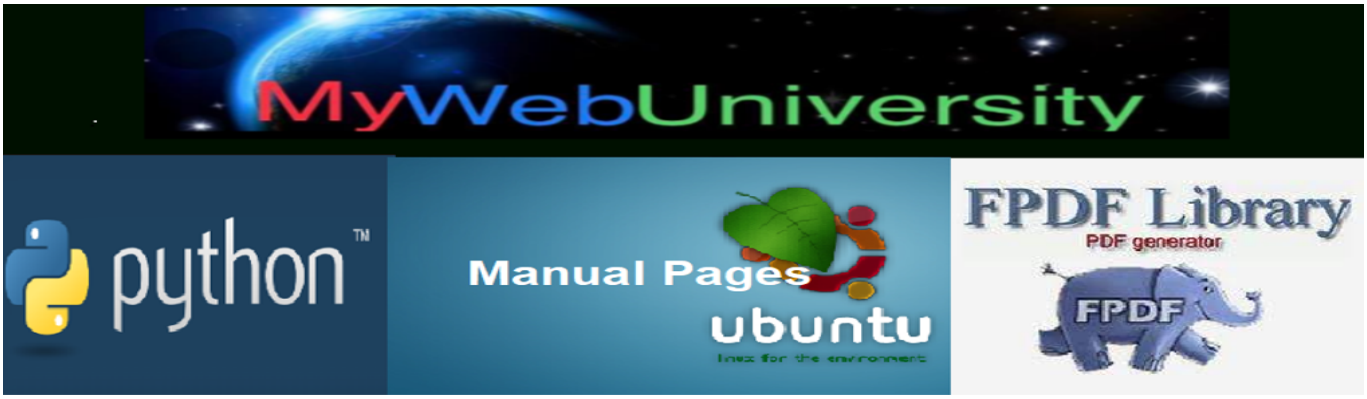

Rocky Enterprise Linux 9.2 Manual Pages on command 'EVP_KEYMGMT-EC.7ssl'

$ man EVP_KEYMGMT-EC.7ssl

EVP_PKEY-EC(7SSL) OpenSSL EVP_PKEY-EC(7SSL)

NAME

EVP_PKEY-EC, EVP_KEYMGMT-EC - EVP_PKEY EC keytype and algorithm support

DESCRIPTION

The EC keytype is implemented in OpenSSL's default provider.

Common EC parameters

The normal way of specifying domain parameters for an EC curve is via the curve name "group". For curves with no curve name, explicit parameters can be used that specify "field-type", "p", "a", "b", "generator" and "order". Explicit parameters are supported for backwards compability reasons, but they are not compliant with multiple standards (including RFC5915) which only allow named curves.

The following KeyGen/Gettable/Import/Export types are available for the built-in EC algorithm:

"group" (OSSL_PKEY_PARAM_GROUP_NAME) <UTF8 string>

The curve name.

"field-type" (OSSL_PKEY_PARAM_EC_FIELD_TYPE) <UTF8 string>

The value should be either "prime-field" or "characteristic-two-field", which correspond to prime field Fp and binary field F2^m.

"p" (OSSL_PKEY_PARAM_EC_P) <unsigned integer>

For a curve over Fp p is the prime for the field. For a curve over F2^m p represents the irreducible polynomial - each bit represents a term in the polynomial. Therefore, there will either be three or five bits set dependent on whether the polynomial is a trinomial or a pentanomial.

"a" (OSSL_PKEY_PARAM_EC_A) <unsigned integer>

"b" (OSSL_PKEY_PARAM_EC_B) <unsigned integer>

"seed" (OSSL_PKEY_PARAM_EC_SEED) <octet string>

a and b represents the coefficients of the curve For Fp: y^2 mod p = x^3 +ax + b mod p OR For F2^m: y^2 + xy = x^3 + ax^2 + b

seed is an optional value that is for information purposes only. It represents the random number seed used to generate the coefficient b from a random number.

"generator" (OSSL_PKEY_PARAM_EC_GENERATOR) <octet string>

"order" (OSSL_PKEY_PARAM_EC_ORDER) <unsigned integer>

"cofactor" (OSSL_PKEY_PARAM_EC_COFACTOR) <unsigned integer>

The generator is a well defined point on the curve chosen for cryptographic operations. The encoding conforms with Sec. 2.3.3 of the SECG SEC 1 ("Elliptic Curve Cryptography") standard. See EC_POINT_oct2point(). Integers used for point multiplications will be between 0 and order - 1. cofactor is an optional value.

order multiplied by the cofactor gives the number of points on the curve.

"decoded-from-explicit" (OSSL_PKEY_PARAM_EC_DECODED_FROM_EXPLICIT_PARAMS) <integer>

Gets a flag indicating wether the key or parameters were decoded from explicit curve parameters. Set to 1 if so or 0 if a named curve was used.

"use-cofactor-flag" (OSSL_PKEY_PARAM_USE_COFACTOR_ECDH) <integer>

Enable Cofactor DH (ECC CDH) if this value is 1, otherwise it uses normal EC DH if the value is zero. The cofactor variant multiplies the shared secret by the EC curve's cofactor (note for some curves the cofactor is 1).

"encoding" (OSSL_PKEY_PARAM_EC_ENCODING) <UTF8 string>

Set the format used for serializing the EC group parameters. Valid values are "explicit" or "named_curve". The default value is "named_curve".

"point-format" (OSSL_PKEY_PARAM_EC_POINT_CONVERSION_FORMAT) <UTF8 string>

Sets or gets the point_conversion_form for the key. For a description of point_conversion_forms please see EC_POINT_new(3). Valid values are "uncompressed" or "compressed". The default value is "uncompressed".

