GnuPG

Logiciel Libre pour le chiffrement et la signature de courriers électroniques

Mercredi 19 Avril
Thomas Petazzoni
http://www.toulibre.org

Pourquoi chiffrer ?

- Envoi de courrier électronique par le protocole **SMTP**, Simple Mail Transport Protocol
- Transit des données **en clair**
- Les intermédiaires sur le réseau voient vos données et peuvent les lire !
- En terme de confidentialité, similaire à une carte postale

Pourquoi chiffrer ?

Petite démonstration avec le logiciel d'écoute réseau Ethereal

Objectif du chiffrement : rendre possible la lecture seulement par le destinataire du courrier électronique

Pourquoi signer ?

- SMTP ne prévoit pas de mécanisme d'authentification de l'expéditeur
- On peut envoyer des courriers électroniques en se faisant passer pour n'importe qui, n'importe quelle adresse
- Absolument aucune garantie sur l'origine d'un courrier électronique

Pourquoi signer ?

Petite démonstration avec un client de courrier électronique

Objectif de la signature : disposer de garanties sur l'expéditeur d'un courrier électronique

Cryptographie symétrique

- Solution la plus simple : utilisation d'un **secret partagé** (mot de passe, méthode)
- Exemple : « on décale de 13 lettres les lettres de chaque mot »
- Algos: *ROT13, XOR, DES, 3DES, AES, Blowfish*
  - En ROT13, espace des clés = 26
  - En AES, 2^128 clés

    «L'âge de l'univers étant de 1010 années, si on suppose qu'il est possible de tester 1000 milliards de clefs par seconde (soit 3.2 1019 clefs par an) il faudra encore plus d'un milliard de fois l'âge de l'univers »
- **Problème : comment se mettre d'accord sur le secret partagé si on ne peut pas se voir ?**

Cryptographie asymétrique

- Utilisation de fonctions mathématiques à sens unique, ou en tout cas très très difficiles à inverser
- Principe : Diffie et Helmann, 1976
- Première implémentation : RSA, Rivest, Shamir et Adelman, 1978
- Chaque partie possède **deux** clés
  - une clé **privée**, que l'on garde secrète
  - une clé **publique**, que l'on diffuse à tout le monde

Chiffrement

Message chiffré

Alphonse chiffre son message avec la clé **publique** de Bob

Bob déchiffre son message avec **sa** clé **privée**

Signature

Message chiffré ou non, accompagné de sa signature

Alphonse signe son message avec **sa** clé **privée**

Bob vérifie la signature du message avec la clé **publique** d'Alphonse

GnuPG

- Logiciel Libre, *http://www.gnupg.org*
- Peut-être utilisé pour chiffrer/signer des fichiers
  ex: paquets Debian et Ubuntu
- Ou des courriers électroniques, directement au sein des clients : Thunderbird, Sylpheed Claws, Kmail, Evolution, Mutt
- Manipulation en ligne de commande (gpg) ou avec des interfaces graphiques (Seahorse, GnomeGPG, Kgpg, Enigmail)

Premières étapes

- Générer son couple clé publiqué / clé privé
  - En ligne de commande: gpg –gen-key
  - Avec interface graphique
- Générer un certificat de révocation, et l'imprimer
  - Pour révoquer la clé en cas de perte/vol
  - gpp –output revoke.asc –gen-revoke mykey
- Lister les clés du trousseau
  - gpg –list-keys
  - Avec interface graphique

Identifier une clé

```
thomas2@crazy:~$ gpg --list-keys --fingerprint
pub  1024D/27AF8427 2004-10-18 GnuPG Test 1
(GnuPG Test 1) <gnupgtest1@free.fr>
Empreinte de la clé = 856D 549D 29E6 1D3A 5BB6
                      4AAC 5B8A 3D98 27AF 8427
sub  1024g/3E4DEF09 2004-10-18
```

- ID de la clé : **27AF8427**
- Empreinte ou *fingerprint* :
  856D 549D 29E6 1D3A 5BB6
  4AAC 5B8A 3D98 27AF 8427

Échange de clés

- Pour pouvoir communiquer, il faut disposer de la clé publique de son correspondant
- Exporter sa clé
  - `gpg --output moi.asc --armor --export foo@bar.org`
  - Graphiquement
- Importer une clé
  - `gpg --import copain.asc`
  - Graphiquement
- Peu pratique ...

Chiffrer, déchiffer un fichier

- Seahorse ne semble pas savoir le faire
- Chiffrer
  - `gpg --output doc.gpg --encrypt --recipient foo@bar.org doc`
- Déchiffrer
  - `gpg --output doc --decrypt doc.gpg`

Signer et vérifier

- Là aussi, Seahorse ne semble pas savoir faire
- Signer
  - `gpg --output doc.sig --detach-sig doc`
- Vérifier la signature
  - `gpg --verify doc.sig doc`

Serveurs de clés

- Pour faciliter les échanges de clés publiques, il existe des **serveurs de clé**, comme *pgp.mit.edu*
- *http://pgp.mit.edu*
- Envoyer sa clé sur un serveur
  - gpg --keyserver pgp.mit.edu
    --send-key moi@moi.org
- Récupérer une clé depuis un serveur
  - gpg --keyserver pgp.mit.edu
    --search-key foo@bar.org
  - Graphiquement

Signature, est-ce suffisant ?

- La signature est prévue pour identifier l'expéditeur
- Pourtant, n'importe qui peut créer une clé publique avec le nom de n'importe qui
- Solutions: **réseau de confiance** ou **tierce partie**
- Réseau de confiance: les utilisateurs **signent les clés** d'autres utilisateurs dans lesquels ils ont confiance
- http://www.cs.uu.nl/people/henkp/henkp/pgp/pathfinder/
- Repose sur le facteur humain, faillible

Signer une clé

- Rencontrer physiquement le détenteur de la clé et vérifier son identité. *Keysigning party* !
- Récupérer sa clé dans son trousseau, vérifier l'empreinte
- Signer la clé
  - gpg --edit-key toto@toto.com, commandes *fpr, sign, quit*
  - graphiquement
- Envoyer la signature sur les serveurs
  - gpg --keyserver pgp.mit.edu --send-key toto@toto.com
  - graphiquement

Mettre à jour son trousseau

- Pour récupérer les nouvelles signatures
  - gpg --keyserver pgp.mit.edu --refresh-keys
  - graphiquement

Courrier électronique

- Démonstration avec deux utilisateurs
  - Thomas Petazzoni <thomas.petazzoni@enix.org>
  - Monsieur Toulibre <monsieur.toulibre@free.fr>
- Sylpheed Claws Gtk 2
  - pour Thomas Petazzoni
  - avec le plugin PGP Mime
- Thunderbird
  - pour Monsieur Toulibre
  - avec le plugin Enigmail

Pour résumer

1) Créer sa clé
2) L'exporter sur un serveur de clé
3) L'utiliser pour signer ses courriers électroniques
4) Mettre l'empreinte sur sa carte de visite, et faire signer sa clé par d'autres utilisateurs

?

Toulibre

http://www.toulibre.org

contact@toulibre.org

Prochaines rencontres

- 20 Avril: Qjelt
- 27 Avril: Conférence de présentation des Logiciels Libres, Centre Culturel Bellegarde
- 17 Mai: Permanence Logiciels Libres aux Musicophages