

# Manuel utilisateur Linux

## **SOMMAIRE:**

Introduction

I. Installations

- Remarque

- Les modules utiles

- GnuPG

II. Créer un volume chiffré

- Créer le volume chiffré

- Montage

- Démontage

- Conclusion

III. Utilisation des scripts

- Créer son trousseau de clé

- Changez la clé par défaut

- Chiffrement des fichiers

- Déchiffrement des fichiers

- Le Gardien

Conclusion

# Introduction

Nous allons vous expliquer à travers cette documentation, comment créer un volume chiffré. Ce volume chiffré sera utilisable comme tout autre volume physique, avec pour particularité d'être hautement sécurisé contre l'intrusion et l'espionnage. La méthode qui vous sera expliquée est celle utilisée par les banques pour le transfert d'informations, tout comme les services secrets.

Nous allons ensuite vous montrer, comment à l'aide de nos scripts, vous serez opérationnels à chiffrer vos fichiers contenus dans ce volume virtuel asymétriquement grâce à l'excellent logiciel GnuPG.

Tout d'abord, il est utile de préciser que notre système d'exploitation est "Ubuntu 7.10 - Gutsy Gibbon - sortie en octobre 2007". La description suivante ne sera donc pas universelle mais le modèle restera le même à quelques détails près, que nous préciserons.

## I. Installations

### **-Remarque**

Il vous faut savoir que si votre volume est créé pour naviguer entre machine linux et windows, les caractères comportant des accents ne seront pas reconnus. Il est donc impératif de choisir un nom de volume sans accents, et obligatoire de choisir un mot de passe contenant uniquement des caractères de base.

Si par contre votre volume est destiné à ne voyager que de machine linux vers d'autre machine linux (idem pour windows), les accents ne posent plus de problème, ils seront reconnus comme caractères normaux.

### **-Les modules utiles**

Le chiffage de données nécessite l'utilisation de commandes qui ne sont peut être pas présentes sur votre ordinateur. Nous allons donc vous guider aux procédés d'installation de ceux ci.

Le chiffage utilise notamment la commande "cryptsetup". Pour l'installer (il n'y a rien de plus simple) il vous suffit d'ouvrir une fenêtre de Terminal (aussi appelée "shell", pour accéder à cette fenêtre allez dans l'onglet d'application de votre bureau et parmi les accessoires vous trouverez le Terminal), et d'y entrer la ligne suivante:

```
$ sudo apt-get install cryptsetup
```

Ici, votre mot de passe de session vous sera demandé. Quelques secondes après, le Terminal vous donnera la taille du module à installer et vous demande si vous souhaitez poursuivre. Répondez par 'O' puis tapez 'entrée' et laissez l'installation suivre son cour. Cette opération prend une petite minute.

Afin de vérifier si celui ci s'est bien installé, tapez la ligne suivante dans le terminal:

```
$ man cryptsetup
```

Une fenêtre de manuel d'utilisation du cryptsetup doit s'ouvrir:

```
stage@stage-desktop: ~
Fichier Édition Affichage Terminal Onglets Aide
CRYPTSETUP(8)      Maintenance Commands      CRYPTSETUP(8)

NAME
  cryptsetup - setup cryptographic volumes for dm-crypt (including LUKS
  extension)

SYNOPSIS
  cryptsetup <options> <action> <action args>

DESCRIPTION
  cryptsetup is used to conveniently setup up dm-crypt managed device-
  mapper mappings. For basic dm-crypt mappings, there are five opera-
  tions.

ACTIONS
  These strings are valid for <action>, followed by their <action args>:

  create <name> <device>

  creates a mapping with <name> backed by device <device>.
  <options> can be [--hash, --cipher, --verify-passphrase, --key-
  file, --key-size, --offset, --skip, --readonly]
```

Manual page cryptsetup(8) line 1

Malheureusement, notre commande n'est pas compatible avec notre noyau (kernel), il suffit alors de télécharger une image noyau. Une fois installée, il sera nécessaire de redémarrer pour que le nouveau noyau s'associe aux commandes. L'utilisation du cryptsetup vous sera donc possible. Pour télécharger ce dernier, il vous faut connaître quel architecture votre machine utilise.

Je vous donne rendez vous sur la page suivante: <http://packages.ubuntu.com/>  
Ici une page contenant une liste de systèmes s'affiche, sélectionnez le votre:

### Ubuntu Packages Search

This site provides you with information about all the packages available in the [Ubuntu](#) Package archive.  
*Please contact [Frank Lichtenheld](#) if you encounter any problems!*

#### Browse through the lists of packages:

- [dapper](#) (6.06LTS)
- [dapper-updates](#)
- [dapper-backports](#)
- [edgy](#) (6.10)
- [edgy-updates](#)
- [edgy-backports](#)
- [feisty](#) (7.04)
- [feisty-updates](#)
- [feisty-backports](#)
- [gutsy](#) (7.10)
- [gutsy-updates](#)
- [gutsy-backports](#)
- [hardy](#)

There is also a list of [packages recently added to hardy](#).

