NAME

cryptsetup-resize − resize an active mapping

SYNOPSIS

cryptsetup *resize* **[<options>] <name>**

DESCRIPTION

Resizes an active mapping <name>.

If −−size (in 512−byte sectors) or −−device−size is not specified, the size is computed from the underlying device. For LUKS, it is the size of the underlying device without the area reserved for the LUKS header (see data payload offset in the **luksDump** command). For a plain crypt device, the whole device size is used.

Note that this does not change the raw device geometry; it just changes how many sectors of the raw device are represented in the mapped device.

If cryptsetup detected a volume key for the active device loaded in the kernel keyring service, the resize action would first try to retrieve the key using a token. Only if it failed, it'd ask for a passphrase to unlock a keyslot (LUKS) or to derive a volume key again (plain mode). The kernel keyring is used by default for LUKS2 devices.

<options> can be [−−size, −−device−size, −−token−id, −−token−only, −−token−type, −−key−slot, −−key−file, −−keyfile−size, −−keyfile−offset, −−timeout, −−disable−external−tokens, −−disable−locks, −−disable−keyring, −−volume−key−keyring, −−verify−passphrase, −−timeout, −−external−tokens−path].

OPTIONS

−−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.

−−device−size *size[units]*

Sets the new size of the device. If unset, the real device size is used.

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).

−−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).

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

For commands that change the LUKS header (e.g., *luksAddKey*), specify the device or file with the LUKS header directly as the LUKS device.

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

−−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.

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−slot, **−S** *<0−N>*
For LUKS operations that add key material, this option allows you to specify which keyslot is selected for the new key.

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.

−−size, **−b** *<number of 512 byte sectors>*
Set the size of the device in sectors of 512 bytes.

−−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.

−−usage
Show short option help.

−−verify−passphrase, **−y**
When interactively asking for a passphrase, ask for it twice and complain if both inputs do not match. Ignored on input from file or stdin.

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

−−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/>.