

GnuPG

Olivier Hoarau (olivier.hoarau@funix.org)

V1.7 du 1 avril 2007

1 Historique du document.....	2
2 Préambule.....	2
3 Présentation.....	2
4 Installation.....	3
5 Comment ça marche.....	3
6 Utilisation.....	4
6.1 Créer une clé.....	4
6.2 Exporter une clé publique.....	6
6.3 Importer une clé publique.....	9
6.4 Vérifier l'empreinte.....	9
6.5 Certifier une clé.....	9
6.6 Chiffrer des données.....	10
6.7 Déchiffrer des données.....	12
6.8 S'authentifier et authentifier.....	16
6.9 Editer ou supprimer des clés.....	17

1 Historique du document

V1.7	01.04.07	Passage à la version 1.4.7, rajout de précisions pour certifier l'intégrité de l'archive
V1.6	30.04.06	Passage à la version 1.4.3
V1.5	08.08.05	Passage à la version 1.4.2, changement de toutes les traces des commandes
V1.4	25.03.05	Passage à la version 1.4.1
V1.3	06.01.04	Passage à la version 1.2.4
V1.2	16.03.03	Passage à la version 1.2.1, il est nécessaire maintenant de signer sa propre clé publique pour qu'elle puisse être importée par d'autres utilisateurs.
V1.1	13.10.02	Passage à la version 1.2.0
V1.0	10.05.02	Création du document

2 Préambule

Ce document présente les moyens de communiquer en toute confidentialité en utilisant **GnuPG**.

La dernière version de ce document est téléchargeable à l'URL <http://www.funix.org>. Ce document peut être reproduit et distribué librement dès lors qu'il n'est pas modifié et qu'il soit toujours fait mention de son origine et de son auteur, si vous avez l'intention de le modifier ou d'y apporter des rajouts, contactez l'auteur pour en faire profiter tout le monde.

Ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni "dans l'état" sans aucune garantie de toute sorte, l'auteur ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

3 Présentation

Le chiffrement est un bon moyen pour assurer la confidentialité et la protection des données que vous transférez sur le net. Un outil comme **GnuPG** ne fait pas qu'encrypter des mails, il peut chiffrer n'importe quel type de données, et permet en outre de pouvoir s'authentifier. **GnuPG** ne fait appel à aucun système de cryptage propriétaire comme **RSA** ou **IDEA**, c'est pourquoi il peut ne pas être compatible avec certaines versions et options de **PGP**.

4 Installation

Je ne présenterai que l'installation avec les sources du programme qu'on peut trouver sur le site officiel de **GnuPG**.

Vous allez récupérer **GnuPG** sur www.gnupg.org l'archive se présente sous la forme d'un tarball **gnupg-1.4.7.tar.bz2**, pour décompresser rien de plus simple:

```
tar xvfj gnupg-1.4.7.tar.bz2
```

Cela va créer un répertoire **gnup-1.4.7** dans le répertoire courant. Dans ce répertoire tapez:

```
./configure
```

On tape maintenant

```
make
```

Et enfin en tant que root

```
make install
```

Cela va créer un répertoire **/usr/local/share/gnupg** contenant un fichier standard de configuration utilisateur qui va s'appeler **options.skel** ainsi qu'une FAQ au format html et les exécutables **gpg** et **gpgv** dans **/usr/local/bin**. Si ces chemins ne vous plaisent pas, taper:

```
./configure -help
```

Et retaper **./configure** avec les arguments qui vont bien.

5 Comment ça marche

Pour chiffrer des données vous allez vous aider d'une clé, pour pouvoir déchiffrer ces données, le destinataire devra disposer de la même clé, en conséquence l'émetteur devra par un moyen

ou un autre donner sa clé au destinataire, le problème est que si quelqu'un obtient cette clé, adieu la confidentialité, n'importe qui obtient la clé peut alors déchiffrer les données.

