

---

## Encrypt an partition with cryptsetup

**By rac**

Published: 26.03.2008 - 21:18

This howto log is done on Ubuntu 7.10 64-bit version but should be fine with most debian based distros

Go to the related link at the end of this article if you want to know what the commands exactly do or if you dont have the same system as I have

Create a partition GParted (aptitude install gparted), Select Filesystem as "not formatted" and note down the device name (e.g. /dev/sdb1)

### Install needed tools

```
aptitude install cryptsetup
```

### Load needed modules

If you have a 32-bit Pentium, you need to load aes-i586 or simply aes (not optimized) instead of aes-x86\_64

```
sudo modprobe aes-x86_64
sudo modprobe dm-crypt
sudo modprobe dm-mod
```

### Create LUKS Filesystem

Kernel 2.6.24 (Hardy):

```
sudo cryptsetup -c aes-xts-plain -y -s 512 luksFormat /dev/sd<Partition>
```

---

Kernel 2.6.20 (Feisty):

```
sudo cryptsetup -c aes-lrw-benbi -y -s 384 luksFormat /dev/sd<Partition>
```

Kernel 2.6.10 (Dapper und Edgy):

```
sudo cryptsetup -c aes-cbc-essiv:sha256 -y -s 256 luksFormat /dev/sd<Partition>
```

In My case the output after Feisty command was

```
root@y:~# cryptsetup -c aes-lrw-benbi -y -s 384 luksFormat /dev/sdb1
WARNING!
=====
This will overwrite data on /dev/sdb1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase:
Verify passphrase:
Command successful.
```

## Formatting your crypto partition

### Open it first (decrypt)

```
root@y:~# cryptsetup luksOpen /dev/sdb1 somename
Enter LUKS passphrase:
key slot 0 unlocked.
Command successful.
```

### Format it with an Ext3 Filesystem

```
root@y:~# mkfs.ext3 /dev/mapper/somename
mke2fs 1.40.2 (12-Jul-2007)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=4096 (log=2)
Fragmentgröße=4096 (log=2)
122109952 Inodes, 244189623 Blöcke
12209481 Blöcke (5.00%) reserviert für den Superuser
erster Datenblock=0
Maximum filesystem blocks=4294967296
7453 Blockgruppen
32768 Blöcke pro Gruppe, 32768 Fragmente pro Gruppe
16384 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
```

---

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,  
4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,  
102400000, 214990848

Schreibe Inode-Tabellen: erledigt

Erstelle Journal (32768 Blöcke): erledigt

Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt

Das Dateisystem wird automatisch alle 26 Mounts bzw. alle 180 Tage überprüft,  
je nachdem, was zuerst eintritt. Veränderbar mit `tune2fs -c` oder `-t`.

### **and close it**

```
root@y:~# cryptsetup luksClose somename
```

### **Create open and close scripts**

Create a directory where you want to mount it to and dont forget to give you write permission to that.

#### **openCrypto.sh**

```
#!/bin/bash  
sudo cryptsetup luksOpen /dev/sdb1 somename  
sudo mount /dev/mapper/somename /mnt/somename
```

#### **closeCrypto.sh**

```
#!/bin/bash  
sudo umount /mnt/somename  
sudo cryptsetup luksClose somename
```

### **Related Links**

- [Blog entry about encrypting a disk](#)
- [Wiki entry from ubuntuusers.de](#)

-->

### **Trackback URL for this post:**

<http://www.2030.tk/trackback/172>

[Creating a transparently encrypted root filesystem](#)

from *Westhoffswelt - Welcome to the real world* on 30 January, 2009 - 02:33

---

This article provides all the necessary explanations to create a fully transparent filesystem encryption for your system root. After you followed the steps of this tutorial all your data will be stored encrypted on your hard drive during your normal d...