Guide pratique de cryptoloop

Version française du guide pratique Cryptoloop HOWTO

Ralf Hölzer

<cryptoloop CHEZ ralfhoelzer POINT com>

Adaptation française : Éric Madesclair

Relecture de la version française : Encolpe Degoute, Bernard Gisbert

Préparation de la publication de la v.f. : Jean-Philippe Guérard

Version : 1.2.fr.1.1

15 août 2005

Historique des versions		
Version 1.2.fr.1.1	2005-08-15	EM, ED, BG, JPG
Quelques mises à jour mineures de présentation.		
Version 1.2.fr.1.0	2005-03-14	EM, ED, BG, JPG
Première traduction française		
Version 1.2	2004-03-12	RH
Ajout d'information sur dm-crypt et sur la sécurité, mise à jour des informations sur loop-AES. *Added information on dm-crypt, updated loop-AES info, added more info on security.*		
Version 1.1	2004-01-24	RH
Mise à jour des informations sur util-linux, Loop-AES, Best Crypt. *Updated information on patching util-linux, Loop-AES, Best Crypt.*		
Version 1.0	2004-01-17	RH
Première version, relu par TM du LDP. *Initial release, reviewed by TM at LDP.*		
Version 0.9	2004-01-15	RH
Mise à jour et conversion au format DocBook XML. *Updated and converted to DocBook XML.*		

Résumé

Ce document explique comment créer un système de fichiers chiffré en utilisant les fonctionnalités de cryptoloop. Cryptoloop est une partie du CryptoAPI des versions 2.6 des noyaux Linux.

Table des matières

1. À propos de ce guide [p 2]
  1.1. Droits d’utilisation et licence *(Copyright and License)* [p 2]
  1.2. Limitations de responsabilité *(Disclaimer)* [p 3]
  1.3. Remerciements / Contributeurs [p 3]
  1.4. Commentaires et corrections [p 3]
2. Introduction [p 4]
3. Configurer le noyau [p 5]
4. Obtenir les outils utilisateurs [p 6]
5. Configurer le périphérique de boucle [p 6]
6. Monter le système de fichiers chiffré [p 8]
7. Utiliser un fichier au lieu d’une partition [p 9]

1. À propos de ce guide

Ce guide pratique explique comment utiliser le périphérique de boucle de chiffrement cryptoloop avec les versions 2.6 des noyaux Linux. Cryptoloop permet de créer un système de fichiers chiffré à l’intérieur d’une partition ou d’un fichier dans un autre système de fichiers. Ce fichier chiffré peut être déplacé sur un CD-ROM, un DVD, une clef mémoire USB, et cætera. Cryptoloop utilise le périphérique de boucle. Ce périphérique est un pseudo-périphérique qui sert comme une « boucle » à travers duquel chaque appel au système de fichier doit passer. De cette façon, les données peuvent être traitées afin d’être chiffrées et déchiffrées. Depuis les noyaux 2.6, l’interface de programmation Crypto a été intégrée directement dans le noyau, et donc la configuration d’un système de fichiers chiffré en est rendue plus facile. Aucun correctif du noyau n’est nécessaire. Par contre une mise à jour de certains utilitaires est requise. Malheureusement, l’utilisation de cryptoloop n’est pas encore bien documentée. Ce guide est un essai pour rendre plus facile pour tout le monde la création d’un système de fichiers chiffré utilisant les fonctionnalités standards de cryptoloop. Cryptoloop est basé sur l’API Crypto des noyaux Linux 2.6. Il ne faut pas confondre avec Loop-AES qui est un projet totalement différent. Cryptoloop est similaire à l’API Crypto qui était disponible comme un correctif séparé pour les versions 2.4 du noyau. Cette nouvelle version n’est pas compatible avec l’ancienne.

1.1. Droits d’utilisation et licence *(Copyright and License)*

Linux est une marque déposée de Linus Torvalds.

Linux is a registered trademark of Linus Torvalds.

1.2. Limitations de responsabilité *(Disclaimer)*

Aucune responsabilité pour le contenu de ces documents ne pourra être acceptée. Utilisez les concepts, exemples et autre contenu à vos propres risques. Il peut y avoir des erreurs et des imprécisions, qui peuvent bien entendu endommager votre système. Procédez avec précaution, et bien que ce soit hautement improbable, l'auteur n'en acceptera aucune responsabilité.

