NAME

cryptsetup-reencrypt - tool for offline LUKS device re-encryption

SYNOPSIS

cryptsetup-reencrypt <options> <device>

DESCRIPTION

Cryptsetup-reencrypt can be used to change reencryption parameters which otherwise require full on-disk data change (re-encryption).

You can regenerate **volume key** (the real key used in on-disk encryption unclocked by passphrase), **cipher**, **cipher mode**.

Cryptsetup-reencrypt reencrypts data on LUKS device in-place. During reencryption process the LUKS device is marked unavailable.

NOTE: If you're looking for LUKS2 online reencryption manual please read cryptsetup(8) man page instead (see reencrypt action). This page is for legacy offline reencryption utility only.

WARNING: The cryptsetup-reencrypt program is not resistant to hardware or kernel failures during reencryption (you can lose your data in this case).

ALWAYS BE SURE YOU HAVE RELIABLE BACKUP BEFORE USING THIS TOOL.
The reencryption can be temporarily suspended (by TERM signal or by using ctrl+c) but you need to retain temporary files named LUKS-<uuid>.[log|org|new]. LUKS device is unavailable until reencryption is finished though.

Current working directory must be writable and temporary files created during reencryption must be present.

For more info about LUKS see cryptsetup(8).

OPTIONS

To start (or continue) re-encryption for <device> use:

cryptsetup-reencrypt <device>

<options> can be [−−batch-mode, −−block-size, −−cipher | −−keep-key, −−debug, −−device-size, −−hash, −−header, −−iter-time | −−pbkdf−force−iterations, −−key-file, −−key-size, −−key-slot, −−keyfile-offset, −−keyfile-size, −−master−key−file, −−tries, −−pbkdf, −−pbkdf−memory, −−pbkdf−parallel, −−progress-frequency, −−use-directio, −−use-random | −−use-urandom, −−use-fsync, −−uuid, −−verbose, −−write-log]

To encrypt data on (not yet encrypted) device, use −−*new* in combination with −−*reduce-device-size* or with −−*header* option for detached header.

To remove encryption from device, use −−*decrypt*.

For detailed description of encryption and key file options see *cryptsetup(8)* man page.

−−batch-mode, −q
Suppresses all warnings and reencryption progress output.

−−block-size, −B *value*
Use re-encryption block size of <value> in MiB.

Values can be between 1 and 64 MiB.

−−cipher, −c <*cipher-spec*>
Set the cipher specification string.

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

−−decrypt
Remove encryption (decrypt already encrypted device and remove LUKS header).

WARNING: This is destructive operation and cannot be reverted.

−−device-size *size[units]*
Instead of real device size, use specified value.

It means that only specified area (from the start of the device to the specified size) will be reencrypted.

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).

WARNING: This is destructive operation.

−−hash, −h <*hash-spec*>
Specifies the hash used in the LUKS1 key setup scheme and volume key digest.

NOTE: if this parameter is not specified, default hash algorithm is always used for new LUKS1 device header.

NOTE: with LUKS2 format this option is only relevant when new keyslot pbkdf algorithm is set to PBKDF2 (see *−−pbkdf*).

−−header <*LUKS header file*>
Use a detached (separated) metadata device or file where the LUKS header is stored. This option allows one to store ciphertext and LUKS header on different devices.

WARNING: There is no check whether the ciphertext device specified actually belongs to the header given. If used with*−−ne w* option, the header file will created (or overwritten). Use with care.

−−iter-time, −i <*milliseconds*>
The number of milliseconds to spend with PBKDF2 passphrase processing for the new LUKS header.

−−keep-key
Do not change encryption key, just reencrypt the LUKS header and keyslots.

This option can be combined only with *−−hash*, *−−iter-time*, *−−pbkdf−force−iterations*, *−−pbkdf* (LUKS2 only), *−−pbkdf−memory* (Argon2i/id and LUKS2 only) and *−−pbkdf−parallel* (Argon2i/id and LUKS2 only) options.

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

WARNING: −−key-file option can be used only if there is only one active keyslot, or alternatively, also if −−key-slot option is specified (then all other keyslots will be disabled in new LUKS device).

If this option is not used, cryptsetup-reencrypt will ask for all active keyslot passphrases.

−−key-size, −s <*bits*>
Set 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.

If you are increasing key size, there must be enough space in the LUKS header for enlarged keyslots (data offset must be large enough) or reencryption cannot be performed.

If there is not enough space for keyslots with new key size, you can destructively shrink device with −−reduce-device-size option.