Ensuite cliquez en fin de page sur la liste complète de paquets (All packages):

lications intégrées facile à utiliser.  
 : [développement pour les bibliothèques](#).  
 rs nécessaires aux développeurs pour utiliser les  
 thèques.  
[ies](#)  
 thèques utilisées par d'autres programmes et fournissant  
 nctionnalités aux programmeurs.

et d'archivage, de surveillance du système, systèmes d'er  
 etc.

**Paquets virtuels**

Paquets virtuels

**Logiciels pour le Web**

Serveurs Web, navigateurs, serveurs mandataires, outils c  
 téléchargement, etc.

**Logiciels pour le système X Window**

Serveurs X, bibliothèques, gestionnaires de fenêtres, ému  
 de terminal et autres applications associées.

[All packages](#)  
 ([liste au format texte compressée](#))

Ici, les modules sont classés par ordre alphabétique, rendez vous à la lettre K où vous  
 trouverez les images kernel. Sélectionnez celle qui correspond à votre machine (pour  
 nous ce sera la version x86 (quatrième de la liste ci dessous):

- [keepassx](#) (0.2.2-2) [[universe](#)]  
 Cross Platform Password Manager
- [kenolaba](#) (4:3.5.8-0ubuntu1)  
 Enolaba board game for KDE
- [kernel-image](#)  
 virtual package provided by [kernel-image-2.6.22-14-cell-dj](#), [kernel-image-2.6.22-14-generic-dj](#), [kernel-image-2.6.22-14-powerpc-dj](#),  
[kernel-image-2.6.22-14-powerpc64-smp-dj](#), [kernel-image-2.6.22-14-386-dj](#)
- [kernel-image-2.6.22-14-386-dj](#) (2.6.22-14.46)  
 Linux kernel binary image for the Debian installer
- [kernel-image-2.6.22-14-cell-dj](#) (2.6.22-14.46)  
 Linux kernel binary image for the Debian installer
- [kernel-image-2.6.22-14-generic-dj](#) (2.6.22-14.46)  
 Linux kernel binary image for the Debian installer
- [kernel-image-2.6.22-14-powerpc-dj](#) (2.6.22-14.46)  
 Linux kernel binary image for the Debian installer
- [kernel-image-2.6.22-14-powerpc64-smp-dj](#) (2.6.22-14.46)  
 Linux kernel binary image for the Debian installer
- [kernel-installer](#)  
 virtual package provided by [base-installer](#)
- [kernel-internals-guide](#) (20020807-3) [[universe](#)]  
 Linux Kernel 2.4 Internals Guide
- [kernel-package](#) (11.001)  
 A utility for building Linux kernel related Debian packages.
- [kernel-patch-adeos](#) (20060329-1) [[universe](#)]  
 ADEOS nanokernel for sharing hardware resources
- [kernel-patch-atopacct](#) (1:1.20-1) [[universe](#)]  
 save additional statistical counters for atop in the record

Cliquez ensuite dans le tableau sur le nom de l'architecture:

## Paquet : kernel-image-2.6.22-14-386-d

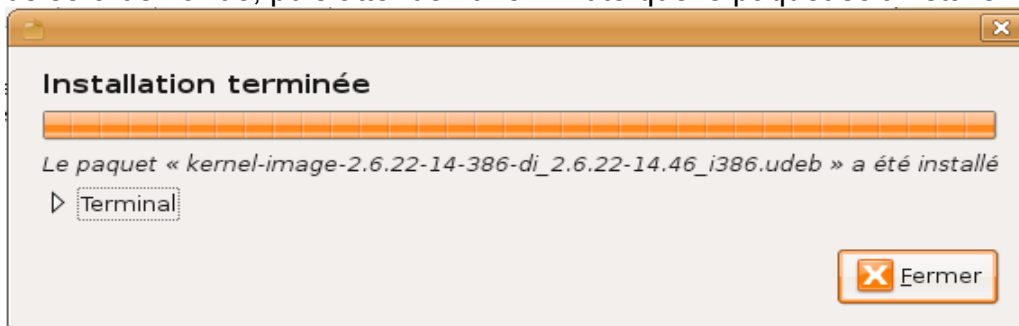
<b>paquet de l'installateur Debian udeb</b>	Link: <hr/>								
Warning: This package is intended for the use in building <a href="#">debian-installer</a> images only. Do not install it on a normal Ubuntu system.	Ubuntu: <ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li></ul>								
<b>Linux kernel binary image for the Debian installer</b>	Download: <ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li></ul>								
This package contains the Linux kernel image for the Debian installer boot images. It does <code>_not_</code> provide a usable kernel for your full Debian system.	Maint: <ul style="list-style-type: none"><li>•</li></ul>								
<b>Télécharger kernel-image-2.6.22-14-386-di</b>	Similar: <hr/>								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Architecture</th> <th style="text-align: left;">Taille du paquet</th> <th style="text-align: left;">Espace occupé</th> <th style="text-align: left;">Fichiers</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;"><b>i386</b></td> <td style="text-align: left;">1,792.5 kB</td> <td style="text-align: left;">2468 kB</td> <td style="text-align: left;">no current information</td> </tr> </tbody> </table>	Architecture	Taille du paquet	Espace occupé	Fichiers	<b>i386</b>	1,792.5 kB	2468 kB	no current information	
Architecture	Taille du paquet	Espace occupé	Fichiers						
<b>i386</b>	1,792.5 kB	2468 kB	no current information						

Celui ci vous redirigera sur une page contenant toutes les sources de téléchargement. Choisissez la première de la catégorie Europe par exemple puis le téléchargement se lancera automatiquement.

