



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'openssl-genpkey.1ssl'

\$ man openssl-genpkey.1ssl

OPENSSL-GENPKEY(1SSL) OpenSSL OPENSSL-GENPKEY(1SSL)

NAME

openssl-genpkey - generate a private key

SYNOPSIS

openssl genpkey [-help] [-out filename] [-outform DER|PEM] [-quiet] [-pass arg] [-cipher]
[-paramfile file] [-algorithm alg] [-pkeyopt opt:value] [-genparam] [-text] [-engine id]
[-provider name] [-provider-path path] [-propquery propq] [-config configfile]

DESCRIPTION

This command generates a private key.

OPTIONS

-help

Print out a usage message.

-out filename

Output the key to the specified file. If this argument is not specified then standard output is used.

-outform DER|PEM

The output format, except when -genparam is given; the default is PEM. See openssl-format-options(1) for details.

When -genparam is given, -outform is ignored.

-quiet

Do not output "status dots" while generating keys.

-pass arg

The output file password source. For more information about the format of arg see

openssl-passphrase-options(1).

-cipher

This option encrypts the private key with the supplied cipher. Any algorithm name accepted by `EVP_get_cipherbyname()` is acceptable such as `des3`.

-algorithm alg

Public key algorithm to use such as RSA, DSA, DH or DHX. If used this option must precede any `-pkeyopt` options. The options `-paramfile` and `-algorithm` are mutually exclusive. Engines may add algorithms in addition to the standard built-in ones.

Valid built-in algorithm names for private key generation are RSA, RSA-PSS, EC, X25519, X448, ED25519 and ED448.

Valid built-in algorithm names for parameter generation (see the `-genparam` option) are DH, DSA and EC.

Note that the algorithm name X9.42 DH may be used as a synonym for DHX keys and PKCS#3 refers to DH Keys. Some options are not shared between DH and DHX keys.

-pkeyopt opt:value

Set the public key algorithm option `opt` to `value`. The precise set of options supported depends on the public key algorithm used and its implementation. See "KEY GENERATION OPTIONS" and "PARAMETER GENERATION OPTIONS" below for more details.

-genparam

Generate a set of parameters instead of a private key. If used this option must precede any `-algorithm`, `-paramfile` or `-pkeyopt` options.

-paramfile filename

Some public key algorithms generate a private key based on a set of parameters. They can be supplied using this option. If this option is used the public key algorithm used is determined by the parameters. If used this option must precede any `-pkeyopt` options. The options `-paramfile` and `-algorithm` are mutually exclusive.

-text

Print an (unencrypted) text representation of private and public keys and parameters along with the PEM or DER structure.

-engine id

See "Engine Options" in `openssl(1)`. This option is deprecated.

-provider name

-provider-path path

-propquery propq

See "Provider Options" in openssl(1), provider(7), and property(7).

-config configfile

See "Configuration Option" in openssl(1).

KEY GENERATION OPTIONS

The options supported by each algorithm and indeed each implementation of an algorithm can vary. The options for the OpenSSL implementations are detailed below. There are no key generation options defined for the X25519, X448, ED25519 or ED448 algorithms.

RSA Key Generation Options

rsa_keygen_bits:numbits

The number of bits in the generated key. If not specified 2048 is used.

rsa_keygen_primes:numprimes

The number of primes in the generated key. If not specified 2 is used.

rsa_keygen_pubexp:value

The RSA public exponent value. This can be a large decimal or hexadecimal value if preceded by "0x". Default value is 65537.

RSA-PSS Key Generation Options

Note: by default an RSA-PSS key has no parameter restrictions.

rsa_keygen_bits:numbits, rsa_keygen_primes:numprimes, rsa_keygen_pubexp:value

These options have the same meaning as the RSA algorithm.

rsa_pss_keygen_md:digest

If set the key is restricted and can only use digest for signing.

rsa_pss_keygen_mgf1_md:digest

If set the key is restricted and can only use digest as it's MGF1 parameter.

rsa_pss_keygen_saltlen:len

If set the key is restricted and len specifies the minimum salt length.

EC Key Generation Options

The EC key generation options can also be used for parameter generation.

ec_paramgen_curve:curve

The EC curve to use. OpenSSL supports NIST curve names such as "P-256".

ec_param_enc:encoding

The encoding to use for parameters. The encoding parameter must be either named_curve or explicit. The default value is named_curve.

DH Key Generation Options

group:name

The paramfile option is not required if a named group is used here. See the "DH Parameter Generation Options" section below.

