

Theme ▼

Table of Contents

[Introduction](#)


[Private \(encrypted\) directory](#)


[Make an encrypted directory with a unique name](#)

[Encrypt your entire home directory](#)

[Migrate to an encrypted home \(post-install\)](#)

[Change your password](#)

 [Graphical method](#)

 [Command line](#)

[Prevent automatic mounting of your encrypted home directory](#)

[Create a new user with an encrypted home directory](#)

[Access your encrypted data from a live CD](#)

[Ecryptfs with a separate home partition](#)

[Using SSH keys an with an encrypted home directory](#)

[Encrypt swap](#)

[Using ecryptfs with Fedora](#)

Dhammapada

**When a brahmin has
crossed beyond duality,
then all the fetters of
such a seer come to an
end.**

Introduction

Encryption is becoming more popular and, IMO, is simply the best way to protect your data from unauthorized access or in the event your laptop is stolen. There are several options for encrypting an entire directory or partition :

- EncFS
- LUKS
- Truecrypt

I wanted to feature Ecryptfs as it is used on Ubuntu as an option (during installation) to encrypt your home directory. This page is intended to review the features of Ecryptfs, including how to set up a private, encrypted directory in Ubuntu (9.04 or higher). As of this writing, Ubuntu 9.04 is an Beta Release, so do not be surprised if you find bugs.

New features in Ubuntu 9.04 (Jaunty):

- The desktop CD allows encryption of your home directory during the installation.
- Encrypt home directories when you create new users.
- Ecryptfs now encrypts file names!

🔓 Your private directory is now decrypted no matter if you log in from the console, ssh, or X (GDM).

Overview

Ecryptfs uses two passwords to decrypt your private directory.

The first is your log in password. This allows your private directory to be automatically decrypted when you log in. When you change your log in password, however, the ecryptfs password is not updated. You unfortunately need to manually the Ecryptfs password (see below).

While this may at first seem inconvenient, in effect it prevents root from accessing your private data by simply changing your user's password.

The second passphrase is called a "mount passphrase". This passphrase is used if you wish to mount your private directory manually.

```
sudo mount -t ecryptfs /home/user_name/.Private  
/home/user_name/Private
```

The mount passphrase, and not your log in password, is used to decrypt your data and is discussed on the Ubuntu Wiki page

(see References below).

Ecryptfs uses 3 directories : .Private, Private , and .Ecryptfs

- .Private = This is where the encrypted data is kept.
- Private = Used as a mount point for .Private. This is where your working directory for your decrypted data.
- .Ecryptfs = This directory contains configuration information and will also be covered later.

How it works

Your data is encrypted in ~/.Private and is decrypted by mounting ~/.Private to ~/Private using Ecryptfs-mount-private. This is performed automatically when you log in (although you can manually encrypt (unmount)/ decrypt (mount)). When you log off, all data in Private is then encrypted to .Private .

If you elect to encrypt your entire home directory the setup is a bit more complex in that ~/.Ecryptfs is a symbolic link to /var/lib/Ecryptfs/user_name/ . This is all set up during the installation.

Private (encrypted) directory

Ecryptfs can be used to create an encrypted directory in your home directory. By default, this directory is called "Private" and is automatically decrypted when you log in. This is a change from Ubuntu 8.10 where the directory was only decrypted automatically if you logged in via GDM (X).

This directory is then automatically encrypted when you log off.

To generate an encrypted directory we first need to install Ecryptfs

```
sudo apt-get install ecryptfs-utils
```

Then simply

```
ecryptfs-setup-private
```

This will asked first for your login password, enter your log in password. You will next be asked to "Enter your mount passphrase [leave blank to generate one]" , leave this blank (hit the enter key) and a random passphrase will be generated.

That is all there is to it. Any data you place in ~/Private will be encrypted in ~/.Private when you log off.

Make an encrypted directory with a unique name

You may wish to use an alternate name to "Private". Although this is easy to do, it is not as automated as Ecryptfs-setup-private and requires root access (via sudo).

To do this, make a new directory such as "secret".

```
mkdir ~/secret  
chmod 700 secret
```

Now mount the directory secret (as root) with the mount command, using Ecryptfs as the filesystem type.

You may use either a single directory (as I do in this example) or two directories (as is default for your Private and .Private directories).

When using a single directory, the contents are encrypted into the same directory when the directory is unmounted.

```
sudo      mount      -t      ecryptfs      ~victoria/secret  
~victoria/secret
```

You will be asked to enter a passphrase and a number of additional questions. Go with the defaults (hit enter) but answer y (yes) to "Enable filename encryption (y/n) [n]:" to encrypt file names.

