NAME

cryptsetup-open, cryptsetup-create, cryptsetup-plainOpen, cryptsetup-luksOpen, cryptsetup-loopaesOpen, cryptsetup-tcryptOpen, cryptsetup-bitlkOpen, cryptsetup-fvault2Open − open an encrypted device and create a mapping with a specified name

SYNOPSIS

cryptsetup *open* **−−type <device_type> [<options>] <device> <name>**

DESCRIPTION

Opens (creates a mapping with) <name> backed by device <device>.

Device type can be *plain*, *luks* (default), *luks1*, *luks2*, *loopaes* or *tcrypt*.

For backward compatibility, there are **open** command aliases:

create (argument−order <name> <device>): open −−type plain
plainOpen: open −−type plain
luksOpen: open −−type luks
loopaesOpen: open −−type loopaes
tcryptOpen: open −−type tcrypt
bitlkOpen: open −−type bitlk

<options> are type−specific and are described below for individual device types. For **create**, the order of the <name> and <device> options is inverted for historical reasons; all other aliases use the standard **<device> <name>** order.

PLAIN

open −−type plain <device> <name> −−cipher <spec> −−key−size <bits> −−hash <alg>
plainOpen <device> <name> (old syntax)
create <name> <device> (OBSOLETE syntax)

Opens (creates a mapping with) <name> backed by device <device>.

You should always specify options −−cipher, −−key−size and (if no keyfile or keyring is used) then also −−hash to avoid incompatibility, as default values can differ in older cryptsetup versions.

The plain format also allows retrieving a volume key from a kernel keyring specified by −−volume−key−keyring. The key in the kernel keyring must be configured before issuing cryptsetup commands, as cryptsetup does not upload any keys to the keyring in plain mode. For subsequent commands (like resize), the user must ensure that the key in the keyring is unchanged. Otherwise, reloading the key can cause data corruption after an unexpected key change.

<options> can be [−−hash, −−cipher, −−sector−size, −−key−file, −−keyfile−size, −−keyfile−offset, −−key−size, −−offset, −−skip, −−device−size, −−size, −−readonly, −−shared, −−allow−discards, −−refresh, −−timeout, −−verify−passphrase, −−iv−large−sectors, −−volume−key−keyring].

EXAMPLES:

To map the encrypted device /dev/sda10 to the decrypted device /dev/mapper/e1, you can use:

cryptsetup open −−type plain −−cipher aes−cbc−essiv:sha256 −−key−size 256 −−hash sha256 /dev/sda10 e1

The decrypted device can then be used as a normal block device to mount a filesystem.

To map a device with a volume key in the preconfigured trusted or encrypted keyring, you need to specify

the keyring with the key and remove the hash specification, for example, to use **%trusted:mykey**:

cryptsetup open −−type plain /dev/sda10 e1 −−volume−key−keyring=%trusted:mykey −−cipher aes−xts−plain64 −−key−size 256

Note that the key size must match the preconfigured key in the keyring.

LUKS

open <device> <name>
open −−type <luks1|luks2> <device> <name> (explicit version request)
luksOpen <device> <name> (old syntax)

Opens the LUKS device <device> and sets up a mapping <name> after successful verification of the supplied passphrase.

First, the passphrase is searched in LUKS2 tokens unprotected by PIN. If such a token does not exist (or fails to unlock keyslot) and the passphrase is not supplied via −−key−file, the command prompts for passphrase interactively.

If there is a valid LUKS2 token but it requires a PIN to unlock the assigned keyslot, it is not used unless one of the following options is added: −−token−only, −−token−type where type matches the desired PIN−protected token or −−token−id with id matching the PIN−protected token.

<**options**> can be [−−key−file, −−keyfile−offset, −−keyfile−size, −−readonly, −−test−passphrase, −−allow−discards, −−header, −−key−slot, −−volume−key−file, −−token−id, −−token−only, −−token−type, −−disable−external−tokens, −−disable−keyring, −−disable−locks, −−type, −−refresh, −−serialize−memory−hard−pbkdf, −−unbound, −−tries, −−timeout, −−verify−passphrase, −−persistent, −−volume−key−keyring, −−link−vk−to−keyring, −−external−tokens−path].

loopAES

open −−type loopaes <device> <name> −−key−file <keyfile>
loopaesOpen <device> <name> −−key−file <keyfile> (old syntax)

Opens the loop−AES <device> and sets up a mapping <name>.

If the key file is encrypted with GnuPG, then you have to use −−key−file=− and decrypt it before use, e.g., like this: **gpg −−decrypt <keyfile> | cryptsetup loopaesOpen −−key−file=− <device> <name>**.

