NAME

hx509 CMS/pkcs7 functions -

Functions

int **hx509_cms_wrap_ContentInfo** (const heim_oid *oid, const heim_octet_string *buf, heim_octet_string *res)

int **hx509_cms_unwrap_ContentInfo** (const heim_octet_string *in, heim_oid *oid, heim_octet_string *out, int *have_data)

int **hx509_cms_unenvelope** (hx509_context context, hx509_certs certs, int flags, const void *data, size_t length, const heim_octet_string *encryptedContent, time_t time_now, heim_oid *contentType, heim_octet_string *content)

int **hx509_cms_envelope_1** (hx509_context context, int flags, hx509_cert cert, const void *data, size_t length, const heim_oid *encryption_type, const heim_oid *contentType, heim_octet_string *content)

int **hx509_cms_verify_signed** (hx509_context context, hx509_verify_ctx ctx, unsigned int flags, const void *data, size_t length, const heim_octet_string *signedContent, hx509_certs pool, heim_oid *contentType, heim_octet_string *content, hx509_certs *signer_certs)

int **hx509_cms_create_signed_1** (hx509_context context, int flags, const heim_oid *eContentType, const void *data, size_t length, const AlgorithmIdentifier *digest_alg, hx509_cert cert, hx509_peer_info peer, hx509_certs anchors, hx509_certs pool, heim_octet_string *signed_data)

Detailed Description

See the **CMS/PKCS7 message functions.** for description and examples.

Function Documentation

int hx509_cms_create_signed_1 (hx509_context context, int flags, const heim_oid * eContentType, const void * data, size_t length, const AlgorithmIdentifier * digest_alg, hx509_cert cert, hx509_peer_info peer, hx509_certs anchors, hx509_certs pool, heim_octet_string * signed_data)

Decode SignedData and verify that the signature is correct.

Parameters:

context A hx509 context.
flags
eContentType the type of the data.
data data to sign
length length of the data that data point to.
digest_alg digest algorithm to use, use NULL to get the default or the peer determined algorithm.
cert certificate to use for sign the data.
peer info about the peer the message to send the message to, like what digest algorithm to use.
anchors trust anchors that the client will use, used to polulate the certificates included in the message

pool certificates to use in try to build the path to the trust anchors.
signed_data the output of the function, free with der_free_octet_string().

int hx509_cms_envelope_1 (hx509_context context, int flags, hx509_cert cert, const void * data, size_t length, const heim_oid * encryption_type, const heim_oid * contentType, heim_octet_string * content)
Encrypt end encode EnvelopedData.

Encrypt and encode EnvelopedData. The data is encrypted with a random key and the the random key is encrypted with the certificates private key. This limits what private key type can be used to RSA.

Parameters:
context A hx509 context.
flags flags to control the behavior.

- HX509_CMS_EV_NO_KU_CHECK - Dont check KU on certificate
- HX509_CMS_EV_ALLOW_WEAK - Allow weak crytpo
- HX509_CMS_EV_ID_NAME - prefer issuer name and serial number

cert Certificate to encrypt the EnvelopedData encryption key with.
data pointer the data to encrypt.
length length of the data that data point to.
encryption_type Encryption cipher to use for the bulk data, use NULL to get default.
contentType type of the data that is encrypted
content the output of the function, free with der_free_octet_string().

int hx509_cms_unenvelope (hx509_context context, hx509_certs certs, int flags, const void * data, size_t length, const heim_octet_string * encryptedContent, time_t time_now, heim_oid * contentType, heim_octet_string * content)
Decode and unencrypt EnvelopedData.

Extract data and parameteres from from the EnvelopedData. Also supports using detached EnvelopedData.

Parameters:
context A hx509 context.
certs Certificate that can decrypt the EnvelopedData encryption key.
flags HX509_CMS_UE flags to control the behavior.
data pointer the structure the contains the DER/BER encoded EnvelopedData stucture.

length length of the data that data point to.
encryptedContent in case of detached signature, this contains the actual encrypted data, othersize its should be NULL.
time_now set the current time, if zero the library uses now as the date.
contentType output type oid, should be freed with der_free_oid().
content the data, free with der_free_octet_string().

int hx509_cms_unwrap_ContentInfo (const heim_octet_string * in, heim_oid * oid, heim_octet_string * out, int * have_data)
Decode an ContentInfo and unwrap data and oid it.

Parameters:
in the encoded buffer.
oid type of the content.
out data to be wrapped.
have_data since the data is optional, this flags show dthe diffrence between no data and the zero length data.

Returns:
Returns an hx509 error code.

int hx509_cms_verify_signed (hx509_context context, hx509_verify_ctx ctx, unsigned int flags, const void * data, size_t length, const heim_octet_string * signedContent, hx509_certs pool, heim_oid * contentType, heim_octet_string * content, hx509_certs * signer_certs)
Decode SignedData and verify that the signature is correct.

Parameters:
context A hx509 context.
ctx a hx509 verify context.
flags to control the behaivor of the function.

- HX509_CMS_VS_NO_KU_CHECK - Don't check KeyUsage
- HX509_CMS_VS_ALLOW_DATA_OID_MISMATCH - allow oid mismatch
- HX509_CMS_VS_ALLOW_ZERO_SIGNER - no signer, see below.

data pointer to CMS SignedData encoded data.
length length of the data that data point to.
signedContent external data used for signature.

pool certificate pool to build certificates paths.
contentType free with der_free_oid().
content the output of the function, free with der_free_octet_string().
signer_certs list of the cerficates used to sign this request, free with **hx509_certs_free()**.

If HX509_CMS_VS_NO_KU_CHECK is set, allow more liberal search for matching certificates by not considering KeyUsage bits on the certificates.

If HX509_CMS_VS_ALLOW_DATA_OID_MISMATCH, allow encapContentInfo mismatch with the oid in signedAttributes (or if no signedAttributes where use, pkcs7-data oid). This is only needed to work with broken CMS implementations that doesn't follow CMS signedAttributes rules.

If HX509_CMS_VS_NO_VALIDATE flags is set, do not verify the signing certificates and leave that up to the caller.

If HX509_CMS_VS_ALLOW_ZERO_SIGNER is set, allow empty SignerInfo (no signatures). If SignedData have no signatures, the function will return 0 with signer_certs set to NULL. Zero signers is allowed by the standard, but since its only useful in corner cases, it make into a flag that the caller have to turn on.

int hx509_cms_wrap_ContentInfo (const heim_oid * oid, const heim_octet_string * buf, heim_octet_string * res)

Wrap data and oid in a ContentInfo and encode it.

Parameters:

oid type of the content.
buf data to be wrapped. If a NULL pointer is passed in, the optional content field in the ContentInfo is not going be filled in.
res the encoded buffer, the result should be freed with der_free_octet_string().

Returns:

Returns an hx509 error code.