Choisissez de l'ouvrir avec l'installateur de paquets GDebi:



Après quelques secondes celui ci va s'ouvrir. Cliquez donc sur "installer". Votre mot de passe vous sera demandé, puis attendez une minute que le paquet soit installé:



Maintenant, redémarrez votre machine proprement afin d'associer votre nouveau noyau aux nouvelles commandes.

## **-GnuPG**

Pour commencer, il vous faut ouvrir le terminal et vous mettre en super-utilisateur (root) et taper la commande suivante:

```
$ apt-get install gnupg
```

Cette opération peut prendre du temps selon la puissance de votre machine et des éléments présents sur celle ci .

## **II.Créer un volume chiffré**

### **-Créer le volume chiffré**

Dans cette partie, nous vous expliquerons en détails comment monter un volume qui sera chiffré ici en AES et hacher en SHA-256. Beaucoup d'autre algorithmes et méthodes de hachage sont disponibles mais celle que nous utiliserons est la plus réputée car elle est la plus difficile a déchiffrer et que celle ci est la plus rapide.

Sachez que toutes les opérations qui suivront en italique seront les commandes à entrer dans la fenêtre de terminal. Afin de ne pas devoir entrer "sudo" en chaque début de commande (nécessaire pour se mettre en superutilisateur), une fois la fenêtre ouverte

entrez la ligne suivante et tapez votre mot de passe:

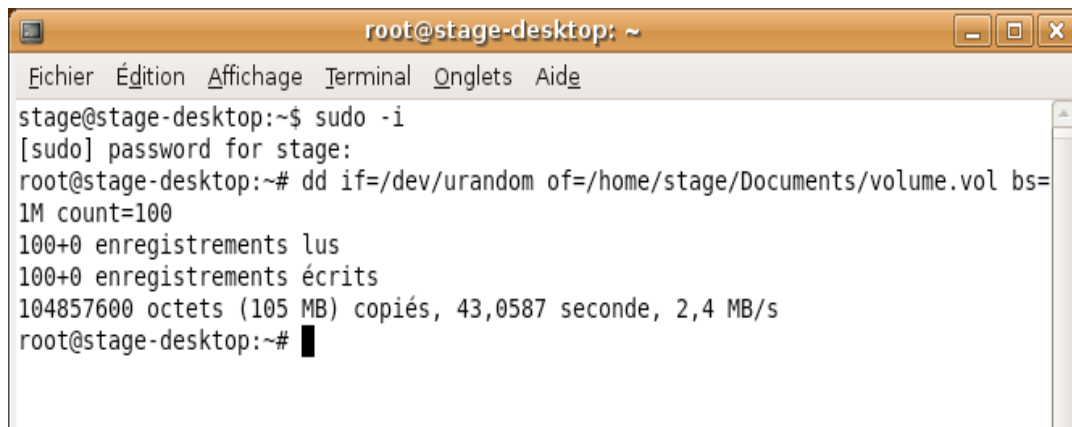
```
$ sudo -i
```

Tout d'abord, nous devons créer un fichier quelconque, ce sera notre volume. Il sera transportable et peut être de la taille que l'on souhaite. Ce fichier que nous allons créer sera basé sur la fonction "urandom", fonction linux qui génère une suite de symboles aléatoires. A la création de ce fichier, plutôt que d'être vide comme n'importe quel autre périphérique vide, il sera déjà rempli ce qui ajoutera une sécurité de plus contre les attaques de déchiffrement.

Pour créer un fichier "volume.vol" ayant une taille de 100Mo dans votre répertoire "Documents" et que votre nom de session est "stage" il vous faut entrer:

```
$ dd if=/dev/urandom of=/home/stage/Documents/volume.vol bs=1M count=100
```

Patiencez le temps que votre fichier soit créé. Plus votre fichier aura une taille importante, plus le temps de création sera conséquent. Vous obtiendrez donc ceci:



```
root@stage-desktop: ~
Fichier Édition Affichage Terminal Onglets Aide
stage@stage-desktop:~$ sudo -i
[sudo] password for stage:
root@stage-desktop:~# dd if=/dev/urandom of=/home/stage/Documents/volume.vol bs=
1M count=100
100+0 enregistrements lus
100+0 enregistrements écrits
104857600 octets (105 MB) copiés, 43,0587 seconde, 2,4 MB/s
root@stage-desktop:~# █
```

Maintenant, associons ce fichier à un des périphériques de boucle, ceux ci se trouvant dans le répertoire "dev" et sont nommée "loop0", "loop1", "loop2"..., jusque "loop7":

```
$ losetup /dev/loop0 /home/stage/Documents/volume.vol
```

Maintenant faisons entrer en jeu le chiffrement et notre module installer précédemment (chapitre I). Cette fonction va créer un "device mapper" de chiffrement, donnez le même nom à celui ci afin de ne pas se tromper:

```
$ cryptsetup -y -c aes-cbc-essiv:sha256 create volume /dev/loop0
```

Ici on vous demandera le mot de passe qui protégera votre volume et qui servira aussi à chiffrer celui ci. Choisissez en un long et compliqué si possible.