−−key-slot, −S <0-MAX>
Specify which key slot is used. For LUKS1, max keyslot number is 7. For LUKS2, it's 31.

WARNING: All other keyslots will be disabled if this option is used.

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

−−keyfile-size, −l
Read a maximum of *value* bytes from the key file. Default is to read the whole file up to the compiled-in maximum.

−−master−key−file
Use new volume (master) key stored in a file.

−−new, −N
Create new header (encrypt not yet encrypted device).

This option must be used together with −−reduce-device-size.

WARNING: This is destructive operation and cannot be reverted.

−−pbkdf
Set Password-Based Key Derivation Function (PBKDF) algorithm for LUKS keyslot. The PBKDF can be: *pbkdf2*, *argon2i* for Argon2i or *argon2id* for Argon2id.

For LUKS1, only *pbkdf2* is accepted (no need to use this option).

−−pbkdf−force−iterations <num>
Avoid PBKDF benchmark and set time cost (iterations) directly.

−−pbkdf−memory <number>
Set the memory cost for PBKDF (for Argon2i/id the number represents kilobytes). Note that it is maximal value, PBKDF benchmark or available physical memory can decrease it. This option is not available for PBKDF2.

−−pbkdf−parallel <number>
Set the parallel cost for PBKDF (number of threads, up to 4). Note that it is maximal value, it is decreased automatically if CPU online count is lower. This option is not available for PBKDF2.

−−progress-frequency <seconds>
Print separate line every <seconds> with reencryption progress.

−−reduce-device-size *size[units]*
Enlarge data offset to specified value by shrinking device size.

This means that last sectors on the original device will be lost, ciphertext data will be effectively shifted by specified number of sectors.

It can be useful if you e.g. added some space to underlying partition (so last sectors contains no data).

For units suffix see −−device-size parameter description.

You cannot shrink device more than by 64 MiB (131072 sectors).

WARNING: This is destructive operation and cannot be reverted. Use with extreme care - shrunk filesystems are usually unrecoverable.

−−tries, −T
Number of retries for invalid passphrase entry.

−−type <type>
Use only while encrypting not yet encrypted device (see −−new).

Specify LUKS version when performing in-place encryption. If the parameter is omitted default value (LUKS1) is used. Type may be one of: **luks** (default), **luks1** or **luks2**.

−−use-directio
Use direct-io (O_DIRECT) for all read/write data operations related to block device undergoing reencryption.

Useful if direct-io operations perform better than normal buffered operations (e.g. in virtual environments).

−−use-fsync
Use fsync call after every written block. This applies for reencryption log files as well.

−−use-random

−−use-urandom
Define which kernel random number generator will be used to create the volume key.

−−uuid <*uuid*>
Use only while resuming an interrupted decryption process (see −−decrypt).

To find out what <*uuid*> to pass look for temporary files LUKS-<uuid>.[|log|org|new] of the interrupted decryption process.

−−verbose, −v
Print more information on command execution.

−−version
Show the program version.

−−write-log
Update log file after every block write. This can slow down reencryption but will minimize data loss in the case of system crash.

RETURN CODES

Cryptsetup-reencrypt returns 0 on success and a non-zero value on error.

Error codes are: 1 wrong parameters, 2 no permission, 3 out of memory, 4 wrong device specified, 5 device already exists or device is busy.

EXAMPLES

Reencrypt /dev/sdb1 (change volume key)
cryptsetup-reencrypt /dev/sdb1

Reencrypt and also change cipher and cipher mode

cryptsetup-reencrypt /dev/sdb1 −c aes-xts-plain64

Add LUKS encryption to not yet encrypted device

First, be sure you have space added to disk.

Or alternatively shrink filesystem in advance.
Here we need 4096 512-bytes sectors (enough for 2x128 bit key).

fdisk −u /dev/sdb # move sdb1 partition end + 4096 sectors (or use resize2fs or tool for your filesystem and shrink it)

cryptsetup-reencrypt /dev/sdb1 −−new −−reduce-device-size 4096S

Remove LUKS encryption completely

cryptsetup-reencrypt /dev/sdb1 −−decrypt

REPORTING BUGS

Report bugs, including ones in the documentation, on the cryptsetup mailing list at <dm-crypt@saout.de> or in the ’Issues’ section on LUKS website. Please attach the output of the failed command with the −−debug option added.

AUTHORS

Cryptsetup-reencrypt was written by Milan Broz <gmazyland@gmail.com>.

COPYRIGHT

SEE ALSO

The project website at **https://gitlab.com/cryptsetup/cryptsetup**