GnuPG

Bien que la méthode décrite se déroule sous Gnu/Linux, c'est la même que sous les autres systèmes.

Attribuer une signature numérique à vos documents, mails et projets secrets ? Chiffrer (= Crypter, mot franglais incorrect) un document encore plus secret, ou alors des photos qui montrent l'existence de votre frère jumeau caché ? Ou simplement vos recettes de cuisine ? La parano's way of life n'est plus réservée aux pro's de la programmation machine à chaud.

En effet GPG (lisez : GNU Privacy Guard) est un outil multifonction, libre et gratuit, qui permet de signer et de chiffrer à peu près tout ce qui se présente sous forme numérique (tout ce qui a un bit, si vous me passez l'expression).

Je vous propose un petit didacticiel “in the Molière's language”, pour être un pro du chiffrement, impressionner vos collègues et vous faire aimer de votre boss sur vos incroyables et nouvelles connaissances sur la certification de données.

Vous trouverez un excellent didacticiel à cette adresse : http://gpglinux.free.fr/gpg.pdf (Pas de https !)

Une autre documentation, au format HTML : http://www.francoz.net/doc/gpg/ - Introduction à GnuPG (Documentation très ancienne)

Théorie

Chiffrement/Signature

Le chiffrement est la fonction la plus connue. Elle permet de rendre un document complètement illisible pour le commun des mortels qui ne connaît pas la phrase magique. Pour ce faire, le créateur de l'information chiffre ses données à l'aide d'un mot de passe convenu avec son interlocuteur, et ce dernier pourra retrouver les documents de son binôme rapidement - le tout - théoriquement - dans une totale intimité numérique.

C'est une méthode qui fait souvent irruption dans les films d'espionnage, d'ailleurs je pense que GPG est connu pour cette fonction dont les néophytes ont dû abuser. Ce n'est pourtant pas la plus “intéressante”.

La signature d'information, c'est un peu le renouveau de la rareté sur internet. En effet, elle permet à

un fichier d'être unique numériquement.

Clairement, la signature numérique permet deux choses : vérifier que les informations reçues proviennent bien de la personne qui nous l'a envoyée et que ces mêmes informations sont reçues dans leur intégralité. Cela permet une multitude de possibilités assez intéressantes dans de nombreux domaines.

Je peux, grâce à cela, m'assurer que les informations que je possède sont “authentiques”, “officielles” et qu'elles n'ont pas été détournées par un tiers malveillant (ou même bienveillant, soit dit en passant).

Est-ce utile pour le commun des mortels ? Oui et non. Oui, parce qu'il faut toujours avoir un système de confidentialité prêt à être utilisé, et non parce que l'utilisateur alpha n'a peut-être pas besoin de vérifier absolument si le message reçu vient bien de l'émetteur qui l'affirme.

Intérêt et mise en garde

Bien sûr, la “parano way of life” peut en laisser beaucoup perplexes. À quoi peut servir le chiffrement à tout va ? Il faut bien comprendre que c'est l'utilisation de première vue qui peut dérouter un peu. Il y a une multitude d'utilisations dérivées possibles, et pas que pour les “geeks”.

Par exemple, la signature numérique peut servir de véritable “certificat d'authenticité” gageant de la provenance, de l'officialité et de l'intégrité de l'information obtenue. De là, on peut imaginer énormément de possibilités, que ce soit de façon professionnelle, comme pour les particuliers. Mais, il y a quand même un brin de paranoïa.

Il faut aussi ne pas oublier que le chiffrement/signature n'est pas qu'une question d'informatique. Autrement dit, il ne faut pas rester que sur le support informatique. Par exemple, concernant les mots de passe d'un document chiffré, ou les signatures publiques (voir plus bas), qu'il faut fournir à l'interlocuteur, il est peu utile de passer par un moyen informatisé comme le mail pour fournir les informations.

Il vous faudra malheureusement accéder à un contact social physique, dans une pièce aux murs de béton armé, pour fournir les précieuses informations de base. Car rien n'est plus sûr que de voir la personne pour convenir d'un mot de passe, ou donner la clef publique, si on veut rester totalement intègre face au principe. C'est blasphématoire de communiquer le mot de passe via mail, messagerie instantanée … bref n'importe quel intermédiaire informatique peu sécurisé. La sécurité totale du chiffrement, de la transmission d'information, est globale dans tout son développement, il faut limiter les intermédiaires, et sécuriser les transmissions physiques/numériques.

Je veux dire qu'il y a une forme de “relativisme” à adopter pour éviter de tomber dans le piège de l'informatique pour l'informatique et ne pas oublier que, si on fait ça, il y a une raison qui vient du monde réel. Maintenant que vous êtes averti, vous pouvez apprendre à vous en servir.