Associons celui ci à un nouveau périphérique de boucle pour pouvoir le formater et le monter comme tout autre périphérique:

```
$ losetup /dev/loop1 /dev/mapper/volume
```

Maintenant, installons un système de fichier sur celui ci, il est préférable d'utiliser un système de fichier compatible Linux et Windows à la fois. C'est pourquoi nous vous recommandons de saisir:

```
$ mkdosfs /dev/loop1
```

Maintenant, créons un dossier dans lequel notre volume sera monté (ce sera dans ce dossier ci qu'il restera à copier les fichiers que vous voulez mettre en sécurité, ce dossier disparaîtra au démontage du volume comme si vous retiriez une clé usb), celui ci sera ici créé dans le dossier que nous appellerons "volume" et se trouvera dans la racine /media:

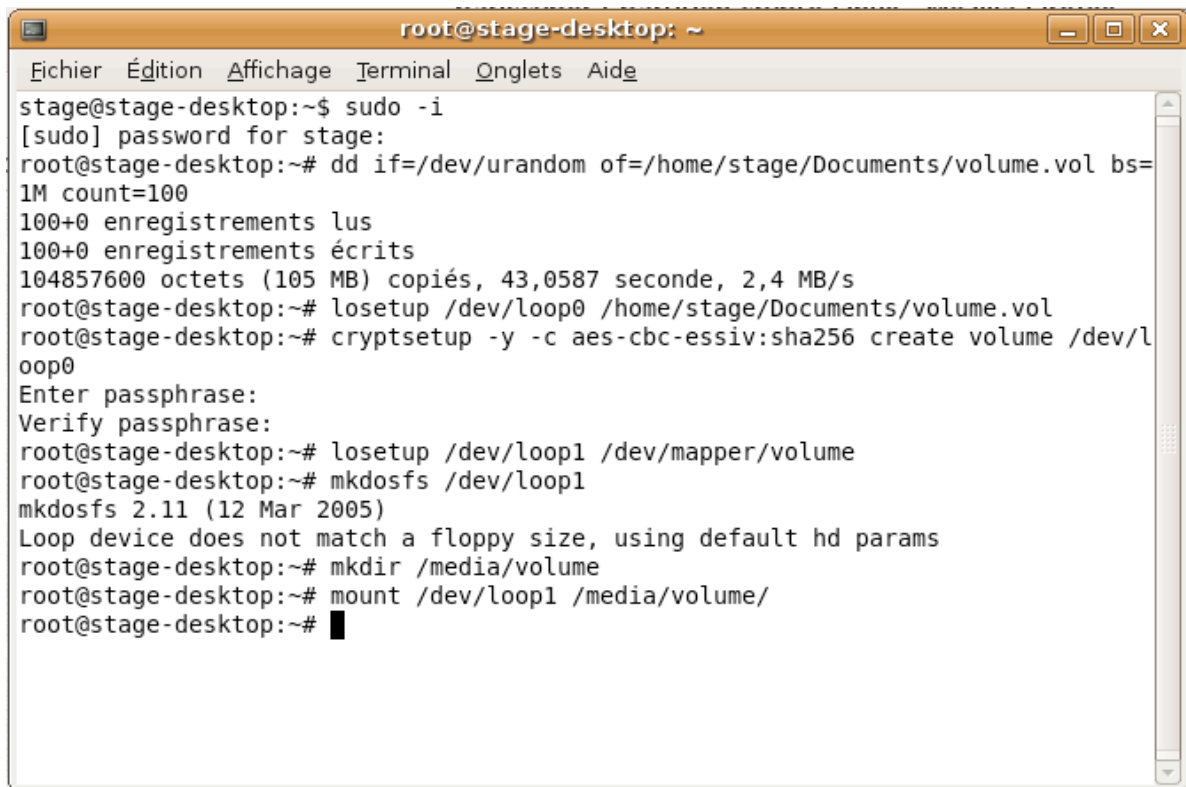
```
$ mkdir /media/volume
```

Maintenant montons notre volume virtuel à l'intérieur de celui ci:

```
$ mount /dev/loop1 /media/volume
```

Voilà, votre volume apparaît normalement sur votre bureau comme un périphérique réel physique. L'utilisation est la même que tout autre dossier ou clé usb.

Pour récapituler, voilà une capture d'écran de ce que vous devez obtenir:



```
root@stage-desktop: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
stage@stage-desktop:~$ sudo -i
[sudo] password for stage:
root@stage-desktop:~# dd if=/dev/urandom of=/home/stage/Documents/volume.vol bs=
1M count=100
100+0 enregistrements lus
100+0 enregistrements écrits
104857600 octets (105 MB) copiés, 43,0587 seconde, 2,4 MB/s
root@stage-desktop:~# losetup /dev/loop0 /home/stage/Documents/volume.vol
root@stage-desktop:~# cryptsetup -y -c aes-cbc-essiv:sha256 create volume /dev/l
oop0
Enter passphrase:
Verify passphrase:
root@stage-desktop:~# losetup /dev/loop1 /dev/mapper/volume
root@stage-desktop:~# mkdosfs /dev/loop1
mkdosfs 2.11 (12 Mar 2005)
Loop device does not match a floppy size, using default hd params
root@stage-desktop:~# mkdir /media/volume
root@stage-desktop:~# mount /dev/loop1 /media/volume/
root@stage-desktop:~# █
```

