cryptsetup Disk encryption

CONCEPTS

> luksFormat destroys all data on the device. Double-check the device name with lsblk first.

Block device /dev/sdb1
↓ open
Mapping /dev/mapper/<name>
↓ mount
Filesystem readable data

> Without the LUKS header the data is gone for good. Keep a header backup somewhere safe, off the disk itself.

> New volumes default to LUKS2. Use --type luks1 only for old tools, such as some bootloaders.

CREATE

cryptsetup luksFormat *<device>*
Encrypt a device with LUKS

cryptsetup luksFormat --type luks1 *<device>*
Use the older LUKS1 format

cryptsetup luksFormat --cipher *<cipher>* **--key-size** *<bits>* *<device>*
Choose cipher and key size

cryptsetup luksFormat --pbkdf argon2id *<device>*
Choose the key derivation function

cryptsetup benchmark
Measure cipher speed

cryptsetup isLuks *<device>*
Exit status 0 if it is LUKS

OPEN AND CLOSE

cryptsetup open *<device>* *<name>*
Unlock and map a device

cryptsetup close *<name>*
Remove the mapping and wipe the key

cryptsetup status *<name>*
Show an active mapping

cryptsetup open --key-file *<file>* *<device>* *<name>*
Unlock with a key file

OPEN AND CLOSE (cont.)

cryptsetup open --readonly *<device>* *<name>*
Unlock read-only

cryptsetup open --allow-discards *<device>* *<name>*
Pass TRIM through to the disk

cryptsetup open --header *<file>* *<device>* *<name>*
Use a detached header

cryptsetup open --test-passphrase *<device>*
Check a passphrase only

cryptsetup luksSuspend *<name>*
Freeze I/O and wipe the key

cryptsetup luksResume *<name>*
Unlock a suspended volume

KEY SLOTS

cryptsetup luksAddKey *<device>*
Add another passphrase

cryptsetup luksAddKey *<device>* *<keyfile>*
Add a key file

cryptsetup luksChangeKey *<device>*
Change a passphrase

cryptsetup luksRemoveKey *<device>*
Remove the passphrase you type

cryptsetup luksKillSlot *<device>* *<slot>*
Remove a slot by number

dd if=/dev/urandom of=*<keyfile>* **bs=512 count=8**
Make a random key file

INSPECT AND HEADER

cryptsetup luksDump *<device>*
Show header, slots and digests

cryptsetup luksUUID *<device>*
Print the UUID

cryptsetup luksHeaderBackup *<device>* **--header-backup-file** *<file>*
Save the header to a file

cryptsetup luksHeaderRestore *<device>* **--header-backup-file** *<file>*
Restore a saved header

INSPECT AND HEADER (cont.)

cryptsetup erase *<device>*
Remove all key slots

blkid *<device>*
Show type crypto_LUKS and UUID

RESIZE

cryptsetup resize *<name>*
Grow mapping to fill the device

resize2fs /dev/mapper/*<name>*
Grow an ext4 filesystem

BOOT AND FSTAB

echo "*<name>* **UUID=***<uuid>* **none luks" >> /etc/crypttab**
Unlock at boot

echo "/dev/mapper/*<name>* *<dir>* **ext4 defaults 0 2" >> /etc/fstab**
Mount after unlock

Common exit codes

Code	Meaning
0	success
1	wrong parameters
2	bad passphrase or no permission
4	wrong device
5	device exists or busy

RECIPES

Encrypt a new USB stick

```
# find the device first
lsblk
cryptsetup luksFormat /dev/sdX1
cryptsetup open /dev/sdX1 usb
mkfs.ext4 /dev/mapper/usb
mount /dev/mapper/usb /mnt
```

Unlock, use, lock

```
cryptsetup open /dev/sdX1 usb
mount /dev/mapper/usb /mnt
# work
umount /mnt
cryptsetup close usb
```

Back up the header

```
cryptsetup luksHeaderBackup /dev/
sdX1 \
  --header-backup-file usb-
header.img
cryptsetup luksDump /dev/sdX1
```

Rotate a passphrase

```
cryptsetup luksAddKey /dev/sdX1
cryptsetup luksRemoveKey /dev/
sdX1
# type the old passphrase to
remove it
```