Créer une clé GPG

À quoi ça sert ?

- À chiffrer un message :
  - **confidentialité** : le contenu du message est protégé des yeux indiscrets par chiffrage à clé publique, seul le destinataire peut le déchiffer grâce à sa clé privée ;
  - **contrôle d'intégrité** : le message lu est bien celui qui a été envoyé, sans modification ni substitution ;
- À signer numériquement :
  - **authentification** : le signataire est bien celui qui a envoyé le message, et il ne peut pas le nier (non répudiation).

Comment ça marche ?

Deux usages principaux, complémentaires : signature et chiffrage

La clé privée sert à signer un message. Lorsque je signe un message avec ma clé secrète, que je suis le seul à posséder et qui est protégée par la passphrase, GnuPG en crée un résumé qui, grâce à ma clé publique, garantit au destinataire que je suis bien le signataire du message, que je suis bien celui que je prétends être.

La clé publique sert à chiffrer le contenu d'un message, à l'image d'un cadenas ouvert. Je chiffre un message avec la clé publique de mon correspondant (je «ferme son cadenas» sur mon message). Lui seul pourra le déchiffrer («ouvrir le cadenas») avec sa clé privée et sa passphrase.

Génération de la paire de clés

Lancer le terminal et taper la commande suivante :

```
gpg --gen-key
```

- Choisissez un type de clé –> 1
- Choisissez la longueur de la clé –> 2048
- Choisissez la durée de validité de la clé, par exemple 5y pour 5 ans.
- Confirmez
- Remplissez les champs demandés :

    - Nom Prénom
    - courriel@fai.domaine
    - commentaire (par exemple votre pseudo)
- Validez
- Entrez une phrase secrète (on ne voit pas ce que l'on tape) à deux reprises pour la confirmer sans erreur.

La phrase secrète doit être complexe, pas trop courte, difficile à deviner mais aisée à mémoriser. Elle sera demandée à chaque processus de signature ou de chiffrement. Une bonne “passphrase”, assez longue, mélangera des lettres majuscules et minuscules, des chiffres et des symboles non alphanumériques.

On peut prendre par exemple un poème (un facile à mémoriser pour vous) et n'utiliser que les initiales et ponctuations. Si l'on veille à ajouter des chiffres, le résultat est en général un mot de passe de qualité.

Ex. le début du poème Deux oiseaux de Jean-Paul Labaisse :

```
Ce sont deux oiseaux blancs qui volent dans le ciel,
Très haut, très loin, au-delà des mortes clairières,
```

donne l'excellente passphrase : *Cs2obqvdlc,Th,tl,a-ddmc,*

On peut aussi utiliser une “vraie” phrase, ordinaire, avec les majuscules, les espaces et la ponctuation, mais attention à être bien en mesure de la taper sans erreur…

La génération de la paire de clés commence.

Un grand nombre d'octets aléatoires doivent être générés, il est préférable que le PC “travaille” pendant ce temps pour lui permettre d'en disposer suffisamment, donc tapez (n'importe quoi) au clavier, bougez la souris… ou, mieux, ouvrez un autre terminal sur la machine pour lancer une commande “gourmande” comme une recherche de fichier sur tout le disque par exemple :

```
find / -name abcdefg.
```

Vérifions en faisant la liste des clés

```
gpg --list-keys

pub   rsa4096/0x4374F0E466582E03 2017-08-26 [SC]
      19C298B226095DD219ED6E544374F0E466582E03
uid                  [  ultime ] Jean Peyratout <jean.peyratout@sud-
ouest.org>
uid                  [  ultime ] Jean Peyratout (clé 4096 bits 2017)
<jean.peyratout@abul.org>
uid                  [  ultime ] Jean Peyratout <jean@sud-ouest.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@free.fr>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@aful.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@laposte.net>
```

```
uid                  [  ultime ] Jean Peyratout <jean.peyratout@ac-
bordeaux.fr>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@scideralle.org>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@terredadeles.org>
uid                  [  ultime ] [jpeg image of size 14964]
sub   rsa4096/0xBF92F66B91148D95 2017-08-26 [E]
```

Ma clé publique a été créée, et elle possède l'identifiant 66582E03. Cet ID n'a que 8 caractères, il existe un risque de confusion avec celui d'une autre clé publique. Aussi, pour limiter le risque d'erreur, on préfère utiliser l'*empreinte digitale* de la clé, appelée *fingerprint*. C'est cette *fingerprint* que l'on échangera avec ses correspondants du réseau de confiance.

