Chapitre 3 : Outil de chiffrement PGP

SÉcurité et Cryptographie
2013-2014

Sup Galilée INFO3

PGP - Pretty Good Privacy

PGP (Pretty Good Privacy) est un système de chiffrement de messages électroniques inventé par Philip Zimmermann.

PGP est un système qui combine des fonctionnalités de la cryptographie à clef publique (algorithme RSA) et de la cryptographie à clef secrète (algorithme IDEA).

PGP - Pretty Good Privacy

Principe : Lorsqu'un utilisateur chiffre un texte avec PGP, les données sont en premier lieu compressées afin de réduire le temps de transmission, et d'économiser de l'espace mémoire. Ensuite, l'opération de chiffrement se fait principalement en deux étapes :

- PGP crée une clef secrète IDEA de façon aléatoire, et chiffre les données avec cette clef,

- PGP chiffre la clef secrète IDEA avec RSA et la clef publique du destinataire, puis transmet les données chiffrées au destinataire.

PGP - Pretty Good Privacy

Le déchiffrement s'opère comme suit :

- Le destinataire déchiffre la clef secrète IDEA avec sa propre clef privée RSA.

- Il déchiffre ensuite le message avec la clef secrète IDEA obtenue.

Plus formellement...

Notons pour F un algorithme de chiffrement, k_e une clef de chiffrement et k_d une clef de déchiffrement :

- $E_F(x, k_e)$ le résultat du chiffrement d'un message x par l'algorithme F et la clef k_e,

- $D_F(y, k_d)$ le résultat du déchiffrement d'un message y par l'algorithme F et la clef k_d.

En particulier, $D_F(E_F(x, k_e), k_d) = x$ si k_d est la clef de déchiffrement associée à la clef de chiffrement k_e.

La clef publique (pour un algorithme à clef publique) d'un individu A est notée k^A_{pub}, et sa clef privée k^A_{priv}. Notons que k^A_{pub} est alors la clef de chiffrement, et k^A_{priv} la clef de déchiffrement.

Plus formellement...

Supposons que A souhaite transmettre un message confidentiel x à B en utilisant PGP. Voici comment cela se déroule (notons que pour IDEA la clef de chiffrement et la clef de déchiffrement sont identiques) :

Phase de chiffrement

- Une clef secrète k IDEA est engéndrée aléatoirement par PGP.
- A calcule $y_1 = E_{IDEA}(x, k)$.
- A calule $y_2 = E_{RSA}(k, k^B_{pub})$.
- A transmet y_1 et y_2 à B.

Phase de déchiffrement

- B calcule $D_{RSA}(y_2, k^B_{priv}) = k$.
- B calcule $D_{IDEA}(y_1, k) = x$.

Et en pratique : GnuPG

GnuPG (Gnu Privacy Guard) est une implémentation open-source de PGP disponible sous Linux, Windows, et Mac OS.

En général GnuPG est installé nativement sous Linux. Il suffit de taper `gpg --help` dans un terminal pour s'en assurer et obtenir les options de GnuPG.

On va simplement s'intéresser au cas où un utilisateur A souhaite transmettre à un utilisateur B un message chiffré avec RSA (la phase de chiffrement avec IDEA est omise).

Pour générer ses clefs RSA : PGP fonctionne par paires de clefs publiques, privées. Pour générer ces deux clefs, il suffit de taper la commande `gpg --gen-key`.

Et en pratique : GnuPG

GnuPG vous demande le type de clef souhaité, puis sa longueur (entre 1024 et 4096 bits en fonction du niveau de sécurité voulu).

Une date d'expiration des clefs est demandée (vous pouvez taper 0 pour indiquer que les clefs sans date d'expiration).

Ensuite de quoi vous devez fournir au logiciel votre nom, votre adresse de messagerie électronique, et un commentaire libre.

Puis vous choisissez une phrase de passe pour accéder à votre clef.

Puis lors de la génération de la clef vous devez fournir du hasard : taper frénétiquement sur le clavier, bouger la souris dans tous les sens. Suite à cela GnuPG génère vos clefs RSA.

Et en pratique : GnuPG

Une paire de clefs prend alors la forme suivante :

```
pub 1024D/1258BA3D 2013-09-19[expire:  2014-09-18]
Empreinte de la clé=295C 13B3 0E27 5B17 8450 9DE60D 1258BA3D
```

L'identifiant de la clef est 1258BA3D.

Il est possible d'oublier sa phrase de passe. Dans ce cas il faut révoquer la clef. En prévision, il faut tout de suite générer un certificat de révocation par `gpg --output revoke.asc --gen-revoke <id de la clef>`

La clef pourra être révoquée en tapant : `gpg --import revoke.asc`

Penser à copier le certificat de révocation pour ne pas le perdre.

Et en pratique : GnuPG

Voyons maintenant comment chiffrer un message. Vous souhaitez envoyer un message à B. Il faut donc obtenir la clef publique (RSA) de B.

Comment publier une clef ? Il faut exporter votre clef précédemment générée par `gpg --output ma-clef.asc --armor --export <id de la clef>`

Vous obtenez alors un fichier ma-clef.asc commençant par BEGIN PGP PUBLIC KEY BLOCK et terminant par END PGP PUBLIC KEY BLOCK.

Pour publier sa clef, il suffit ensuite de copier votre fichier par exemple sur votre page web et sur des serveurs de clefs.

Et en pratique : GnuPG

Revenons à l'envoi d'un message chiffré à B. Une fois obtenue la clef de B, vous l'ajoutez à votre "trousseau de clefs" par `gpg --import clef-de-B.asc`

Cette clef pourra être utilisée autant que souhaitée pour communiquer avec B. Pour chiffrer le message :

`gpg --recipient <nom d'utilisateur de B> --encrypt <message>`

Le fichier message.gpg contiendra le résultat du chiffrement de votre message par la clef publique de B.

Pour déchiffrer un message que l'on vous a transmis : `gpg --decrypt message.gpg`

Et en pratique : GnuPG

Maintenant le contraire : prouver qu'un message vient bien de vous.

Pour signer un message : `gpg --sign message`