## -Montage

Pour remonter votre volume, quatre étapes sont nécessaires. Localisez votre fichier et votre dossier de montage (pour nous, notre fichier sera "volume.vol" dans "Documents" et pour le dossier ce sera "volume" dans "media"). Il faut associer ce fichier:

```
$ losetup /dev/loop0 /home/stage/Documents/volume.vol
```

Puis créons le "device mapper" pour le déchiffrement:

```
$ cryptsetup -c aes-cbc-essiv:sha256 create volume /dev/loop0
```

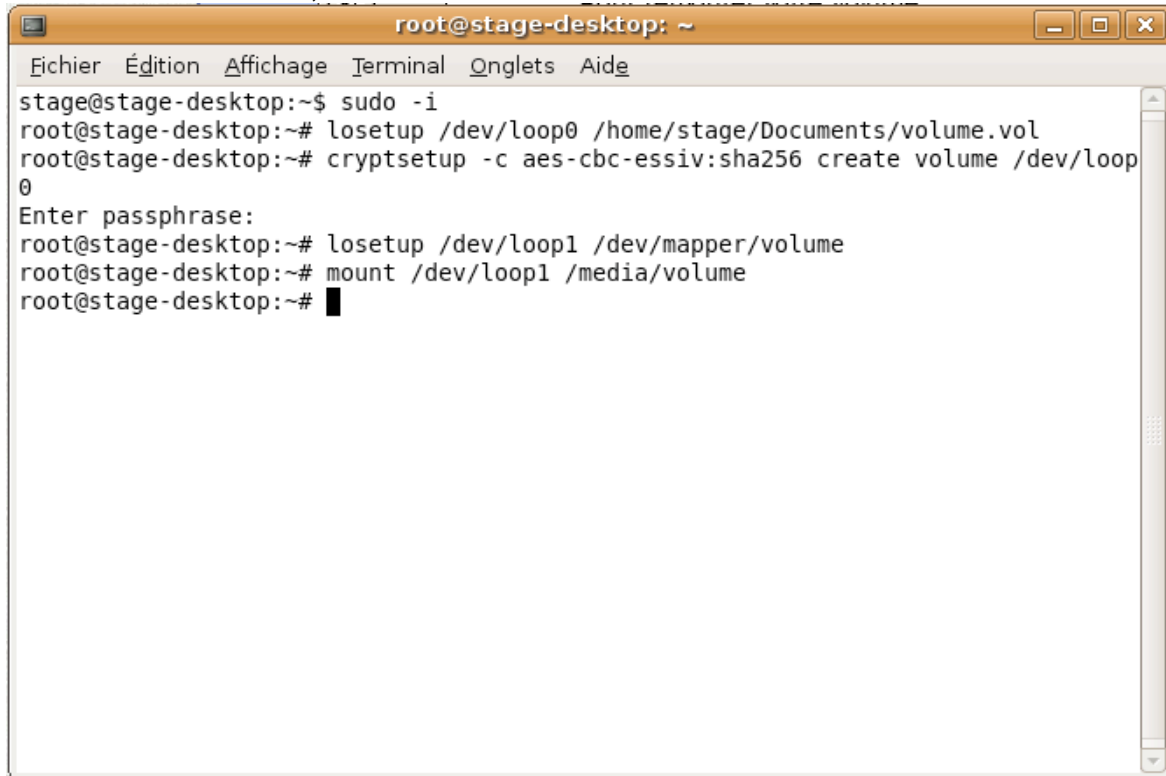
Associons ce dernier à un autre périphérique de boucle:

```
$ losetup /dev/loop1 /dev/mapper/volume
```

Et pour finir, montons ce volume:

```
$ mount /dev/loop1 /media/volume
```

Voilà notre volume monté, voici l'écran de terminal normalement obtenu:



```
root@stage-desktop: ~  
Fichier Édition Affichage Terminal Onglets Aide  
stage@stage-desktop:~$ sudo -i  
root@stage-desktop:~# losetup /dev/loop0 /home/stage/Documents/volume.vol  
root@stage-desktop:~# cryptsetup -c aes-cbc-essiv:sha256 create volume /dev/loop  
0  
Enter passphrase:  
root@stage-desktop:~# losetup /dev/loop1 /dev/mapper/volume  
root@stage-desktop:~# mount /dev/loop1 /media/volume  
root@stage-desktop:~# █
```

## **-Démontage**

Pour démonter votre volume afin de le transporter ou de l'envoyer, il suffit de réaliser les commandes inverses, c'est à dire détacher tous les périphériques de boucle ainsi que retirer le "device mapper" qui a été créé.

