TP1 - Sécurité

Loïc MAYOL

8 décembre 2018

Fonctionnement de GnuPG

GnuPG est une implémentation libre du standard **OpenPGP** définit dans la RFC4880 [1]. GnuPG nous permet de chiffrer et de signer nos communications et est disponible dans tous les systèmes d'exploitations GNU/Linux. Il peut très bien être utilisé grâce à une interface graphique ou bien directement en ligne de commande [2]. D'après le `man` de Linux [3], il existe deux versions principales à GnuPG :

- GnuPG 2.x qui supporte les algorithmes de chiffrement dit "modernes",
- GnuPG 1.x, toujours utilisé si notre machine ne supporte pas GnuPG 2.x ou bien si l'on souhaite utiliser des méthodes "périmées".

En tant qu'implémentation libre de OpenPGP, GnuPG propose les mêmes fonctionnalités que celui-ci, soit :

- signature numérique (digital signature),
- chiffrement.

Fonctionnement du chiffrement

GnuPG combine donc le chiffrement symétrique des clés et le chiffrement de clé publique (chiffrement asymétrique). Lors du chiffrement d'un message, le message est d'abord chiffré en utilisant un algorithme de chiffrement symétrique. La clé utilisée par l'algorithme de chiffrement est une "clé de session" générée aléatoirement. Cette clé est chiffrée avec la clé publique du destinataire puis transmise avec le message. Chaque "clé de session" est unique et n'est utilisée qu'une seule fois.

Fonctionnement de la signature

La signature numérique utilise un hashé et un algorithme de chiffrement asymétrique. Le fichier que l'on souhaite signé est hashé en utilisant notre clé privée afin de générer la signature qui sera attachée au fichier. Lorsque le fichier signé sera vérifié, il faudra aussi vérifier la signature afin d'en certifier son authenticité.

Services de GnuPG

Lors du TP1, nous avons utilisé la commande `gpg`. Il est donc possible de générer des clés grâce à l'option `--gen-key`. Durant la génération, un nom, une adresse email et une passphrase nous est demandé afin d'identifier et construire la clé. De plus, afin d'augmenter l'entropie, il nous est aussi demandé d'utiliser le clavier, de bouger la souris ou bien d'utiliser le disque dur. Les clés publiques et privées sont alors placées dans le répertoire caché `.gnupg/`. Il nous est aussi possible de consulter nos clés en utilisant l'option `-k` (ou bien avec `--list-key`).

En cas de perte de passphrase ou si notre sécurité est compromise, on peut annuler la validité de la paire de clés crées. Pour cela, on utilise un certificat de révocation que l'on peut créer grâce à l'option `--gen-revoke`. Une fois le certificat créé, il suffira d'utiliser `--import` pour révoquer la clé (on utilisera la même option pour importer les clés de nos camarades).

Il est possible de partager notre clé publique grâce à l'option `--export`, on peut ajouter l'option `--armor` afin que notre clé soit générée en ASCII-armor (ou Radix-64 encoding [1]) (Listing 1) et non en binaire nous permettant ainsi de la partager facilement dans un mail.

```
$ m14009868@L-024110A009-11:~$ gpg --export --armor

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQGiBFwHymsRBACS7w7bHmtqx/rkLFBpcrhoyKqak3vinAH66Ni/CUYk3Ig+GuVh
cDxkfjoy4J1SPvuEu3HZ81ZsF7lLJ61j3U/gaubQHrlo2Z/SjVx/R6I5LVNJYaLn
[...]
QyQCbmJjG0qISQQYEQIACQUCXAfKlgIbDAAKCRBZ5GjN4nYHnOyqAJ9o9epVRHmN
dvTcoPQgkuSJZS52eACgqtUjll8i21B6CsssbqjlLEspflQ=
=z5Fo
-----END PGP PUBLIC KEY BLOCK-----
```

Listing 1 – Clé publique PGP

Pour chiffrer un fichier, on peut utiliser l'option `--encrypt` en précisant le destinataire grâce à `--recipient`. Comme précedemment, l'utilisation de `--armor` nous permettra d'avoir des caractères lisibles en ASCII-armor. Notre destinataire utilisera l'option `--decrypt` afin de déchiffrer le message, il pourra même utiliser `--output` pour préciser le fichier dans lequel il doit l'enregistrer.

Il est aussi possible de signer un fichier en utilisant l'option `--clearsign` et de vérifier celle-ci grâce à `--verify` (Listing 2) en précisant le fichier de notre signature.

```
$ m14009868@L-024110A009-11:~$ gpg --verify essai.txt.asc

gpg: Signature faite le ven. 07 déc. 2018 16:26:31 CET avec la clef DSA d'identifiant
     ACC6841A
gpg: Bonne signature de   Frank   RIBERY (La routourne) <frank.ribery@etu.univ-amu.fr
    >
```

Listing 2 – Vérification de signature

De plus, des serveurs de clés publiques existent afin que l'on puisse échanger nos clés. Il suffit juste de charger la clé sur le serveur et n'importe qui peut la récupérer et l'utiliser afin de chiffrer un message. Des sites proposent de nous simplifier la tâche en chargeant le fichier contenant notre clé publique directement sur ceux-ci et nous permettant une recherche facile [4].

Références

[1] Callas, et al (2007) **RFC4880 - OpenPGP Message Format**, IETF.
Disponible sur : https://www.ietf.org/rfc/rfc4880.txt.

[2] Contributeurs (2018) **GNU Privacy Guard**, Wikipédia.
Disponible sur : https://www.wikiwand.com/fr/GNU_Privacy_Guard.

[3] GnuPG (2017) **GnuPG - gpg man page**, GnuPG Projet.
Disponible sur : https://www.gnupg.org/documentation/manpage.html

[4] MIT (2018) **MIT PGP Key Server**, MIT.
Disponible sur : https://pgp.mit.edu/