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

Tous les droits d'auteur sont détenus par leurs propriétaires respectifs, sauf mention contraire expresse. L'utilisation d'un terme dans ce document ne doit pas être vue comme affectant la valeur d'une marque de fabrique ou d'une marque de service. La mention de produits particuliers ou de marques ne doit pas être considérée comme un acte d'approbation.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

1.3. Remerciements / Contributeurs

Je remercie les personnes suivantes pour l'aide qu'elles m'ont apporté dans la rédaction de ce guide :

- Dennis Kaledin
- Binh Nguyen
- David Lawyer
- Tabatha Marshall
- Kian Spongsveen

1.4. Commentaires et corrections

Les réactions sur ce document sont les bienvenues. Envoyez en anglais vos ajouts, commentaires et critiques à l'adresse suivante : <`cryptoloop CHEZ ralfhoelzer POINT com`>.

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à <`commentaires CHEZ traduc POINT org`> en précisant son titre, sa date et sa version.

2. Introduction

Il existe plusieurs alternatives à l'utilisation de cryptoloop. Loop-AES (http://loop-aes.sourceforge.net) est probablement le plus connu. Il fournit des fonctions similaires à cryptoloop. Aes-loop est probablement plus mature que cryptoloop et est aussi plus rapide (environ deux fois plus rapide, selon l'auteur de loop-AES) parce qu'il utilise une implémentation hautement optimisée d'un assembleur pour AES. Cela ne veut pas dire que cryptoloop est lent. Je n'ai pas noté de différence significative de vitesse entre une partition chiffrée avec cryptoloop et une partition non chiffrée durant mon travail avec un nombre normal d'entrée/sortie (I/O). A moins que les performances I/O soient extrêmement importantes pour vous, cryptoloop est suffisant. Loop-AES offre certaines fonctionnalités qui ne sont pas encore présentes dans l'implémentation noyau de cryptoloop. Loop-AES demande de modifier certains utilitaires utilisateur (mount, losetup) et ces modifications sont incompatibles avec une utilisation de cryptoloop. Vous ne pouvez pas utiliser cryptoloop et Loop-AES en même temps.

En terme de sécurité, cryptoloop est bon. La clef est généralement créée depuis un mot de passe et un hachage de cette clef est utilisé comme pour les clefs AES. Ce qui laisse la possibilité d'une attaque à texte clair connu (« *known-plaintext attack* »). Loop-AES est de ce point de vue supérieur, parce qu'il génère une clef aléatoire et chiffre cette clé séparément, rendant une attaque par texte clair connu plus difficile. Loop-AES utilise aussi un mode à plusieurs clefs, où chaque secteur est chiffré avec 64 clefs AES différentes. En général, une attaque sur votre mot de passe peut réussir, si vous choisissez un mot de passe trop simple. Pour être sécurisé, votre mot de passe doit avoir au moins 20 caractères. Cependant une attaque par force brute sur votre mot de passe sera sans doute plus facile que d'essayer directement sur les données chiffrées par AES.

Les fonctionnalités de cryptoloop dans le noyau standard fournissent une implémentation stable et propre sans avoir besoin de rajouter des correctifs. Puisqu'il est encore récent, il peut ne pas avoir suffisamment de critique en terme de sécurité. Vous devez décider par vous-même ce qui vous convient.

Important

Cryptoloop est considéré comme obsolète dans les derniers noyaux 2.6. Il semble qu'il ne sera plus maintenu encore longtemps. Le successeur de cryptoloop sera dm-crypt. Dm-crypt est disponible dans le noyau principal 2.6.4. Cryptoloop sera encore disponible dans le noyau pendant encore quelque temps, mais dm-crypt devrait devenir la méthode choisit pour le chiffrement des disques durs dans le futur. Dm-crypt est basé sur le device mapper et offre plus ou moins les même fonctionnalités que cryptoloop. Il est encore très récent et il existe encore peu d'outils utilisateur simples, disponibles. Le code de dm-crypt est considéré comme plus propre que celui de cryptoloop, il existe des différences notables entre les deux. Par exemple, la création d'un système de fichiers chiffré autrement que dans un fichier, nécessite encore de passer par un périphérique de boucle, mais la gestion est encore en développement.

Il existe d'autres outils vous permettant de créer un système de fichiers chiffrés. BestCrypt est un produit commercial de Jetico. Il vous permet de créer un récipient chiffré et d'utiliser un large éventail d'algorithme de chiffrement. Il offre également certains dispositifs ingénieux tels que les récipients cachés. Il est disponible pour les systèmes Windows et Linux, ce qui fait qu'il est préférable pour l'échange de récipient chiffré entre Window et Linux. BestCrypt se compile maintenant très bien sur les noyaux 2.6. Cryptoloop peut également créer des récipients pouvant être déplacés, en créant un système de fichiers chiffré dans un fichier comme décrit ci-dessous. Je ne sais pas comment accéder à un fichier chiffré avec cryptoloop depuis un autre système d'exploitation comme Windows. Dans ce cas, BestCrypt est peut être votre seul choix.

