Faculté des Sciences de Limoges 2002–2003
DESS Sécurité de l'Information
Cryptographie

Travaux pratiques, séance 1

Utilisation de GnuPG

1. GnuPG

1. — Télécharger GnuPG, l'implémentation du standard OpenPGP par GNU :

```
ftp://ftp.gnupg.org/pub/gcrypt/gnupg/gnupg-1.2.1.tar.gz
```

2. — Compiler GnuPG :

```
$ tar -xzf gnupg-1.2.1.tar.gz
$ cd gnupg-1.2.1
$ ./configure
$ make
$ make check
$ su
$ make install
```

3. — Générer une paire de clés (publique/privée) par la commande `gpg --gen-key`. Choisir une longueur de 1024 bits et une `passphrase` d'entropie suffisante (au minimum 60 bits).

Utiliser la commande `gpg --list-keys` pour voir les clés publiques obtenues et `gpg --list-secret-keys` pour voir les clés privées. La commande `gpg --gen-key` génère en fait par défault deux paires de clés : une clé DSA (D) pour signer et une clé El Gamal (g) pour chiffrer. Par précaution, on peut aussi préparer un certificat de révocation (qui pourra être utile si votre clé secrète est détruite ou que vous oubliez la passphrase) par la commande `gpg --gen-revoke --armor untel > revoke.asc`.

4. — Extraire votre clé publique (en fait, les deux clés publiques) dans un fichier avec, par exemple,

```
$ gpg --export --armor untel > mykey.asc
```

Echanger par mail vos clés publiques. Importer les clés reçues.

```
$ gpg --import keys.asc
```

Refaire `gpg --list-sigs` pour voir les clés importées. Vérifier les empreintes (extraites en utilisant la commande `gpg --fingerprint untelautre`) visuellement auprès des personnes qui vous ont envoyé leurs clés et signer ensuite ces clés avec la vôtre :

```
$ gpg --sign-key untelautre
```

Essayer alors les commandes `gpg --list-sigs` et `gpg --check-sigs`.

5. — Chiffrer des fichiers avec votre clé publique

```
$ gpg -r untel --encrypt fichier-clair
```

Vous pouvez préciser le(s) destinataire(s) avec l'option -r, ou bien en répondant aux questions.

Déchiffrer avec votre clé secrète (`gpg secretfile.gpg`) et vérifier que le fichier obtenu est identique au clair original.

Chiffrer des fichiers avec la clé publique d'un ou plusieurs correspondant(s). Remarque : lorsque vous chiffrez un fichier avec la clé publique d'autres destinataires, vous n'aurez pas la possibilité de déchiffrer vous-même le fichier obtenu, sauf si vous vous êtes inclus dans la liste des destinataires. Envoyer le(s) fichier(s) chiffré(s) au(x) destinataire(s) (en attachement, ou utiliser l'option `--armor` lors du chiffrement).

Déchiffrer avec votre clé secrète les messages reçus.

6. — Signer des fichiers (option `--sign`) avec votre clé secrète. Vérifier la signature et restaurer le fichier original (`gpg signfile.gpg`). Comparer avec la signature "en clair" (option `--clearsign`). Envoyer des fichiers signés à vos correspondants. Vérifier les signatures des fichiers reçus. Fabriquer aussi des signatures "détachées" (option `--detach-sign`). Faire `man gpg` pour avoir la syntaxe exacte de toutes ces options.

7. — Echanger vos certificats. Il suffit pour cela de ré-extraire les clés qu'on vous a envoyées, elles seront alors accompagnées des certificats que vous avez créés. Renvoyer le tout aux propriétaires des clés.

8. — Combiner chiffrement et signature.

9. — Récupérer la clé de Werner Koch (numéro `57548DCD`) sur le serveur `www.keyserver.net`. Vérifier la signature de `gnupg-1.2.1.tar.gz` (disponible sur le site de GnuPG).

10. — Déposer votre clé publique (avec d'éventuelles signatures) sur le serveur `www.keyserver.net` :

```
$ gpg --keyserver www.keyserver.net --send-keys untel
```

(Ou bien exporter votre clé dans un fichier et copiez-collez le sur le site du serveur).

11. — Signer la clé `EA893E0A` de François Arnault trouvée sur le serveur `www.keyserver.net` (empreinte complète : 09E6 8C0F B35D D187 94D5 0DAE A3E0 A2F1 EA89 3E0A). Examiner le résultat avec la commande `gpg --list-sigs`. Extraire la clé et ses certificats en faisant `gpg --export --armor arnault > arnault.asc`. Examiner le contenu du fichier par `gpg --verbose arnault.asc`. Envoyer les certificats à `arnault@unilim.fr`, ou déposez-les sur le serveur de clés (merci).

12. — Examiner et éditer les valeurs de confiance de votre clé personnelle et des clés publiques que vous avez recueillies, avec la commande `gpg --edit-key untel` et diverses sous-commandes (`help`, `trust`, ...).

Deux caractères séparés par un `/` décrivent les valeurs de confiance associées à chaque clé. Ceux de droite indiquent si l'origine prétendue de la clé correspondante doit être considérée comme exacte. Leur valeur par ordre de confiance décroissante peut être `u`, `f`, `m`, `eq-` (non calculée), `n` (négative). Il sont calculés par GnuPG.

Ceux de gauche indiquent à quel point les certificats que cette clé signe doivent être pris au sérieux. Leur valeur peut être `f`, `m`, `q`, `n`, par ordre de crédit décroissant. Elles peuvent être modifiées à l'aide de la sous-commande `trust` et servent à calculer les caractères de droite.

Lorsque GnuPG utilise une clé publique, il calcule sa valeur de confiance et, si elle n'est pas totale (`u` ou `f`) vous prévient que l'authenticité de la clé n'est pas certaine.

Référence

Le "GNU Privacy Handbook" : `www.gnupg.org/gph/en/manual.{htm,pdf}`

2. pgp4pine

1. — Récupérer l'interface `pgp4pine` sur le serveur `http://pgp4pine.flatline.de`.

2. — Installer :

```
$ tar -xzf pgp4pine-1.76.tar.gz
$ cd pgp4pine-1.76
$ ./configure
$ cp stamp-h stamp-h.in
$ make
$ make install
$ cp /usr/local/doc/pgp4pine/pgp4pinerc.example ~/.pgp4pinerc
```

Recopier le fichier `/usr/local/doc/pgp4pine/pgp4pinerc.example` en `~/.pgp4pinerc` et éditer le pour qu'il convienne à vos besoins. En particulier, il doit contenir les lignes suivantes :

```
profile_list=gpg
profile_gpg_tmpfile=~/mail/pgp4pine.tmp
profile_gpg_my_address=untel@truffe.unilim.fr
profile_gpg_gpgbin=/usr/local/bin/gpg
```

Dans le menu `Setup` de `pine`, choisir `Config` et indiquer (près de la fin de liste) :

```
display-filters = _BEGINNING("-----BEGIN PGP")_ /usr/local/bin/pgp4pine -d -i _TMPFILE_
sending-filters = /usr/local/bin/pgp4pine -e -i _TMPFILE_ -r _RECIPIENTS_
```

3. — Utiliser `pine` pour envoyer des messages signés et/ou chiffrés. Pour cela, après avoir fait “Send” (`^X`), faire “Next Filter” (`^N`) et répondre aux questions.