Affichons la *fingerprint* avec la commande **gpg –fingerprint**

```
gpg --fingerprint
/home/jean/.gnupg/pubring.gpg
------------------------------
ub   rsa4096/0x4374F0E466582E03 2017-08-26 [SC]
      Empreinte de la clef = 19C2 98B2 2609 5DD2 19ED  6E54 4374 F0E4 6658
2E03
uid                  [  ultime ] Jean Peyratout <jean.peyratout@sud-
ouest.org>
uid                  [  ultime ] Jean Peyratout (clé 4096 bits 2017)
<jean.peyratout@abul.org>
uid                  [  ultime ] Jean Peyratout <jean@sud-ouest.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@free.fr>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@aful.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@laposte.net>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@ac-
bordeaux.fr>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@scideralle.org>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@terredadeles.org>
uid                  [  ultime ] [jpeg image of size 14964]
sub   rsa4096/0xBF92F66B91148D95 2017-08-26 [E]
```

Voilà l'empreinte de ma clé publique : 19C2 98B2 2609 5DD2 19ED 6E54 4374 F0E4 6658 2E03 Elle se termine par les 8 caractères 66582E03 vus précédemment.

Notez votre empreinte, elle vous servira plus tard pour échanger avec vos partenaires dans le réseau de confiance. On note en général sur une carte de visite ou une languette de papier la *fingerprint* de sa clé publique pour pouvoir la donner facilement et éviter les erreurs de retranscription manuelle.

GPG avec des outils graphiques

On peut aussi utiliser un logiciel graphique pour gérer ses clés GPG, comme Enigmail (extension du courrielleur Thunderbird, c'est celui que j'utilise au quotidien), Seahorse, GnomePGP, GPA, KGpg…

Qu'est-ce qui a été généré ?

Nous avons généré deux clés interdépendantes et ne fonctionnant pas l'une sans l'autre.

- L'une d'elles est la **clé publique**, que l'on va mettre à disposition de tout le monde en la téléversant sur un serveur de clefs.
- L'autre est la **clé secrète**, qu'on doit absolument être le seul à conserver et qui est protégée par la passphrase.

Ces clés sont enregistrées par défaut dans le répertoire .gnupg de l'utilisateur ~/.gnupg/. À l'intérieur de ce répertoire se trouvent deux fichiers, pubring et secring, avec respectivement les clés publiques et secrètes.

Pour utiliser une image, on peut dire que la clé publique est un cadenas ouvert, mis à la disposition de tous. N'importe qui peut le verrouiller sans votre intervention. Mais seule la clé privée correspondante, celle que vous êtes seul(e) à posséder, est capable d'ouvrir ce cadenas.

Générer les certificats de révocation

Avant d'aller plus loin, il faut créer des certificats de révocation. Si votre clé s'avérait compromise, perdue ou périmée, ce certificat de révocation vous permettra de le signaler à tous (*mon_id* est l'identifiant à 8 caractères vu au début, mais on peut aussi utiliser le nom ou le pseudo choisi).

La commande est la suivante :

```
gpg --gen-revoke mon_id
```

Pour récupérer directement la clé de révocation sous un fichier revoque.macle que l'on pourra garder en sécurité hors de son ordinateur :

```
gpg --gen-revoke mon_id >.gnupg/revoque.macle
```

Révoquer une clé avec un certificat de révocation

Pour révoquer une clé, il faut donc avoir généré un certificat de révocation comme indiqué ci-dessus.

On commence par importer le certificat :

```
gpg --import /.gnupg/revoque.macle
```

Puis on envoie la clé de révocation sur un serveur :

```
gpg --send-key mon_id
```

Publier la clé publique sur un serveur de clefs

Une fois vos clés générées, il faut stocker votre clé publique sur un serveur de clés pour que vos interlocuteurs puissent la trouver s'ils souhaitent vous envoyer un courriel chiffré.

Nous allons utiliser le serveur http://pgp.mit.edu du MIT (*Massachusetts Institute of Technology*), il y en a bien d'autres, notamment http://www.pgp.net/pgpnet/wwwkeys.html ou encore https://keys.openpgp.org. La plupart des serveurs de clés GPG/PGP sont interconnectés et se réactualisent mutuellement au bout d'un certain temps.

```
gpg --keyserver pgp.mit.edu --send-keys  mon_id
```

Après quelques heures la clé publique sera répliquée sur les autres serveurs de clés.

Réseaux de confiance

Au-delà des aspects techniques, le cœur de l'utilisation des clés GPG est leur authenticité, garantie par le réseau de confiance.

Lorsqu'on rencontre quelqu'un avec qui on veut avoir des échanges utilisant les clés GPG, on échange ses *fingerprint*.