Ces étapes doivent être faites dans l'ordre bien sûr. Commençons donc par démonter notre volume "volume" monté dans le dossier "volume":

```
$ umount /media/volume
```

Maintenant qu'il est démonté il faut détacher le loop1:

```
$ losetup -d /dev/loop1
```

Maintenant retirons le device mapper:

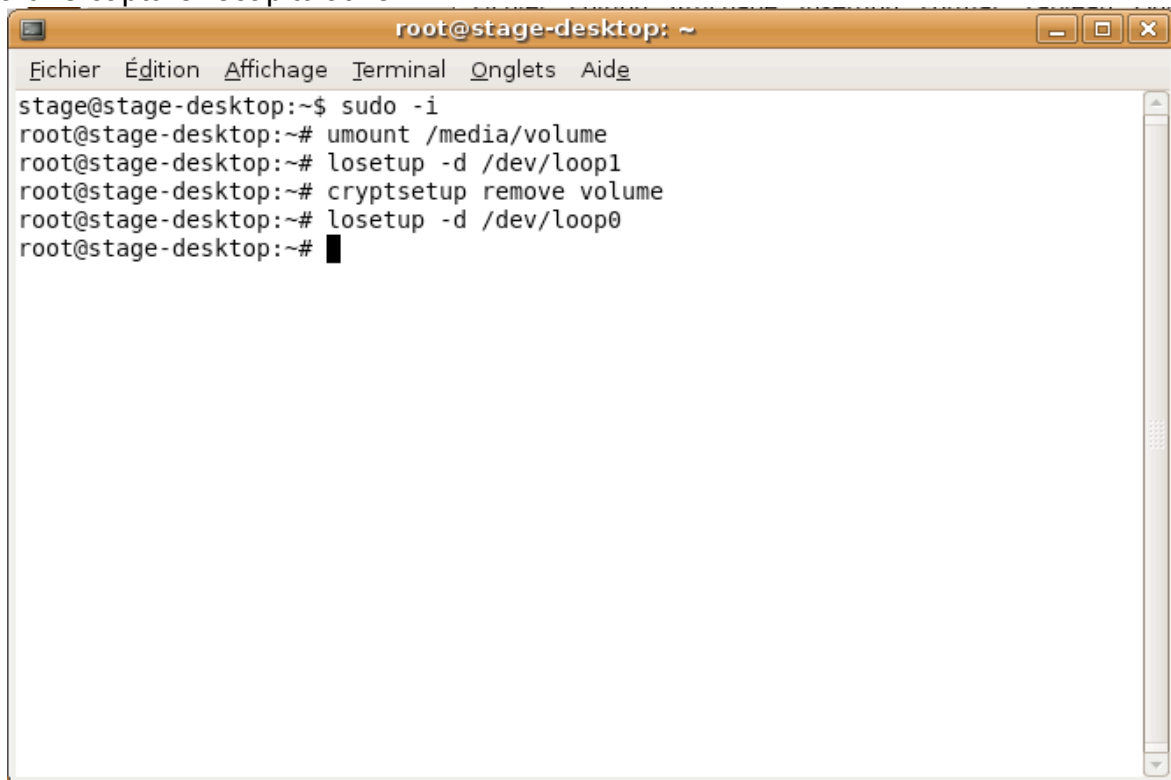
```
$ cryptsetup remove volume
```

Enfin détachons le dernier périphérique de boucle associé:

```
$ losetup -d /dev/loop0
```



Notre volume est maintenant entièrement démonté. Il ne reste plus que le fichier créé au départ. Ce fichier est inviolable et vous pouvez le monter et le démonter à volonté. voilà une capture récapitulative:



```
root@stage-desktop: ~  
Fichier Édition Affichage Terminal Onglets Aide  
stage@stage-desktop:~$ sudo -i  
root@stage-desktop:~# umount /media/volume  
root@stage-desktop:~# losetup -d /dev/loop1  
root@stage-desktop:~# cryptsetup remove volume  
root@stage-desktop:~# losetup -d /dev/loop0  
root@stage-desktop:~# █
```

### **-Conclusion**

Vous êtes maintenant dans l'entière capacité de créer un volume virtuel chiffré, qui vous servira à protéger vos données confidentielles, contre l'espionnage. Dans le chapitre suivant vous trouverez le moyen de les rendre complètement inviolables.

## **III.Utilisation des Scripts**

Avant toutes choses, placez vos scripts dans un emplacement où ils resteront, comme un dossier "Scripts" dans votre /home/user. Afin d'éviter de devoir se placer dans le dossier contenant ces scripts pour les exécuter, entrez:

```
$ PATH=$PATH"/home/user/Scripts/"
```

Vous pouvez donc maintenant exécuter vos scripts de n'importe quel emplacement.

### **-Créer son trousseau de clé**

La méthode de chiffage utilisée par nos scripts étant asymétrique, nous devons donc créer notre trousseau de clé (Publique/Privée).

