NAME

EVP_PKEY_derive_init, 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_set_peer(EVP_PKEY_CTX *ctx, EVP_PKEY *peer);
int EVP_PKEY_derive(EVP_PKEY_CTX *ctx, unsigned char *key, size_t *keylen);
```

DESCRIPTION

The *EVP_PKEY_derive_init()* function initializes a public key algorithm context using key **pkey** for shared secret derivation.

The *EVP_PKEY_derive_set_peer()* function sets the peer key: this will normally be a public key.

The *EVP_PKEY_derive()* derives a shared secret using **ctx**. If**k ey** 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.

EXAMPLE

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

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

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

ctx = EVP_PKEY_CTX_new(pkey);
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)*,

HISTORY

These functions were first added to OpenSSL 1.0.0.