

**NAME**

`CMS_encrypt` - create a CMS envelopedData structure

**SYNOPSIS**

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

```
CMS_ContentInfo *CMS_encrypt(STACK_OF(X509) *certs, BIO *in, const EVP_CIPHER *c
```

**DESCRIPTION**

`CMS_encrypt()` creates and returns a CMS 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.

**NOTES**

Only certificates carrying RSA keys are supported 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.

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 `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()`.

**NOTES**

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 ASN1BIO 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()`.

**RETURN VALUES**

`CMS_encrypt()` returns 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**

*CMS\_decrypt()* was added to OpenSSL 0.9.8 The **CMS\_STREAM** flag was first supported in OpenSSL 1.0.0.