Ouvrez donc une fenêtre de Terminal puis entrez:

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

Une suite de question vous sera posé. Dans l'ordre entrez:

```
-1
```

```
-1024
```

-0  
 -y  
 -"votre nom"  
 -"votre adresse e-mail"  
 -"facultatif"  
 -(Ici le mot de passe qui vous servira à déchiffrer vos fichiers vous est demandé, conservez-le précieusement!)  
 -O (si vous ne vous êtes pas trompé dans les informations précédentes)  
 Votre trousseau de clé va donc se créer, pendant sa création veillez à bouger votre souris afin de créer une clé plus sécurisée (votre clé sera générée à partir des mouvements de votre souris).  
 La création doit donc se terminer par l'écran suivant:

```

-----
nombres aléatoires une meilleure chance d'avoir assez d'entropie.
+++++
+++++>+++++.....+
++++^^^
gpg: /home/stage/.gnupg/trustdb.gpg: base de confiance créée
gpg: clé 0AF1C157 marquée comme ayant une confiance ultime.
les clés publique et secrète ont été créées et signées.

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/0AF1C157 2007-10-22
    Empreinte de la clé = 8F99 F075 347D F697 CC21 2000 8E83 6207 0AF1 C157
uid          stage <stage.desktop@gmail.com>
sub 1024g/C0494F5B 2007-10-22
stage@stage-desktop:~$

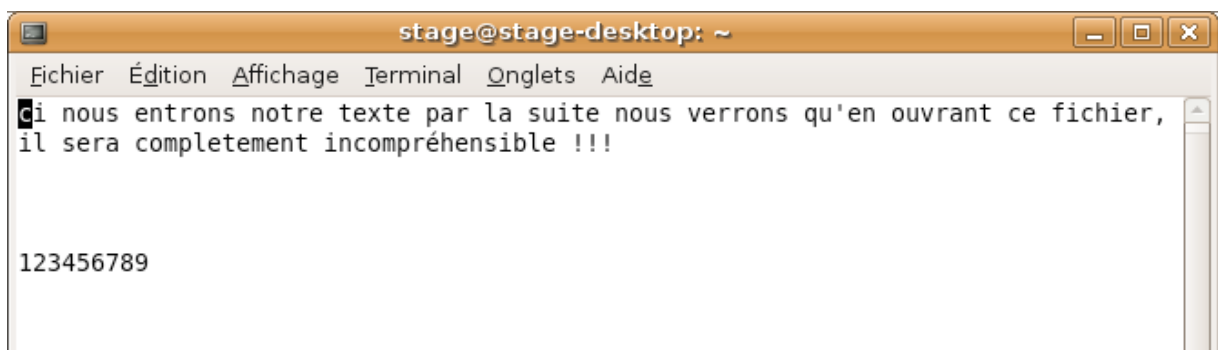
```

## -Changez la clé par défaut

Une fois votre trousseau généré, lancez DefaultKey.bash. Ce dernier va configurer GPG pour le chiffrement.

## -Chiffrement des fichiers

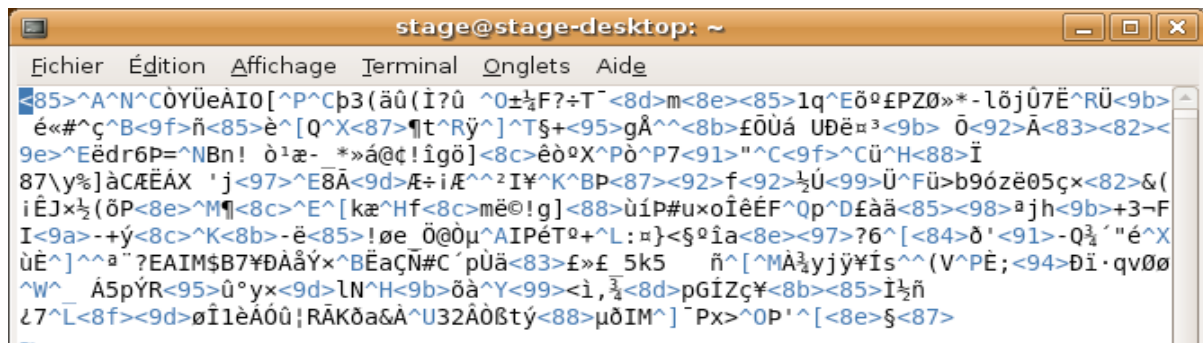
Nous allons maintenant vous expliquer comment chiffrer vos fichiers. Supposons que votre volume virtuel chiffré soit créé et monté sur /media/volume, que sur ce dernier vous ayez sauvegardé un fichier texte nommé "documentation.txt". Celui-ci contiendra:



Pour le chiffrer, il vous suffit d'ouvrir un Terminal, puis de taper la ligne suivante:  
\$ GPGiser.bash /media/volume/documentation.txt

Maintenant, observez votre fichier, vous pouvez observer que votre fichier a conservé son icône de ".txt" mais qu'un cadenas est venu s'ajouter à celui ci. De plus, l'extension elle aussi a vu son premier caractère se transformer en "3".

Le fichier semble donc comme tout autre fichier mais en l'ouvrant avec un éditeur de texte on remarque bien que celui est chiffré:



Votre fichier est maintenant totalement sécurisé, la seule personne à pouvoir le déchiffrer est vous-même à l'aide de votre mot de passe créé lors de la création de votre trousseau de clés.

Maintenant, admettons que dans votre disque, résident des centaines de fichiers... Pas facile de tout chiffrer de cette manière là, c'est pourquoi nous avons créé le script "GPGiserDossier.bash". Ce script vous permet de chiffrer tous les fichiers internes à un dossier. Par exemple un dossier "Documents" sur votre volume peut être chiffré de la manière suivante:

```
$ GPGiserDossier.bash /media/volume/Documents
```

Votre dossier est maintenant chiffré de A à Z.

