GnuPG

Olivier Hoarau (olivier.hoarau@funix.org)

V1.9 du 15 octobre 2015

1Historique du document....2
2Préambule....2
3Présentation....3
4Installation....3
5Comment ça marche....4
6Utilisation....5
6.1Créer une clé....5
6.2Exporter une clé publique....6
6.3Importer une clé publique....8
6.4Vérifier l'empreinte....9
6.5Certifier une clé....9
6.6Chiffrer des données....10
6.7Déchiffrer des données....11
6.8S'authentifier et authentifier....15
6.9Editer ou supprimer des clés....16
7Intégration de GnuPG à thunderbird....17

1 Historique du document

V2.0 24.12.16 Passage à la version 2.0.30

V1.9 15.10.15 Passage à la version 2.0.29

V1.8 03.11.08 Passage à 1.4.9, intégration à Thunderbird

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.

- **Partage des Conditions Initiales à l'Identique** — Si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Par ailleurs 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**, on installera préalablement les packages suivants **lib64pth-devel**, **lib64assuan-devel** et **lib64ksba-devel**.

Vous allez récupérer **GnuPG** sur www.gnupg.org, l'archive se présente sous la forme d'un tarball qu'on décompresse en tapant

tar xvfj gnupg-2.0.30.tar.bz2

Cela va créer un répertoire **gnup-2.0.30** dans le répertoire courant. Dans ce répertoire tapez:

./configure

voilà le résultat

GnuPG v2.0.30 has been configured as follows:

Revision: 83cae8c (33738)
Platform: GNU/Linux (x86_64-unknown-linux-gnu)

OpenPGP: yes
S/MIME: yes
Agent: yes
Smartcard: yes (without internal CCID driver)
Gpgtar: no

Protect tool: (default)
Default agent: (default)
Default pinentry: (default)
Default scdaemon: (default)
Default dirmngr: (default)

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 **gpg-conf.skel** ainsi qu'une FAQ au format **html** et les exécutables **gpg2** 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 privés 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 receviez soit bien celle de votre destinataire, pour cela vous devez certifier la clé publique, vous ne devez en aucun cas certifiez 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]$ gpg2 --gen-key

voilà le résultat

gpg (GnuPG) 2.0.30; Copyright (C) 2015 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.

Sélectionnez le type de clef désiré :
(1) RSA et RSA (par défaut)
(2) DSA et Elgamal
(3) DSA (signature seule)
(4) RSA (signature seule)
Quel est votre choix ? 2
les clefs DSA peuvent faire une taille comprise entre 1024 et 3072 bits.
Quelle taille de clef désirez-vous ? (2048)
La taille demandée est 2048 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
0 = la clef n'expire pas
<n> = la clef expire dans n jours
<n>w = la clef expire dans n semaines
<n>m = la clef expire dans n mois
<n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 0
La clef n'expire pas du tout
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : Olivier Hoarau
Adresse électronique : olivier.hoarau@funix.org
Commentaire :
Vous avez sélectionné cette identité :
« Olivier Hoarau <olivier.hoarau@funix.org> »

Faut-il modifier le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? O
Une phrase secrète est nécessaire pour protéger votre clef secrète.

on saisit alors sa phrase secrète dans une fenêtre

De nombreux octets aléatoires doivent être générés. 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 nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: Attention : certains programmes OpenPGP ne peuvent pas gérer de clef DSA avec cette taille de hachage
De nombreux octets aléatoires doivent être générés. 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 nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: clef 5038487B marquée de confiance ultime.
les clefs publique et secrète ont été créées et signées.

gpg: vérification de la base de confiance
gpg: 3 marginale(s) nécessaire(s), 1 complète(s) nécessaire(s), modèle de confiance PGP
gpg: profondeur : 0 valables : 1 signées : 0
confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
pub 2048D/5038487B 2015-10-15
Empreinte de la clef = A215 5B62 FC02 A0C4 5122 444A 0860 5DFD 5038 487B
uid [ultime] Olivier Hoarau <olivier.hoarau@funix.org>
sub 2048g/D445AE4D 2015-10-15

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.

Attention, s'il existe une version préinstallé de **GnuPG** sur votre système, vous pouvez soit la supprimer, l'écraser, ou alors spécifier le chemin de **gpg** version tarball (**/usr/local/bin** pour une installation par défaut).

6.2 Exporter une clé publique

Pour exporter une clé, vous devez signer votre propre clé publique pour qu'elle puisse être importée par vos interlocuteurs.