L'utilisation des clés publiques et privées résout ce problème. La clé publique comme son nom l'indique est publique et peut être largement diffusée sur le net, l'autre clé est privée, elle ne doit en aucun cas être communiquée à quelqu'un et doit rester secrète, elle est uniquement disponible pour son propriétaire et seulement. Maintenant l'émetteur va chiffrer son message au moyen de la clé publique qui appartient au destinataire, ce dernier déchiffrera son message avec sa clé privée, et le tour est joué.

Par conséquent le point crucial du système est que vous ne communiquiez en aucun cas votre clé privée, vous devez faire en sorte que le fichier et répertoire contenant votre clé privée soient d'accès hautement restrictifs.

Le risque maintenant du système est que la clé publique du destinataire que vous détenez ne soit pas la bonne mais appartienne à quelqu'un d'autre, ou que quelqu'un se soit fait passer pour votre destinataire (ce qui revient au même) et ait donné sa clé publique. Pour parer à cela, il faut ABSOLUMENT être sûr sans la moindre ambiguïté que la clé publique que vous recevez soit bien celle de votre destinataire, pour cela vous devez certifier la clé publique, vous ne devez en aucun cas certifier une clé publique si vous avez des doutes sur son origine.

GnuPG permet en outre de pouvoir s'authentifier ou d'authentifier des personnes. Pour s'authentifier, il suffit de crypter un message avec votre clé privée, n'importe qui possédant votre clé publique pourra alors le déchiffrer, comme vous êtes le seul à pouvoir émettre ce message, cela va vous authentifier parfaitement auprès des autres. Un risque cependant c'est que quelqu'un vous ait piqué votre clé privée.

6 Utilisation

6.1 Créer une clé

Pour créer une clé on doit taper la commande suivante

```
[olivier@asterix olivier]$ gpg --gen-key
```

voilà le résultat

gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.

This program comes with ABSOLUTELY NO WARRANTY.

This is free software, and you are welcome to redistribute it under certain conditions. See the file COPYING for details.

gpg: répertoire `/home/olivier/.gnupg' créé

gpg: nouveau fichier de configuration `/home/olivier/.gnupg/gpg.conf' créé

gpg: AVERTISSEMENT: les options de `/home/olivier/.gnupg/gpg.conf' ne sont pas encore actives cette fois

gpg: le porte-clés `/home/olivier/.gnupg/secring.gpg` a été créé

gpg: le porte-clés `/home/olivier/.gnupg/pubring.gpg` a été créé

Sélectionnez le type de clé désiré:

(1) DSA et Elgamal (par défaut)

(2) DSA (signature seule)

(5) RSA (signature seule)

Votre choix ? 1

La paire de clés DSA fera 1024 bits.

les clés ELG-E peuvent faire entre 1024 et 4096 bits de longueur.

Quelle taille de clé désirez-vous ? (2048)

La taille demandée est 2048 bits

Spécifiez combien de temps cette clé devrait être valide.

0 = la clé n'expire pas

<n> = la clé expire dans n jours

<n>w = la clé expire dans n semaines

<n>m = la clé expire dans n mois

<n>y = la clé expire dans n années

La clé est valide pour ? (0) 0

La clé n'expire pas du tout

Est-ce correct ? (o/N) o

Vous avez besoin d'un nom d'utilisateur pour identifier votre clé; le programme le construit à partir du nom réel, d'un commentaire et d'une adresse e-mail de cette manière:

« Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de> »

Nom réel: Olivier Hoarau

Adresse e-mail: olivier.hoarau@funix.org

Commentaire:

Vous avez sélectionné ce nom d'utilisateur:

"Olivier Hoarau <olivier.hoarau@funix.org>"

Changer le (N)om, le (C)ommentaire, l'(E)-mail ou (O)K/(Q)uitter ? O

Vous avez besoin d'une phrase de passe pour protéger votre clé secrète.