Fonctionnement et principes

Le système de clefs

GPG possède un système de clefs asymétriques. Bien que cela puisse paraître barbare, c'est assez

simple, mais il faut le comprendre pour bien utiliser l'ensemble.

Lorsque que vous créez votre “profil” (c'est-à-dire, lorsque vous créez l'ensemble des données pour signer/chiffrer les données de votre nom), deux clefs sont créées : la clef publique et la clef privée.

La clef privée doit rester confidentielle. Elle vous permettra d'exporter votre profil sur d'autres machines, c'est elle qui contient toutes les informations vous concernant (chiffrées, évidemment).

La clef publique est celle que vous devez fournir à vos correspondants. Elle permet à ces derniers de reconnaître l'authenticité de vos mails, et de chiffrer des informations pour votre utilisation.

Il faut savoir qu'une clef publique peut ne pas forcément être attribuée à une seule personne (physique), mais aussi à un groupe de personnes (comme une personne morale).

Signer

Lorsque vous signez une information, vous tapez votre code. GPG crée un fichier unique associé à vos informations. Il est unique. Autrement dit si vous modifiez le document signé, le fichier de signature ne sera plus valide. Même dans le cas où vous modifiez le document et annulez les modifications, le fichier signature ne sera plus valide.

Le fichier signature doit être fait avec la version finale du document à signer, il ne doit plus changer d'un bit après la signature.

Ensuite vous envoyez votre document à une personne. Cette dernière doit posséder votre clef publique. Elle se servira de cette dernière avec GPG pour vérifier que le document est (encore une fois au bit près) conforme à ce que vous lui avez envoyé, et, surtout, pour vérifier qu'il vient bien de vous (car vous seul pouvez générer ce fichier signature avec votre clef privée).

Il y a véritablement une question d'unicité dans la signature numérique. Si le contenu signé est modifié ne serait-ce que de rien du tout (il suffit de l'ouvrir pour modifier les informations contenues), alors il ne sera plus authentique, ni certifié.

Cette méthode n'empêche pas la lecture du contenu, elle permet de vérifier que ce contenu est authentique, et livré dans son intégralité par rapport à la version de celui qui l'a composé. Attention : si la signature comporte bien une date, cette dernière n'est pas fiable. Il suffit en effet de changer la date de son ordinateur pour pouvoir générer des signatures post ou antidatées. Une signature certifie donc l'authenticité des données qu'elle signe, mais pas à une date donnée.

À savoir : il existe des serveurs sécurisés pour stocker les clefs publiques. C'est plus pratique pour que n'importe qui puisse vérifier la validité du document. Vous pouvez aussi mettre ce fichier sur votre site internet, mais le plus sécurisé reste de transmettre la clef … par un contact social.

Chiffrer

Cette méthode permet de diffuser des informations qui, en plus d'être certifiées authentiques et intégrales, ne sont pas lisibles par quiconque.

Lorsque vous chiffrez un document, vous spécifiez la clef publique appartenant à la personne (morale ou physique) qui est le destinataire. Seule cette personne (qui possède également le mot de passe)

pourra déchiffrer les informations. Vous pouvez aussi chiffrer « dans le vide », c'est-à-dire que le fichier n'est pas destiné à quelqu'un en particulier, mais à toute personne connaissant le mot de passe, mais c'est évidemment moins sécurisé.

C'est donc beaucoup plus confidentiel. Bien que l'on puisse croire à première vue que c'est destiné à la communication entre deux personnes uniquement, ce n'est pas forcément vrai : un profil peut appartenir à un groupe de personnes (association, entreprise), donc chaque ordinateur de ce groupe peut posséder le même compte, mais la sécurité est plus controversée dans ce cas-là.

Annexe

Il existe beaucoup d'applications qui utilisent GPG, et, donc, la signature et le chiffrement. La messagerie instantané Jabber, ou encore, Mozilla Thunderbird avec son module enigmail, pour les mails… Tous ces logiciels facilitent l'utilisation des outils de GPG pour une utilisation courante.

Pratique

Il faut savoir que la majorité de l'initialisation se fait en ligne de commande (ici en tout cas). C'est ce que je préfère, dans la parano's way of life : on n'aime pas les interfaces graphiques. Mais vous en trouverez un large nombre qui vous faciliteront la vie.

Créer ses clefs

Une fois GPG installé, vous pouvez créer vos clefs. C'est très simple, il suffit de lancer une console et taper :

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

A partir de ce moment vous allez avoir un certain nombre de questions.

