TP GPG (*GNU Privacy Guard)*

Simon Bjornson

1. GPG et le chiffrement symétrique

1.1. Vérifications

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. GPG est une version libre du logiciel PGP (Pretty Good Privacy) crée par Philip Zimmermann. Bien qu'il existe des clients graphique (GPA par exemple), nous allons utilisez l'outil le plus basique (mais aussi le plus puissant) : le programme gpg en mode texte.

Dans un terminal vérifiez que le logiciel est bien installé avec la commande

$ gpg --version

1.2. Chiffrement

Si vous partagez une clé secrète avec votre destinataire, vous pouvez utiliser GPG pour faire de la cryptographie symétrique. Créer un fichier « testgpg.txt ».
Pour crypter le fichier "testgpg.txt", il faut utiliser la commande :

$ gpg --symmetric --armour testgpg.txt

Ceci créera un fichier "testgpg.txt.asc" contenant le fichier chiffré. Cette méthode est préférée quand vous voulez par exemple envoyer le fichier par email. Vous pouvez également créer un fichier crypté binaire. La commande correspondante est

$ gpg --symmetric testgpg

Ceci créera un fichier binaire "testgpg.gpg" contenant le fichier binaire chiffré. Utilisez cette commande pour créer un fichier crypté. Vérifier qu'en modifiant un tout petit peu la clé ou le contenu du fichier, le contenu du fichier crypté change énormément. Vous pouvez pour ceci utiliser un gros fichier texte et regarder ce qui se passe dans le fichier crypté en l'ouvrant avec un éditeur de texte (gedit ou autre).

1.3. Déchiffrement

Pour déchiffrer, il suffit d'utiliser :

$ gpg --decrypt testgpg.txt.asc

(Bien entendu, " testgpg.txt.asc" sera remplacé par " testgpg.txt.gpg" si le fichier a été chiffré en binaire)

1. Envoyez un fichier crypté à votre voisin.
2. Quelle est la clé de chiffrement ?
3. Décryptez le fichier chiffré de la question précédente et vérifier que vous obtenez le bon message. (Vous pouvez échanger la clé à l'oral.)
4. Essayez de décrypter le fichier en utilisant une clé erronée (une seule lettre de différence), que se passe-t-il ?

5. À votre avis, que peut-on faire pour déchiffrer le fichier si on a perdu la clé ?

2. Gestion des clés publique / privée

2.1. Création des clés

Les clés sont stockées dans un répertoire **caché** de votre répertoire personnel : .gnupg. Vous êtes la seule personne à avoir accès à ce fichier. De plus, vos clés sont protégées par un mot de passe pour renforcer la sécurité. 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 (30 jours)
- Mettez votre vrai nom (ou au moins vos initiales) et choisissez ``BTS SIO'' 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.

Pour vérifier que les clés ont bien été créées, utilisez la commande

$ gpg --list-keys

Vous devriez obtenir quelque chose du genre

```
$HOME/.gnupg/pubring.gpg
pub   2048R/729616C2 2012-03-20 [expire: 2013-05-19]
uid                  Jean DUPONT (BTS SIO) <jean.dupont@xxxxx.fr>
sub   2048R/215D7425 2012-03-20 [expire: 2013-05-19]
```

qui vous indique que vous avez une clé principale (ligne "pub") qui expire le 19 mai ; et une sous-clé (ligne "sub") qui expire aussi le 19 mai. La ligne "uid" vous donne l'identité de l'utilisateur correspondant. La clé principale est utilisée pour les signatures, et la sous-clé pour le chiffrement.

Créez votre propre clé et vérifiez son existence.

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. Ceci est d'autant plus important si vous avez distribué votre clé publique...

2.2. Certificat de révocation

Lors de la création d'une clé importante, il est impératif de créer un *certificat de révocation*. C'est ceci qui vous permettra de faire savoir que votre clé ne doit plus être utilisée... Un tel certificat pourra servir dans le cas où vous perdez votre clé privée, ou bien vous avez perdu votre passphrase, ou on vous a volé votre ordinateur...