See man ecryptfs for an explanation of the options.

FYI: ~username is short hand for /home/username

Place any data you wish encrypted in ~/secret .

```
sudo umount ./secret
```

To make it easier you can give the options you desire to the mount command with the -o flag (all one line)

```
sudo      mount      ./secret      ./secret      -o  
key=passphrase,ecryptfs_cipher=aes,  
ecryptfs_key_bytes=16,ecryptfs_passthrough=no,  
ecryptfs_enable_filename_crypto=yes
```

Either make an alias in ~/.bashrc or make a custom launcher.

There are two caveats to this method:

The custom directory will NOT automatically be decrypted when you log in.

You may use more than one password.

If you use more than one password, you will have more than one set of data. Only data encrypted with password_1 will be decrypted when you mount with password_1. Data encrypted with password_2 will remain encrypted.

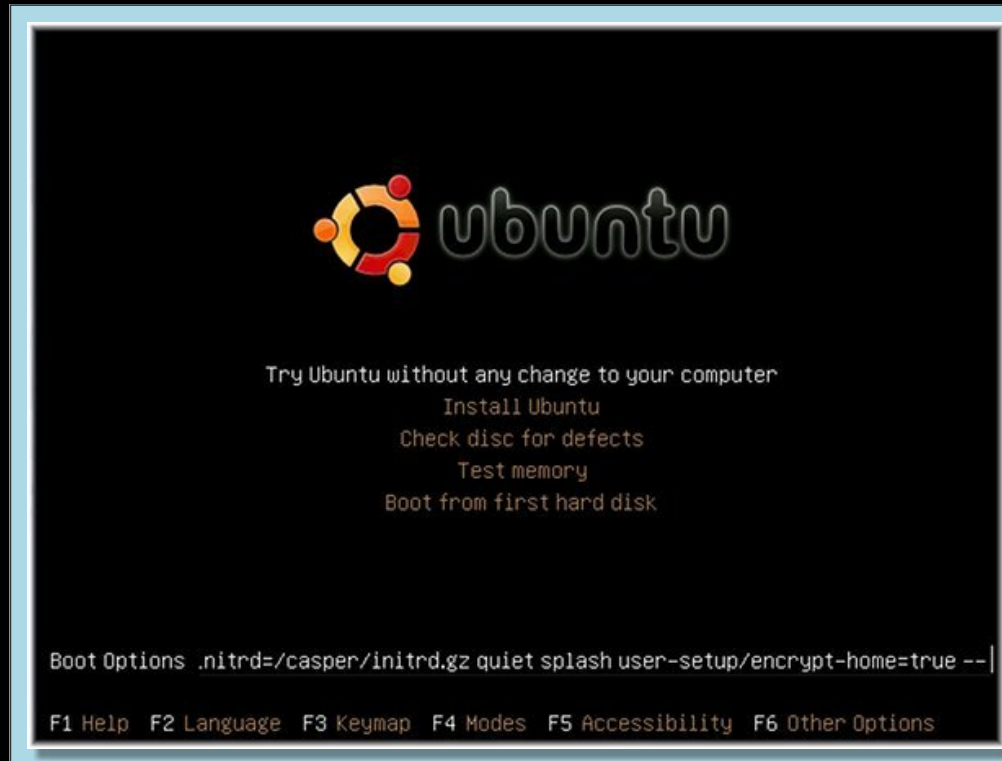
Encrypt your entire home directory

This is now an option on both the alternate and desktop (live) CD's.

With the alternate CD you will be given the option to encrypt your home directory as part of the installation, just after you create your first user.

This can be done with the desktop CD with a "cheat code".