- **Le choix du chiffrement** : Le choix (1) RSA/RSA est conseillé (DSA ou Elgamal l'étaient dans les versions plus anciennes). Même si vous ne faites que signer, je pense que c'est le choix à faire.
- **La longueur de la clef** : Théoriquement, plus la longueur de la clef est grande, plus la sécurité est forte. Le standard étant 2048 et le maximum 4096. Plus la clef est longue, plus elle a du poids ; il ne me semble pas que ce soit un souci, de nos jours, où on parle d'avoir des connexions à fibres optiques pour tous. Personnellement j'ai pris 4096 (parano's way of life). À savoir : plus la clef est longue, plus l'ordinateur a du mal à la générer : pendant la génération, pour qu'elle soit efficace et donne un résultat hautement sécurisé, il faut utiliser son ordinateur, bouger la souris, taper du texte, bref … « mettre des artéfacts » pendant la génération.
- **Le temps de la clef** : Intuitivement, on aurait tendance à mettre « illimité ». Pourtant, je ne pense pas que ce soit la bonne solution. Sachant que vous pouvez perdre votre clef privée, ou vous la faire voler, qui sait (on peut quand même révoquer publiquement une clef, cf plus loin), ou tout simplement si vous oubliez votre mot de passe. Donc, le mieux, pour la sécurité, reste de prolonger régulièrement la durée de votre clef, avant que celle-ci n'expire. Tout dépend de votre utilisation et de votre paranoïa quant au chiffrement de vos données. Il faut y réfléchir avant hein :o.

- **Identité** : Toutes les questions de nom, courriel, commentaires (on met généralement le site). C'est important, étant donné que ce sont les informations qui apparaîtront lors de la vérification des signatures.
- **Mot de passe** : Forcément, très important, l'idéal serait 3anc3dlk56nld12qn#@ius, mais ce n'est pas très simple à retenir. Plus simple : une phrase relativement longue, avec des caractères spéciaux. Plus la complexité est grande, plus la sécurité est grande. Et ne pas oublier … de ne pas oublier ce mot de passe ; auquel cas il faudra re-générer une clef (d'où l'utilité de créer des clefs périodiques).

À partir de ce moment, les clefs se génèrent. Dès qu'il n'y a plus de . et de +, c'est que tout s'est bien passé. Il vous informe des informations concernant vos clefs publique/privée, et vous donne « l'empreinte ». Cette empreinte permettra de retrouver votre clef publique, dans le serveur PGP, pour une vérification plus aisée et automatique.

Gérer son porte-clef

Gérer son porte clef, c'est exporter sa clef publique pour la diffuser, exporter sa clef privée pour la sauvegarder, c'est aussi importer les clefs publiques de ses connaissances.

Pour connaître la liste des clefs publiques que vous possédez :

```
gpg --list-key
```

Pour connaître la liste des clefs privées :

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

Pour supprimer une clef publique :

```
gpg --delete-key la-clef-publique
```

Et pour la clef privée :

```
gpg --delete-secret-key la-clef-privée
```

Export

Pour que vos amis super secrets puissent vérifier la validité des informations que vous leur envoyez, il faut leur donner la clef publique !

```
gpg --export --armor votre-clef > ~/clefpublique.asc
```

Cette commande génèrera un fichier .asc, qui est, finalement, un fichier texte. De là, plusieurs solutions s'offrent à vous.

- Soit vous la transmettez à un serveur sécurisé (comme https://pgp.mit.edu), en faisant un copier-coller du fichier texte produit dans la fenêtre mise à disposition sur ledit site. Cela permettra à tous de s'y référer de façon rapide.

- Soit vous mettez ce fichier sur votre site internet personnel, et vous donnez le lien. Cette méthode n'est pas très sûre pour une multitude de raisons. Par contre, elle permet de mettre à jour régulièrement la clef et, donc, d'assurer un changement simple et régulier.
- Mano à mano, méthode la plus sécurisée : vous donnez vous même le fichier à la personne qui désire vous identifier - par contact social (argh !) - sur une clef USB, que vous avez pris soin de chiffrer, avec un mot de passe de 10 lignes, que vous indiquez à la personne à ce même moment, dans le parking d'un centre commercial, aux alentours de 2h du matin.

Vous l'aurez compris, il faut limiter absolument le nombre d'intermédiaires non sécurisés. Rien que l'export en fichier sur votre disque n'est pas sécurisé ! Pour cela, il existe une commande qui permet d'exporter votre clef de façon sécurisée, directement sur le serveur :

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

Import

Un fois créée, vous ou vos collègues de la DST doivent pouvoir mettre la clef publique dans leur trousseau de clefs !

Vous récupérez le fichier .asc en question, et faites, tout simplement :

```
gpg --import clef.asc
```

Après, faites un petit –list-key, pour vérifier qu'elle est bien présente.

