NAME

openssl−genpkey, genpkey − generate a private key

SYNOPSIS

openssl genpkey [**−help**] [**−out filename**] [**−outform PEM|DER**] [**−pass arg**] [***−cipher***] [**−engine id**] [**−paramfile file**] [**−algorithm alg**] [**−pkeyopt opt:value**] [**−genparam**] [**−text**]

DESCRIPTION

The **genpkey** 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

This specifies the output format DER or PEM. The default format is PEM.

−pass arg

The output file password source. For more information about the format of **arg** see the **PASS PHRASE ARGUMENTS** section in **openssl** (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**.

−engine id

Specifying an engine (by its unique **id** string) will cause **genpkey** to attempt to obtain a functional reference to the specified engine, thus initialising it if needed. The engine will then be set as the default for all available algorithms. If used this option should precede all other options.

−algorithm alg

Public key algorithm to use such as RSA, DSA or DH. 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 the DH algorithm. These are identical and do not indicate the type of parameters that will be generated. Use the **dh_paramgen_type** option to indicate whether PKCS#3 or X9.42 DH parameters are required. See ''DH Parameter Generation Options'' below for more details.

−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.

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''.

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

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

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.

DH Parameter Generation Options

dh_paramgen_prime_len:numbits
The number of bits in the prime parameter **p**. The default is 2048.

dh_paramgen_subprime_len:numbits
The number of bits in the sub prime parameter **q**. The default is 256 if the prime is at least 2048 bits long or 160 otherwise. Only relevant if used in conjunction with the **dh_paramgen_type** option to generate X9.42 DH parameters.

dh_paramgen_generator:value
The value to use for the generator **g**. The default is 2.

dh_paramgen_type:value
The type of DH parameters to generate. Use 0 for PKCS#3 DH and 1 for X9.42 DH. The default is 0.

dh_rfc5114:num
If this option is set, then the appropriate RFC5114 parameters are used instead of generating new parameters. The value **num** can take the values 1, 2 or 3 corresponding to RFC5114 DH parameters consisting of 1024 bit group with 160 bit subgroup, 2048 bit group with 224 bit subgroup and 2048 bit group with 256 bit subgroup as mentioned in RFC5114 sections 2.1, 2.2 and 2.3 respectively. If present this overrides all other DH parameter options.

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:

```
openssl genpkey -genparam -algorithm DSA -out dsap.pem \
    -pkeyopt dsa_paramgen_bits:2048
```

Generate DSA key from parameters:

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

Generate 2048 bit DH parameters:

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

Generate 2048 bit X9.42 DH parameters:

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

Output RFC5114 2048 bit DH parameters with 224 bit subgroup:

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

Generate DH key from parameters:

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

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
```

HISTORY

The ability to use NIST curve names, and to generate an EC key directly, were added in OpenSSL 1.0.2. The ability to generate X25519 keys was added in OpenSSL 1.1.0. The ability to generate X448, ED25519 and ED448 keys was added in OpenSSL 1.1.1.

COPYRIGHT