The loop−AES extension cannot use the direct input of the key file on the real terminal because the keys are separated by end−of−line, and only part of the multi−key file would be read. If you need it in script, just use the pipe redirection: **echo $keyfile | cryptsetup loopaesOpen −−key−file=− <device> <name>**.

Use −−keyfile−size to specify the proper key length if needed.

Use −−offset to specify device offset. Note that the units need to be specified in terms of 512−byte sectors.

Use −−skip to specify the IV offset. If the original device used an offset but did not use it in IV sector calculations, you must explicitly use −−skip 0 in addition to the offset parameter.

Use −−hash to override the default hash function for passphrase hashing (otherwise it is detected according to key size).

<**options**> can be [−−cipher, −−key−file, −−keyfile−size, −−keyfile−offset, −−key−size, −−offset, −−skip, −−hash, −−readonly, −−allow−discards, −−refresh].

TrueCrypt and VeraCrypt

open −−type tcrypt <device> <name>
tcryptOpen <device> <name> (old syntax)

Opens the TCRYPT (TrueCrypt and VeraCrypt compatible) <device> and sets up a mapping <name>.

The −−key−file option allows a combination of file content with the passphrase The −−key−file option can be repeated. Note that using keyfiles differs from LUKS keyfile logic.

If −−cipher or −−hash options are used, only cipher chains or PBKDF2 variants with the specified hash algorithms are checked. This could speed up unlocking the device (but also reveals some information about the container).

If you use −−header in combination with hidden or system options, the header file must contain specific headers in the same positions as the original encrypted container.

Option −−allow−discards cannot be combined with option −−tcrypt−hidden. For normal mapping, it can cause the destruction of hidden volume (hidden volume appears as unused space for outer volume, so this space can be discarded).

<options> can be [−−key−file, −−tcrypt−hidden, −−tcrypt−system, −−tcrypt−backup, −−readonly, −−test−passphrase, −−allow−discards, −−veracrypt (ignored), −−disable−veracrypt, −−veracrypt−pim, −−veracrypt−query−pim, −−header, −−cipher, −−hash, −−tries, −−timeout, −−verify−passphrase].

BitLocker

open −−type bitlk <device> <name>
bitlkOpen <device> <name> (old syntax)

Opens the BITLK (a BitLocker compatible) <device> and sets up a mapping <name>.

Note that −−test−passphrase doesn’t work with −−volume−key−file because we cannot check whether the provided volume key is correct for this device. When using −−volume−key−file, the device will be opened even if the provided key is incorrect.

<options> can be [−−key−file, −−keyfile−offset, −−keyfile−size, −−key−size, −−readonly, −−test−passphrase, −−allow−discards −−volume−key−file, −−tries, −−timeout, −−verify−passphrase].

FileVault2

open −−type fvault2 <device> <name>
fvault2Open <device> <name> (old syntax)

Opens the FVAULT2 (a FileVault2 compatible) <device> and sets up a mapping <name>.

<options> can be [−−key−file, −−keyfile−offset, −−keyfile−size, −−key−size, −−readonly, −−test−passphrase, −−allow−discards −−volume−key−file, −−tries, −−timeout, −−verify−passphrase].

OPTIONS

−−allow−discards

Allow the use of discard (TRIM) requests for the device. This is also not supported for LUKS2 devices with data integrity protection.

WARNING: This command can have a negative security impact because it can make filesystem−level operations visible on the physical device. For example, information leaking filesystem type, used space, etc., may be extractable from the physical device if the discarded blocks can be located later. If in doubt, do not use it.

A kernel version of 3.1 or later is needed. For earlier kernels, this option is ignored.

−−batch−mode, **−q**
Suppresses all confirmation questions. Use with care!

If the −−verify−passphrase option is not specified, this option also switches off the passphrase verification.

−−cipher, **−c** <*cipher−spec*>
Set the cipher specification string for the *plain* device type.

For the *tcrypt* device type, it restricts checked cipher chains when looking for the header.

cryptsetup −−help shows the compiled−in defaults.

If a hash is part of the cipher specification, then it is used as part of the IV generation. For example, ESSIV needs a hash function, while "plain64" does not and hence none is specified.

For XTS mode, you can optionally set a key size of 512 bits with the −s option. Key size for XTS mode is twice that for other modes for the same security level.

−−debug or **−−debug−json**
Run in debug mode with full diagnostic logs. Debug output lines are always prefixed by #.

If −−debug−json is used, additional LUKS2 JSON data structures are printed.

