NAME

EVP_OpenInit, EVP_OpenUpdate, EVP_OpenFinal - EVP envelope decryption

SYNOPSIS

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

int EVP_OpenInit(EVP_CIPHER_CTX *ctx, EVP_CIPHER *type, unsigned char *ek,
                 int ekl, unsigned char *iv, EVP_PKEY *priv);
int EVP_OpenUpdate(EVP_CIPHER_CTX *ctx, unsigned char *out,
                   int *outl, unsigned char *in, int inl);
int EVP_OpenFinal(EVP_CIPHER_CTX *ctx, unsigned char *out, int *outl);
```

DESCRIPTION

The EVP envelope routines are a high-level interface to envelope decryption. They decrypt a public key encrypted symmetric key and then decrypt data using it.

EVP_OpenInit() initializes a cipher context **ctx** for decryption with cipher **type**. It decrypts the encrypted symmetric key of length **ekl** bytes passed in the **ek** parameter using the private key **priv**. The IV is supplied in the **iv** parameter.

EVP_OpenUpdate() and **EVP_OpenFinal()** have exactly the same properties as the **EVP_DecryptUpdate()** and **EVP_DecryptFinal()** routines, as documented on the **EVP_EncryptInit**(3) manual page.

NOTES

It is possible to call **EVP_OpenInit()** twice in the same way as **EVP_DecryptInit()**. The first call should have **priv** set to NULL and (after setting any cipher parameters) it should be called again with **type** set to NULL.

If the cipher passed in the **type** parameter is a variable length cipher then the key length will be set to the value of the recovered key length. If the cipher is a fixed length cipher then the recovered key length must match the fixed cipher length.

RETURN VALUES

EVP_OpenInit() returns 0 on error or a non zero integer (actually the recovered secret key size) if successful.

EVP_OpenUpdate() returns 1 for success or 0 for failure.

EVP_OpenFinal() returns 0 if the decrypt failed or 1 for success.

SEE ALSO

evp(7), **RAND_bytes**(3), **EVP_EncryptInit**(3), **EVP_SealInit**(3)

COPYRIGHT