NAME

d2i_ECPKParameters, i2d_ECPKParameters, d2i_ECPKParameters_bio, i2d_ECPKParameters_bio, d2i_ECPKParameters_fp, i2d_ECPKParameters_fp, ECPKParameters_print, ECPKParameters_print_fp - Functions for decoding and encoding ASN1 representations of elliptic curve entities

SYNOPSIS

```
#include <openssl/ec.h>

EC_GROUP *d2i_ECPKParameters(EC_GROUP **px, const unsigned char **in, long len);
int i2d_ECPKParameters(const EC_GROUP *x, unsigned char **out);
#define d2i_ECPKParameters_bio(bp,x) ASN1_d2i_bio_of(EC_GROUP,NULL,d2i_ECPKParameters,bp,x)
#define i2d_ECPKParameters_bio(bp,x) ASN1_i2d_bio_of_const(EC_GROUP,i2d_ECPKParameters,bp,x)
#define d2i_ECPKParameters_fp(fp,x) (EC_GROUP *)ASN1_d2i_fp(NULL, \
(char *(*)())d2i_ECPKParameters,(fp),(unsigned char **)(x))
#define i2d_ECPKParameters_fp(fp,x) ASN1_i2d_fp(i2d_ECPKParameters,(fp), \
(unsigned char *)(x))
int ECPKParameters_print(BIO *bp, const EC_GROUP *x, int off);
int ECPKParameters_print_fp(FILE *fp, const EC_GROUP *x, int off);
```

DESCRIPTION

The ECPKParameters encode and decode routines encode and parse the public parameters for an **EC_GROUP** structure, which represents a curve.

d2i_ECPKParameters() attempts to decode **len** bytes at ***in**. If successful a pointer to the **EC_GROUP** structure is returned. If an error occurred then **NULL** is returned. If **px** is not **NULL** then the returned structure is written to ***px**. If ***px** is not **NULL** then it is assumed that ***px** contains a valid **EC_GROUP** structure and an attempt is made to reuse it. If the call is successful ***in** is incremented to the byte following the parsed data.

i2d_ECPKParameters() encodes the structure pointed to by **x** into DER format. If **out** is not **NULL** is writes the DER encoded data to the buffer at ***out**, and increments it to point after the data just written. If the return value is negative an error occurred, otherwise it returns the length of the encoded data.

If ***out** is **NULL** memory will be allocated for a buffer and the encoded data written to it. In this case ***out** is not incremented and it points to the start of the data just written.

d2i_ECPKParameters_bio() is similar to *d2i_ECPKParameters()* except it attempts to parse data from BIO **bp**.

d2i_ECPKParameters_fp() is similar to *d2i_ECPKParameters()* except it attempts to parse data from FILE pointer **fp**.

i2d_ECPKParameters_bio() is similar to *i2d_ECPKParameters()* except it writes the encoding of the structure **x** to BIO **bp** and it returns 1 for success and 0 for failure.

i2d_ECPKParameters_fp() is similar to *i2d_ECPKParameters()* except it writes the encoding of the structure **x** to BIO **bp** and it returns 1 for success and 0 for failure.

These functions are very similar to the X509 functions described in *d2i_X509(3)*, where further notes and examples are available.

The ECPKParameters_print and ECPKParameters_print_fp functions print a human-readable output of the public parameters of the EC_GROUP to **bp** or **fp**. The output lines are indented by **off** spaces.

RETURN VALUES

d2i_ECPKParameters(), *d2i_ECPKParameters_bio()* and *d2i_ECPKParameters_fp()* return a valid **EC_GROUP** structure or **NULL** if an error occurs.

i2d_ECPKParameters() returns the number of bytes successfully encoded or a negative value if an error occurs.

i2d_ECPKParameters_bio(), *i2d_ECPKParameters_fp()*, ECPKParameters_print and ECPKParameters_print_fp return 1 for success and 0 if an error occurs.

SEE ALSO

crypto(3), *ec(3)*, *EC_GROUP_new(3)*, *EC_GROUP_copy(3)*, *EC_POINT_new(3)*, *EC_POINT_add(3)*, *EC_KEY_new(3)*, *EC_GFp_simple_method(3)*, *d2i_X509(3)*