NAME

hx509 verification functions -

Functions

void **hx509_context_set_missing_revoke** (hx509_context context, int flag)
int **hx509_verify_init_ctx** (hx509_context context, hx509_verify_ctx *ctx)
void **hx509_verify_destroy_ctx** (hx509_verify_ctx ctx)
void **hx509_verify_attach_anchors** (hx509_verify_ctx ctx, hx509_certs set)
void **hx509_verify_attach_revoke** (hx509_verify_ctx ctx, hx509_revoke_ctx revoke_ctx)
void **hx509_verify_set_time** (hx509_verify_ctx ctx, time_t t)
void **hx509_verify_set_max_depth** (hx509_verify_ctx ctx, unsigned int max_depth)
void **hx509_verify_set_proxy_certificate** (hx509_verify_ctx ctx, int boolean)
void **hx509_verify_set_strict_rfc3280_verification** (hx509_verify_ctx ctx, int boolean)
int **hx509_verify_path** (hx509_context context, hx509_verify_ctx ctx, hx509_cert cert, hx509_certs pool)
int **hx509_ocsp_verify** (hx509_context context, time_t now, hx509_cert cert, int flags, const void *data, size_t length, time_t *expiration)
int **hx509_crl_alloc** (hx509_context context, hx509_crl *crl)
int **hx509_crl_add_revoked_certs** (hx509_context context, hx509_crl crl, hx509_certs certs)
int **hx509_crl_lifetime** (hx509_context context, hx509_crl crl, int delta)
void **hx509_crl_free** (hx509_context context, hx509_crl *crl)
int **hx509_crl_sign** (hx509_context context, hx509_cert signer, hx509_crl crl, heim_octet_string *os)

Detailed Description

Function Documentation

void hx509_context_set_missing_revoke (hx509_context context, int flag)

Selects if the **hx509_revoke_verify()** function is going to require the existans of a revokation method (OCSP, CRL) or not. Note that **hx509_verify_path()**, **hx509_cms_verify_signed()**, and other function call **hx509_revoke_verify()**.

Parameters:

context hx509 context to change the flag for.
flag zero, revokation method required, non zero missing revokation method ok

int hx509_crl_add_revoked_certs (hx509_context context, hx509_crl crl, hx509_certs certs)

Add revoked certificate to an CRL context.

Parameters:

context a hx509 context.
crl the CRL to add the revoked certificate to.
certs keyset of certificate to revoke.

Returns:
An hx509 error code, see **hx509_get_error_string()**.

int hx509_crl_alloc (hx509_context context, hx509_crl * crl)
Create a CRL context. Use **hx509_crl_free()** to free the CRL context.

Parameters:
context a hx509 context.
crl return pointer to a newly allocated CRL context.

Returns:
An hx509 error code, see **hx509_get_error_string()**.

void hx509_crl_free (hx509_context context, hx509_crl * crl)
Free a CRL context.

Parameters:
context a hx509 context.
crl a CRL context to free.

int hx509_crl_lifetime (hx509_context context, hx509_crl crl, int delta)
Set the lifetime of a CRL context.

Parameters:
context a hx509 context.
crl a CRL context
delta delta time the certificate is valid, library adds the current time to this.

Returns:
An hx509 error code, see **hx509_get_error_string()**.

int hx509_crl_sign (hx509_context context, hx509_cert signer, hx509_crl crl, heim_octet_string * os)
Sign a CRL and return an encode certificate.

Parameters:
context a hx509 context.
signer certificate to sign the CRL with
crl the CRL to sign
os return the signed and encoded CRL, free with free_heim_octet_string()

Returns:

An hx509 error code, see **hx509_get_error_string()**.

int hx509_ocsp_verify (hx509_context context, time_t now, hx509_cert cert, int flags, const void * data, size_t length, time_t * expiration)

Verify that the certificate is part of the OCSP reply and it's not expired. Doesn't verify signature the OCSP reply or it's done by a authorized sender, that is assumed to be already done.

Parameters:

context a hx509 context
now the time right now, if 0, use the current time.
cert the certificate to verify
flags flags control the behavior
data pointer to the encode ocsp reply
length the length of the encode ocsp reply
expiration return the time the OCSP will expire and need to be rechecked.

Returns:

An hx509 error code, see **hx509_get_error_string()**.

void hx509_verify_attach_anchors (hx509_verify_ctx ctx, hx509_certs set)

Set the trust anchors in the verification context, makes an reference to the keyset, so the consumer can free the keyset independent of the destruction of the verification context (ctx). If there already is a keyset attached, it's released.

Parameters:

ctx a verification context
set a keyset containing the trust anchors.

void hx509_verify_attach_revoke (hx509_verify_ctx ctx, hx509_revoke_ctx revoke_ctx)

Attach an revocation context to the verfication context, , makes an reference to the revoke context, so the consumer can free the revoke context independent of the destruction of the verification context. If there is no revoke context, the verification process is NOT going to check any verification status.

Parameters:

ctx a verification context.
revoke_ctx a revoke context.

void hx509_verify_destroy_ctx (hx509_verify_ctx ctx)

Free an hx509 verification context.

Parameters:

ctx the context to be freed.

int hx509_verify_init_ctx (hx509_context context, hx509_verify_ctx * ctx)

Allocate an verification context that is used fo control the verification process.

Parameters:

context A hx509 context.

ctx returns a pointer to a hx509_verify_ctx object.

Returns:

An hx509 error code, see **hx509_get_error_string()**.

int hx509_verify_path (hx509_context context, hx509_verify_ctx ctx, hx509_cert cert, hx509_certs pool)

Build and verify the path for the certificate to the trust anchor specified in the verify context. The path is constructed from the certificate, the pool and the trust anchors.

Parameters:

context A hx509 context.

ctx A hx509 verification context.

cert the certificate to build the path from.

pool A keyset of certificates to build the chain from.

Returns:

An hx509 error code, see **hx509_get_error_string()**.

void hx509_verify_set_max_depth (hx509_verify_ctx ctx, unsigned int max_depth)

Set the maximum depth of the certificate chain that the path builder is going to try.

Parameters:

ctx a verification context

max_depth maxium depth of the certificate chain, include trust anchor.

void hx509_verify_set_proxy_certificate (hx509_verify_ctx ctx, int boolean)

Allow or deny the use of proxy certificates

Parameters:

ctx a verification context

boolean if non zero, allow proxy certificates.

void hx509_verify_set_strict_rfc3280_verification (hx509_verify_ctx ctx, int boolean)

Select strict RFC3280 verification of certificiates. This means checking key usage on CA certificates, this will make version 1 certificiates unusable.

Parameters:

ctx a verification context

boolean if non zero, use strict verification.

void hx509_verify_set_time (hx509_verify_ctx ctx, time_t t)

Set the clock time the the verification process is going to use. Used to check certificate in the past and future time. If not set the current time will be used.

Parameters:

ctx a verification context.

t the time the verifiation is using.