Entrez la phrase de passe:

Un grand nombre d'octets aléatoires doit être généré. Vous devriez faire autre-chose (taper au clavier, déplacer la souris, utiliser les disques) pendant la génération de nombres premiers; cela donne au générateur de

```
+++++.+++++.+++++..+++++ ++++++.+++++.+++++
+++++ ++++++ ++++++..+++++..+++++ ++++++ ++++++ +
++++..+++++.+++++ ++++++ ++++++ ++++++ ++++++ +.....+++++
```

+++++.+++++.+++++
+++++.+++++.+++++.+++++.+++++
+++++.+++++.+++++.+++++.+++.
++++>++++>++++.....>....++++.<++++.....
.....<++++.....>++++.....
..>++++.....<++++.....
.....<++++...>..++++.....++++^ ^ ^

gpg: vérifier la base de confiance
gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle
de confiance PGP
gpg: profondeur: 0 valide: 1 signé: 0
confiance: 0-. 0g. 0n. 0m. 0f. 1u
pub 1024D/B7FDA962 2006-04-29

Vous remarquerez qu'éventuellement on peut donner un temps de validité pour la clé. Cela va créer dans votre home directory un répertoire `.gnupg` contenant votre clé publique (**pubring.gpg**) et votre clé privée (**secring.gpg**). Inutile de chercher à éditer ces fichiers, c'est du binaire.

6.2 Exporter une clé publique

```
[olivier@asterix olivier]$ gpg --edit-key olivier
```

voilà le résultat

**gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.**

La clé secrète est disponible.

**pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: ultime validité: ultime
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>**

Commande> trust

**pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: ultime validité: ultime
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>**

**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 ? 5

Voulez-vous vraiment donner une confiance ultime à cette clé ? (o/N) o

**pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: ultime validité: ultime
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>**

Commande> quit

Pour exporter une clé, maintenant rien de plus simple

gpg --export --armor > public-key-ohoarau.asc

Cela va donner le fichier suivant.

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v1.4.7 (GNU/Linux)

```
mQGIBEYPgooRBACqiAhDZJw8PapB+DTDsxk69xgpxXEWUCr2H6Khnc2uVx6wkQY
3
IJtVAGexFNQtuyAfyxpyGDbNM/NdFVTT1urL0g+TFG1zeF2HGAjHlqjV9xEUR/eH
O3Kv9AN/cTQ2tUL8Nh3gEdO9PBsZzI4HTjYJIMYwIK/nvHVNUiyBNHaMUwCgnWe4
neuS7uogAY5fOlrnz+jVoDMD+QF8UIBETU2zObCtGRRhS0z36o5NU0J2bIO/s8au
N/ToX1GR/kKYjH3VHgZ8bPq4Qgy9QI9Uul32VUmuivRErbileGA5BvyCHcs6rtjI
07LO+CXJDFK6JULkJfZB4XE10EojfFUQQblbaqSWwhH7fwqUe/ADZorYPgEiKWbv
EZdjA/99woNDgRgM6z8Knuwix3eUVVutPLp0Ax9WfUmF88joFPtAUkoXAmh783nq
nqRWS05kImg9CAxMqxfFnKtheGjaX3IYYTgwDgSEnuHRCQ6aMaC8iIrmPlml9tz4
H6CRyTlnbTqZLWxnTvzRV20KEp9bk3Y49Cc6T+KZoAhyOWPnQLQpT2xpdmlliciBI
b2FyYXUgPG9saXZpZXIuaG9hcmF1QGZ1bml4Lm9yZz6IYAQTEQIAIAUCRg+CigIb
AwYLCQgHAwIEFQIIAwQWAgMBAh4BAheAAAJEC+JbVGDP+m3rBUAnAn47s6Im
8Oe
FJPqWqzSs3sByMowAJ0VAe9wj6TkNwXaFnmZYYkzGa3jUrKCDQRGD4KKEAgA2v
YT
CFBK1ainyyNK/s14nBR3aNBVNYHiq89525dDBqGux9IySudr13EVCbhZMxTMFUO4
bHF5GWHXoO4X59FnXenyMyYhKF2WpXCLUB3IGPMHg6eG2SM5agrIzAj/FDo+/5t
H
gp4AdVpzspaFVX0tgQXv2vBST87F9PHdfG6Tw5kCFsB0oZNb6GoW1bcpl1ddiwEu
4T3avkr0JLowiwUaSp3Guh8eRmDIICuDPY3pv6RCdgVDbv95rWXtUABTDDUHBuBk
3uF43yUxwdl0gCCdEhAK4e1fXRXri3/svThkGR9j6rsgdSze3Boq0jT9grqh36T/
9KPinavYFQauKeptvwADBwgAjib9i1fssYmwSODPCsxWwMWX4RM8uEUGfvVp3clz
jmhUIIUUk3+h0EzP2WaRSK0PFZCEvswnrzTNFlvGJ4Y9jY55tiEJ2KDoH8MGhApN
ieXIeF/YnwrsGe+QJrEIP5ag3I1ZzNh3jg/hsdK0NEVj69/oi6HqXecA/IONe4RY
sYfQIV21zg07D2wJD4QYVfY4cs0nGWfV2qlrJLT5Mj2Jl+HF1wc9TM5pRFdmgaUa
8/xdVO1dvkFWXeRFHtCj0957e9ZSRitsJaY8z5KUdhIFT2XfifnfKe8bCarHgkuS
WXlfMna/1Z1x6mEb3rrtZc8r0Y5SS+szyHgUx4ITG/KgoYhJBBgRAgAJBQJGD4KK
AhsMAAoJEC+JbVGDP+m3n7UANjXksFwySPKyvn1ncPL8jFC0ona1AJ9oUMPgcD2v
1z6ahCQZ4RxJnzhDiA==
=aodt
-----END PGP PUBLIC KEY BLOCK-----
```