[olivier@asterix olivier]$ gpg2 --edit-key olivier

voilà le résultat

gpg (GnuPG) 2.0.30; Copyright (C) 2015 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 clef secrète est disponible.

pub 2048D/5038487B créé : 2015-10-15 expire : jamais utilisation : SC
confiance : ultime validité : ultime
sub 2048g/D445AE4D créé : 2015-10-15 expire : jamais utilisation : E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>

gpg> trust
pub 2048D/5038487B créé : 2015-10-15 expire : jamais utilisation : SC
confiance : ultime validité : ultime
sub 2048g/D445AE4D créé : 2015-10-15 expire : jamais utilisation : E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>

Décidez maintenant de la confiance que vous portez en cet utilisateur pour vérifier les clefs des autres utilisateurs (en regardant les passeports, en vérifiant les empreintes depuis diverses sources, etc.)

1 = je ne sais pas ou n'ai pas d'avis
2 = je ne fais PAS confiance
3 = je fais très légèrement confiance
4 = je fais entièrement confiance
5 = j'attribue une confiance ultime
m = retour au menu principal

Quelle est votre décision ? 5
Voulez-vous vraiment attribuer une confiance ultime à cette clef ? (o/N) o

pub 2048D/5038487B créé : 2015-10-15 expire : jamais utilisation : SC
confiance : ultime validité : ultime
sub 2048g/D445AE4D créé : 2015-10-15 expire : jamais utilisation : E
[ultime] (1). Olivier Hoarau <olivier.hoarau@funix.org>

Pour exporter une clé, maintenant rien de plus simple

gpg2 --export --armor > public-key-ohoarau.asc

Cela va donner le fichier suivant.

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQMuBFHe9bURCACxKJ6UguvZ5/DfgldACFTuSWKT1TTV/CwtjfZgqT4SNM146ee
d
xwT1x3qmvge4Gtpo1O/L/RIbNd9LUEPsgve+uOFizXO1qqprgZfrBWcQGFH0DyLl
YyDZnSvLpVL19AT0x8oaRrqCMZI14xrTU0oetIYn1cpePK/4ZFowc7msD4tXkfv/
Q/iX4Mj4vYJGmjN0Jxh4K7e4PIvsPfd9EQoLbKYauJ5oPqbP/BSB7hoRLZhQy4PN
z1jyEj7oPZRTElNy39CCKV0dEdwz/e4qmWH4Qgm/bt2qQe+DFxdMAaFzFnf2kJEk
1OIJOgu5z/J8nuoyKbrFopbK3ejef0UDM6r3AQDEUz0DyzUK7I5KXz7EUKBTE+yd

