NAME

EVP_PKEY_decapsulate_init, EVP_PKEY_decapsulate - Key decapsulation using a KEM algorithm with a private key

SYNOPSIS

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

 int EVP_PKEY_decapsulate_init(EVP_PKEY_CTX *ctx, const OSSL_PARAM params[]);
 int EVP_PKEY_decapsulate(EVP_PKEY_CTX *ctx,
                          unsigned char *unwrapped, size_t *unwrappedlen,
                          const unsigned char *wrapped, size_t wrappedlen);
```

DESCRIPTION

The **EVP_PKEY_decapsulate_init()** function initializes a private key algorithm context *ctx* for a decapsulation operation and then sets the *params* on the context in the same way as calling **EVP_PKEY_CTX_set_params**(3). Note that *ctx* usually is produced using **EVP_PKEY_CTX_new_from_pkey**(3), specifying the private key to use.

The **EVP_PKEY_decapsulate()** function performs a private key decapsulation operation using *ctx*. The data to be decapsulated is specified using the *wrapped* and *wrappedlen* parameters. If *unwrapped* is NULL then the maximum size of the output secret buffer is written to **unwrappedlen*. If *unwrapped* is not NULL and the call is successful then the decapsulated secret data is written to *unwrapped* and the amount of data written to **unwrappedlen*.

NOTES

After the call to **EVP_PKEY_decapsulate_init()** algorithm-specific parameters for the operation may be set or modified using **EVP_PKEY_CTX_set_params**(3).

RETURN VALUES

EVP_PKEY_decapsulate_init() and **EVP_PKEY_decapsulate()** 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 private key algorithm.

EXAMPLES

Decapsulate data using RSA:

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

 /*
  * NB: assumes rsa_priv_key is an RSA private key,
  * and that in, inlen are already set up to contain encapsulated data.
  */

 EVP_PKEY_CTX *ctx = NULL;
 size_t secretlen = 0;
 unsigned char *secret = NULL;;

 ctx = EVP_PKEY_CTX_new_from_pkey(libctx, rsa_priv_key, NULL);
 if (ctx = NULL)
     /* Error */
 if (EVP_PKEY_decapsulate_init(ctx, NULL) <= 0)
     /* Error */

 /* Set the mode - only 'RSASVE' is currently supported */
 if (EVP_PKEY_CTX_set_kem_op(ctx, "RSASVE") <= 0)
     /* Error */

 /* Determine buffer length */
 if (EVP_PKEY_decapsulate(ctx, NULL, &secretlen, in, inlen) <= 0)
     /* Error */

 secret = OPENSSL_malloc(secretlen);
 if (secret == NULL)
     /* malloc failure */

 /* Decapsulated secret data is secretlen bytes long */
 if (EVP_PKEY_decapsulaterctx, secret, &secretlen, in, inlen) <= 0)
     /* Error */
```

SEE ALSO

EVP_PKEY_CTX_new_from_pkey(3), **EVP_PKEY_encapsulate**(3), **EVP_KEM-RSA**(7),

HISTORY

These functions were added in OpenSSL 3.0.

COPYRIGHT