Cela vous permet par exemple d'obtenir ma clé publique si vous voulez m'envoyer des messages chiffrés pour exercice.

6.3 Importer une clé publique

Pour importer une clé publique d'une personne, il suffit que ce dernier vous l'expédie par un moyen ou un autre. Une fois obtenue, vous devez la rentrer dans votre base de données des clés publiques de vos destinataires potentiels de messages chiffrés.

```
[olivier@obelix temp]$ gpg --import public-key-vhoarau.asc
```

voilà le résultat

```
gpg: clé 2CB452D2: clé publique « Véronique Hoarau <veronique.hoarau@funix.org> »  
importée
```

```
gpg:   Quantité totale traitée: 1
```

```
gpg:       importée: 1
```

Cette base de données de destinataires potentiels est dans votre répertoire **.gnupg** et se nomme **trustdb.gpg** (fichier binaire).

6.4 Vérifier l'empreinte

Bon le problème est que n'importe qui peut dire qu'une clé publique lui appartient alors que ce n'est pas le cas. La solution pourrait être de vérifier qu'on a la bonne clé en passant un coup de fil au propriétaire, mais ce n'est pas très pratique de lire un fichier signature. Y a une solution plus simple, l'empreinte (fingerprint), c'est une séquence de chiffres hexadécimaux qui identifie de manière unique la clé publique. Il ne peut y avoir deux empreintes identiques. Il est ainsi plus facile de vérifier l'empreinte d'une clé par téléphone.

Pour afficher l'empreinte d'une clé, il suffit de taper:

```
gpg --fingerprint veronique
```

```
pub 1024D/2CB452D2 2007-04-01
```

```
    Empreinte de la clé = 2F15 A16A 284A 347B B517 C6D5 9E39 979C 2CB4 52D2
```

```
uid          Véronique Hoarau <veronique.hoarau@funix.org>
```

```
sub 2048g/5776C745 2007-04-01
```

C'est quand même plus facile de vérifier avec l'empreinte.