hN1GgI+Inu25dr3Xcwf+LK/8Up1nmfiSQramW53jUl2Cehy/TD13+QZd31nLxsjh
1f4ZUAGj47PizaUqa2+ycOEenp2mobTyKFdXEIA1el/RIjMj5raLQllrYKanTR01
3fxChMIOKNoX9CF5qr7keyRHaZHPG85Ae8iM5ADSNxrRHE/BLrcoH/z8NUoJ2Gmf
jFNHT5WPXewl0SsQUrPWoiIdjJO9KQUBZDL1I+pwKAQ9kIrx02A+xW1W+e91NmXv
7q27G6lStzaUG02pPqMeSzlX0E7BA/4/WotwUv8lO1PbcBbl4BFOKhkeDG+eER2u
rYzRQYq3q0nG5RaTCJy0yJiv+ENkLiNfn4494yRaNgf/e8iNxSHuOfNAmgQ+NqMj
2qHQonvQHuepcHN1WaLcNPAfNxUZ3K+eI+2qZ6ftYEJq/pH7Yk7g0VN4cIrMSTeC
Y9ne56NeZYNgE8tKbCzDyPfr3Txz1+2sm7AV59B+TG2zVFnlIuMC0YGAvO1wxkVE
hLQQppJ+zcfuQrVD0LHObiH1mVmaLugRWJzK5dorD6QKy7dIFkdnqd89cJ3t3op6
FwSNDJjQ304bgTtlxKudzH1oaL/W86LvYv4qwgitH1fgdPnWq3bjXrMal2rTlNcg
C7DImggFPfYXXSnN+Cjl6WtNJqtnwOJjeDWUKbKb4EcqMj7X4E5CwddRdRVHoE
pH
T7QpT2xpdmllciBIb2FyYXUgPG9saXZpZXIuaG9hcmF1QGZ1bml4Lm9yZz6IegQT
EQgAIgUCUd71tQIbAwYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQKo
pgu/H8
Xw/0SwD/XJ5x8cD0ZB1onzpowLHpY21zv6ra5uZkZoJePoFNfe4A/j4SxMCZbm3j
OuY1RAXYfSdkf61/9R+SYJWecHc2NfthuQINBFHe9bUQCACVsbo4lKoST+sWxViw
C3bycOD1wtKyjPsAysdN9cFegEIPQ0d62q4+TLZHHGtAMtD6pR/n5GnGhJQWd/rD
/esx8IKzQkOjRlcA1HST4UqVNakUpDcxAQgMkHXvHRLw6RvRbiBpXQq/oYT/l64I
oYZeKDq3IRL1eJLm1PlfHOIh7ekkTUvvWBPVygKOmTnA9Y9/YLQQjc+gvzSp1OB
k
ydJMph7sM3GuKuLNg8D6NM6lXL6NQ1Yb7Tm7B+/HvBLs5bshDlxfwmnz7Cm2gm
MP
dB13MtcKaaqSkU5pwSJz3n+NEUag1uMIKwQj6QSbWdEDfu7GMcSRA3/o6L76QqeR
izKfAAMFB/0SagiSXGL735qVvcErCoZWk2jeWoAxsrLfLWGR2PymuKUycs9qRei7
oZE13YZhRDL4IgVEZYBEIByliiaflgjgtw7rPApI4zfzt3oo3zpnVs/JILuyeZyx
0TXYwA6LtqhX7z/J4uJ61bJCtKMXstUWIhtGfZnlN94q8qgpjc5sdoAyQue9Yj9U
Gb7HBmwL4wNdAEmdAYDO2q165F9/RYBhMqXGEkvMHU08xWJykMtH3zHYNp
CZXIbg
B1YOyXGtmIBP20DLR6zsj2B7TjJL+nT7HCSH0A5owTNHgfCcTeb77GyPrr1+grXG
t6ihyH0Hlk9+HVZdR3L+6tVEhoE2EvfviGEEGBEIAAkFAlHe9bUCGwwACgkQKopg
u/H8Xw9SGQD/eqUVqfTC9aFDVTGCotHHCs2rLDSv3Hn+BW7nk5zPMUkA/2znhA7
g
mgAYsUA6HS5iXqx2VAwhbdH0syDgHhiS1chL
=yIcf
-----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]$ gpg2 --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:

gpg2 --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]$ gpg2 --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

gpg2 --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]$ gpg2 -sear 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 2048 bits DSA, ID F1FC5F0F, créée le 2013-07-11**

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 v2**

**hQIOAyHK030iiG12EAf/bBo87wzNsobaoKusS13oEtaDy6BU4Nn56/ujOeelmimL
HqIJuVs/KhseIbC7hu20eSReyZqJ4dfbmdUZcWgWIXI2pZCyPkzKJm7YUhEGlgTT**

c8cRd/Q+sB5xohjHv2+4+lYYOUC5gOHzi8vfUKK4yL9GS5kCSfeOPKtTab55ozc7
3yUerZXdQx/GWxJb2qsem2NkP+VG0iQ/QbXGLYl2Nyg/l+jtZ/fSPfSVRjLtL2Kj

(...)

1nsouDehNUHCYZzvWm7Ep36ugnQbyRcJBuEOy7DlVFbQHpRAVtmjzsFhsw7tMur6
Wm8fC78TBiTgukfHj+7vI2fN7IiSHmGGWAF372G9wY61c4EFSpP5OK2EPmX0Jqv8
m9KkAfwKz/orjEPZupeo2ojlTIM6nc8j/55F5WWB6xzozke0OwZrSwRsFzPQO5wz
E11tedOn5bHDJfBHakv0Gcck9uQnwmiLwOGxqsg7pmagHHkUb7BSrTejdnOA7HvK
X6hYzAANuB988jKJ3zxBIdE+TlaTgy0b9cpMGIU7ltZxTcvzm1B912z0yGoEKq3+
xB6q044HYzT61SElOqYhDdGDWb7SsM0=
=tLQT
-----END PGP MESSAGE-----

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

[olivier@obelix olivier]$ gpg2 -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 nomée 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]$ gpg2 -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.