Pour créer un certificat de révocation, il faut faire :

$ gpg --output revoke.txt --gen-revoke uid

où "uid" est l'identité de la clé concernée.

Cette commande génèrera un fichier revoke.txt

Attention : ce fichier permet de supprimer votre clé... Générez un certificat de révocation de votre clé.

Où le sauvegardez-vous ?

2.3. Partage des clés

2.3.1. À la main

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

$ gpg --output cle.asc --export --armour uid

où "uid" est l'identité (l'adresse email par exemple) de la clé concernée et "cle.asc" le nom du fichier qui contiendra la clé en ASCII.

Si vous voulez exporter la clé en binaire, il faut utiliser :

$ gpg --output cle.gpg --export uid

À l'inverse, pour importer une clé (en binaire ou en ASCII) contenue dans le fichier "cle.asc", il suffit d'utiliser la commande

$ gpg --import cle.asc

1. Échangez vos clés avec votre voisin en exportant la vôtre et important la sienne.
2. Vérifiez qu'une nouvelle clé apparaît dans la liste affichée par "$ gpg --list-keys".

2.3.2. Avec un annuaire

Pour partager les clés à grande échelle, on utilise plutôt un *serveur de clés*. Il existe de nombreux serveurs publics, comme celui du MIT : pgp.mit.edu. Ces serveurs de clés sont tous interconnectés, et il est pour cette raison impossible de pirater ces annuaires... (Il faudrait pour ceci arriver à tous les pirater en même temps !)

$ gpg --keyserver nom_du_serveur --send-key 0xnnnnnnnn

où nnnnnnnn est le numéro de votre clé, càd les chiffres apparaissant après le 1024D sur la première ligne de votre entrée lors de la commande gpg --list-keys. (729616C2 dans mon cas)

Vous pouvez également télécharger des clés sur le serveur avec

$ gpg --keyserver nom_du_serveur --recv-key 0xnnnnnnnn

2.3.3. Vérifier et contre-signer une clé

Chaque clé possède une « empreinte digitale ». Quand vous récupérez une clé, il est important de vérifier cette empreinte... Cette empreinte est suffisamment petite pour être facilement transmissible (carte de visite etc.)

Par exemple :

4EB0 814A 796F A631 F2E3 3004 277B 0890 7396 16C2

Pour trouver l'empreinte d'une clé, vous pouvez utiliser

$ gpg --fingerprint

qui listera toutes les clés connues avec leur empreinte. Si vous mettez une chaîne de caractères à la fin de la commande, cela ne listera que les clés qui contiennent la chaîne en question. (Pratique quand vous avez beaucoup de clés.)

Une fois que vous avez vérifié une clé, vous pouvez l'authentifier pour dire « je fais confiance à cette clé... » On parle de *contre-signature*. La commande est simplement :

$ gpg --sign-key uid

où "uid" est l'identité de la clé à authentifier.

> Vérifiez l'empreinte des clés que vous avez récupérées, et authentifiez les.

3. Signature électronique

3.1. Signature

Maintenant que vous avez des clés, vous pouvez signer des messages. Pour cela, il faut utiliser :

$ gpg --clearsign fichier

pour signer le fichier "fichier". Ceci créera un nouveau fichier "fichier.asc" qui contiendra le fichier original avec une signature vous authentifiant.

La commande

$ gpg --detach-sign fichier

permet elle de créer uniquement une signature (binaire) pour le fichier en question. Cette signature (fichier "fichier.sig") devra être envoyé avec le fichier original.

Pour un fichier texte, la première méthode est préférable. (Sauf si c'est un email et que votre logiciel gère les signatures en pièce jointe...) Pour un fichier binaire, il faut mieux utiliser la seconde méthode.

3.2. Vérification

Pour vérifier un fichier signé, on utilise

$ gpg --verify fichier.asc

Il faut bien entendu pour cela disposer de la clé publique de la personne qui a signé le document.

