NAME

mount.cryptfs − mount an encrypted filesystem

SYNOPSIS

mount.cryptfs *<DEVICE> <DIRECTORY>* [**−fnsv**] [**−o** *<OPTIONS>*]

DESCRIPTION

The **mount.cryptfs**(8) utility mounts an encrypted filesystem. It is usually invoked indirectly by the **mount**(8) command when the filesystem type is specified to be **cryptfs** (using either the **−t** option or the filesystem type field in the */etc/fstab* file).

If the **mount.cryptfs**(8) utility is invoked by an ordinary user (and not by root), only the directory, the verbosity level (the **−v** options) and the mount option **ro** will be retained. The device and the mount options (excluding the mount option **ro**) will be reverted to the values specified in the */etc/fstab* file and the user's permissions to mount the directory will be checked. This is done to prevent ordinary users from circumventing mounting restrictions issued by the system administrator.

The mounting will be done using the following steps:

- If at least one of the mount options **encryption**, **loop**, **offset** and **sizelimit** is specified, a new loop device will be set up unless the device and the loop device specified by the **loop** mount option are the same.
- If the mount option **linear** is specified, a new logical volume will be set up using a dm-linear device mapper device.
- If at least one of the mount options **cipher**, **crypt** and **luks** is specified, a new cryptographic volume will be set up using a dm-crypt device mapper device.
- If at least one of the mount options **numsectors** and **startsector** is specified and neither a new logical volume nor a new cryptographic volume has been set up, a new logical volume will be set up using a dm-linear device mapper device.
- The four steps above will be repeated for secondary device layers if corresponding options for them are specified.
- If the mount option **fsck** is specified, the filesystem will be checked unless at least one of the mount options **remount** and **ro** is also specified.
- The filesystem will be mounted.
- The */etc/mtab* file will be updated unless the **−n** option is specified.

There are two distinct purposes for setting up loop devices. Firstly, they can be used for mounting filesystem image files instead of mounting block devices. Secondly, they can be used for encryption. In the latter case the loop-AES kernel patch is pretty much required because the standard loop device encryption algorithms are severely limited.

Secondary device layers can be used for key management purposes, for instance. Normally, they should not be used because their use complicates filesystem management but does not make the encryption any stronger. Device and encryption key mount options for different device layers can be specified by suffixing mount options with device layer indices. An optional index **1** can be used for the primary device layer and a mandatory index **2** to **9** must be used for secondary device layers.

The purpose of updating the */etc/mtab* file is to make the **umount**(8) command to be able to unmount the filesystem, to remove assosiated device mapper devices and to release assosiated loop devices either directly or by invoking the **umount.cryptfs**(8) utility.

OPERANDS

<DEVICE>

A block device or a filesystem image file containing the encrypted filesystem to be mounted.

<DIRECTORY>

A directory to which the encrypted filesystem will be mounted.

OPTIONS

−f
Fake the mount. Do everything except the actual system call.
See the **−f** option in **mount**(8).

−n
Do not update the */etc/mtab* file.
See the **−n** option in **mount**(8).

−o *<OPTIONS>*
Set mount options using a comma separated option list.

−s
Tolerate sloppy mount instead of failing. Ignore mount options not supported by the filesystem type. Not all filesystem types support this option.
See the **−s** option in **mount**(8).

−v
Increase the verbosity level.
See the **−v** option in **mount**(8).

MOUNT OPTIONS

PREFIXED MOUNT OPTIONS

cryptfs_*<OPTION>*[=*<ARGUMENT>*]
The **cryptfs_** prefix will be ignored. This can be used for passing unaltered options to the **mount.cryptfs**(8) utility through commands which parse mount options.

ENCRYPTION KEY MOUNT OPTIONS

keyfile=*<PLAINTEXT−KEY−FILE>*
Use the content of a file as an encryption key or as an encryption key passphrase or pass the pathname of the file as an operand to a key script.

keyhash=*<KEY−HASH>*
Set the encryption key hash algorithm.
See the **−−hash** option in **cryptsetup**(8).
See the **−H** option in **losetup**(8) from loop-AES.

keysize=*<KEY−SIZE>*
Set the encryption key size.
See the **−−key−size** option in **cryptsetup**(8).
See the **−k** option in **losetup**(8) from util-linux-ng.

keyscript=*<KEY−SCRIPT>*
Execute a script and use the output as an encryption key or as an encryption key passphrase. If a plaintext key file is also specified, it will be passed as an operand to the key script.

gpgkey=*<GNUPG−KEY−FILE>*
Decrypt the content of a **gpg**(1) encrypted file and use the plaintext as an encryption key.

gpghome=*<GNUPG−HOME−DIRECTORY>*
Set the GnuPG home directory.
See the **−−homedir** option in **gpg**(1).

sslkey=*<OPENSSL−KEY−FILE>*
Decrypt the content of an **openssl**(1ssl) encrypted file and use the plaintext as an encryption key.

sslcipher=*<SSL−CIPHER>*
Set the OpenSSL enryption algorithm.
See **enc**(1ssl) from OpenSSL.

sslhash=*<SSL−HASH>*
Set the OpenSSL message digest algorithm.
See the **−md** option in **enc**(1ssl) from OpenSSL.

timeout=*<TIMEOUT>*
Set how quickly cryptographic volume passphrase prompts will timeout and how quickly key script and key decryption processes will be killed.