gpg2 --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:

gpg2 --sign-key olivier

Redéchiffrons à nouveau notre fameux message

[veronique@obelix temp]$ gpg2 -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]$ gpg2 --edit-key olivier

voilà le résultat

**gpg (GnuPG) 2.0.30; Copyright (C) 2015 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.**

**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 redéchiffre maintenant le fichier

[veronique@asterix temp]$ gpg2 -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

gpg2 -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]$ gpg2 --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-2.0.30/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**

gpg2 --import /usr/local/linux/securite/gnupg-2.0.30/doc/samplekeys.asc

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

gpg: clef 5B0358A2 : clef publique « Werner Koch <wk@gnupg.org> » importée
gpg: clef B2D7795E : clef publique « Philip R. Zimmermann <prz@mit.edu> » importée
gpg: clef 57548DCD : clef publique « Werner Koch (gnupg sig) <dd9jn@gnu.org> » importée
gpg: clef 1CE0C630 : clef publique « Werner Koch (dist sig) <dd9jn@gnu.org> » importée
gpg: clef 1E42B367 : clef publique « Werner Koch <wk@gnupg.org> » importée
gpg: clef 87978569 : clef publique « Marcus Brinkmann <Marcus.Brinkmann@ruhr-uni-bochum.de> » importée
gpg: clef 4F25E3B6 : clef publique « Werner Koch (dist sig) » importée
gpg: clef 99242560 : clef publique « David M. Shaw <dshaw@jabberwocky.com> » importée
gpg: Quantité totale traitée : 8
gpg: importées : 8 (RSA: 3)
gpg: 3 marginale(s) nécessaire(s), 1 complète(s) nécessaire(s),
modèle de confiance PGP
gpg: profondeur : 0 valables : 1 signées : 0
confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.

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

gpg2 --verify gnupg-2.0.30.tar.bz2.sig

cela donne cela

gpg: assuming signed data in './gnupg-2.0.30.tar.bz2'
gpg: Signature faite le mar. 08 sept. 2015 16:38:22 CEST avec la clef RSA d'identifiant 4F25E3B6
gpg: Bonne signature de « Werner Koch (dist sig) » [inconnu]
gpg: Attention : cette clef n'est pas certifiée avec une signature de confiance.
gpg: Rien n'indique que la signature appartient à son propriétaire.
Empreinte de clef principale : D869 2123 C406 5DEA 5E0F 3AB5 249B 39D2 4F25 E3B6
gpg: Signature faite le mer. 09 sept. 2015 12:30:24 CEST avec la clef RSA d'identifiant 33BD3F06
gpg: Impossible de vérifier la signature : Pas de clef publique

Les fichiers **gnupg-2.2.29.tar.bz2.sig** et **gnupg-2.0.30.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:

gpg2 --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)

