

**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**

Copyright 2008-2020 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at [<https://www.openssl.org/source/license.html>](https://www.openssl.org/source/license.html).