Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'EVP_PKEY_derive_set_peer.3ossl' command

$ man EVP_PKEY_derive_set_peer.3ossl

EVP_PKEY_DERIVE(3ossl) OpenSSL EVP_PKEY_DERIVE(3ossl)

NAME

EVP_PKEY_derive_init, EVP_PKEY_derive_init_ex,

EVP_PKEY_derive_set_peer_ex, EVP_PKEY_derive_set_peer, EVP_PKEY_derive

- derive public key algorithm shared secret

SYNOPSIS

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

int EVP_PKEY_derive_init(EVP_PKEY_CTX *ctx);
int EVP_PKEY_derive_init_ex(EVP_PKEY_CTX *ctx, const OSSL_PARAM params[]);
int EVP_PKEY_derive_set_peer_ex(EVP_PKEY_CTX *ctx, EVP_PKEY *peer,
                                int validate_peer);
int EVP_PKEY_derive_set_peer(EVP_PKEY_CTX *ctx, EVP_PKEY *peer);
int EVP_PKEY_derive(EVP_PKEY_CTX *ctx, unsigned char *key, size_t *keylen);
```

DESCRIPTION

EVP_PKEY_derive_init() initializes a public key algorithm context ctx

for shared secret derivation using the algorithm given when the context

was created using EVP_PKEY_CTX_new(3) or variants thereof. The

algorithm is used to fetch a EVP_KEYEXCH method implicitly, see

"Implicit fetch" in provider(7) for more information about implicit

fetches.

EVP_PKEY_derive_init_ex() is the same as EVP_PKEY_derive_init() but additionally sets the passed parameters params on the context before returning.

EVP_PKEY_derive_set_peer_ex() sets the peer key: this will normally be a public key. The validate_peer will validate the public key if this value is non zero.

EVP_PKEY_derive_set_peer() is similiar to EVP_PKEY_derive_set_peer_ex() with validate_peer set to 1.

EVP_PKEY_derive() derives a shared secret using ctx. If key is NULL then the maximum size of the output buffer is written to the keylen parameter. If key is not NULL then before the call the keylen parameter should contain the length of the key buffer, if the call is successful the shared secret is written to key and the amount of data written to keylen.

NOTES

After the call to EVP_PKEY_derive_init(), algorithm specific control operations can be performed to set any appropriate parameters for the operation.

The function EVP_PKEY_derive() can be called more than once on the same context if several operations are performed using the same parameters.

RETURN VALUES

EVP_PKEY_derive_init() and EVP_PKEY_derive() return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

EXAMPLES

Derive shared secret (for example DH or EC keys):

```
#include <openssl/evp.h>
#include <openssl/rsa.h>

EVP_PKEY_CTX *ctx;
ENGINE *eng;
unsigned char *skey;
size_t skeylen;
EVP_PKEY *pkey, *peerkey;
/* NB: assumes pkey, eng, peerkey have been already set up */

ctx = EVP_PKEY_CTX_new(pkey, eng);
if (!ctx)
    /* Error occurred */
if (EVP_PKEY_derive_init(ctx) <= 0)
    /* Error */
if (EVP_PKEY_derive_set_peer(ctx, peerkey) <= 0)
    /* Error */

/* Determine buffer length */
if (EVP_PKEY_derive(ctx, NULL, &skeylen) <= 0)
    /* Error */

skey = OPENSSL_malloc(skeylen);

if (!skey)
    /* malloc failure */

if (EVP_PKEY_derive(ctx, skey, &skeylen) <= 0)
    /* Error */
```

```
/* Shared secret is skey bytes written to buffer skey */
```

SEE ALSO

EVP_PKEY_CTX_new(3), EVP_PKEY_encrypt(3), EVP_PKEY_decrypt(3), EVP_PKEY_sign(3), EVP_PKEY_verify(3), EVP_PKEY_verify_recover(3), EVP_KEYEXCH_fetch(3)

HISTORY

The EVP_PKEY_derive_init(), EVP_PKEY_derive_set_peer() and EVP_PKEY_derive() functions were originally added in OpenSSL 1.0.0.

The EVP_PKEY_derive_init_ex() and EVP_PKEY_derive_set_peer_ex() functions were added in OpenSSL 3.0.

COPYRIGHT