NAME

gbde - operation and management utility for Geom Based Disk Encryption

SYNOPSIS

gbde attach *destination* [**-k** *keyfile*] [**-l** *lockfile*] [**-p** *pass-phrase*]
gbde detach *destination*
gbde init *destination* [**-i**] [**-f** *filename*] [**-K** *new-keyfile*] [**-L** *new-lockfile*] [**-P** *new-pass-phrase*]
gbde setkey *destination* [**-n** *key*] [**-k** *keyfile*] [**-l** *lockfile*] [**-p** *pass-phrase*] [**-K** *new-keyfile*]
[**-L** *new-lockfile*] [**-P** *new-pass-phrase*]
gbde nuke *destination* [**-n** *key*] [**-k** *keyfile*] [**-l** *lockfile*] [**-p** *pass-phrase*]
gbde destroy *destination* [**-k** *keyfile*] [**-l** *lockfile*] [**-p** *pass-phrase*]

DESCRIPTION

NOTICE: Please be aware that this code has not yet received much review and analysis by qualified cryptographers and therefore should be considered a slightly suspect experimental facility.

We cannot at this point guarantee that the on-disk format will not change in response to reviews or bug-fixes, so potential users are advised to be prepared that dump(8)/restore(8) based migrations may be called for in the future.

The **gbde** utility is the only official operation and management interface for the gbde(4) GEOM based disk encryption kernel facility. The interaction between the **gbde** utility and the kernel part is not a published interface.

The operational aspect consists of two subcommands: one to open and attach a device to the in-kernel cryptographic **gbde** module (**attach**), and one to close and detach a device (**detach**).

The management part allows initialization of the master key and lock sectors on a device (**init**), initialization and replacement of pass-phrases (**setkey**), and key invalidation (**nuke**) and blackening (**destroy**) functions.

The **-l** *lockfile* argument is used to supply the lock selector data. If no **-l** option is specified, the first sector is used for this purpose.

The **-L** *new-lockfile* argument specifies the lock selector file for the key initialized with the **init** subcommand or modified with the **setkey** subcommand.

The **-n** *key* argument can be used to specify to which of the four keys the operation applies. A value of 1 to 4 selects the specified key, a value of 0 (the default) means "this key" (i.e., the key used to gain access to the device) and a value of -1 means "all keys".

The **-f** *filename* specifies an optional parameter file for use under initialization.

Alternatively, the **-i** option toggles an interactive mode where a template file with descriptions of the parameters can be interactively edited.

The **-p** *pass-phrase* argument specifies the pass-phrase used for opening the device. If not specified, the controlling terminal will be used to prompt the user for the pass-phrase. Be aware that using this option may expose the pass-phrase to other users who happen to run ps(1) or similar while the command is running.

The **-P** *new-pass-phrase* argument can be used to specify the new pass-phrase to the **init** and **setkey** subcommands. If not specified, the user is prompted for the new pass-phrase on the controlling terminal. Be aware that using this option may expose the pass-phrase to other users who happen to run ps(1) or similar while the command is running.

The **-k** *keyfile* argument specifies a key file to be used in combination with the pass-phrase (whether the pass-phrase is specified on the command line or entered from the terminal) for opening the device. The device will only be opened if the contents of the key file and the pass-phrase are both correct.

The **-K** *new-keyfile* argument can be used to specify a new key file to the **init** and **setkey** subcommands. If not specified, no key file will be used (even if one was previously used).

EXAMPLES

To initialize a device, using default parameters:

```
gbde init /dev/ada0s1f -L /etc/ada0s1f.lock
```

To attach an encrypted device:

```
gbde attach ada0s1f -l /etc/ada0s1f.lock
```

The encrypted device has the suffix *.bde* so a typical command to create and mount a file system would be:

```
newfs /dev/ada0s1f.bde
mount /dev/ada0s1f.bde /secret
```

To detach an encrypted device:

```
gbde detach ada0s1f
```

Please notice that detaching an encrypted device corresponds to physically removing it, do not forget to unmount the file system first.

To initialize the second key using a detached lockfile and a trivial pass-phrase:

```
gbde setkey ada0s1f -n 2 -P foo -L key2.lockfile
```

To invalidate your own masterkey:

```
gbde nuke ada0s1f
```

This will overwrite your masterkey sector with zeros, and results in a diagnostic if you try to use the key again. You can also destroy the other three copies of the masterkey with the -n argument.

You can also invalidate your masterkey without leaving a tell-tale sector full of zeros:

```
gbde destroy ada0s1f
```

This will overwrite the information fields in your masterkey sector, encrypt it and write it back. You get a (different) diagnostic if you try to use it.

SEE ALSO

gbde(4), geom(4)

HISTORY

This software was developed for the FreeBSD Project by Poul-Henning Kamp and NAI Labs, the Security Research Division of Network Associates, Inc. under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"), as part of the DARPA CHATS research program. **gbde** first appeared in FreeBSD 5.0.

AUTHORS

Poul-Henning Kamp <*phk@FreeBSD.org*>

BUGS

The cryptographic algorithms and the overall design have not been attacked mercilessly for over 10 years by a gang of cryptoanalysts.