## NAME

openssl−ec, ec − EC key processing

## SYNOPSIS

**openssl ec** [**−help**] [**−inform PEM|DER**] [**−outform PEM|DER**] [**−in filename**] [**−passin arg**] [**−out filename**] [**−passout arg**] [**−des**] [**−des3**] [**−idea**] [**−text**] [**−noout**] [**−param_out**] [**−pubin**] [**−pubout**] [**−conv_form arg**] [**−param_enc arg**] [**−no_public**] [**−check**] [**−engine id**]

## DESCRIPTION

The **ec** command processes EC keys. They can be converted between various forms and their components printed out. **Note** OpenSSL uses the private key format specified in 'SEC 1: Elliptic Curve Cryptography' (http://www.secg.org/). To convert an OpenSSL EC private key into the PKCS#8 private key format use the **pkcs8** command.

## OPTIONS

**−help**

Print out a usage message.

**−inform DER|PEM**

This specifies the input format. The **DER** option with a private key uses an ASN.1 DER encoded SEC1 private key. When used with a public key it uses the SubjectPublicKeyInfo structure as specified in RFC 3280. The **PEM** form is the default format: it consists of the **DER** format base64 encoded with additional header and footer lines. In the case of a private key PKCS#8 format is also accepted.

**−outform DER|PEM**

This specifies the output format, the options have the same meaning and default as the **−inform** option.

**−in filename**

This specifies the input filename to read a key from or standard input if this option is not specified. If the key is encrypted a pass phrase will be prompted for.

**−passin arg**

The input file password source. For more information about the format of **arg** see the **PASS PHRASE ARGUMENTS** section in **openssl** (1).

**−out filename**

This specifies the output filename to write a key to or standard output by is not specified. If any encryption options are set then a pass phrase will be prompted for. The output filename should **not** be the same as the input filename.

**−passout arg**

The output file password source. For more information about the format of **arg** see the **PASS PHRASE ARGUMENTS** section in **openssl** (1).

**−des|−des3|−idea**

These options encrypt the private key with the DES, triple DES, IDEA or any other cipher supported by OpenSSL before outputting it. A pass phrase is prompted for. If none of these options is specified the key is written in plain text. This means that using the **ec** utility to read in an encrypted key with no encryption option can be used to remove the pass phrase from a key, or by setting the encryption options it can be use to add or change the pass phrase. These options can only be used with PEM format output files.

**−text**

Prints out the public, private key components and parameters.

**−noout**

This option prevents output of the encoded version of the key.

**−pubin**

By default, a private key is read from the input file. With this option a public key is read instead.

**–pubout**

> By default a private key is output. With this option a public key will be output instead. This option is automatically set if the input is a public key.

**–conv_form**

> This specifies how the points on the elliptic curve are converted into octet strings. Possible values are: **compressed** (the default value), **uncompressed** and **hybrid**. For more information regarding the point conversion forms please read the X9.62 standard. **Note** Due to patent issues the **compressed** option is disabled by default for binary curves and can be enabled by defining the preprocessor macro **OPENSSL_EC_BIN_PT_COMP** at compile time.

**–param_enc arg**

> This specifies how the elliptic curve parameters are encoded. Possible value are: **named_curve**, i.e. the ec parameters are specified by an OID, or **explicit** where the ec parameters are explicitly given (see RFC 3279 for the definition of the EC parameters structures). The default value is **named_curve**. **Note** the **implicitlyCA** alternative, as specified in RFC 3279, is currently not implemented in OpenSSL.

**–no_public**

> This option omits the public key components from the private key output.

**–check**

> This option checks the consistency of an EC private or public key.

**–engine id**

> Specifying an engine (by its unique **id** string) will cause **ec** 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.

## NOTES

The PEM private key format uses the header and footer lines:

```
-----BEGIN EC PRIVATE KEY-----
-----END EC PRIVATE KEY-----
```

The PEM public key format uses the header and footer lines:

```
-----BEGIN PUBLIC KEY-----
-----END PUBLIC KEY-----
```

## EXAMPLES

To encrypt a private key using triple DES:

```
openssl ec -in key.pem -des3 -out keyout.pem
```

To convert a private key from PEM to DER format:

```
openssl ec -in key.pem -outform DER -out keyout.der
```

To print out the components of a private key to standard output:

```
openssl ec -in key.pem -text -noout
```

To just output the public part of a private key:

```
openssl ec -in key.pem -pubout -out pubkey.pem
```

To change the parameters encoding to **explicit**:

```
openssl ec -in key.pem -param_enc explicit -out keyout.pem
```

To change the point conversion form to **compressed**:

```
openssl ec -in key.pem -conv_form compressed -out keyout.pem
```

## SEE ALSO

**ecparam** (1), **dsa** (1), **rsa** (1)

**COPYRIGHT**