L'échange doit être précédé par la **vérification de l'identité réelle des personnes grâce aux papiers officiels d'identité**. C'est indispensable ! Puis on échange les *fingerprint*.

Une fois la *fingerprint* de votre correspondant récupérée, on télécharge sa clé publique à partir d'un serveur de clés en utilisant les huit derniers caractères de l'empreinte, ici la mienne sur le serveur pgp.mit.edu :

```
gpg --keyserver pgp.mit.edu --recv-keys 66582E03
```

Relancer la commande gpg –fingerprint pour afficher la *fingerprint* correspondant à la clé que l'on vient de télécharger sur le serveur.

```
gpg --fingerprint 66582E03

ub   rsa4096/0x4374F0E466582E03 2017-08-26 [SC]
      Empreinte de la clef = 19C2 98B2 2609 5DD2 19ED  6E54 4374 F0E4 6658
2E03
uid                  [  ultime ] Jean Peyratout <jean.peyratout@sud-
ouest.org>
uid                  [  ultime ] Jean Peyratout (clé 4096 bits 2017)
<jean.peyratout@abul.org>
uid                  [  ultime ] Jean Peyratout <jean@sud-ouest.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@free.fr>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@aful.org>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@laposte.net>
uid                  [  ultime ] Jean Peyratout <jean.peyratout@ac-
```

```
bordeaux.fr>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@scideralle.org>
uid                  [  ultime ] Jean Peyratout
<jean.peyratout@terredadeles.org>
uid                  [  ultime ] [jpeg image of size 14964]
sub   rsa4096/0xBF92F66B91148D95 2017-08-26 [E]
```

Il ne reste plus qu'à comparer le résultat obtenu à l'écran avec l'empreinte indiquée sur la carte de visite. Si les deux concordent, alors vous êtes vraiment sûr·e que la clé publique qui se trouve maintenant dans votre trousseau est bonne, qu'elle correspond à la personne avec laquelle l'échange physique des cartes et la vérification des papiers d'identité ont été faits.

Vous pouvez “signer” cette clé publique avec votre propre clé.

Éditons la clé reçue, avec l'id (8 caractères) ou le nom. Les deux commandes

gpg - -edit-key peyratout ou

gpg - -edit-key 66582E03 sont équivalentes.

```
gpg --edit-key son_id
 blahblah nom, ID, date, etc.
Commande>
```

La signature se fait en deux parties :

- trust pour indiquer votre niveau de confiance en la personne.
- sign pour indiquer comment la vérification de l'identité a été faite.

```
Commande> trust
Décidez maintenant à quel point vous avez confiance en cet utilisateur
pour qu'il vérifie les clés des autres utilisateurs (vous pouvez
vérifier son passeport, vérifier les empreintes de plusieurs sources
différentes, etc.)
  1 = ne sais pas ou ne dirai pas
  2 = je ne fais PAS confiance
  3 = je crois marginalement
  4 = je fais entièrement confiance
  5 = je donne une confiance ultime
  m = retour au menu principal
Votre décision ?
```

Indiquez comment vous avez confiance en cet utilisateur (pour le fait de vérifier avec soin l'identité de ses correspondants GPG).

Puis passons à la façon dont s'est déroulée la vérification de la clé

```
Commande> sign
Signer vraiment tous les noms d'utilisateurs ? (o/N)
```

Indiquez avec quel soin l'identité de la personne a été vérifiée. Entrez votre choix, puis entrez votre passphrase pour signer la clé.

La commande **save** permet d'enregistrer les changements.

La commande **quit** permet de sortir du mode interactif. Si des changements n'ont pas été enregistrés, GnuPG propose de le faire avant de quitter.

On peut aussi utiliser la commande :

```
gpg --sign-key son_id
```

Vous avez maintenant signé la clé de votre correspondant, vous pouvez donc exporter sa clé publique pour signaler au reste du monde que vous l'avez signée.

```
gpg --keyserver pgp.mit.edu --send-keys
```

Ainsi ceux qui font confiance à votre clé pour savoir que vous vérifiez avec attention l'identité de ceux auxquels vous signez leur clé GPG pourront également faire confiance à leurs clés, *modulo* les niveaux de confiance que vous avez vous-même accordés.

Attention, l'appartenance au réseau de confiance n'est pas une garantie de bonne foi, c'est seulement un indice de validité de l'identité de la personne. Le nombre de signatures d'une clé n'est pas important, c'est la qualité de ceux qui se sont engagés en la signant qui compte.