Il existe d'autres outils commerciaux de chiffrement de disque comme PGP, mais à ma connaissance il n'y a pas de support Linux pour eux.

3. Configurer le noyau

Avant de pouvoir utiliser cryptoloop, vous devez activer certaines options dans le noyau. Vous avez la possibilité de compiler cryptoloop comme module ou alors directement dans le noyau. Les étapes suivantes activeront cryptoloop en tant que module. Si vous n'êtes pas familiarisé avec la compilation d'un noyau 2.6, vous devriez consulter le Guide pratique du noyau Linux [1 [p 9]]. Les instructions suivantes présentent seulement les principales étapes de la compilation du noyau.

1. Allez dans le répertoire contenant l'arborescence des sources du noyau (généralement `/usr/src/linux/`) et commencez la configuration :

   ```
   make menuconfig
   ```

2. Activez la gestion du périphérique de boucle. Cochez « Loopback device support » sous :

   ```
   Device Drivers -> Block Devices -> Loopback device support
   ```

3. Dans la même section activez la gestion de cryptoloop. Cette option devient accessible dès que vous avez activé la gestion du périphérique de bouclage.

4. Activez l'API cryptographic en allant dans le menu « Cryptographic options » depuis le menu principal. Vous pouvez sans risque choisir la plupart des algorithmes ici. Je vous recommande d'activer les suivants :

   ```
   -- Cryptographic API
    <*>   HMAC support
    < >   Null algorithms
    <*>   MD4 digest algorithm
    <*>   MD5 digest algorithm
    <*>   SHA1 digest algorithm
    <*>   SHA256 digest algorithm
    <*>   SHA384 and SHA512 digest algorithms
    <*>   DES and Triple DES EDE cipher algorithms
    <*>   Blowfish cipher algorithm
    <*>   Twofish cipher algorithm
    <*>   Serpent cipher algorithm
    <*>   AES cipher algorithms
    <*>   CAST5 (CAST-128) cipher algorithm
    <*>   CAST6 (CAST-256) cipher algorithm
    <*>   Deflate compression algorithm
    < >   Testing module
   ```

   Si vous avez décidé de les compiler en module, assurez-vous de charger les modules appropriés (cryptoloop, aes, et cætera) au démarrage avant de continuer.

5. Construisez le noyau et les modules et installez les. Par exemple, si vous utilisez lilo sur une machine x86, vous pouvez tapez les commandes suivantes :

```
make
make modules_install
cp arch/i386/boot/bzImage /boot/kernel-2.6.1
lilo
```

6. Chargez les modules nécessaires au démarrage. Cela peut être fait de différente façon suivant les distributions. Par exemple sur la distribution Gentoo, les modules peuvent être ajoutés dans le fichier `/etc/modules.autoload/kernel-2.6`. Si vous avez compilé cryptoloop en tant que module, il doit être chargé en premier. Le module du périphérique de boucle sera alors automatiquement chargé. Vous pouvez aussi charger manuellement le module avec la commande suivante :

   ```
   modprobe cryptoloop
   ```

4. Obtenir les outils utilisateurs

Le pilote cryptoloop demande une mise à jour des outils utilisateur pour pouvoir créer et monter le système de fichiers chiffré. Une mise à jour du paquet util-linux est nécessaire et peut être obtenu à l'adresse suivante : http://ftp.cwi.nl/aeb/util-linux/util-linux-2.12.tar.gz. La version la plus récente est la version 2.12. Il y aura sûrement de nouvelles versions introduisant probablement des changements majeurs. Pour être sûr vous devriez vérifier les mises à jours de ce guide pratique avant de mettre à jour vers une version plus récente. Malheureusement, il existe plusieurs correctifs pour le paquet util-linux. Il y a différentes façons de créer et de monter une partition chiffrée. Afin d'utiliser la version 2.12 de l'utilitaire util-linux avec un noyau 2.6, les deux correctifs suivants doivent être appliqués :

1. Correctif combiné losetup : `losetup-combined.patch`
2. Correctif util-linux 2.6 : `util-linux-2.12-kernel-2.6.patch`

Téléchargez le paquetage util-linux et les deux correctifs disponibles. Premièrement extraire l'archive util-linux et appliquez les correctifs :

```
tar xvfz util-linux-2.12.tar.gz

cd util-linux-2.12

patch -p1 < /chemin_vers_le_correctif/losetup-combined.patch

patch -p1 < /chemin_vers_le_correctif/util-linux-2.12-kernel-2.6.patch
```

Après avoir appliqué les correctifs, compilez et installez util-linux en suivant les instructions du fichier `INSTALL`.