6.5 Certifier une clé

Si vous êtes absolument sûr que la clé publique que vous venez de recevoir appartient bien au destinataire (par vérification de l'empreinte par exemple) et seulement dans ce cas là, vous pouvez certifier sa clé:

```
[olivier@obelix olivier]$ gpg --sign-key veronique
```

voilà le résultat

pub 1024D/2CB452D2 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: inconnu validité: inconnu
sub 2048g/5776C745 créé: 2007-04-01 expire: jamais utilisation: E
[inconnue] (1). Véronique Hoarau <veronique.hoarau@funix.org>

pub 1024D/2CB452D2 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: inconnu validité: inconnu
Empreinte de la clé principale: 2F15 A16A 284A 347B B517 C6D5 9E39 979C 2CB4
52D2

Véronique Hoarau <veronique.hoarau@funix.org>

Etes-vous vraiment sûr(e) que vous voulez signer cette clé
avec votre clé « Olivier Hoarau <olivier.hoarau@funix.org> » (83A7E9B7)

Signer réellement ? (o/N)

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Olivier Hoarau <olivier.hoarau@funix.org> »
clé de 1024 bits DSA, ID 83A7E9B7, créée le 2007-04-01

Entrez la phrase de passe:

Pour info la syntaxe de **gpg** avec l'option **--sign-key** est

gpg --sign-key UID

Avec UID une chaîne de caractère compris dans le nom de la personne ou son email. On verra
plus loin que ce n'est pas suffisant pour certifier une clé avec une confiance absolue.

6.6 Chiffrer des données

Maintenant on va envoyer un message **toto.txt** chiffré à l'utilisateur **veronique**, pour cela on va
taper

[olivier@obelix olivier]\$ **gpg -s veronique toto.txt**

voilà le résultat

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Olivier Hoarau <olivier.hoarau@funix.org> »
clé de 1024 bits DSA, ID 83A7E9B7, créée le 2007-04-01

Entrez la phrase de passe:

gpg: vérifier la base de confiance

gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle de confiance PGP

gpg: profondeur: 0 valide: 1 signé: 1

confiance: 0-. 0g. 0n. 0m. 0f. 1u

gpg: profondeur: 1 valide: 1 signé: 0

confiance: 1-. 0g. 0n. 0m. 0f. 0

Avec les options :

- **s** pour certifier, en effet comme votre clé est publique n'importe qui pourrait se faire passer pour vous en vous empruntant votre clé publique, dans ce cas il faut passer par l'étape de certification des clés publiques pour déchiffrer les données.

- **e** pour chiffrer,

- **a** pour créer un fichier .asc prêt à être envoyé en fichier attaché par email

- **r** suivi du destinataire du message

et enfin **toto.txt** le nom du fichier à chiffrer, celui contenant:

FUNIX - <http://funix.free.fr>

Mettez un pingouin dans votre PC

Cela va créer un fichier **toto.txt.asc** qui aura cette tête là:

-----BEGIN PGP MESSAGE-----

Version: GnuPG v1.4.7 (GNU/Linux)

**hQIOAyHK030iiG12EAf/bBo87wzNsobaoKusS13oEtaDy6BU4Nn56/ujOeelmimL
HqIJuVs/KhseIbC7hu20eSReyZqJ4dfbmdUZcWgWIXI2pZCyPkzKJm7YUhEGlgTT
c8cRd/Q+sB5xohjHv2+4+IYYOUC5gOHzi8vfUKK4yL9GS5kCSfeOPKtTab55ozc7
3yUerZXdQx/GWxJb2qsem2NkP+VG0iQ/QbXGLYl2Nyg/l+jtZ/fSPfSVRjLtL2Kj
4Lw+DeJtZLMDZCbl+p9k9FcPKtwxvgNkUnxH8+JVKhQuS0p3ifayZhJfJD6TCg9+r
59wWzwWRBp7zv9clKazd1b3BfGAYw6kbkwNv6N4LXwf/Yb7tD99C471pDBSq6YVJ
dVgABBKliCn3Emt2ZTQKmt4ea7R3yPMIVR5duoZxZVslPRoytVS5l2HF/ftSITpe
vvdG3Z1Aj7rxeaMJMzQvf3yBDTHqxLnORB/J3NAW7drB3/b71bIrr/1YjR6zJ50l
6wjml95SIanLWcaAae5ZueBVtv3/jaRYrSlxSvTOI803tANbUwjFIO9YsU3bpFyP
1nsouDehNUHCYZzvWm7Ep36ugnQbyRcJBuEOy7DIVFbQHPRAVtmjzsFhsw7tMur6
Wm8fC78TBTgukfHj+7vI2fN7liSHmGGWAF372G9wY61c4EFSpP5OK2EPmX0Jqv8
m9KKaFwKz/orjEPZupeo2ojlTIM6nc8j/55F5WWB6xzozke0OwZrSwRsFzPQO5wz
E11tedOn5bHDJfBHakv0Gcck9uQnwmilWOGxqsg7pmagHHkUb7BSrTejdnOA7HvK
X6hYzAANuB988jKJ3zxBIde+TlaTgy0b9cpMGIU7ltZxTcvzm1B912z0yGoEKq3+
xB6q044HYzT6lSElOqYhDdGDWb7SsM0=
=tLQT**

