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.

CREATE

`cryptsetup luksFormat <device>`
Encrypt a device with 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

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

KEY SLOTS

`cryptsetup luksAddKey <device>`
Add another passphrase

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

KEY SLOTS (cont.)

`cryptsetup luksChangeKey <device>`
Change a passphrase

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

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

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

`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