PARAMETER GENERATION OPTIONS

The options supported by each algorithm and indeed each implementation of an algorithm can vary. The options for the OpenSSL implementations are detailed below.

DSA Parameter Generation Options

dsa_paramgen_bits:numbits

The number of bits in the generated prime. If not specified 2048 is used.

dsa_paramgen_q_bits:numbits

qbits:numbits

The number of bits in the q parameter. Must be one of 160, 224 or 256. If not specified 224 is used.

dsa_paramgen_md:digest

digest:digest

The digest to use during parameter generation. Must be one of sha1, sha224 or sha256.

If set, then the number of bits in q will match the output size of the specified digest and the dsa_paramgen_q_bits parameter will be ignored. If not set, then a digest will be used that gives an output matching the number of bits in q, i.e. sha1 if q length is 160, sha224 if it 224 or sha256 if it is 256.

properties:query

The digest property query string to use when fetching a digest from a provider.

type:type

The type of generation to use. Set this to 1 to use legacy FIPS186-2 parameter generation. The default of 0 uses FIPS186-4 parameter generation.

gindex:index

The index to use for canonical generation and verification of the generator g. Set this to a positive value ranging from 0..255 to use this mode. Larger values will only use the bottom byte. This index must then be reused during key validation to verify the value of g. If this value is not set then g is not verifiable. The default value is -1.

hexseed:seed

The seed data to use instead of generating a random seed internally. This should be used for testing purposes only. This will either produced fixed values for the generated parameters OR it will fail if the seed did not generate valid primes.

DH Parameter Generation Options

For most use cases it is recommended to use the group option rather than the type options.

Note that the group option is not used by default if no parameter generation options are specified.

group:name

dh_param:name

Use a named DH group to select constant values for the DH parameters. All other options will be ignored if this value is set.

Valid values that are associated with the algorithm of "DH" are: "ffdhe2048", "ffdhe3072", "ffdhe4096", "ffdhe6144", "ffdhe8192", "modp_1536", "modp_2048", "modp_3072", "modp_4096", "modp_6144", "modp_8192".

Valid values that are associated with the algorithm of "DHX" are the RFC5114 names "dh_1024_160", "dh_2048_224", "dh_2048_256".

dh_rfc5114:num

If this option is set, then the appropriate RFC5114 parameters are used instead of generating new parameters. The value num can be one of 1, 2 or 3 that are equivalent to using the option group with one of "dh_1024_160", "dh_2048_224" or "dh_2048_256".

All other options will be ignored if this value is set.

pbits:numbits

dh_paramgen_prime_len:numbits

The number of bits in the prime parameter p. The default is 2048.

qbits:numbits

dh_paramgen_subprime_len:numbits

The number of bits in the sub prime parameter q. The default is 224. Only relevant if used in conjunction with the dh_paramgen_type option to generate DHX parameters.

safeprime-generator:value

dh_paramgen_generator:value

The value to use for the generator g. The default is 2. The algorithm option must be "DH" for this parameter to be used.

type:string

The type name of DH parameters to generate. Valid values are:

"generator"

Use a safe prime generator with the option `safeprime_generator`. The algorithm option must be "DH".

"fips186_4"

FIPS186-4 parameter generation. The algorithm option must be "DHX".

"fips186_2"

FIPS186-4 parameter generation. The algorithm option must be "DHX".

"group"

Can be used with the option `pbits` to select one of "ffdhe2048", "ffdhe3072", "ffdhe4096", "ffdhe6144" or "ffdhe8192". The algorithm option must be "DH".

"default"

Selects a default type based on the algorithm. This is used by the OpenSSL default provider to set the type for backwards compatibility. If algorithm is "DH" then "generator" is used. If algorithm is "DHX" then "fips186_2" is used.

`dh_paramgen_type:value`

The type of DH parameters to generate. Valid values are 0, 1, 2 or 3 which correspond to setting the option `type` to "generator", "fips186_2", "fips186_4" or "group".

`digest:digest`

The digest to use during parameter generation. Must be one of sha1, sha224 or sha256. If set, then the number of bits in `qbits` will match the output size of the specified digest and the `qbits` parameter will be ignored. If not set, then a digest will be used that gives an output matching the number of bits in `q`, i.e. sha1 if `q` length is 160, sha224 if it is 224 or sha256 if it is 256. This is only used by "fips186_4" and "fips186_2" key generation.

`properties:query`

The digest property query string to use when fetching a digest from a provider. This is only used by "fips186_4" and "fips186_2" key generation.