-----END PGP MESSAGE-----

Dans le cas où la personne n'est pas certifiée cela donne cela:

```
[olivier@obelix olivier]$ gpg -sear benjamin signature
```

cela donne

**Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Olivier Hoarau <olivier.hoarau@funix.org> »
clé de 1024 bits DSA, ID 83A7E9B7, créée le 2007-04-01**

**gpg: 6D597A58: Rien ne dit que la clé appartient vraiment à l'utilisateur
nommé.**

**pub 2048g/6D597A58 2005-08-08 Benjamin Hoarau <benjamin.hoarau@free.fr>
Empreinte de la clé principale: 8038 48D2 2AAE 5655 535E 6FEB 9520 70B4 5F61 5291
Empreinte de la sous-clé: DD3A 4CC8 9E1C E826 8663 C762 A5C7 3E2C 6D59 7A58**

**Il n'est PAS certain que la clé appartient à la personne nommée dans
le nom d'utilisateur. Si vous savez **vraiment** ce que vous faites,
vous pouvez répondre oui à la prochaine question.**

Utiliser cette clé quand même ? (o/N) o

6.7 Déchiffrer des données

L'utilisateur **veronique** pour déchiffrer le message envoyé va devoir taper :

```
[veronique@obelix temp]$ gpg -d toto.txt.asc
```

voilà le résultat

**Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Véronique Hoarau <veronique.hoarau@funix.org> »
clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01 (ID clé principale 2CB452D2)**

Entrez la phrase de passe

**gpg: chiffré avec une clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01
« Véronique Hoarau <veronique.hoarau@funix.org> »**

**FUNIX - <http://funix.free.fr>
Mettez un pingouin dans votre PC**

gpg: Signature faite le dim 01 avr 2007 12:13:04 CEST avec la clé DSA ID 83A7E9B7
gpg: Impossible de vérifier la signature: clé publique non trouvée

Vous constatez alors qu'on a le message **Impossible de vérifier la signature: clé publique non trouvée** car veronique n'a pas la clé publique d'olivier et donc n'a pu la certifier et donc même si elle peut déchiffrer le message, ne peut assurer que l'envoyeur est bien olivier. On va arranger cela, olivier doit lui donner sa clé publique, et veronique doit la mettre dans sa base de données.

gpg --import public-key-ohoarau.asc

Redéchiffrons le message pour voir maintenant:

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Véronique Hoarau <veronique.hoarau@funix.org> »
clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01 (ID clé principale 2CB452D2)

Entrez la phrase de passe

gpg: chiffré avec une clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01
« Véronique Hoarau <veronique.hoarau@funix.org> »

FUNIX - <http://www.funix.org>
Mettez un pingouin dans votre PC

gpg: Signature faite le dim 01 avr 2007 12:13:04 CEST avec la clé DSA ID 83A7E9B7
gpg: Bonne signature de « Olivier Hoarau <olivier.hoarau@funix.org> »
gpg: ATTENTION: Cette clé n'est pas certifiée avec une signature de confiance !
gpg: Rien ne dit que la signature appartient à son propriétaire.
Empreinte de clé principale: 112C 669C 28C8 75AF 5B17 7167 2F89 6D51 83A7 E9B7