Il faut savoir un petit détail pour les étourdis, si vous perdez votre clef privée, vous ne pourrez pas la retrouver en réimportant votre clef publique. Il faut exporter votre clef privée, mais ce n'est pas du tout sécuritaire : il faut chiffrer le fichier .asc, avec un mot de passe que vous n'oublierez jamais, ou, mieux, se l'imprimer, si vous avez un scanner capable de faire de l'OCR, ou le cacher dans un autre fichier … Bref, vous l'aurez compris, exporter sa clef privée n'est pas une bonne chose, il vaut mieux prendre ses précautions (clef périodique, faire une clef pour révoquer, avoir un système stable et durable, etc..)

Pour pouvoir réimporter sa clef privée, il faut l'exporter avec la commande suivant :

```
gpg --export-secret-key -a > fichier
```

et l'importer avec cette commande :

```
gpg --import --allow-secret-key-import fichier
```

Cette opération est relativement peu sécurisée, il vaut mieux l'éviter.

Détruire sa clef

Pour une raison ou une autre, on peut regretter sa clef. Et vous l'avez diffusée sur tous les serveurs ! Il existe un moyen simple : il faut révoquer votre clef. Autrement dit, vous créez une clef qui, mise sur les même serveurs, « annoncera » la fin de validité de votre clef publique. Très pratique quand on a

créé une clef à durée de vie illimitée.

```
gpg --gen-revoke votre-clef
```

Elle créera un certificat pour révoquer votre clef, à mettre ensuite sur les serveurs de clefs. Autant vous dire qu'il est intelligent de créer ce certificat, avant que vous n'en ayez besoin. Il est aussi important de ne pas le dévoiler, même si c'est moins important que la clef privée (on ne pourra pas signer à votre place, mais on pourra vous empêcher de signer).

Ensuite, pour l'appliquer, vous importez le fichier via :

```
gpg --import fichier
```

Et, ensuite, exportez-la au serveur de clefs :

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

Si tout s'est bien passé, votre clef est maintenant invalide.

Signer ses informations

Nous attaquons une partie intéressante : la signature électronique ! Il existe deux méthodes, il faut étudier laquelle convient le mieux à votre utilisation.

Avec un fichier joint

Pour signer un fichier, il faut taper :

```
gpg --default-key votre-clef --armor --detach-sign fichier
```

Cela créera un fichier portant le même nom que celui à signer avec une extension .asc. Ensuite, il vous suffira d'envoyer votre fichier avec lequel vous allez joindre le fichier .asc.

Pour vérifier que le fichier est certifié authentique, il suffit de taper (si, bien sûr, vous avez la clef publique de votre correspondant) :

```
gpg --verify fichier.asc fichier
```

Signature intégrée

Vous pouvez intégrer dans le fichier même (par exemple les messages, les fichier textes, etc.), la signature électronique. Pour cela :

```
gpg --default-key votre-clef --clearsign fichier
```

La signature se trouvera à l'intérieur du fichier. Vous le remarquerez rapidement, votre texte sera

entouré des balises GPG, avec le morceau de la signature.

Pour vérifier l'authenticité :

```
gpg --verify le-fichier-signé
```

Bien sûr, c'est la version manuelle ; des logiciels font ce genre de travail parfaitement, comme enigmail, pour signer les mails.

Chiffrer ses informations

Méthode via clefs

Allons-y ! Produisons des fichiers hautement sécurisés (du moins, théoriquement).

```
gpg --recipient la-clef-du-recepteur --encrypt fichier
```

Cela vous pondra un fichier .gpg, illisible de façon binaire.

Mais, vous pouvez chiffrer l'information de façon ascii, souvent plus recommandé, de cette façon :

```
gpg --recipient clef-du-recepteur --armor --encrypt fichier
```

À ce moment là, vous aurez un fichier .asc, qui est lisible par un éditeur de texte, mais, bien sûr, incompréhensible. C'est plus conseillé, puisque plus souple. Vous pouvez faire un copier-coller du fichier via mail, ou autre, s'il est chiffré en ascii. Les deux façons sont aussi sécurisées l'une que l'autre.

Mais, la première est moins gourmande en poids (relatif), et la seconde est plus simple d'exportation.

Dans les deux cas, seule la personne possédant la clef indiquée dans la commande pourra déchiffrer les informations.

Pour déchiffrer :

```
gpg --decrypt fichier > fichier
```

Tout simplement, il demandera le mot de passe de l'utilisateur, et transformera le fichier .gpg en fichier lisible.

Méthode de chiffrement symétrique

Autre moyen de chiffrer/déchiffrer : Le chiffrement symétrique, c'est-à-dire qui n'utilise pas le système de clef publique/privée, on vous demande un mot de passe que votre interlocuteur doit connaître.

`gpg -c fichier` pour chiffrer

`gpg -o fichier -d fichier.gpg` pour déchiffrer.

Rappelons que la meilleure façon de donner un mot de passe à quelqu'un, c'est par un contact SOCIAL !