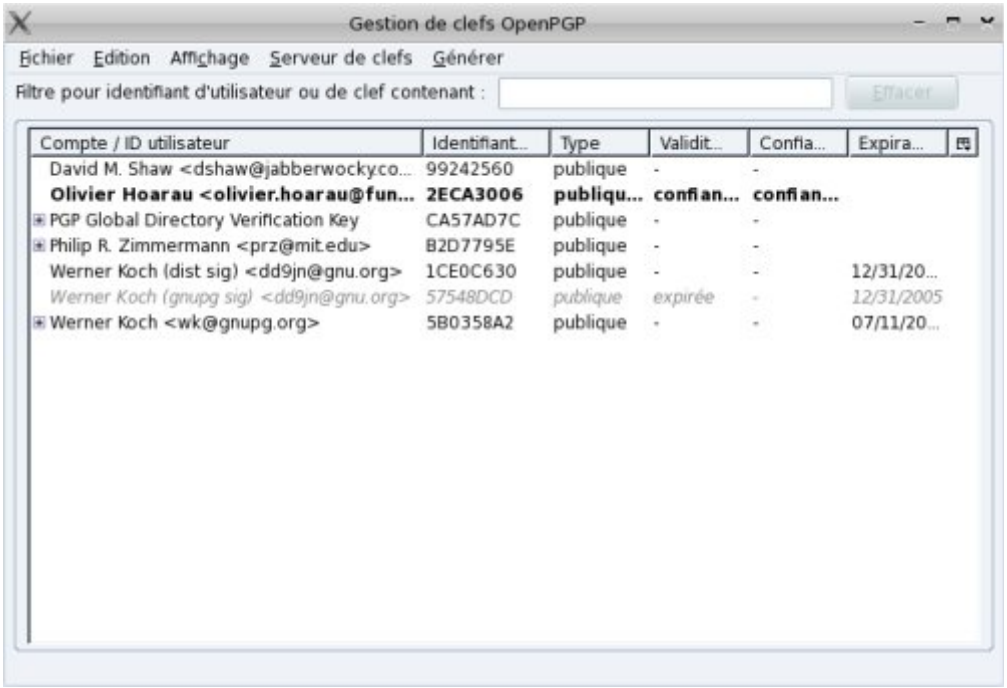
gpg2 --delete-key UID

Editer la clé publique d'un utilisateur particulier

gpg2 --edit-key UID

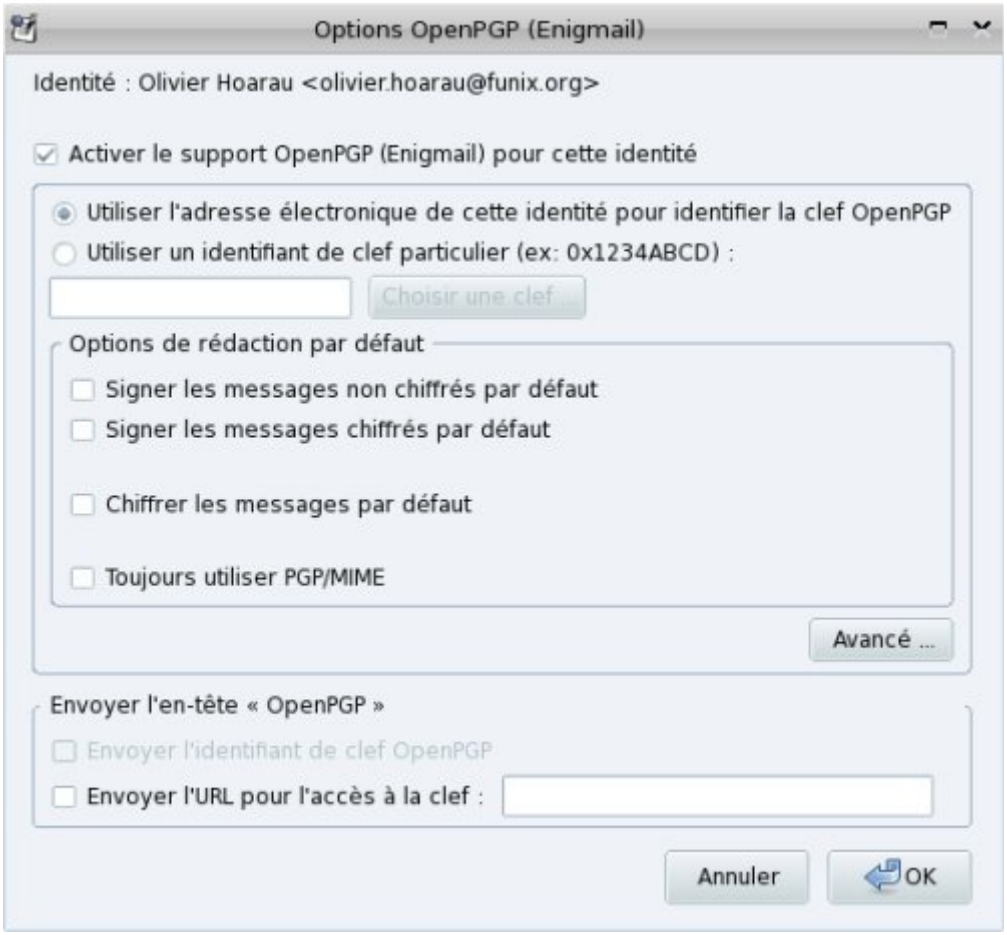
7 Intégration de GnuPG à thunderbird

Il suffit d'installer le plugin **enigmail** (Menu **outils** puis **Modules complémentaires**). Cela va créer dans la barre de menu **OpenPGP**, voilà ce que ça donne en choisissant **Gestion des clefs**

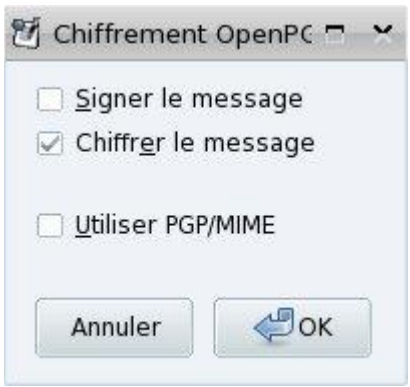


on peut charger des nouvelles clefs à partir du menu **Fichier** ou en régénérer avec le menu **Générer**. Pour signer une clef ou définir le niveau de confiance il suffit de la sélectionner puis d'accéder à la fonction à partir du menu accessible avec le bouton droit de la souris.

Maintenant pour chiffrer un mail il suffit de l'éditer (ou de le créer) vous avez une icône **OpenPGP**, en cliquant dessus on obtient



puis quand on clique sur OK on obtient



Attention il faut que vous disposiez de la clef publique de votre destinataire.