

NAME

cryptsetup-benchmark – benchmarks ciphers and KDF

SYNOPSIS

cryptsetup benchmark [**<options>**]

DESCRIPTION

Benchmarks, ciphers and KDF (key derivation function). Without parameters, it tries to measure a few common configurations.

To benchmark other ciphers or modes, specify **--cipher** and **--key-size** options.

To benchmark PBKDF you need to specify **--pbkdf** or **--hash** with optional cost parameters **--iter-time**, **--pbkdf-memory** or **--pbkdf-parallel**.

This benchmark uses memory only and is only informative. You cannot directly predict real storage encryption speed from it.

For testing block ciphers, this benchmark requires the kernel userspace crypto API to be available. If you are configuring the kernel yourself, enable "User-space interface for symmetric key cipher algorithms" in "Cryptographic API" section (CRYPTO_USER_API_SKCIPHER .config option).

<options> can be [**--cipher**, **--key-size**, **--hash**, **--pbkdf**, **--iter-time**, **--pbkdf-memory**, **--pbkdf-parallel**].

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.

--cipher, -c <cipher-spec>

Set the cipher specification string.

--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.

--hash, -h <hash-spec>

The specified hash is used for PBKDF2 and the AF splitter.

--help, -?

Show help text and default parameters.

--iter-time, -i <number of milliseconds>

The number of milliseconds to spend with PBKDF passphrase processing. Specifying 0 as a parameter selects the compiled-in default.

--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 *open* *--type plain* or *luksFormat*. All other LUKS actions will use the key size specified in the LUKS header. Use *cryptsetup --help* to show the compiled-in defaults.

--pbkdf <PBKDF spec>

Set Password–Based Key Derivation Function (PBKDF) algorithm for LUKS keyslot. The PBKDF can be: *pbkdf2* (for PBKDF2 according to RFC2898), *argon2i* for Argon2i or *argon2id* for Argon2id (see Argon2 <<https://www.cryptolux.org/index.php/Argon2>> for more info).

For LUKS1, only PBKDF2 is accepted (no need to use this option). The default PBKDF for LUKS2 is set during compilation time and is available in the *cryptsetup --help* output.

A PBKDF is used for increasing the dictionary and brute–force attack cost for keyslot passwords. The parameters can be time, memory and parallel cost.

For PBKDF2, only the time cost (number of iterations) applies. For Argon2i/id, there is also memory cost (memory required during the process of key derivation) and parallel cost (number of threads that run in parallel during the key derivation).

Note that increasing memory cost also increases time, so the final parameter values are measured by a benchmark. The benchmark tries to find iteration time (*--iter-time*) with required memory cost *--pbkdf-memory*. If it is not possible, the memory cost is decreased as well. The parallel cost *--pbkdf-parallel* is constant and is checked against available CPU cores.

You can see all PBKDF parameters for a particular LUKS2 keyslot with the **cryptsetup-luksDump**(8) command.

If you do not want to use benchmark and want to specify all parameters directly, use *--pbkdf-force-iterations* with *--pbkdf-memory* and *--pbkdf-parallel*. This will override the values without benchmarking. Note it can cause extremely long unlocking time or cause out–of–memory conditions with unconditional process termination. Use only in specific cases, for example, if you know that the formatted device will be used on some small embedded system.

MINIMAL AND MAXIMAL PBKDF COSTS: For **PBKDF2**, the minimum iteration count is 1000 and the maximum is 4294967295 (maximum for 32–bit unsigned integer). Memory and parallel costs are not supported for PBKDF2. For **Argon2i** and **Argon2id**, the minimum iteration count (CPU cost) is 4, and the maximum is 4294967295 (maximum for a 32–bit unsigned integer). Minimum memory cost is 32 KiB and maximum is 4 GiB. If the memory cost parameter is benchmarked (not specified by a parameter), it is always in the range from 64 MiB to 1 GiB. Memory cost above 1GiB (up to the 4GiB maximum) can be setup only by the *--pbkdf-memory* parameter. The parallel cost minimum is 1 and maximum 4 (if enough CPU cores are available, otherwise it is decreased by the available CPU cores).

WARNING: Increasing PBKDF computational costs above the mentioned limits provides negligible additional security improvement. While elevated costs significantly increase brute–force overhead, they offer negligible protection against dictionary attacks. The marginal cost increase for processing an entire dictionary remains fundamentally insufficient.

The hardcoded PBKDF limits represent engineered trade–offs between cryptographic security and operational usability. LUKS maintains portability and must be used within a reasonable time on resource–constrained systems.

Cryptsetup deliberately restricts maximum memory cost (4 GiB) and parallel cost (4) parameters due to architectural limitations (like embedded and legacy systems).

PBKDF memory cost mandates actual physical RAM allocation with intensive write operations that must remain in physical RAM. Any swap usage results in unacceptable performance degradation. Memory management often overcommits allocations beyond available physical memory, expecting most allocated memory to remain unused. In such situations, as PBKDF always uses all allocated memory, it frequently causes out-of-memory failures that abort cryptsetup operations.

--pbkdf-memory *number*

Set the memory cost for PBKDF (for Argon2i/id, the number represents kilobytes). Note that it is the 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 the maximal value; it is decreased automatically if the CPU online count is lower. This option is not available for PBKDF2.

--usage

Show short option help.

--version, -V

Show the program version.

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/>>.