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 'PKCS7_encrypt_ex.3ossl' command

$ man PKCS7_encrypt_ex.3ossl

PKCS7_ENCRYPT(3ossl) OpenSSL PKCS7_ENCRYPT(3ossl)

NAME

PKCS7_encrypt_ex, PKCS7_encrypt - create a PKCS#7 envelopedData structure

SYNOPSIS

```
#include <openssl/pkcs7.h>

PKCS7 *PKCS7_encrypt_ex(STACK_OF(X509) *certs, BIO *in,
                        const EVP_CIPHER *cipher, int flags,
                        OSSL_LIB_CTX *libctx, const char *propq);
PKCS7 *PKCS7_encrypt(STACK_OF(X509) *certs, BIO *in, const EVP_CIPHER *cipher,
                     int flags);
```

DESCRIPTION

PKCS7_encrypt_ex() creates and returns a PKCS#7 envelopedData structure. certs is a list of recipient certificates. in is the content to be encrypted. cipher is the symmetric cipher to use. flags is an optional set of flags. The library context libctx and the property query propq are used when retrieving algorithms from providers.

Only RSA keys are supported in PKCS#7 and envelopedData so the recipient certificates supplied to this function must all contain RSA public keys, though they do not have to be signed using the RSA algorithm.

EVP_des_ede3_cbc() (triple DES) is the algorithm of choice for S/MIME use because most clients will support it.

Some old "export grade" clients may only support weak encryption using 40 or 64 bit RC2. These can be used by passing EVP_rc2_40_cbc() and EVP_rc2_64_cbc() respectively.

The algorithm passed in the cipher parameter must support ASN1 encoding of its parameters.

Many browsers implement a "sign and encrypt" option which is simply an S/MIME envelopedData containing an S/MIME signed message. This can be readily produced by storing the S/MIME signed message in a memory BIO and passing it to PKCS7_encrypt().

The following flags can be passed in the flags parameter.

If the PKCS7_TEXT flag is set MIME headers for type text/plain are prepended to the data.

Normally the supplied content is translated into MIME canonical format (as required by the S/MIME specifications) if PKCS7_BINARY is set no translation occurs. This option should be used if the supplied data is in binary format otherwise the translation will corrupt it. If PKCS7_BINARY is set then PKCS7_TEXT is ignored.

If the PKCS7_STREAM flag is set a partial PKCS7 structure is output suitable for streaming I/O: no data is read from the BIO in.

If the flag PKCS7_STREAM is set the returned PKCS7 structure is not complete and outputting its contents via a function that does not properly finalize the PKCS7 structure will give unpredictable results.

Several functions including SMIME_write_PKCS7(), i2d_PKCS7_bio_stream(), PEM_write_bio_PKCS7_stream() finalize the structure. Alternatively finalization can be performed by obtaining the streaming ASN1 BIO directly using BIO_new_PKCS7().

PKCS7_encrypt() is similar to PKCS7_encrypt_ex() but uses default values of NULL for the library context libctx and the property query propq.

RETURN VALUES

PKCS7_encrypt_ex() and PKCS7_encrypt() return either a PKCS7 structure or NULL if an error occurred. The error can be obtained from ERR_get_error(3).

SEE ALSO

ERR_get_error(3), PKCS7_decrypt(3)

HISTORY

The function PKCS7_encrypt_ex() was added in OpenSSL 3.0.

The PKCS7_STREAM flag was added in OpenSSL 1.0.0.

COPYRIGHT

<https://www.openssl.org/source/license.html>.