See the **−−timeout** option in **cryptsetup**(8).

tries=<*TRIES*>

Set how many times cryptographic volume passphrase prompt will be repeated, at most.
See the **−−tries** option in **cryptsetup**(8).

If a passphrase will be required but a plaintext key file is not specified, the passphrase will either be extracted from a source defined by a password environment variable **PASSWD**, **PASSWD_FD** or **PASSWD_FILE** or read from the standard input or from the terminal.

These mount options can be suffixed with a device layer index **1** to **9** (the default is **1**).

BASIC LOOP DEVICE MOUNT OPTIONS

loop[=[<*LOOP−DEVICE*>]]

Setup a loop device. Optionally, use the specified loop device.

offset=<*OFFSET*>

Set the loop device data start offset in bytes. Setup a loop device.
See the **−o** option in **losetup**(8).

sizelimit=<*SIZE−LIMIT*>

Set the loop device size in bytes. Setup a loop device.
See the **−s** option in **losetup**(8) from loop-AES.

These mount options can (and usually should) be suffixed with a device layer index **1** to **9** (the default is **1**). See also LOOP DEVICES below.

LOOP DEVICE ENCRYPTION MOUNT OPTIONS

encryption=<*ENCRYPTION*>

Set the loop device encryption algorithm. Setup a loop device.
See the **−e** option in **losetup**(8).

itercountk=<*THOUSAND−ITERATIONS*>

Set the loop device key hash iteration count.
See the **−C** option in **losetup**(8) from loop-AES.

loinit=<*LOINIT*>

Pass a value to a loop device cipher transfer function.
See the **−I** option in **losetup**(8) from loop-AES.

pseed=<*PSEED*>

Set the loop device key hash seed.
See the **−S** option in **losetup**(8) from loop-AES.

These mount options can (and usually should) be suffixed with a device layer index **1** to **9** (the default is **1**). See also LOOP DEVICES below.

LOOP DEVICE WORKAROUND MOUNT OPTIONS

dev=<*ORIG−DEVICE*>

Set the original device to be used while updating the */etc/mtab* file. This can be used if one wants to use the **mount**(8) command to setup loop devices (in which case the <*DEVICE*> operand will be a loop device) but still wants that the **mount.cryptfs**(8) utility uses the original device while updating the */etc/mtab* file in order to ensure that ordinary users can unmount the filesystems they have mounted.

loop0 Assume that the mount option **loop**=<*DEVICE*> was specified even if it is not passed to the **mount.cryptfs**(8) utility (older versions of the **mount**(8) command do not pass the **loop** mount option).

See also LOOP DEVICES below.

LOGICAL VOLUME MOUNT OPTIONS

linear[=[<*DM−LINEAR−DEVICE*>]]
Setup a logical volume. Optionally, use the specified dm-linear device mapper device.

numsectors=<*SECTOR−COUNT*>
Set the volume size in sectors. Setup a logical or a cryptographic volume.
See the **−−size** option in **cryptsetup**(8).

startsector=<*START−SECTOR*>
Set the volume data start offset in sectors. Setup a logical or a cryptographic volume.
See the **−−offset** option in **cryptsetup**(8).

These mount options can be suffixed with a device layer index **1** to **9** (the default is **1**).

CRYPTOGRAPHIC VOLUME MOUNT OPTIONS

crypt[=[<*DM−CRYPT−DEVICE*>]]
Setup a cryptographic volume. Optionally, use the specified dm-crypt device mapper device.

cipher=<*CIPHER*>
Set the cryptographic volume encryption algorithm. Setup a cryptographic volume without using LUKS (see the **crypt** and **noluks** mount options).
See the **−−cipher** option in **cryptsetup**(8).

ivoffset=<*OFFSET*>
Set cryptographic volume IV start offset. Do not setup a cryptographic volume using LUKS (see the **noluks** mount options).
See the **−−skip** option in **cryptsetup**(8).

luks Setup a cryptographic volume using LUKS. This is the default if a cryptographic volume will be set up.

noluks Do not setup a cryptographic volume using LUKS. This does not, however, prevent a cryptographic volume from being set up. Only the use LUKS is prevented.

autoluks
If the device is a LUKS partition, setup a cryptographic volume using LUKS (see the **luks** mount options). If the device is not a LUKS partition, do not setup a cryptographic volume using LUKS (see the **noluks** mount options).

These mount options can be suffixed with a device layer index **1** to **9** (the default is **1**).

FILESYSTEM MOUNT OPTIONS

fstype=<*FS−TYPE*>
Set the filesystem type to be used while checking the filesystem and while mounting the filesystem.
See the **−t** option in **fsck**(8).
See the **−t** option in **mount**(8).

fsck Check the filesystem before mounting it read-write.

nofsck Do not check the filesystem. This is the default.