Je recommande d'utiliser le Linux Gentoo, car ces correctifs sont automatiquement appliqués quand apparaissent les correctifs pour util-linux. D'autres distributions peuvent avoir aussi des versions de util-linux disponibles et avoir aussi appliqué ces correctifs.

5. Configurer le périphérique de boucle

Cryptoloop peut aussi bien être utilisé sur un fichier que sur un système de fichiers entier. La suite décrit la façon de le configurer sur une partition particulière. cette partition peut être n'importe quelle partition que vous désirez, l'exemple utilisé dans la suite est `/dev/sda1`. J'ai choisit d'utiliser l’algorithme de chiffrement AES, mais vous pouvez le remplacer par n’importe quel autre algorithme activé dans votre noyau. Vous pouvez obtenir une liste de tous les algorithmes gérés par votre noyau actuel en éditant le fichier `/proc/crypto`. Le livre de Bruce Scheier, *Applied Cryptography and Practical Cryptography*, est un excellent ouvrage présentant les différents algorithmes de chiffrement. Les algorithmes AES et Serpent sont probablement le choix le plus raisonnable. AES a été beaucoup utilisé pour le chiffrement et aucune vulnérabilité sérieuse n’a été trouvée depuis longtemps. Serpent n’a pas encore été beaucoup analysé, mais il est considéré être un peu plus sécurisé que AES. Cependant, Serpent est par contre plus lent que AES. N’utilisez pas DES, il est le plus lent et le moins sécurisé. Triple-Des peut aussi être une possibilité, mais AES est probablement plus sécurisé et plus rapide, donc il n’existe pas réellement de raison d’utiliser triple-DES.

1. Il est recommandé de formater la partition et de la remplir avec des données aléatoires avant de créer le système de fichiers chiffré dessus. Ce qui rendra plus difficile à un pirate la détection des signatures dans votre partition chiffrée.

   Attention !

   Faite très attention au nom de la partition que vous inscrivez. Si vous faites une erreur, vous pouvez facilement écraser une autre partition avec des données aléatoires.

   Une partition remplie avec des données aléatoires peut être obtenue grâce à la commande suivante :

   ```
   dd if=/dev/urandom of=/dev/sda1 bs=1M
   ```

   Vous pouvez avoir un message d’erreur indiquant que le périphérique est plein. Vous pouvez l’ignorer.

2. Sélectionnez un algorithme et une taille de clef. Une liste d’algorithmes gérée par votre noyau peut être obtenu depuis le fichier `/proc/crypto`. Je recommande l’utilisation de AES avec une clef de 256 bits.

3. Configurez le périphérique de boucle. Vous pouvez utiliser la commande **losetup** provenant du paquet util-linux. La commande suivante crée un système de fichiers chiffré sur le périphérique de boucle 0 en utilisant l’algorithme de chiffrage AES avec une clef de 256 bits sur le périphérique `/dev/sda1` :

   ```
   losetup -e aes-256 /dev/loop0 /dev/sda1
   ```

   La commande vous demandera un mot de passe. Choisissez un mot de passe robuste et essayez de vous en souvenir sans utiliser un post-it collé à votre moniteur. C’est le mauvais côté de l’utilisation de cryptoloop. Puisque le mot de passe est haché pour créer la clef de chiffrement, il n’est pas facile après de changer de mot de passe. La meilleur manière de changer un mot de passe est de créer une nouvelle partition ou un nouveau fichier chiffré et de déplacer toute les données dessus. Pour cette raison, soyez sûr de choisir un mot de passe robuste dès le début. AES peut être un algorithme de chiffrement très robuste, mais si vous choisissez un mauvais mot de passe, alors la sécurité ne sera pas bonne.

   Si la commande **losetup** échoue avec le message d’erreur `INVALID ARGUMENT`, le problème vient du paquet util-linux. Assurez-vous d’avoir suivi les instructions précédentes sur la façon d’installer le correctif pour util-linux. Les anciennes versions ainsi que les versions non corrigés utilisent une méthode différente pour passer la taille de la clef et ne fonctionnent pas avec l’API

Crypto 2.6.

