

**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 verification 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 certificates. This means checking key usage on CA certificates, this will make version 1 certificates unuseable.

**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 verification is using.