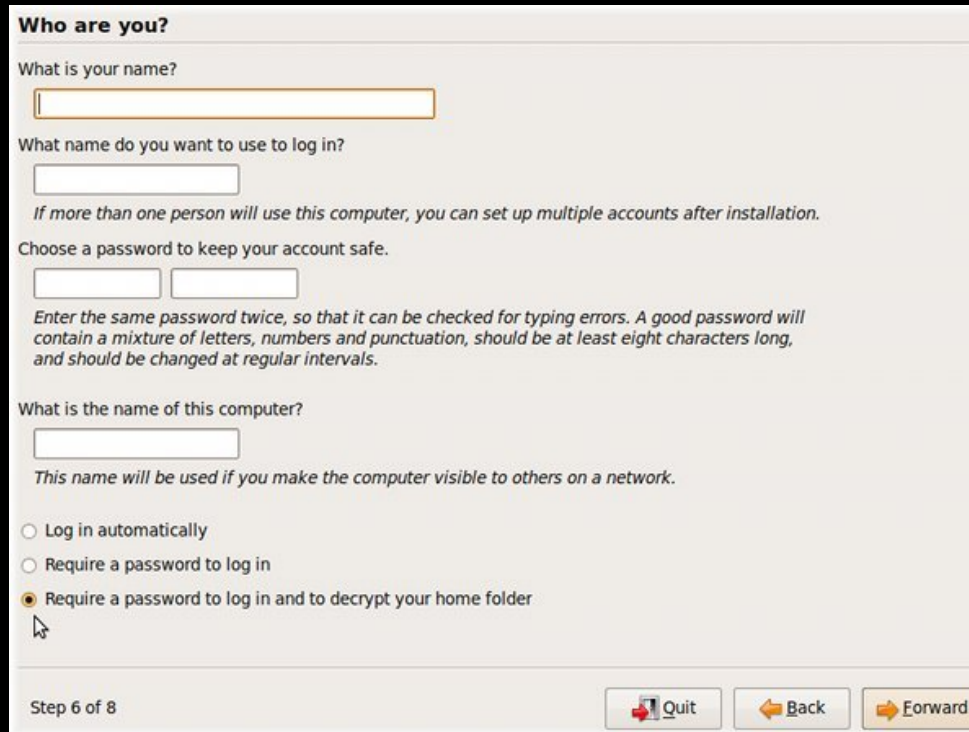
Boot the desktop CD. At the very first screen, just after you select your language, hit F6, then the Esc key. This will allow you to edit the options line. Use the arrow keys on the key board to position the cursor between the word "splash" and the "--" at the end of the line.



Add `user-setup/encrypt-home=true` , be sure there is a space between the cheat code and the --

Hit the enter key to continue booting.

As you install, at the screen where you enter your user name and password, you now have a new option, "Require a password to log in and decrypt your home folder".



The image shows a screenshot of the Ubuntu installer's 'Who are you?' screen. It contains several input fields for name, login name, and password, along with a section for selecting login options. The 'Require a password to log in and to decrypt your home folder' option is selected. Navigation buttons for 'Quit', 'Back', and 'Forward' are at the bottom right, and 'Step 6 of 8' is at the bottom left.

Who are you?

What is your name?

What name do you want to use to log in?

If more than one person will use this computer, you can set up multiple accounts after installation.

Choose a password to keep your account safe.

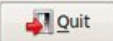

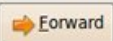
Enter the same password twice, so that it can be checked for typing errors. A good password will contain a mixture of letters, numbers and punctuation, should be at least eight characters long, and should be changed at regular intervals.

What is the name of this computer?

This name will be used if you make the computer visible to others on a network.

☐ Log in automatically
☐ Require a password to log in
☒ Require a password to log in and to decrypt your home folder

Step 6 of 8

Congratulations, your home directory is now encrypted.

Migrate (encrypt) your home directory (post-install)

If you did not choose to encrypt your home directory at the time of installation, it is possible to encrypt your home directory at a later time with "ecryptfs-migrate-home". I would advise you back up any data first in the event there is a problem, then run:

```
sudo ecryptfs-migrate-home -u user
```

Where user is the user name to migrate. The user to be migrated must not be logged in at the time, so to migrate your administrative user boot to recovery mode.

You will see output similar to this:

```
=====
Some Important Notes!
```

```
1. The file encryption appears to have completed
successfully, however,
test          MUST          LOGIN          IMMEDIATELY,
_BEFORE_THE_NEXT_REBOOT_,
TO COMPLETE THE MIGRATION!!!
```

```
2. If test can log in and read and write their
files, then the migration is complete,
and you should remove /home/user.7y3X0vjM.
Otherwise, restore /home/user.7y3X0vjM back to
/home/test.
```

3. user should also run 'ecryptfs-unwrap-passphrase' and record their randomly generated mount passphrase as soon as possible.

4. To ensure the integrity of all encrypted data on this system, you should also encrypted swap space with 'ecryptfs-setup-swap'.

=====

As outlined above, have the user log in and test the encryption. If it seems to be in working order, delete the backup directory and reboot. If there is a problem, restore the user's home directory from the backup.