`gindex:index`

The index to use for canonical generation and verification of the generator `g`. Set this to a positive value ranging from 0..255 to use this mode. Larger values will only use the bottom byte. This index must then be reused during key validation to verify the value of `g`. If this value is not set then `g` is not verifiable. The default value

is -1. This is only used by "fips186_4" and "fips186_2" key generation.

hexseed:seed

The seed seed data to use instead of generating a random seed internally. This should be used for testing purposes only. This will either produced fixed values for the generated parameters OR it will fail if the seed did not generate valid primes. This is only used by "fips186_4" and "fips186_2" key generation.

EC Parameter Generation Options

The EC parameter generation options are the same as for key generation. See "EC Key Generation Options" above.

NOTES

The use of the genpkey program is encouraged over the algorithm specific utilities because additional algorithm options and ENGINE provided algorithms can be used.

EXAMPLES

Generate an RSA private key using default parameters:

```
openssl genpkey -algorithm RSA -out key.pem
```

Encrypt output private key using 128 bit AES and the passphrase "hello":

```
openssl genpkey -algorithm RSA -out key.pem -aes-128-cbc -pass pass:hello
```

Generate a 2048 bit RSA key using 3 as the public exponent:

```
openssl genpkey -algorithm RSA -out key.pem \  
-pkeyopt rsa_keygen_bits:2048 -pkeyopt rsa_keygen_pubexp:3
```

Generate 2048 bit DSA parameters that can be validated: The output values for gindex and seed are required for key validation purposes and are not saved to the output pem file).

```
openssl genpkey -genparam -algorithm DSA -out dsap.pem -pkeyopt pbits:2048 \  
-pkeyopt qbits:224 -pkeyopt digest:SHA256 -pkeyopt gindex:1 -text
```

Generate DSA key from parameters:

```
openssl genpkey -paramfile dsap.pem -out dsakey.pem
```

Generate 4096 bit DH Key using safe prime group ffdhe4096:

```
openssl genpkey -algorithm DH -out dhkey.pem -pkeyopt group:ffdhe4096
```

Generate 2048 bit X9.42 DH key with 256 bit subgroup using RFC5114 group3:

```
openssl genpkey -algorithm DHX -out dhkey.pem -pkeyopt dh_rfc5114:3
```

Generate a DH key using a DH parameters file:

```
openssl genpkey -paramfile dhp.pem -out dhkey.pem
```

Output DH parameters for safe prime group ffdhe2048:

```
openssl genpkey -genparam -algorithm DH -out dhp.pem -pkeyopt group:ffdhe2048
```

Output 2048 bit X9.42 DH parameters with 224 bit subgroup using RFC5114 group2:

```
openssl genpkey -genparam -algorithm DHX -out dhp.pem -pkeyopt dh_rfc5114:2
```

Output 2048 bit X9.42 DH parameters with 224 bit subgroup using FIP186-4 keygen:

```
openssl genpkey -genparam -algorithm DHX -out dhp.pem -text \  
-pkeyopt pbits:2048 -pkeyopt qbits:224 -pkeyopt digest:SHA256 \  
-pkeyopt gindex:1 -pkeyopt dh_paramgen_type:2
```

Output 1024 bit X9.42 DH parameters with 160 bit subgroup using FIP186-2 keygen:

```
openssl genpkey -genparam -algorithm DHX -out dhp.pem -text \  
-pkeyopt pbits:1024 -pkeyopt qbits:160 -pkeyopt digest:SHA1 \  
-pkeyopt gindex:1 -pkeyopt dh_paramgen_type:1
```

Output 2048 bit DH parameters:

```
openssl genpkey -genparam -algorithm DH -out dhp.pem \  
-pkeyopt dh_paramgen_prime_len:2048
```

Output 2048 bit DH parameters using a generator:

```
openssl genpkey -genparam -algorithm DH -out dhp.pem \  
-pkeyopt dh_paramgen_prime_len:2048 \  
-pkeyopt dh_paramgen_type:1
```

Generate EC parameters:

```
openssl genpkey -genparam -algorithm EC -out ecp.pem \  
-pkeyopt ec_paramgen_curve:secp384r1 \  
-pkeyopt ec_param_enc:named_curve
```

Generate EC key from parameters:

```
openssl genpkey -paramfile ecp.pem -out eckey.pem
```

Generate EC key directly:

```
openssl genpkey -algorithm EC -out eckey.pem \  
-pkeyopt ec_paramgen_curve:P-384 \  
-pkeyopt ec_param_enc:named_curve
```

Generate an X25519 private key:

```
openssl genpkey -algorithm X25519 -out xkey.pem
```

Generate an ED448 private key:

```
openssl genpkey -algorithm ED448 -out xkey.pem
```

