ses signatures
- les clés GPG de Marc SCHAEFERet ses signaturesd'une de ses clés.

14 La révocation

Le problème
- une signature d'un certificat peut devoir être annulée avant son terme (durée de validité)
- p.ex. parce que la clé administrative ou racine a été compromise ! !

La solution
- un serveur maintenant la liste des certificats révoqués (CRL), les clients s'y connectent périodiquement
- risque de déni de service.

15 SSL/TLS

Secure Socket Layer / Transport Layer Security
- aussi appelé chiffrement couche application (7)
- sécurise des protocoles de couche supérieure (HTTP, POP, IMAP, SMTP, etc)
- basé sur la chaîne de confiance
- certificats de serveurs (et/ou de clients) (authentification du serveur et/ou du client)
- chiffrement des données (confidentialité)
- signature (intégrité)
- outils : openssl

15.1 Notes

L'outil SSH, fort pratique pour la connexion distante sécurisée (successeur de telnet et rlogin) ou l'exécution distante de commandes (successeur de rsh et rexec), et l'établissement de tunnels TCP chiffrés (accès via un seul port ouvert à plusieurs services) utilise la bibliothèque OpenSSL, même s'il utilise son propre protocole et dispose de sa propre gestion des clés. Il permet également de traverser de manière inverse des firewalls (tunnels inverses).

Le HTTPS est du HTTP sur SSL/TLS, usuellement sur le port 443 (plutôt que le port 80 de l'HTTP).

16 GPG/PGP

- chiffrement/déchiffrement symétrique
- chiffrement/déchiffrement/signature/vérification asymétrique
- utilisé principalement pour l'e-mail et la signature des logiciels (sources, packaging Debian/RPM, etc)
- basé sur le réseau de confiance
  - serveurs de clé (keyserver)

16.1 Notes

Commandes principales

fonctioncommande

générer une paire de clés gpg –gen-key

générer une révocation gpg –gen-revoke

télécharger la clé publique ID 7F76BFC9 d'un serveur de clé gpg –recv-keys 7F76BFC9

envoyer la clé publique ID 7F76BFC9 au serveur de clé gpg –send-keys 7F76BFC9

obtenir l'empreinte (finger print) de la clé gpg –fingerprint 7F76BFC9

signer une clé gpg –sign-key 7F76BFC9

vérifier une signature gpg –verify SIG-FILE FILE

chiffrer gpg -e < file > file.gpg

déchiffrer gpg < file.gpg > file

17 Vos questions

Vos questions ?

18 Références

- http ://www.sebsauvage.net/comprendre/ssl/

19 Remerciements

- ce document est en partie basé sur
  - Introduction to Public Key Cryptographic systems with the practical example of GNU Privacy Guard (GPG), Marc SCHAEFER, 19 juin 2002, Open Business Lunch Berne, /ch/open
  - Introduction à la cryptographie, Frédéric SCHUTZ et Marc SCHAEFER, 2005, cours sécurité postgrade ES, ESNIG
  - cours HTTPS/SSL, David GRUENENWALD, 2009, cours Applications Internet II, HE-Arc ingénierie.
-