Il faut de temps en temps mettre à jour son trousseau de clés publiques pour savoir si certaines ont été signées ou révoquées.

```
gpg --keyserver pgp.mit.edu --refresh-keys
```

Des *keysigning parties* sont organisées à l'occasion de rencontres pour étendre le réseau de confiance. À cette occasion on échange ses *fingerprint* (d'où l'intérêt de la carte de visite ou de la bandelette de papier préparée à l'avance) après avoir vérifié avec soin l'identité de ses interlocuteurs, y compris quelqu'un qu'on côtoie depuis longtemps. Par commodité de langage on parle souvent d'"échange de clés".

Ajouter des identités à une clé

À ce niveau vous avez pu vous rendre compte que la paire de clés créée correspond à un nom, une adresse courriel et un commentaire. Mais il est inutile de créer une clé par adresse courriel, il suffit simplement de rajouter des identités (les trois informations nom réel, adresse courriel, commentaire) à votre clé personnelle pour pouvoir signer plusieurs adresses courriel avec la même clé GPG.

Tout comme pour signer une clé, la commande d'édition est la suivante :

```
gpg --edit-key mon_id
```

Le logiciel affiche les informations dont il dispose puis vous propose de saisir une commande.

Voilà ce que ça donne dans mon cas (avec jean ou 6A82BA76 à la place de `mon_id`) :

```
gpg --edit-key jean
gpg (GnuPG) 1.4.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

La clé secrète est disponible.

sec  rsa4096/0x4374F0E466582E03
     créé : 2017-08-26  expire : jamais       utilisation : SC
     confiance : ultime        validité : ultime
ssb  rsa4096/0xBF92F66B91148D95
     créé : 2017-08-26  expire : jamais       utilisation : E
[  ultime ] (1). Jean Peyratout <jean.peyratout@sud-ouest.org>
[  ultime ] (2)  Jean Peyratout (clé 4096 bits 2017)
<jean.peyratout@abul.org>
[  ultime ] (3)  Jean Peyratout <jean@sud-ouest.org>
[  ultime ] (4)  Jean Peyratout <jean.peyratout@free.fr>
[  ultime ] (5)  Jean Peyratout <jean.peyratout@aful.org>
[  ultime ] (6)  Jean Peyratout <jean.peyratout@laposte.net>
[  ultime ] (7)  Jean Peyratout <jean.peyratout@ac-bordeaux.fr>
[  ultime ] (8)  Jean Peyratout <jean.peyratout@scideralle.org>
[  ultime ] (9)  Jean Peyratout <jean.peyratout@terredadeles.org>
[  ultime ] (10)  [jpeg image of size 14964]

Commande>
```

Cette fois-ci, la commande à entrer est `adduid`

```
Commande> adduid
```

Il vous est ensuite demandé de remplir les mêmes champs que lors de la création de la clé, c'est à dire le nom (au moins 5 caractères), une adresse courriel et un commentaire (facultatif).

Puis la commande `save` vous permet d'enregistrer les changements, et enfin la commande `quit` permet de sortir du mode interactif.

```
Commande> save
```

```
Commande> quit
```

Pour finir, envoyez votre clé modifiée au serveur de clés (ici celui du MIT) :

```
gpg --keyserver pgp.mit.edu --send-keys mon_id
```

Merci à Guillaume Subiron

Résumé des principales commandes

- Générer une paire de clés :

```
gpg --gen-key
```

- Générer un certificat de révocation :

```
gpg --gen-revoke mon_id_clef
```

- Publier sur un serveur :

```
gpg --keyserver pgp.mit.edu --send-keys mon_id_clef
```

- Récupérer une clef publique sur un serveur :

```
gpg --keyserver pgp.mit.edu --recv-keys son_id_clef
```

- Calculer la fingerprint d'une clé :

```
gpg --fingerprint id_clef
```

- Lister les clés présentes :

```
gpg --list-keys
```

- Éditer une clé :

```
gpg --edit-key id_clef
```

  - `Commande> trust`
  - `Commande> sign`
  - `Commande> quit`

- Signer une clé :

```
gpg --sign-key id_clef
```

Liens

De nombreuses ressources sont disponibles sur le Web :

- http://fr.wikipedia.org/wiki/GNU_Privacy_Guard
- http://www.francoz.net/doc/gpg/
- http://rqchp.ca/?lang=FR&pageId=65&
- GnuPG http://doc.ubuntu-fr.org/gnupg

- Enigmail http://doc.ubuntu-fr.org/enigmail avec Thunderbird
- FireGPG http://fr.getfiregpg.org/index.html avec Firefox

Revenir au sommaire du Mainteneur de paquet