−−device−size *size[units]*
Instead of the real device size, use the specified value. Usable only with *plain* device type.

If no unit suffix is specified, the size is in bytes.

Unit suffix can be S for 512 byte sectors, K/M/G/T (or KiB, MiB, GiB, TiB) for units with 1024 base or KB/MB/GB/TB for 1000 base (SI scale).

−−disable−external−tokens
Disable loading of plugins for external LUKS2 tokens.

−−disable−keyring
Do not load the volume key in the kernel keyring; store it directly in the dm−crypt target instead. This option is supported only for the LUKS2 type.

−−disable−locks
Disable lock protection for metadata on disk. This option is valid only for LUKS2 and is ignored for other formats.

WARNING: Do not use this option unless you run cryptsetup in a restricted environment where locking is impossible to perform (where /run directory cannot be used).

−−disable−veracrypt
This option can be used to disable VeraCrypt compatible mode (only TrueCrypt devices are recognized). See the *TCRYPT* section in **cryptsetup**(8) for more info.

−−external−tokens−path <*absolute path*>
Override the system directory path where cryptsetup searches for external token handlers (or token

plugins). It must be an absolute path (starting with '/' character).

−−hash, **−h** *<hash−spec>*
Specifies the passphrase hash. Applies to *plain* and *loopaes* device types only.

For the *tcrypt* device type, it restricts the checked PBKDF2 variants when looking for the header.

−−header *<device or file storing the LUKS header>*
Specify a detached (separated) metadata device or file where the header is stored.

WARNING: There is no check whether the ciphertext device specified actually belongs to the header given. In fact, you can specify an arbitrary device as the ciphertext device with the −−header option. Use with care.

−−help, **−?**
Show help text and default parameters.

−−iv−large−sectors
Count Initialization Vector (IV) in larger sector size (if set) instead of 512−byte sectors. This option can be used only with the *plain* device type.

This option does not have any performance or security impact; use it only for accessing incompatible existing disk images from other systems that require this option.

−−key−description *text*
Set the key description in the keyring that will be used for passphrase retrieval.

−−key−file, **−d** *file*
Read the passphrase from the file.

If the name given is "−", then the passphrase will be read from stdin. In this case, reading will not stop at newline characters.

With *plain* device type, the passphrase obtained via −−key−file option is passed directly in dm−crypt. Unlike the interactive mode (stdin), where the digest of the passphrase is passed in dm−crypt instead.

See section *NOTES ON PASSPHRASE PROCESSING* in **cryptsetup**(8) for more information.

−−keyfile−offset *value*
Skip *value* bytes at the beginning of the key file.

−−keyfile−size, **−l** *value*
Read a maximum of *value* bytes from the key file. The default is to read the whole file up to the compiled−in maximum that can be queried with −−help. Supplying more data than the compiled−in maximum aborts the operation.

This option is useful to cut trailing newlines, for example. If −−keyfile−offset is also given, the size count starts after the offset.

−−key−size, **−s** *bits*
Sets key size in *bits*. The argument has to be a multiple of 8. The possible key sizes are limited by the cipher and mode used.

See /proc/crypto for more information. Note that the key size in /proc/crypto is stated in bytes.

This option can be used for *plain* and *luks* devices. For LUKS2 devices in reencryption, you may use the parameter twice to specify both old and new volume key sizes. Each −−key−size option corresponds to the respective −−volume−key−file parameter (also allowed to be used up to two times).

−−key−slot, **−S** *<0−N>*
This option selects a specific keyslot to compare the passphrase against. If the given passphrase would only matches a different keyslot, the operation fails.

The maximum number of keyslots depends on the LUKS version. LUKS1 can have up to 8 keyslots. LUKS2 can have up to 32 keyslots based on keyslot area size and key size, but a valid keyslot ID can always be between 0 and 31 for LUKS2.

−−link−vk−to−keyring *<keyring description>::<key description>*
Link the volume key in a keyring with the specified key name. The volume key is linked only if the requested action is successfully finished (with −−test−passphrase, the verified volume key is linked in a keyring without taking further action).

The *<keyring description>* string has to contain the existing kernel keyring description. The keyring name may be optionally prefixed with "%:" or "%keyring:" type descriptions. Or, the keyring may also be specified directly by numeric key id. Also, special keyring notations starting with "@" may be used to select existing predefined kernel keyrings.

The string "::" is a delimiter used to separate the keyring description and key description.

