NAME

genpkey - generate a private key

SYNOPSIS

openssl genpkey [**-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

-out filename

the output filename. If this argument is not specified then standard output is used.

-outform DER|PEM

This specifies the output format DER or 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.

-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** below for more details.

-genparam

generate a set of parameters instead of a private key. If used this option must precede and **-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 and **-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 algorith and indeed each implementation of an algorithm can vary. The options for the OpenSSL implementations are detailed below.

RSA KEY GENERATION OPTIONS

rsa_keygen_bits:numbits

The number of bits in the generated key. If not specified 1024 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.

DSA PARAMETER GENERATION OPTIONS

dsa_paramgen_bits:numbits
The number of bits in the generated parameters. If not specified 1024 is used.

DH PARAMETER GENERATION OPTIONS

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

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

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.

EC PARAMETER GENERATION OPTIONS

ec_paramgen_curve:curve
the EC curve to use.

GOST2001 KEY GENERATION AND PARAMETER OPTIONS

Gost 2001 support is not enabled by default. To enable this algorithm, one should load the ccgost engine in the OpenSSL configuration file. See README.gost file in the engines/ccgost directiry of the source distribution for more details.

Use of a parameter file for the GOST R 34.10 algorithm is optional. Parameters can be specified during key generation directly as well as during generation of parameter file.

paramset:name
Specifies GOST R 34.10-2001 parameter set according to RFC 4357. Parameter set can be specified using abbreviated name, object short name or numeric OID. Following parameter sets are supported:

```
paramset OID Usage
A 1.2.643.2.2.35.1 Signature
B 1.2.643.2.2.35.2 Signature
C 1.2.643.2.2.35.3 Signature
XA 1.2.643.2.2.36.0 Key exchange
XB 1.2.643.2.2.36.1 Key exchange
test 1.2.643.2.2.35.0 Test purposes
```

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 1024 bit DSA parameters:

```
openssl genpkey -genparam -algorithm DSA -out dsap.pem \
-pkeyopt dsa_paramgen_bits:1024
```

Generate DSA key from parameters:

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

Generate 1024 bit DH parameters:

```
openssl genpkey -genparam -algorithm DH -out dhp.pem \
-pkeyopt dh_paramgen_prime_len:1024
```

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
```