NAME

CMS_get0_RecipientInfos, CMS_RecipientInfo_type, CMS_RecipientInfo_ktri_get0_signer_id, CMS_RecipientInfo_ktri_cert_cmp, CMS_RecipientInfo_set0_pkey, CMS_RecipientInfo_kekri_get0_id, CMS_RecipientInfo_kari_set0_pkey_and_peer, CMS_RecipientInfo_kari_set0_pkey, CMS_RecipientInfo_kekri_id_cmp, CMS_RecipientInfo_set0_key, CMS_RecipientInfo_decrypt, CMS_RecipientInfo_encrypt - CMS envelopedData RecipientInfo routines

SYNOPSIS

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

STACK_OF(CMS_RecipientInfo) *CMS_get0_RecipientInfos(CMS_ContentInfo *cms);
int CMS_RecipientInfo_type(CMS_RecipientInfo *ri);

int CMS_RecipientInfo_ktri_get0_signer_id(CMS_RecipientInfo *ri,
                                          ASN1_OCTET_STRING **keyid,
                                          X509_NAME **issuer,
                                          ASN1_INTEGER **sno);
int CMS_RecipientInfo_ktri_cert_cmp(CMS_RecipientInfo *ri, X509 *cert);
int CMS_RecipientInfo_set0_pkey(CMS_RecipientInfo *ri, EVP_PKEY *pkey);
int CMS_RecipientInfo_kari_set0_pkey_and_peer(CMS_RecipientInfo *ri,
                                              EVP_PKEY *pk, X509 *peer);
int CMS_RecipientInfo_kari_set0_pkey(CMS_RecipientInfo *ri, EVP_PKEY *pk);
int CMS_RecipientInfo_kekri_get0_id(CMS_RecipientInfo *ri, X509_ALGOR **palg,
                                    ASN1_OCTET_STRING **pid,
                                    ASN1_GENERALIZEDTIME **pdate,
                                    ASN1_OBJECT **potherid,
                                    ASN1_TYPE **pothertype);
int CMS_RecipientInfo_kekri_id_cmp(CMS_RecipientInfo *ri,
                                   const unsigned char *id, size_t idlen);
int CMS_RecipientInfo_set0_key(CMS_RecipientInfo *ri,
                               unsigned char *key, size_t keylen);

int CMS_RecipientInfo_decrypt(CMS_ContentInfo *cms, CMS_RecipientInfo *ri);
int CMS_RecipientInfo_encrypt(CMS_ContentInfo *cms, CMS_RecipientInfo *ri);
```

DESCRIPTION

The function **CMS_get0_RecipientInfos()** returns all the CMS_RecipientInfo structures associated with a CMS EnvelopedData structure.

CMS_RecipientInfo_type() returns the type of CMS_RecipientInfo structure **ri**. It will currently return CMS_RECIPINFO_TRANS, CMS_RECIPINFO_AGREE, CMS_RECIPINFO_KEK, CMS_RECIPINFO_PASS, or CMS_RECIPINFO_OTHER.

CMS_RecipientInfo_ktri_get0_signer_id() retrieves the certificate recipient identifier associated with a specific CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_TRANS. Either the keyidentifier will be set in **keyid** or **both** issuer name and serial number in **issuer** and **sno**.

CMS_RecipientInfo_ktri_cert_cmp() compares the certificate **cert** against the CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_TRANS. It returns zero if the comparison is successful and non zero if not.

CMS_RecipientInfo_set0_pkey() associates the private key **pkey** with the CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_TRANS.

CMS_RecipientInfo_kari_set0_pkey_and_peer() associates the private key **pkey** and peer certificate **peer** with the CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_AGREE.

CMS_RecipientInfo_kari_set0_pkey() associates the private key **pkey** with the CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_AGREE.

CMS_RecipientInfo_kekri_get0_id() retrieves the key information from the CMS_RecipientInfo structure **ri** which must be of type CMS_RECIPINFO_KEK. Any of the remaining parameters can be NULL if the application is not interested in the value of a field. Where a field is optional and absent NULL will be written to the corresponding parameter. The keyEncryptionAlgorithm field is written to **palg**, the **keyIdentifier** field is written to **pid**, the **date** field if present is written to **pdate**, if the **other** field is present the components **keyAttrId** and **keyAttr** are written to parameters **potherid** and **pothertype**.

CMS_RecipientInfo_kekri_id_cmp() compares the ID in the **id** and **idlen** parameters against the **keyIdentifier** CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_KEK. It returns zero if the comparison is successful and non zero if not.

CMS_RecipientInfo_set0_key() associates the symmetric key **key** of length **keylen** with the CMS_RecipientInfo structure **ri**, which must be of type CMS_RECIPINFO_KEK.

CMS_RecipientInfo_decrypt() attempts to decrypt CMS_RecipientInfo structure **ri** in structure **cms**. A key must have been associated with the structure first.

CMS_RecipientInfo_encrypt() attempts to encrypt CMS_RecipientInfo structure **ri** in structure **cms**. A

key must have been associated with the structure first and the content encryption key must be available: for example by a previous call to **CMS_RecipientInfo_decrypt()**.

NOTES

The main purpose of these functions is to enable an application to lookup recipient keys using any appropriate technique when the simpler method of **CMS_decrypt()** is not appropriate.

In typical usage and application will retrieve all CMS_RecipientInfo structures using **CMS_get0_RecipientInfos()** and check the type of each using **CMS_RecipientInfo_type()**. Depending on the type the CMS_RecipientInfo structure can be ignored or its key identifier data retrieved using an appropriate function. Then if the corresponding secret or private key can be obtained by any appropriate means it can then associated with the structure and **CMS_RecipientInfo_decrypt()** called. If successful **CMS_decrypt()** can be called with a NULL key to decrypt the enveloped content.

The **CMS_RecipientInfo_encrypt()** can be used to add a new recipient to an existing enveloped data structure. Typically an application will first decrypt an appropriate CMS_RecipientInfo structure to make the content encrypt key available, it will then add a new recipient using a function such as **CMS_add1_recipient_cert()** and finally encrypt the content encryption key using **CMS_RecipientInfo_encrypt()**.

RETURN VALUES

CMS_get0_RecipientInfos() returns all CMS_RecipientInfo structures, or NULL if an error occurs.

CMS_RecipientInfo_ktri_get0_signer_id(), **CMS_RecipientInfo_set0_pkey()**, **CMS_RecipientInfo_kekri_get0_id()**, **CMS_RecipientInfo_set0_key()** and **CMS_RecipientInfo_decrypt()** return 1 for success or 0 if an error occurs.
CMS_RecipientInfo_encrypt() return 1 for success or 0 if an error occurs.

CMS_RecipientInfo_ktri_cert_cmp() and **CMS_RecipientInfo_kekri_cmp()** return 0 for a successful comparison and non zero otherwise.

Any error can be obtained from **ERR_get_error**(3).

SEE ALSO

ERR_get_error(3), **CMS_decrypt**(3)

HISTORY

CMS_RecipientInfo_kari_set0_pkey_and_peer and **CMS_RecipientInfo_kari_set0_pkey** were added in OpenSSL 3.0.

COPYRIGHT