"group-check" (OSSL_PKEY_PARAM_EC_GROUP_CHECK_TYPE) <UTF8 string>

Sets or Gets the type of group check done when EVP_PKEY_param_check() is called. Valid values are "default", "named" and "named-nist". The "named" type checks that the domain parameters match the inbuilt curve parameters, "named-nist" is similiar but

also checks that the named curve is a nist curve. The "default" type does domain parameter validation for the OpenSSL default provider, but is equivalent to "named-nist" for the OpenSSL fips provider.

"include-public" (OSSL_PKEY_PARAM_EC_INCLUDE_PUBLIC) <integer>

Setting this value to 0 indicates that the public key should not be included when encoding the private key. The default value of 1 will include the public key.

See also EVP_KEYEXCH-ECDH(7) for the related OSSL_EXCHANGE_PARAM_EC_ECDH_COFACTOR_MODE parameter that can be set on a per-operation basis.

"pub" (OSSL_PKEY_PARAM_PUB_KEY) <octet string>

The public key value in EC point format.

"priv" (OSSL_PKEY_PARAM_PRIV_KEY) <unsigned integer>

The private key value.

"encoded-pub-key" (OSSL_PKEY_PARAM_ENCODED_PUBLIC_KEY) <octet string>

Used for getting and setting the encoding of an EC public key. The public key is expected to be a point conforming to Sec. 2.3.4 of the SECG SEC 1 ("Elliptic Curve Cryptography") standard.

"qx" (OSSL_PKEY_PARAM_EC_PUB_X) <unsigned integer>

Used for getting the EC public key X component.

"qy" (OSSL_PKEY_PARAM_EC_PUB_Y) <unsigned integer>

Used for getting the EC public key Y component.

(OSSL_PKEY_PARAM_DEFAULT_DIGEST) <UTF8 string>

Getter that returns the default digest name. (Currently returns "SHA256" as of OpenSSL 3.0).

The following Gettable types are also available for the built-in EC algorithm:

"basis-type" (OSSL_PKEY_PARAM_EC_CHAR2_TYPE) <UTF8 string>

Supports the values "tpBasis" for a trinomial or "ppBasis" for a pentanomial. This field is only used for a binary field F2^m.

"m" (OSSL_PKEY_PARAM_EC_CHAR2_M) <integer>

"tp" (OSSL_PKEY_PARAM_EC_CHAR2_TP_BASIS) <integer>

"k1" (OSSL_PKEY_PARAM_EC_CHAR2_PP_K1) <integer>

"k2" (OSSL_PKEY_PARAM_EC_CHAR2_PP_K2) <integer>

"k3" (OSSL_PKEY_PARAM_EC_CHAR2_PP_K3) <integer>

These fields are only used for a binary field F2^m. m is the degree of the binary

field.

tp is the middle bit of a trinomial so its value must be in the range m > tp > 0.

k1, k2 and k3 are used to get the middle bits of a pentanomial such that m > k3 > k2 > k1 > 0

EXAMPLES

An EVP_PKEY context can be obtained by calling:

```
EVP_PKEY_CTX *pctx =
    EVP_PKEY_CTX_new_from_name(NULL, "EC", NULL);
```

An EVP_PKEY ECDSA or ECDH key can be generated with a "P-256" named group by calling:

```
pkey = EVP_EC_gen("P-256");
```

or like this:

```
EVP_PKEY *key = NULL;
OSSL_PARAM params[2];
EVP_PKEY_CTX *gctx =
    EVP_PKEY_CTX_new_from_name(NULL, "EC", NULL);
EVP_PKEY_keygen_init(gctx);
params[0] = OSSL_PARAM_construct_utf8_string(OSSL_PKEY_PARAM_GROUP_NAME,
                                             "P-256", 0);
params[1] = OSSL_PARAM_construct_end();
EVP_PKEY_CTX_set_params(gctx, params);
EVP_PKEY_generate(gctx, &key);
EVP_PKEY_print_private(bio_out, key, 0, NULL);
...
EVP_PKEY_free(key);
EVP_PKEY_CTX_free(gctx);
```

An EVP_PKEY EC CDH (Cofactor Diffie-Hellman) key can be generated with a "K-571" named group by calling:

```
int use_cdh = 1;
EVP_PKEY *key = NULL;
OSSL_PARAM params[3];
EVP_PKEY_CTX *gctx =
    EVP_PKEY_CTX_new_from_name(NULL, "EC", NULL);
EVP_PKEY_keygen_init(gctx);
```

```
params[0] = OSSL_PARAM_construct_utf8_string(OSSL_PKEY_PARAM_GROUP_NAME,
                                             "K-571", 0);
/*
 * This curve has a cofactor that is not 1 - so setting CDH mode changes
 * the behaviour. For many curves the cofactor is 1 - so setting this has
 * no effect.
 */
params[1] = OSSL_PARAM_construct_int(OSSL_PKEY_PARAM_USE_COFACTOR_ECDH,
                                     &use_cdh);
params[2] = OSSL_PARAM_construct_end();
EVP_PKEY_CTX_set_params(gctx, params);
EVP_PKEY_generate(gctx, &key);
EVP_PKEY_print_private(bio_out, key, 0, NULL);
...
EVP_PKEY_free(key);
EVP_PKEY_CTX_free(gctx);
```

SEE ALSO

EVP_EC_gen(3), EVP_KEYMGMT(3), EVP_PKEY(3), provider-keymgmt(7), EVP_SIGNATURE-ECDSA(7), EVP_KEYEXCH-ECDH(7)

COPYRIGHT