STANDARD MOUNT OPTIONS

remount
Remount an already mounted filesystem. Do not setup devices. Do not check the filesystem.

ro Mount the filesystem read-only. Setup read-only logical and cryptographic volumes.

rw Mount the filesystem read-write. Setup read-write logical and cryptographic volumes.

user, **users**
Ignored. Note that the **mount**(8) command maps these mount options to mount options **nodev**, **noexec** and **nosuid** before invoking the **mount.cryptfs**(8) utility.

nouser, **nousers**

Ignored.

Other standard mount options will be passed to the encrypted filesystem.

FILESYSTEM SPECIFIC MOUNT OPTIONS

Filesystem specific mount options will be passed to the encrypted filesystem.

ENVIRONMENT VARIABLES

PASSWD

A passphrase to be used as an encryption key or as an encryption key passphrase unless a plaintext key file is specified.

PASSWD_FD

A file descriptor of an open file whose content is to be used as an encryption key or as an encryption key passphrase unless a plaintext key file is specified.

PASSWD_FILE

A pathname of a file whose content is to be used as an encryption key or as an encryption key passphrase unless a plaintext key file is specified.

FILES

<LOOP−DEVICE>, */dev/loop<NUMBER>*

A loop device to be used if a loop device will be set up.

<DM−LINEAR−DEVICE>, */dev/mapper/cryptfs−<MAJOR>.<MINOR>*[*.<INODE>*]*−<LAYER>−linear*

A dm-linear device mapper device to be used if a logical volume will be set up.

<DM−CRYPT−DEVICE>, */dev/mapper/cryptfs−<MAJOR>.<MINOR>*[*.<INODE>*]*−<LAYER>−crypt*

A dm-crypt device mapper device to be used if a cryptographic volume will be set up.

/etc/fstab

A file containing static filesystem entries.

/etc/mtab

A file containing entries for mounted filesystems.

LOOP DEVICES

When the **mount.cryptfs**(8) utility is invoked indirectly by the **mount**(8) command, loop devices can be set up either by the **mount**(8) command or by the **mount.cryptfs**(8) utility (or in a rare case by both of them) but it is usually advisable to let the loop devices be set up by the **mount.cryptfs**(8) utility.

If the **mount**(8) command is invoked using a command similar to

```
mount −t cryptfs −o noauto,users,exec,crypt,loop1 <FILE> <DIRECTORY>
```

then the **mount**(8) command will invoke the **mount.cryptfs**(8) utility using a command similar to

```
mount.cryptfs <FILE> <DIRECTORY> −o rw,noauto,nodev,nosuid,users,crypt,loop1
```

and the **mount.cryptfs**(8) utility will setup a loop device and everything will work as expected. The mount options **loop1**, **cryptfs_loop** and **cryptfs_loop1** can be used interchangeably.

On the other hand, if the **mount**(8) command is invoked using a command similar to

```
mount −t cryptfs −o noauto,users,exec,crypt,loop <FILE> <DIRECTORY>
```

then the **mount**(8) command will setup a loop device and will invoke the **mount.cryptfs**(8) utility using a command similar to

```
mount.cryptfs <LOOP−DEVICE> <DIRECTORY> −o rw,noauto,nodev,nosuid,users,crypt
          −o loop=<LOOP−DEVICE>
```

in the case of newer versions of the **mount**(8) command and

```
mount.cryptfs <LOOP−DEVICE> <DIRECTORY> −o rw,noauto,nodev,nosuid,users,crypt
```

in the case of older versions of the **mount**(8) command. The original device *<FILE>* is not specified at all in any case and the **loop** mount option is not specified in the case of older versions of the **mount**(8) command.

The loss of the **loop** mount option is a severe problem because as a result the **mount.cryptfs**(8) command will not write the **loop** mount option to the */etc/mtab* file and therefore the **umount**(8) command will not release the loop device. This problem can however be worked around using the **loop0** mount option.

The loss of the original device *<FILE>* is a much less severe problem. If the filesystem is mounted by root, the loss is mainly aesthetic. But if the filesystem is mounted by an ordinary user using an */etc/fstab* entry similar to

```
<FILE> <DIRECTORY> cryptfs noauto,users,exec,crypt,loop,loop0 0 0
```

and a command similar to

```
mount <DIRECTORY>
```

the loss prevents the ordinary user from unmounting the filesystem because the device specified in the */etc/fstab* file is *<FILE>* and the device specified in the */etc/mtab* file will be *<LOOP−DEVICE>* and the **umount**(8) command does not permit ordinary users to unmount filesystems when there are such conflicts. This problem can be worked around using an */etc/fstab* entry similar to

```
<FILE> <DIRECTORY> cryptfs noauto,users,exec,crypt,dev0=<FILE>,loop,loop0 0 0
```

but now the original device *<FILE>* must be specified twice.

It is thus usually advisable to let the loop devices be set up by the **mount.cryptfs**(8) utility.

SEE ALSO

umount.cryptfs(8), **cryptsetup**(8), **dmsetup**(8), **fsck**(8), **gpg**(1), **losetup**(8), **mount**(8) **openssl**(1ssl)

AUTHOR

Eero Häkkinen