La clé publique n'étant pas certifiée par veronique, on est absolument pas sûr qu'elle appartienne bien à olivier. Cependant si veronique est absolument sûre qu'olivier est bien le propriétaire de cette clé publique, elle peut la certifier en faisant:

gpg --sign-key olivier

Redéchiffrons à nouveau notre fameux message

[veronique@obelix temp]\$ gpg -d toto.txt.asc

voilà le résultat

**Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Véronique Hoarau <veronique.hoarau@funix.org> »
clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01 (ID clé principale 2CB452D2)**

Entrez la phrase de passe

**gpg: chiffré avec une clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01
« Véronique Hoarau <veronique.hoarau@funix.org> »**

**FUNIX - <http://www.funix.org>
Mettez un pingouin dans votre PC**

**gpg: Signature faite le dim 01 avr 2007 12:13:04 CEST avec la clé DSA ID 83A7E9B7
gpg: vérifier la base de confiance
gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle
de confiance PGP
gpg: profondeur: 0 valide: 1 signé: 1
confiance: 0-. 0g. 0n. 0m. 0f. 1u
gpg: profondeur: 1 valide: 1 signé: 0
confiance: 1-. 0g. 0n. 0m. 0f. 0u
gpg: Bonne signature de « Olivier Hoarau <olivier.hoarau@funix.org> »**

Le doute subsiste même si on a certifié sans ambiguïté l'authenticité de la clé publique, il manque la signature de confiance. Pour ce faire, on doit taper :

[veronique@asterix temp]\$ gpg --edit-key olivier

voilà le résultat

**gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.**

**pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: inconnu validité: entière
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[entière] (1). Olivier Hoarau <olivier.hoarau@funix.org>**

**Commande> sign
« Olivier Hoarau <olivier.hoarau@funix.org> » a déjà été signé par la clé 2CB452D2
Rien à signer avec la clé 2CB452D2**

Commande> trust

pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: inconnu validité: entière
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[entière] (1). Olivier Hoarau <olivier.hoarau@funix.org>

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 ? 5

Voulez-vous vraiment donner une confiance ultime à cette clé ? (o/N) o

pub 1024D/83A7E9B7 créé: 2007-04-01 expire: jamais utilisation: SC
confiance: ultime validité: entière
sub 2048g/9B601F37 créé: 2007-04-01 expire: jamais utilisation: E
[entière] (1). Olivier Hoarau <olivier.hoarau@funix.org>

Notez que la validité affichée pour la clé n'est pas nécessairement correcte tant que vous n'avez pas relancé le programme.

Commande> quit

On redécrypte maintenant le fichier

[veronique@asterix temp]\$ gpg -d toto.txt.asc

voilà le résultat

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Véronique Hoarau <veronique.hoarau@funix.org> »
clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01 (ID clé principale 2CB452D2)

Entrez la phrase de passe:

gpg: chiffré avec une clé de 2048 bits ELG-E, ID 5776C745, créée le 2007-04-01
« Véronique Hoarau <veronique.hoarau@funix.org> »

FUNIX - <http://www.funix.org>

Mettez un pingouin dans votre PC

gpg: Signature faite le dim 01 avr 2007 12:13:04 CEST avec la clé DSA ID 83A7E9B7

gpg: vérifier la base de confiance

**gpg: 3 marginale(s) nécessaires, 1 complète(s) nécessaires, modèle
de confiance PGP**

gpg: profondeur: 0 valide: 2 signé: 0

confiance: 0-. 0g. 0n. 0m. 0f. 2u

gpg: Bonne signature de « Olivier Hoarau <olivier.hoarau@funix.org> »

Cette fois-ci c'est bon !