<key description> part describes key type and key name of volume key linked in the keyring described in *<keyring description>*. The type may be specified by adding a "%<type_name>:" prefix in front of the key name. If the type is missing, the default *user* type is applied. If the key of the same name and type already exists (already linked in the keyring), it will get replaced in the process.

See also the **KEY IDENTIFIERS** section of **keyctl**(1).

−−offset, **−o** *<number of 512 byte sectors>*
Start offset in the backend device in 512−byte sectors. This option is only relevant with plain or loopaes device types.

−−perf−high_priority
Set dm−crypt workqueues and the writer thread to high priority. This improves throughput and latency of dm−crypt while degrading the general responsiveness of the system.

This option is available only for low−level dm−crypt performance tuning, use only if you need a change to the default dm−crypt behaviour. Needs kernel 6.10 or later.

−−perf−no_read_workqueue, **−−perf−no_write_workqueue**
Bypass dm−crypt internal workqueue and process read or write requests synchronously.

These options are available only for low−level dm−crypt performance tuning, use only if you need a change to the default dm−crypt behaviour. Needs kernel 5.9 or later.

−−perf−same_cpu_crypt
Perform encryption using the same CPU on which that IO was submitted. The default is to use an unbound workqueue so that encryption work is automatically balanced between available CPUs.

This option is available only for low−level dm−crypt performance tuning, use only if you need a change to the default dm−crypt behaviour.

−−perf−submit_from_crypt_cpus
Disable offloading writes to a separate thread after encryption. There are some situations where offloading write bios from the encryption threads to a single thread degrades performance significantly. The default is to offload write bios to the same thread.

This option is available only for low−level dm−crypt performance tuning, use only if you need a change to the default dm−crypt behaviour.

−−persistent
If used with LUKS2 devices and activation commands like *open* or *refresh*, the specified activation flags are persistently written into metadata and used next time automatically, even for normal activation. (No need to use cryptab or other system configuration files.)

If you need to remove a persistent flag, use −−persistent without the flag you want to remove (e.g., to disable the persistently stored discard flag, use −−persistent without −−allow−discards).

Only −−allow−discards, −−perf−same_cpu_crypt, −−perf−submit_from_crypt_cpus, −−perf−no_read_workqueue, −−perf−no_write_workqueue and −−integrity−no−journal can be stored persistently.

−−readonly, **−r**
Set up a read−only mapping.

−−refresh
Refreshes an active device with a new set of parameters. See **cryptsetup−refresh**(8) for more details.

−−sector−size *bytes*
Set encryption sector size for use with *plain* device type. It must be a power of two and in the 512 − 4096 bytes range. The default encryption sector size is 512 bytes.

Increasing sector size from 512 to 4096 bytes can provide better performance on most modern storage devices and with some hardware encryption accelerators.

Note that using a sector size larger than the underlying storage device's physical sector size may result in data corruption during unexpected power failures. A power failure during write operations may result in only partial completion of the encryption sector write, leaving encrypted data in an inconsistent state that cannot be properly decrypted.

−−serialize−memory−hard−pbkdf
Use a global lock to serialize unlocking of keyslots using memory−hard PBKDF.

This is a workaround for a specific situation when multiple devices are activated in parallel, and the system, instead of reporting out of memory, starts unconditionally stop processes using the out−of−memory killer.

DO NOT USE this switch until you are implementing the boot environment with parallel devices activation!

−−shared
Creates an additional mapping for one common ciphertext device. Arbitrary mappings are supported. This option is only relevant for the *plain* device type. Use −−offset, −−size and −−skip to specify the mapped area.

−−size, **−b** <*number of 512 byte sectors*>

Set the size of the device in sectors of 512 bytes. Usable only with *plain* device type.

−−skip, **−p** *<number of 512 byte sectors>*
Start offset used in IV calculation in 512−byte sectors (how many sectors of the encrypted data to skip at the beginning). This option is only relevant with plain or loopaes device types.

Hence, if −−offset *n*, and −−skip *s*, sector *n* (the first sector of the encrypted device) will get a sector number of *s* for the IV calculation.

−−tcrypt−backup, **−−tcrypt−hidden**, **−−tcrypt−system**
Specify which TrueCrypt on−disk header will be used to open the device. See the *TCRYPT* section in **cryptsetup**(8) for more info.

Using a system−encrypted device with the −−tcrypt−system option requires specific settings to work as expected.

TrueCrypt/VeraCrypt supports full system encryption (only a partition table is not encrypted) or system partition encryption (only a system partition is encrypted). The metadata header then contains the offset and size of the encrypted area. Cryptsetup needs to know the specific partition offset to calculate encryption parameters. To properly map a partition, you must specify a real partition device so cryptsetup can calculate this offset.

