Rocky Enterprise Linux 9.2 Manual Pages on command 'cryptsetup-loopaesOpen.8'

$ man cryptsetup-loopaesOpen.8

CRYPTSETUP-OPEN(8) Maintenance Commands CRYPTSETUP-OPEN(8)

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>

plainOpen <device> <name> (old syntax)

create <name> <device> (OBSOLETE syntax)

Opens (creates a mapping with) <name> backed by device <device>. <options> can be [--hash, --cipher, --verify-passphrase, --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].

Example: 'cryptsetup open --type plain /dev/sda10 e1' maps the raw encrypted device /dev/sda10 to the mapped (decrypted) device /dev/mapper/e1, which can then be mounted, fsck-ed or have a filesystem created on it.

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 LUKS tokens. If it?s not found in any token and also the passphrase is not supplied via --key-file, the command prompts for it interactively.

<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].

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>

WARNING: 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 number of 512 byte sectors.

Use --skip to specify the IV offset. If the original device used an

offset and but did not use it in IV sector calculations, you have to

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>.

<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].

The keyfile parameter allows a combination of file content with the passphrase and can be repeated. Note that using keyfiles is compatible with TCRYPT and is different 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 it reveals some information about the container).

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

WARNING: 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).

BitLocker

open --type bitlk <device> <name>

bitlkOpen <device> <name> (old syntax)

Opens the BITLK (a BitLocker 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].

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

--type <device-type>

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

--hash, -h <hash-spec>

Specifies the passphrase hash. Applies to plain and loopaes device types only.

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

--cipher, -c <cipher-spec>

Set the cipher specification string for plain device type.

For tcrypt device type it restricts checked cipher chains when looking for 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.

--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.

--key-file, -d name

Read the passphrase from file.

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

NOTE: With plain device type, the passphrase obtained via --key-file option is passed directly in dm-crypt. Unlike the interactive mode (stdin) where digest (--hash option) 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.

--volume-key-file, --master-key-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.

--key-slot, -S <0-N>

This option selects a specific key-slot to compare the passphrase against. If the given passphrase would only match a different key-slot, the operation fails.

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

--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 key-size in /proc/crypto is stated in bytes.

This option can be used for plain device type only.

--size, -b <number of 512 byte sectors>

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

--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.

--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.

--device-size size[units]

Instead of real device size, use 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).

--readonly, -r

set up a read-only mapping.

--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.

--timeout, -t <number of seconds>

The number of seconds to wait before 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.

--tries, -T

How often the input of the passphrase shall be retried. The default is 3 tries.

--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.

--perf-same_cpu_crypt

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

NOTE: This option is available only for low-level dm-crypt performance tuning, use only if you need a change to default dm-crypt behaviour. Needs kernel 4.0 or later.

--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.

NOTE: This option is available only for low-level dm-crypt performance tuning, use only if you need a change to default dm-crypt behaviour. Needs kernel 4.0 or later.

--perf-no_read_workqueue, --perf-no_write_workqueue

Bypass dm-crypt internal workqueue and process read or write requests synchronously.

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

--test-passphrase

Do not activate the device, just verify passphrase. The device

mapping name is not mandatory if this option is used.

--header <device or file storing the LUKS header>

Specify 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.

--disable-external-tokens

Disable loading of plugins for external LUKS2 tokens.

--disable-locks

Disable lock protection for metadata on disk. This option is valid only for LUKS2 and 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-keyring

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

--token-id

Specify what token to use. If omitted, all available tokens will be checked before proceeding further with passphrase prompt.

--token-only

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

--token-type type

Restrict tokens eligible for operation to specific token type.

Mostly useful when no --token-id is specified.

--sector-size bytes

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

512 bytes.

Note that if sector size is higher than underlying device hardware sector, using this option can increase risk on incomplete sector writes during a power fail.

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

--iv-large-sectors

Count Initialization Vector (IV) in larger sector size (if set) instead of 512 bytes sectors. This option can be used only with plain device type.

NOTE: 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.

--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 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.

--refresh

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

--unbound

Allowed only together with --test-passphrase parameter, it allows one to test passphrase for unbound LUKS2 keyslot. Otherwise,

unbound keyslot passphrase can be tested only when specific keyslot is selected via --key-slot parameter.

--tcrypt-hidden, --tcrypt-system, --tcrypt-backup

Specify which TrueCrypt on-disk header will be used to open the device. See TCRYPT section in cryptsetup(8) for more info.

--veracrypt

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

--disable-veracrypt

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

--veracrypt-pim, --veracrypt-query-pim

Use a custom Personal Iteration Multiplier (PIM) for VeraCrypt device. See TCRYPT section in cryptsetup(8) for more info.

--serialize-memory-hard-pbkdf

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

NOTE: This is (ugly) workaround for a specific situation when multiple devices are activated in parallel and system instead of reporting out of memory starts unconditionally stop processes using out-of-memory killer.

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

--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.

--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.

--version, -V

Show the program version.

--usage

Show short option help.

--help, -?

Show help text and default parameters. == 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 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/>.

cryptsetup 2.6.0 2022-12-14 CRYPTSETUP-OPEN(8)