info614 : Mathématiques pour l'informatique
TP : signature électronique et chiffrement avec GPG

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : http://www.lama.univ-savoie.fr/~hyvernat/
wiki : http://www.lama.univ-savoie.fr/wiki

Partie -1 : consignes

Le but de ce TP est de vous familiariser avec la notion de signature électronique et de chiffrement clé publique / clé privée en utilisant le logiciel GPG (Gnu Privacy Guard : `http://www.gnupg.org/`). GPG est une version libre du logiciel PGP (Pretty Good Privacy : `http://www.pgp.com/`) crée par Philip Zimmermann.

Pour ce TP, vous devrez m'envoyer deux mails ainsi qu'un petit rapport.

TOUS les emails que vous m'enverrez devront avoir comme sujet "`[info-614] TP` *???*", où les *???* peuvent être n'importe quoi.

Le rapport devra être un fichier texte (ou un fichier pdf). Les rapports envoyés dans un autre format (OpenOffice, rtf, Word) ne seront pas lus. (Ce n'est pas une blague.)

Faites attention lors de la rédaction du rapport : je ne veux pas lire la suite des commandes que vous avez utilisées ; et je ne veux pas lire un roman fleuve de 40 pages !

Vous pouvez travailler en binomes pour le rapport ; mais chacun doit m'envoyer les 2 mails (signés et cryptés) : seul le rapport ne sera fait qu'en un seul exemplaire.

Partie 0 : préliminaires

▷ Pour commencer le TP, redémarrez votre machine sous Linux et identifiez vous.

Dans un terminal, vérifiez que le logiciel est bien installé avec la commande "`$ gpg --version`"

▷ Installez l'extension Firefox "FireGPG"

`http://getfiregpg.org/`

pour pouvoir utiliser facilement GPG à partir de Firefox. (À partir du webmail par exemple...)

Je vous conseille cependant d'utiliser l'interface en ligne de commandes dans un premier temps.

Partie 1 : GPG et la cryptographie symétrique

Vous pouvez utiliser GPG pour faire une cryptographie symétrique (si vous partagez une clé avec votre destinataire) avec la commande

- "`$ gpg --symmetric` *fichier*" : ceci créera un fichier binaire *fichier.gpg* contenant le fichier chiffré,
- "`$ gpg --symmetric --armour` *fichier*" : ceci créera un fichier ASCII *fichier.asc* contenant le fichier chiffré,

▷ Essayez cette commande pour envoyer un message crypté à votre voisin. (Vous pouvez échanger votre clé secrète à l'oral.)

▷ Décryptez le fichier chiffré avec la commande "`$ gpg --decrypt` *fichier.asc*" (ou "*fichier.gpg*").

Partie 2 : création des clés, stockage des clés

Question 1. Création de votre clé publique/clé privée.

▷ Dans un terminal, vérifiez s'il existe un répertoire "`$HOME/.gnupg/`". (S'il existe, regardez les fichiers qu'il contient ainsi que leurs dates de de modification.

▷ Pour créer votre propre clé publique/clé privée, il faut utiliser la commande

"`$ gpg --gen-key`"

- Créez vos clés en acceptant les choix par défaut, sauf pour la durée de validité de vos clés : comme il s'agit d'un premier essai, je vous conseille de ne créer qu'une clé temporaire (15 jours). Vous pourrez toujours recréer des clés quand vous vous serez familiariser avec le fonctionnement GPG...
- Mettez votre vrai nom (ou au moins vos initiales) et choisissez "info-614" comme commentaire. Choisissez une adresse email que vous consultez régulièrement...
- Choisissez une "passphrase" sûre et dont vous vous rappellerez... Elle vous servira à chaque fois que vous aurez à utiliser votre clé privée.

Remarque : si GPG vous dis qu'il n'a pas assez d'entropie, vous pouvez utiliser la commande "`$ find / -name a`" dans un autre terminal et attendre un peu...

▷ Allez vérifier l'existence du fichier "`$HOME/.gnupg/`" et des fichiers qu'il contient (ou comparez les dates de modification des fichiers).

▷ Pour vérifiez que les clés ont bien été créées, utilisez la commande "`$ gpg --list-keys`". Vous devriez obtenir quelque chose du genre

```
pub   1024D/7383EB1A 2008-05-12 [expires: 2008-05-27]
uid                  Pierre .H. (info-614) <pierre.hyvernat@univ-savoie.fr>
sub   2048g/B727F99A 2008-05-12 [expires: 2008-05-27]
```

qui vous indique que vous avez une clé principale (ligne "`pub`") qui expire le 12 mai ; et une sous-clé (ligne "`sub`") qui expire aussi le 12 mai. La ligne "`uid`" vous donne l'identité de l'utilisateur correspondant.

La clé principale est utilisé pour les signatures, et la sous-clé pour le chiffrement.

Remarque : lors de la création d'une clé de plus longue durée, il est impératif de créer des certificats de révocation avec "`$ gpg --output revoke.txt --gen-revoke` *uid*". C'est ça qui vous permettra de faire savoir que votre clé ne doit plus être utilisée... (Si vous perdez votre clé privée par exemple.)

Attention : votre clé secrète doit rester secrète. Si quelqu'un y a accès, il peut usurper votre identité et lire les messages chiffrés qui vous sont adressés. Votre passphrase doit en garantir la sécurité, car c'est la seul protection que vous avez si quelqu'un peut accéder à votre compte... Choisissez donc une passphrase sûre, et ne la dévoilez à personne.

Question 2. Échange de clés.