1. Comparez les signatures d'un document original
2. Vérifiez la signature d'un message que votre voisin vous enverra.
3. Vérifiez la signature d'un message qui a été modifié après signature. Que se passe-t-il ?

4. Chiffrement

4.1. Chiffrement

Le principe est le même que pour la signature : on utilise

$ gpg --encrypt --armour fichier

pour obtenir un fichier ASCII "fichier.asc" contenant le fichier original crypté. GPG nous demandera les destinataires, à choisir parmi les gens dont on possède les clés publiques.

Si on veut obtenir un fichier binaire, la commande devient :

$ gpg --encrypt fichier

pour obtenir un fichier "fichier.gpg" contenant le fichier original crypté.

4.2. Déchiffrement

Toujours pareil : pour décrypter, on utilise

$ gpg --decrypt fichier

Testez le cryptage / décryptage avec votre voisin...

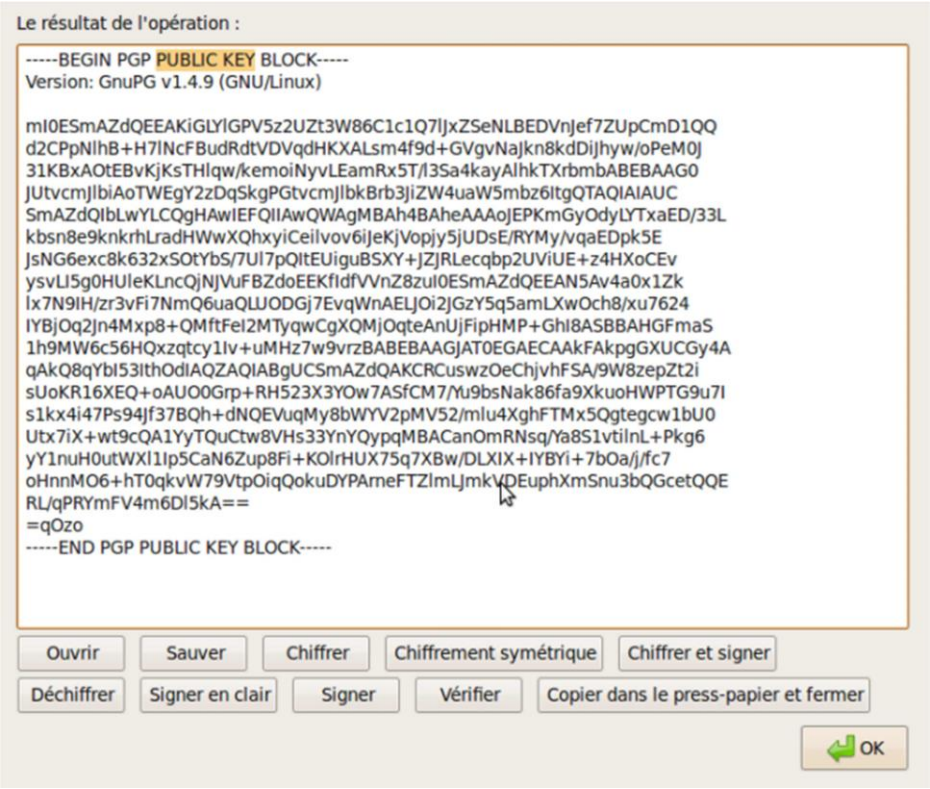
5. En graphique sous Windows

GPG est a la base un programme qui s'utilise dans une console, ce qui rebute les moins puristes. Mais il existe une multitude d'interfaces graphiques et de plugins utilisant GPG, peu importe que vous utilisiez **Outlook, Thunderbird ou Gmail** en direct, vous trouverez toujours le petit soft GPG qui vous permettra de chiffrer vos emails.

Installez **FireGPG**. Il s'agit d'un plugin pour Firefox qui permet de chiffrer n'importe quels messages dans un webmail. (ou n'importe quel champs texte dans firefox en fait... donc y compris des messages sur des forums and co). Ce n'est qu'un exemple parmi d'autres. Une fois que c'est installé, vous avez accès à différentes options via le menu "**Outils -> FireGPG**" ou via un clic droit dans une zone de texte dans un webmail par exemple.

