Rocky Enterprise Linux 9.2 Manual Pages on command 'CMS_decrypt.3ossl'

$ man CMS_decrypt.3ossl

CMS_DECRYPT(3ossl) OpenSSL CMS_DECRYPT(3ossl)

NAME

CMS_decrypt, CMS_decrypt_set1_pkey_and_peer, CMS_decrypt_set1_pkey - decrypt content from a CMS envelopedData structure

SYNOPSIS

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

int CMS_decrypt(CMS_ContentInfo *cms, EVP_PKEY *pkey, X509 *cert,
                BIO *dcont, BIO *out, unsigned int flags);
int CMS_decrypt_set1_pkey_and_peer(CMS_ContentInfo *cms,
                EVP_PKEY *pk, X509 *cert, X509 *peer);
int CMS_decrypt_set1_pkey(CMS_ContentInfo *cms, EVP_PKEY *pk, X509 *cert);
```

DESCRIPTION

CMS_decrypt() extracts and decrypts the content from a CMS EnvelopedData or AuthEnvelopedData structure. pkey is the private key

of the recipient, cert is the recipient's certificate, out is a BIO to write the content to and flags is an optional set of flags.

The dcont parameter is used in the rare case where the encrypted content is detached. It will normally be set to NULL.

CMS_decrypt_set1_pkey_and_peer() associates the private key pkey, the corresponding certificate cert and the originator certificate peer with the CMS_ContentInfo structure cms.

CMS_decrypt_set1_pkey() associates the private key pkey, corresponding certificate cert with the CMS_ContentInfo structure cms.

NOTES

Although the recipients certificate is not needed to decrypt the data it is needed to locate the appropriate (of possible several) recipients in the CMS structure.

If cert is set to NULL all possible recipients are tried. This case however is problematic. To thwart the MMA attack (Bleichenbacher's attack on PKCS #1 v1.5 RSA padding) all recipients are tried whether they succeed or not. If no recipient succeeds then a random symmetric key is used to decrypt the content: this will typically output garbage and may (but is not guaranteed to) ultimately return a padding error only. If CMS_decrypt() just returned an error when all recipient encrypted keys failed to decrypt an attacker could use this in a timing attack. If the special flag CMS_DEBUG_DECRYPT is set then the above behaviour is modified and an error is returned if no recipient encrypted key can be decrypted without generating a random content encryption key. Applications should use this flag with extreme caution especially in automated gateways as it can leave them open to attack.

It is possible to determine the correct recipient key by other means

(for example looking them up in a database) and setting them in the CMS structure in advance using the CMS utility functions such as CMS_set1_pkey(). In this case both cert and pkey should be set to NULL.

To process KEKRecipientInfo types CMS_set1_key() or CMS_RecipientInfo_set0_key() and CMS_RecipientInfo_decrypt() should be called before CMS_decrypt() and cert and pkey set to NULL.

The following flags can be passed in the flags parameter.

If the CMS_TEXT flag is set MIME headers for type text/plain are deleted from the content. If the content is not of type text/plain then an error is returned.

RETURN VALUES

CMS_decrypt() returns either 1 for success or 0 for failure. The error can be obtained from ERR_get_error(3)

BUGS

The lack of single pass processing and the need to hold all data in memory as mentioned in CMS_verify() also applies to CMS_decrypt().

SEE ALSO

ERR_get_error(3), CMS_encrypt(3)

HISTORY

CMS_decrypt_set1_pkey_and_peer was added in OpenSSL 3.0.

COPYRIGHT

in the file LICENSE in the source distribution or at

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