6.8 S'authentifier et authentifier

Si vous voulez vous authentifier auprès des autres personnes, créer un fichier **signature** quelconque du style:

Veronique Hoarau
veronique.hoarau@fnac.net

Puis taper

gpg -sa signature

voilà le résultat

Vous avez besoin d'une phrase de passe pour déverrouiller la
clé secrète pour l'utilisateur: « Véronique Hoarau <veronique.hoarau@funix.org> »
clé de 1024 bits DSA, ID 2CB452D2, créée le 2007-04-01

Maintenant si vous voulez authentifier quelqu'un qui vous a envoyé sa signature cryptée avec sa clé privée, il vous faudra sa clé publique et taper:

[olivier@obelix temp]\$ gpg --verify signature.asc

voilà le résultat

gpg: Signature faite le dim 01 avr 2007 12:25:46 CEST avec la clé DSA ID 2CB452D2
gpg: Bonne signature de « Véronique Hoarau <veronique.hoarau@funix.org> »

Au message **Bonne signature** on voit bien que la signature appartient bien à la personne dont vous avez la clé publique. Vous pouvez obtenir des warnings dans le cas où la personne n'est pas certifiée.

D'une autre manière quand vous récupérerez des fichiers sous internet (des packages par exemple), c'est un bon moyen de voir que la package vient bien du créateur et n'est pas un package détournée avec des backdoors à l'intérieur.

Ainsi on peut trouver le tarball de **GnuPG** signé, ainsi que la clé publique du projet **GnuPG** (fichier **samplekeys.asc** qu'on trouve sous **/gnupg-1.4.7/doc**), vous pouvez ainsi authentifier l'origine du package avec le fichier signature qu'on peut trouver en téléchargement sur le site de **GnuPG**, tout d'abord il faut importer la clé publique de **GnuPG**

```
gpg --import /usr/local/linux/securite/gnupg-1.4.7/doc/samplekeys.asc
```

vous mettez le chemin absolu de **gnupg**, voilà le résultat

```
gpg: clé 57548DCD: clé publique « Werner Koch (gnupg sig) <dd9jn@gnu.org> » importée  
gpg: clé 5B0358A2: clé publique « Werner Koch <wk@gnupg.org> » importée  
gpg: clé B2D7795E: clé publique « Philip R. Zimmermann <prz@mit.edu> » importée  
gpg: clé 99242560: clé publique « David M. Shaw <dshaw@jabberwocky.com> » importée  
gpg: clé CA57AD7C: clé publique « PGP Global Directory Verification Key » importée  
gpg: clé 1CE0C630: clé publique « Werner Koch (dist sig) <dd9jn@gnu.org> » importée  
gpg:      Quantité totale traitée: 6  
gpg:      importée: 6 (RSA: 3)
```

et maintenant pour vérifier l'intégrité de l'archive on tapera

```
gpg --verify gnupg-1.4.7.tar.bz2.sig
```

cela donne cela

```
gpg: Signature faite le lun 05 mar 2007 10:54:17 CET avec la clé RSA ID 1CE0C630  
gpg: Bonne signature de « Werner Koch (dist sig) <dd9jn@gnu.org> »  
gpg: ATTENTION: Cette clé n'est pas certifiée avec une signature de confiance !  
gpg:      Rien ne dit que la signature appartient à son propriétaire.  
Empreinte de clé principale: 7B96 D396 E647 1601 754B E4DB 53B6 20D0 1CE0 C630
```

Les fichiers **gnupg-1.4.7.tar.bz2.sig** et **gnupg-1.4.7.tar.bz2** doivent se trouver dans le même répertoire.

6.9 Editer ou supprimer des clés

Pour lister les clés publiques de votre base de données:

gpg --list-key

voilà le résultat

/export/home/olivier/.gnupg/pubring.gpg

pub 1024D/83A7E9B7 2007-04-01

uid Olivier Hoarau <olivier.hoarau@funix.org>

sub 2048g/9B601F37 2007-04-01

pub 1024D/2CB452D2 2007-04-01

uid Véronique Hoarau <veronique.hoarau@funix.org>

sub 2048g/5776C745 2007-04-01

pub 1024D/5F615291 2005-08-08

uid Benjamin Hoarau <benjamin.hoarau@free.fr>

sub 2048g/6D597A58 2005-08-08

Pour supprimer une clé publique d'un utilisateur (si par exemple on lui a volé sa clé privée)

gpg --delete-key UID

Editer la clé publique d'un utilisateur particulier

gpg --edit-key UID