1ère étape - Créer votre clé !

Allez dans ce menu et cliquez sur "Gestionnaire de clés" puis cliquez sur le bouton "Nouvelle clé" pour générer votre paire de clés. C'est aussi à cet endroit que vous importerez les clés publiques de vos amis. Renseignez les champs... Plus le cryptage est fort et plus la clé mettra du temps à se générer. Et surtout, mettez un vrai mot de passe qui tient la route avec des majuscules, des minuscules, des chiffres. Cliquez sur "Générer la clé". Firefox va alors se bloquer, patientez. Votre paire de clés est maintenant créée. Importez les clés publiques de vos amis et surtout donnez leur la votre (et mettez la en signature de vos emails). Pour connaitre votre clé publique, toujours dans le menu de FireGPG, cliquez sur "Exporter" (ou importer pour ajouter les clés de vos contacts)



On voit clairement que la clé commence par -- BEGIN PGP PUBLIC KEY BLOCK -- ce qui nous confirme qu'il s'agit bien de votre clé publique.

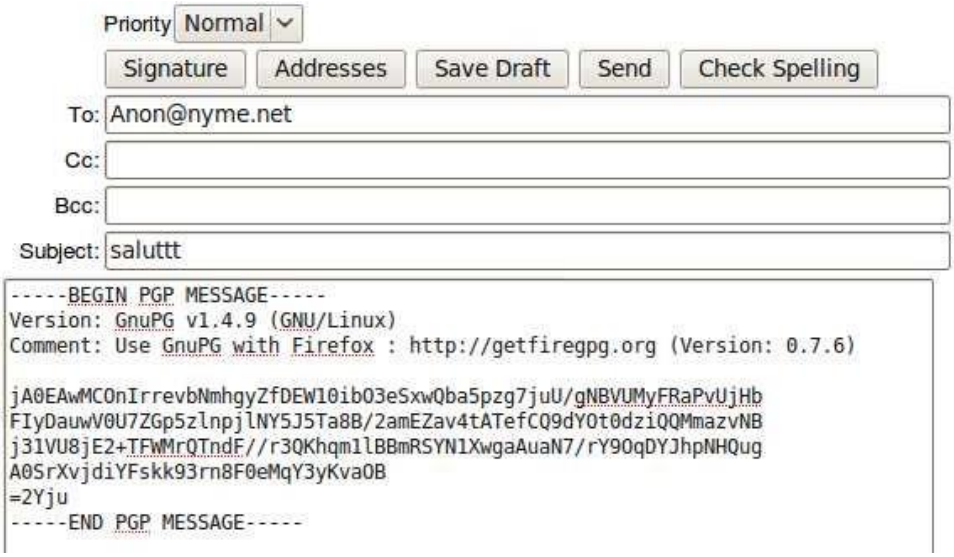
Vous pouvez donc la diffuser sur votre site ou dans vos emails en guise de signature.

2ème étape - Chiffrage / signature de notre premier email

Ensuite, 2 choix s'offrent à vous... Signer ou chiffrer votre message. Le signer, ça permet de le laisser en clair mais d'en attester l'origine. Quand un de vos emails est signé, votre destinataire peut facilement savoir si c'est bien vous qui l'avez envoyé. Le chiffrement lui, ne permet pas de voir le message.

Prenez un webmail classique...

Faites un clic droit et dans le menu FireGPG, sélectionnez l'option "Chiffrer". On vous demande alors un mot de passe et de sélectionner la clé publique de votre destinataire. Votre message apparaitra ensuite complètement chiffré et lisible seulement par votre destinataire. Vous pouvez donc l'envoyer sans crainte. (et pour le chiffrement des pièces jointes, utilisez un outil GPG standard pour les fichiers)



Lorsque vous recevez un message chiffré, il ne vous reste plus qu'à le déchiffrer via l'option qui va bien. Maintenant, la balle est dans votre camp.