While you can use a full device as a parameter (/dev/sdb), always prefer to specify the partition you want to map (/dev/sdb1), as only the system partition mode can be detected this way.

For mapping images (stored in a file), you can use the additional −−header option with the real partition device. If the −−header is used (and it is different from the data image), cryptsetup expects that the data image contains a snapshot of the data partition only.

If −−header is not used (or points to the same image), cryptsetup expects that the image contains a full disk (including the partition table). This can map a full encrypted area that is not directly mountable as a filesystem. Please prefer creating a loop device with partitions (**losetup −P**, see **losetup**(8) man page) and use a real partition (/dev/loopXp1) as the device parameter.

−−test−passphrase
Do not activate the device, just verify the passphrase. The device mapping name is not mandatory if this option is used.

−−timeout, **−t** *seconds*
The number of seconds to wait before a timeout on passphrase input via terminal. It is relevant every time a passphrase is asked. It has no effect if used in conjunction with −−key−file.

This option is useful when the system should not stall if the user does not input a passphrase, e.g., during boot. The default is a value of 0 seconds, which means to wait forever.

−−token−id
Specify what token to use and allow the token PIN prompt to take precedence over the interactive keyslot passphrase prompt. If omitted, all available tokens (not protected by PIN) will be checked before proceeding further with the passphrase prompt.

−−token−only
Do not proceed further with the action if the token−based keyslot unlock failed. Without the option, the action asks for a passphrase to proceed further.

It allows LUKS2 tokens protected by PIN to take precedence over the interactive keyslot passphrase prompt.

−−token−type *type*
Restrict tokens eligible for operation to a specific token *type*. Mostly useful when no −−token−id is specified.

It allows LUKS2 *type* tokens protected by PIN to take precedence over the interactive keyslot passphrase prompt.

−−tries, **−T**
How often the input of the passphrase shall be retried. The default is 3 tries.

−−type *type*
Specifies required device type, for more info, read the *BASIC ACTIONS* section in **cryptsetup**(8).

−−unbound
Allowed only together with −−test−passphrase parameter, it allows one to test the passphrase for an unbound LUKS2 keyslot. Otherwise, an unbound keyslot passphrase can be tested only when a specific keyslot is selected via −−key−slot parameter.

−−usage
Show short option help.

−−veracrypt
This option is ignored as VeraCrypt compatible mode is supported by default.

−−veracrypt−pim, **−−veracrypt−query−pim**
Use a custom Personal Iteration Multiplier (PIM) for the VeraCrypt device. See the *TCRYPT* section in **cryptsetup**(8) for more info.

−−verify−passphrase, **−y**
When interactively asking for a passphrase, ask for it twice and complain if both inputs do not match. Advised when creating a *plain* type mapping for the first time. Ignored on input from file or stdin.

−−version, **−V**
Show the program version.

−−volume−key−file *file*, −−master−key−file file (OBSOLETE alias)
Use a volume key stored in a file.

This allows one to open *luks* and *bitlk* device types without giving a passphrase.

For devices in reencryption, the option may be used twice to specify both old and new volume keys. When using the option twice, make sure you pair each −−volume−key−file option with the respective −−key−size parameter as well.

−−volume−key−keyring <*key description*>
Use a volume key stored in a keyring. This allows one to open *luks* and *plain* device types without giving a passphrase.

For LUKS, the key and associated type have to be readable from userspace so that the volume key digest may be verified before activation. For devices in reencryption, the option may be used twice to specify both old and new volume keys.

For PLAIN type, the user must ensure that the key in the keyring is unchanged since activation. Otherwise, reloading the key can cause data corruption after an unexpected key change.

The <*key description*> uses keyctl−compatible syntax. This can either be a numeric key ID or a string name in the format *%<key type>:<key name>*. See also the **KEY IDENTIFIERS** section of **keyctl**(1). When no *%<key type>:* prefix is specified, we assume the key type is *user* (default type).

REPORTING BUGS

Report bugs at cryptsetup mailing list <cryptsetup@lists.linux.dev> or in Issues project section <https://gitlab.com/cryptsetup/cryptsetup/−/issues/new>.

Please attach the output of the failed command with −−debug option added.

SEE ALSO

Cryptsetup FAQ <https://gitlab.com/cryptsetup/cryptsetup/wikis/FrequentlyAskedQuestions>

cryptsetup(8), **integritysetup**(8) and **veritysetup**(8)

CRYPTSETUP

Part of cryptsetup project <https://gitlab.com/cryptsetup/cryptsetup/>.