Pour envoyer votre clé publique à quelqu'un, vous pouvez commencer par l'exporter avec la commande

- "`$ gpg --output` *fichier.asc* `--export --armour` *uid*" (le fichier *fichier.asc* contiendra la clé en ASCII)
- "`$ gpg --output` *fichier.gpg* `--export` *uid*" (le fichier *fichier.gpg* contiendra la clé en binaire)

Pour importer une clé (en binaire ou en ASCII) contenue dans le fichier *fichier.gpg*, il suffit d'utiliser la commande "`$ gpg --import` *fichier.gpg*".

▷ Échangez votre clé avec votre voisin.

Question 3. Serveur de clés.

▷ On va maintenant envoyer notre clé à un *serveur de clés* pour que tout le monde puisse y avoir accès. Utilisez la commande

"`$ gpg --keyserver pgp.mit.edu --send-key 0x`*nnnnnnnn*"

où *nnnnnnnn* est le numéro de votre clé (les chiffres apparaissant après le "`1024D` sur la première ligne du résultat de `gpg --list-keys` ; dans mon cas, c'est 7383EB1A...)

Remarque : cette commande peut prendre un peu de temps.

▷ En allant sur la page web `http://pgp.mit.edu/`, vérifiez que votre clé a bien été rajoutée.

Question 4. Serveurs de clés, empreintes.

Vous pouvez maintenant récupérer des clés sur le serveur `http://pgp.mit.edu/` : il suffit de récupérer le numéro de la clé, et d'utiliser la commande

"`$ gpg --keyserver pgp.mit.edu --recv-key 0x`*nnnnnnnn*"

▷ Recherchez ma clé, et importez la grâce à la commande précédente. (Vous pouvez aussi en profiter pour importer d'autres clés...)

Pour garantir que vous avez bien récupéré la bonne clé, et pas celle de quelqu'un qui essaie de se faire passer pour moi, je vous donne *l'empreinte* de ma clé :

`E880 1531 9828 AF61 24D5  25CB 82B2 7B5B A173 3724`

▷ En utilisant la commande "`$ gpg --fingerprint`", vérifiez que vous avez bien récupéré ma clé.

▷ Chacun d'entre vous doit me donner, en main propre (c'est à dire en venant l'écrire sur le papier que j'aurais emmené) son empreinte de clé.

Partie 3 : Signature et chiffrement

Question 1. Signature électronique.

Pour signer un fichier texte, vous pouvez utiliser les commandes

- "`$ gpg --clearsign` *fichier*" : cela crée un fichier "`.asc`" qui contient le fichier original, plus la signature,
- "`$ gpg --detach-sign` *fichier*" qui crée un fichier "`.sig`" qui contient juste la signature (en binaire).

Pour un fichier texte, la première méthode est préférable*.

▷ Quelle est la clé utilisée pour une signature ?

Envoyez moi la réponse dans un petit email signé en utilisant le webmail : il suffit de faire un copier-coller du contenu du fichier "`.asc`".

Pour vérifier une signature, il suffit d'utiliser la commande "`$ gpg --verify` *fichier*" où *fichier* est le fichier signé ("`.asc`") ou la signature ("`.sig`").

▷ Vérifiez la signature d'un message que votre voisin vous enverra.

Que se passe-t'il si vous modifiez le message après l'avoir signé ?

Question 2. Chiffrement.

Pour crypter un document, il faut utiliser la commande

- "`$ gpg --encrypt` *fichier*" pour obtenir un fichier binaire *fichier.gpg* contenant le fichier original chiffré,

* Sauf si c'est un email, et que votre logiciel de messagerie gère les signatures en pièce jointe.

- "`$ gpg --encrypt --armour` *fichier*" pour obtenir un fichier ASCII *fichier.asc* contenant le fichier original chiffré,

Pour un fichier texte envoyé par email, la seconde méthode est préférée.

▷ Quelle est la clé utilisée lors du chiffrement ?

Envoyez moi un mail crypté pour me donner la réponse.

Pour décrypter un document, il suffit d'utiliser la commande "`$ gpg --decrypt` *fichier.gpg*"

▷ Testez le décryptage avec votre voisin...

Partie 4 : contre-signature des clés, réseau de confiance

Question 1. Contre-signature d'une clé.

Lorsque vous êtes sûr qu'une clé est correcte (càd qu'elle appartient bien à son propriétaire), vous pouvez y apposer votre signature : c'est un moyen de dire "*j'ai vérifié cette signature*".

Pour faire ça, on utilise la commande "`$ gpg --sign-key` *uid*".

Attention : il ne faut signer une clé que lorsqu'on est sûr de sa correction ! (En vérifiant son empreinte par exemple.)

Vous pouvez ensuite renvoyer la clé signée à son propriétaire, pour qu'il puisse faire savoir que vous l'avez contre-signée.

Encore mieux, vous pouvez la renvoyer sur un serveur de clés avec la commande

"`$ gpg --keyserver pgp.mit.edu --send-key` *uid*"

▷ Contre-signez les clés que vous avez vérifiées, en commençant par la mienne ; répercutez tous ces changements sur le serveur de clés `pgp.mit.edu`.

Question 2. Confiance et réseau de confiance.

▷ En allant lire les parties 3.2, 4.1.4 et 4.2 du manuel de GPG :

`http://www.gnupg.org/gph/fr/manual.html`,

décrivez succinctement le concept de *réseau de confiance*.

Donnez les intérêts de cette chose là, et discutez.

▷ Envoyez moi votre rapports en pièce jointe d'un email. Votre rapport (le fichier) devra être signé et crypté.

Le corps de votre email devra comporter :
- les noms et prénoms des membres du groupes
- les passphrase que vous avez choisies
- des commentaires sur le TP et/ou le cours en général

En plus du rapport, mettez également en pièce jointe vos clés privées obtenue avec la commande

"`$ gpg --output` *nom.asc* `--export-secret-key --armour` *uid*"