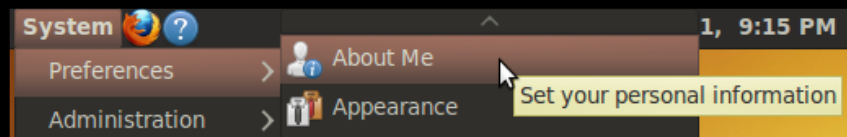
Change your passphrase to mount your encrypted private directory or home

When you are using an encrypted home directory, you must change your Ecryptfs passphrase and log in password at the same time. This is accomplished either via the graphical interface (easiest) or from the command line (not difficult).

Graphical Interface


Change your password from the graphical interface under:

System -> Preferences -> About Me



Click the "Change password ..." box :

About Live session user

**Live session user**

User name: ubuntu
[Change Password...](#)

Contact

Address

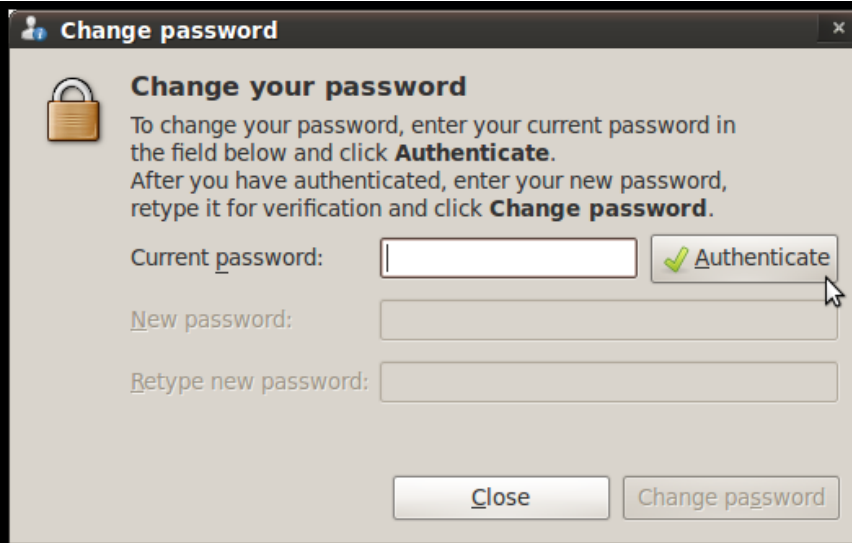
Personal Info

Home
Address:
City:
Zip/Postal code:
State/Province:
P.O. box:
Country:

Work

[Close](#)

Enter your current password, click "Authenticate", enter and confirm your new password.



Finally, click "Change password".

Command line

If you change your (login) password from the command line, you will notice the passphrase to mount your encrypted home directory is updated.

```
passwd your_user_name
```

If you change your password as root (or if root changes your password), the passphrase to mount your encrypted home will NOT be updated. This is

good news in that it keeps root from accessing your data simply by changing your user's password and logging in as your user.

```
# This will fail  
sudo passwd your_user_name
```

Prevent your encrypted private directory or home from being mounted automatically

With your home directory mounted (decrypted), simply delete
~/.ecryptfs/auto-mount

This is an empty file and you can recreate it with

```
touch ~/.ecryptfs/auto-mount
```

This will re-enable "automagic" decryption of your home directory when you log in.

Create a new user with an encrypted home directory

Simply use `adduser` with the `--encrypt-home` options (no graphical option yet).

```
sudo adduser --encrypt-home new_user_name
```

Access your encrypted data from a live CD

Starting with Ubuntu 11.04 there is a utility, `ecryptfs-recover-private`, to automate recovery of data from an encrypted `$HOME` directory.

Start by booting a live CD (Ubuntu 11.04 or higher), and mount your Ubuntu root partition. If you have a separate home partition, you will need to mount that as well. `ecryptfs-recover-private` will search mounted file systems for `.Private` `ecryptfs` crypts and interactively give you the opportunity to decrypt the

data. You still need to know the password you used to encrypt the data.