## -Déchiffrage des fichiers

Vous voilà donc face à votre volume chiffré à l'intérieur duquel vous avez chiffré vos fichiers à l'aide du script "GPGiser.bash". Afin de déchiffrer vos fichiers, il vous suffit d'utiliser le script deGPGiser.bash suivi du fichier à déchiffrer:

```
$ deGPGiser.bash /media/volume/fichier.3xt
```

## -Le gardien

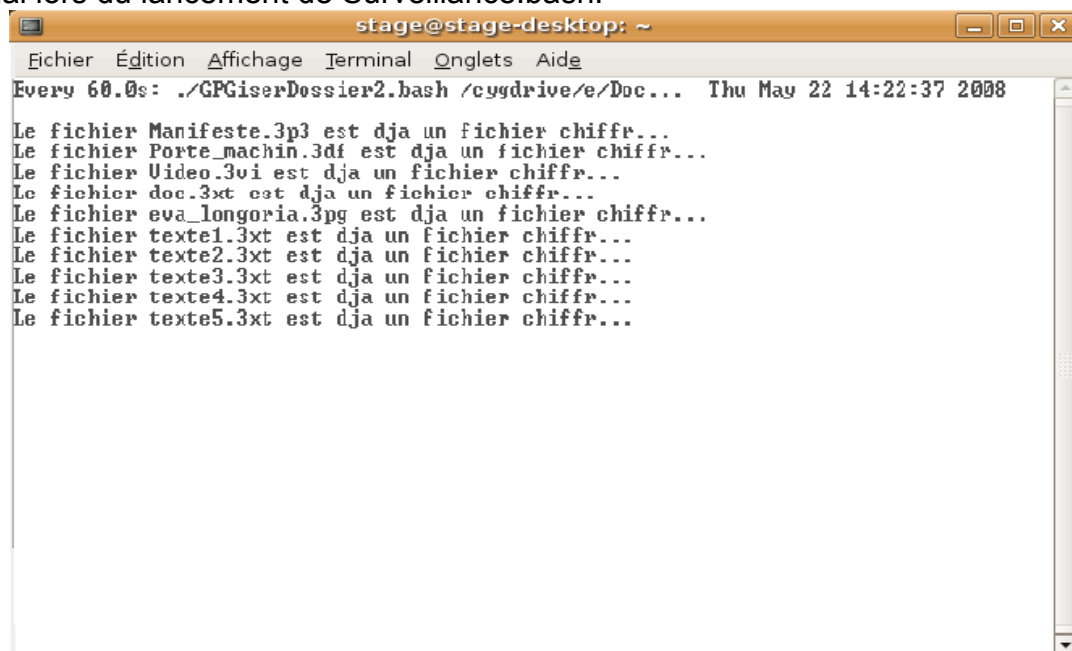
En utilisant nos programmes, vous vous rendez vite compte que sécuriser ses fichiers est simple, mais qu'il ne faut pas oublier qu'en ouvrant un fichier chiffré celui-ci se déchiffre avant l'ouverture, c'est pourquoi nous avons développé un script nommé "Surveillance.bash".

Celui-ci a pour rôle d'observer le contenu du dossier sélectionné toutes les 60 secondes, et chiffrer les fichiers qui auraient pu être oubliés. Lors de l'ouverture d'un dossier contenant des fichiers chiffrés, lancez Surveillance.bash et vous n'aurez plus à vous soucier du chiffrement de ceux-ci après les avoir ouverts.

Pour lancer Surveillance.bash entrez ceci dans bash.exe comme pour les opérations précédentes:

\$ Surveillance.bash /cygdrive/e/Documents

Terminal lors du lancement de Surveillance.bash:



```
stage@stage-desktop: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide
Every 60.0s: ./GPGiserDossier2.bash /cygdrive/e/Doc... Thu May 22 14:22:37 2008
Le fichier Manifeste.3p3 est dja un fichier chiffre...
Le fichier Porte_machin.3df est dja un fichier chiffre...
Le fichier Uideo.3vi est dja un fichier chiffre...
Le fichier doc.3xt est dja un fichier chiffre...
Le fichier eva_longoria.3pg est dja un fichier chiffre...
Le fichier texte1.3xt est dja un fichier chiffre...
Le fichier texte2.3xt est dja un fichier chiffre...
Le fichier texte3.3xt est dja un fichier chiffre...
Le fichier texte4.3xt est dja un fichier chiffre...
Le fichier texte5.3xt est dja un fichier chiffre...
```

## Conclusion:

Ce manuel vous permet donc de sécuriser vos fichiers sensibles, ou personnels contre le vol ou le piratage informatique. Vos données peuvent être interceptées par n'importe qui, personne ne sera capable de voir ce qu'il se trouve à l'intérieur. A savoir que le volume chiffré en AES256 avec la méthode de hachage SHA256 reste complètement inviolable pour le moment, et se trouve être la méthode utilisée par la NSA (National Security Agency). En ajoutant le chiffage fichier par fichier nous ajoutons donc un sécurité supplémentaire rendant nos fichiers complètement secrets.