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 'CMS_encrypt_ex.3ossl' command

$ man CMS_encrypt_ex.3ossl

CMS_ENCRYPT(3ossl) OpenSSL CMS_ENCRYPT(3ossl)

NAME

CMS_encrypt_ex, CMS_encrypt - create a CMS envelopedData structure

SYNOPSIS

```
#include <openssl/cms.h>
CMS_ContentInfo *CMS_encrypt_ex(STACK_OF(X509) *certs, BIO *in,
                                const EVP_CIPHER *cipher, unsigned int flags,
                                OSSL_LIB_CTX *libctx, const char *propq);
CMS_ContentInfo *CMS_encrypt(STACK_OF(X509) *certs, BIO *in,
                             const EVP_CIPHER *cipher, unsigned int flags);
```

DESCRIPTION

CMS_encrypt_ex() creates and returns a CMS EnvelopedData or AuthEnvelopedData 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 internally when retrieving algorithms from providers.

Only certificates carrying RSA, Diffie-Hellman or EC keys are supported by this function.

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

The algorithm passed in the cipher parameter must support ASN1 encoding of its parameters. If the cipher mode is GCM, then an AuthEnvelopedData

structure containing MAC is used. Otherwise an EnvelopedData structure is used. Currently the AES variants with GCM mode are the only supported AEAD algorithms.

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 CMS_encrypt().

The following flags can be passed in the flags parameter.

If the CMS_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 CMS_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 CMS_BINARY is set then CMS_TEXT is ignored.

OpenSSL will by default identify recipient certificates using issuer name and serial number. If CMS_USE_KEYID is set it will use the subject key identifier value instead. An error occurs if all recipient certificates do not have a subject key identifier extension.

If the CMS_STREAM flag is set a partial CMS_ContentInfo structure is returned suitable for streaming I/O: no data is read from the BIO in.

If the CMS_PARTIAL flag is set a partial CMS_ContentInfo structure is returned to which additional recipients and attributes can be added before finalization.

The data being encrypted is included in the CMS_ContentInfo structure, unless CMS_DETACHED is set in which case it is omitted. This is rarely used in practice and is not supported by SMIME_write_CMS().

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

Several functions including SMIME_write_CMS(), i2d_CMS_bio_stream(), PEM_write_bio_CMS_stream() finalize the structure. Alternatively

finalization can be performed by obtaining the streaming ASN1 BIO directly using BIO_new_CMS().

The recipients specified in certs use a CMS KeyTransRecipientInfo info structure. KEKRecipientInfo is also supported using the flag CMS_PARTIAL and CMS_add0_recipient_key().

The parameter certs may be NULL if CMS_PARTIAL is set and recipients added later using CMS_add1_recipient_cert() or CMS_add0_recipient_key().

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

RETURN VALUES

CMS_encrypt_ex() and CMS_encrypt() return either a CMS_ContentInfo structure or NULL if an error occurred. The error can be obtained from ERR_get_error(3).

SEE ALSO

ERR_get_error(3), CMS_decrypt(3)

HISTORY

The function CMS_encrypt_ex() was added in OpenSSL 3.0.

The CMS_STREAM flag was first supported in OpenSSL 1.0.0.

COPYRIGHT