4. Créez un système de fichier. Vous pouvez choisir le type de système de fichier que vous souhaitez. L'exemple suivant crée un système de fichiers ext3 sur le périphérique de boucle :

```
mkfs.ext3 /dev/loop0
```

5. Montez le système de fichiers chiffré. Premièrement vous devez créer un point de montage, par exemple `/mnt/crypto` :

```
mkdir /mnt/crypto
```

   Ensuite vous devez monter le système de fichiers. À ce niveau, vous devez indiquer explicitement à la commande mount le périphérique de boucle utilisé :

```
mount -t ext3 /dev/loop0 /mnt/crypto
```

6. Vous pouvez maintenant jouer avec votre fichier chiffré jusqu'à l'ennui.

7. Démontez le système de fichiers. Après avoir suffisamment joué avec, démontez le système de fichiers :

```
umount /mnt/crypto
```

8. Détachez le périphérique de boucle. Le périphérique de boucle est encore attaché à votre partition, détachez le avec la commande :

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

6. Monter le système de fichiers chiffré

Pour toutes les opérations sur le périphérique cryptoloop, il est important que les modules nécessaires soient chargés. Vous avez besoin de charger au moins le module cryptoloop et les modules pour chaque algorithme de chiffrement avec la commande **modprobe**. Si les options ont été compilées directement dans le noyau, alors ce n'est pas nécessaire.

Afin de monter le système de fichiers chiffré créé ci-dessus, vous pouvez employer la commande **mount** du paquet util-linux.

```
mount -t ext3 /dev/sda1 /mnt/crypto/ -o encryption=aes-256
```

Le mot de passe vous sera demandé et ensuite le système de fichiers sera monté comme n'importe quel autre. L'option « encryption » indique que le périphérique contient un système de fichiers chiffré, la sélection du périphérique de boucle se fera automatiquement parmi les périphériques disponibles.

Quand vous aurez fini de travailler, vous pouvez le démonter avec la commande :

```
umount /mnt/crypto
```

Vous pouvez ajouter la ligne suivante dans le fichier `/etc/fstab` :

```
/dev/sda1            /mnt/crypto     ext3         noauto,encryption=aes-256      0 0
```

Maintenant vous pouvez simplement monter le périphérique avec :

```
mount /mnt/crypto
```

C'est tout. Amusez-vous.

7. Utiliser un fichier au lieu d'une partition

Il est très facile de créer un système de fichiers chiffré directement à l'intérieur d'un fichier. Ceci est particulièrement intéressant, si vous souhaitez sauvegarder ce fichier sur un DVD, ou sur tout autre média. Il vous sera aussi très facile de déplacer ce fichier sur d'autres machines.

Pour commencer, créez un fichier de 100 Mo contenant des données aléatoires en utilisant la commande suivante :

```
dd if=/dev/urandom of=/mestrucs.aes bs=1k count=100000
```

Si vous souhaitez changer la taille du fichier, changez la valeur de `count`. La commande précédente crée 100 000 blocs d'une taille de 1 ko, mais vous pouvez mettre la valeur que vous désirez. Assurez-vous juste qu'elle ne soit pas trop petite et qu'il pourra contenir le système de fichiers que vous avez choisi. Vous pouvez remplacer `/mestrucs.aes` par n'importe quel nom de fichier du moment que vous avez suffisamment de place libre sur la partition.

Vous pouvez créer le système de fichiers chiffré dans ce fichier, de la même façon que nous l'avons fait précédemment :

```
losetup -e aes-256 /dev/loop0 /mestrucs.aes
```

Maintenant créez le système de fichiers :

```
mkfs.ext3 /dev/loop0
```

et montez le :

```
mount -t ext3 /dev/loop0 /mnt/crypto
```

Pour finir, démontez et détachez le périphérique de boucle

```
umount /mnt/crypto
losetup -d /dev/loop0
```

Vous pourrez alors monter le système de fichiers plus tard avec la commande :

```
mount /mestrucs.aes /mnt/crypto -oencryption=aes-256
```

Si vous souhaitez déplacer le fichier ou le graver sur un CD-ROM ou un DVD, assurez-vous qu'il ait bien été *démonté* avant.

[1 [p 5]] N.D.T. : ce document est obsolète. Un document de remplacement est en cours de préparation par le Projet de documentation Linux (LDP). En attendant, vous pouvez consulter l'article de la Gazette Linux « Compiler le noyau Linux » ou le Guide pratique de reconstruction du noyau Linux (en anglais) de Kwan Lowe.