Assuming Ubuntu is installed into /dev/sda1/

```
ubuntu@ubuntu:~$ sudo mount /dev/sda1 /mnt

#####
# Separate home #
#####

# SKIP THIS STEP IF YOU DO NOT HAVE A SEPARATE HOME
# DIRECTORY
ubuntu@ubuntu:~$ sudo mount /dev/sda2 /mnt/home
```

Then run ecryptfs-recover-private

```
sudo ecryptfs-recover-private
```

Your encrypted \$HOME will be mounted ro in /tmp

You can then copy the data using nautilus (as root).

```
gksu nautilus
```

```
ubuntu@ubuntu:~$ sudo mount /dev/sda1 /mnt
ubuntu@ubuntu:~$ sudo ecryptfs-recover-private
INFO: Searching for encrypted private directories (this might
take a while)...
```

```
INFO: Found [/mnt/home/.ecryptfs/cryptothelow/.Private].  
Try to recover this directory? [Y/n]: Y  
INFO: Found your wrapped-passphrase  
Do you know your LOGIN passphrase? [Y/n] Y  
INFO: Enter your LOGIN passphrase...  
Passphrase:  
Inserted auth tok with sig [fa0516369a9d60dd] into the user  
session keyring  
INFO: Success! Private data mounted read-only at  
[/tmp/ecryptfs.yxyLYWVG].  
ubuntu@ubuntu:~$ gksu nautilus /tmp/ecryptfs.yxyLYWVG
```

For additional information see: [Dustin Kirkland's Blog](#)

Using Ecryptfs with a separate /home partition

One downside of encryption is that using a separate /home partition is more difficult and there are as of yet no automated tools on the installation CD (alternate or desktop) to automatically preserve and configure your Ecryptfs encrypted /home directories.

I advise you back up your data, install, then restore your data.

Using SSH keys with an encrypted home directory

By default, ssh uses `~/.ssh/authorized_keys` to log in. This file will not be available if your home directory is encrypted.

I suggest you move the file to an alternate location. Using any editor, open `/etc/ssh/sshd_config` and find the line `"#AuthorizedKeysFile .ssh/authorized_keys"`, uncomment the line and assign a new location, using a full path.

```
AuthorizedKeysFile /etc/ssh/.authorized_keys
```

Place our public keys in the new file.

Using Ecryptfs to encrypt swap

Note: Encrypting swap may break hibernation and sleep.

In addition to ecryptfs-utils you need cryptsetup

```
sudo apt-get install ecryptfs-utils cryptsetup
```

Then use ecryptfs-setup-swap

```
sudo ecryptfs-setup-swap
```

This will unmount your swap partition, encrypt it, and remount it.

A new entry for your encrypted swap is automatically generated in /etc/fstab , but unfortunately as of this writing ,the old entry is not removed and you must remove it manually (or suffer error messages when you boot).

Using any editor, edit fstab as root (gksu gedit /etc/fstab)

The old swap starts with either UUID=xxx-yyy-zzz or /dev/sdxy, remove that line.

The new swap is identified by /dev/mapper/cryptswap , keep this line.

That's it, your swap is now encrypted and will mount automatically when you boot.

I would like to thank Dustin Kirkland for his contributions to Ecryptfs and his blog. His blog in particular is the best source of up to date information on Ecryptfs.

Using ecryptfs with Fedora

In Fedora 15 , ecryptfs can be used to encrypt data. By default it uses the directories ~/Private and ~/.Private to do so.

pam can be configured to decrypt ~/Private when you log in, but I could not get an encrypted home directory working the way it does in Ubuntu.

1. Install ecryptfs-utils if needed:

```
yum install ecryptfs-utils
```

2. Using the command line or graphical tools, add your user to the ecryptfs group.

```
usermod -a -G ecryptfs your_user
```

3. Set up your encrypted directory, as a user run:

by default this uses ~/Private and ~/.Private

4. You can now mount the Private directory with

```
ecryptfs-mount-private
```

And unmount (encrypt) the data using

```
ecryptfs-umount-private
```

5. You can have your Private directory automatically decrypted when you log in by configuring pam

Using any editor, as root, open /etc/sysconfig/authconfig and uncomment the ecryptfs line:

```
USEECRYPTFS=yes
```

Then run authconfig-tui --updateall

```
authconfig-tui --updateall
```

6. You can move any sensitive data (~/.mozilla , ~/.ssh , etc) into ~/Private and use links.

```
mv ~/.mozilla ~/.ssh Private  
ln -s ~/Private/.mozilla ~/.mozilla  
ln -s ~/Private/.ssh ~/.ssh
```

Although if you wish to ssh in, you should move your authorized keys outside of the Encrypted ~/Private as outlined above .

References

- [Ecryptfs Home Page](#)
- [Ecryptfs FAQ](#)
- [Ubuntu Wiki Ecryptfs page](#)
- [Tombuntu's ecryptfs blog](#)
- [ubuntugeek ecryptfs](#)
- [Debian Ecryptfs](#)
- [Dustin Kirkland's blog - Encrypted Home directories](#)
- [Dustin Kirkland's blog - Mounting Ecryptfs directories with a live CD](#)

