NAME

Heimdal Kerberos 5 cryptography functions -

Functions

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_enctype_valid** (krb5_context context, krb5_enctype etype)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_cksumtype_to_enctype** (krb5_context context, krb5_cksumtype ctype, krb5_enctype *etype)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_encrypt_iov_ivec** (krb5_context context, krb5_crypto crypto, unsigned usage, **krb5_crypto_iov** *data, int num_data, void *ivec)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_decrypt_iov_ivec** (krb5_context context, krb5_crypto crypto, unsigned usage, **krb5_crypto_iov** *data, unsigned int num_data, void *ivec)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_create_checksum_iov** (krb5_context context, krb5_crypto crypto, unsigned usage, **krb5_crypto_iov** *data, unsigned int num_data, krb5_cksumtype *type)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_verify_checksum_iov** (krb5_context context, krb5_crypto crypto, unsigned usage, **krb5_crypto_iov** *data, unsigned int num_data, krb5_cksumtype *type)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_init** (krb5_context context, const krb5_keyblock *key, krb5_enctype etype, krb5_crypto *crypto)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_destroy** (krb5_context context, krb5_crypto crypto)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_getblocksize** (krb5_context context, krb5_crypto crypto, size_t *blocksize)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_getenctype** (krb5_context context, krb5_crypto crypto, krb5_enctype *enctype)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_getpadsize** (krb5_context context, krb5_crypto crypto, size_t *padsize)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_getconfoundersize** (krb5_context context, krb5_crypto crypto, size_t *confoundersize)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_enctype_disable** (krb5_context context, krb5_enctype enctype)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_enctype_enable** (krb5_context context, krb5_enctype enctype)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_allow_weak_crypto** (krb5_context context, krb5_boolean enable)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_random_to_key** (krb5_context context, krb5_enctype type, const void *data, size_t size, krb5_keyblock *key)

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_crypto_fx_cf2** (krb5_context context, const krb5_crypto crypto1, const krb5_crypto crypto2, krb5_data *pepper1, krb5_data *pepper2,

krb5_enctype enctype, krb5_keyblock *res)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_generate_subkey_extended** (krb5_context context, const krb5_keyblock *key, krb5_enctype etype, krb5_keyblock **subkey)
KRB5_LIB_FUNCTION void KRB5_LIB_CALL **krb5_keyblock_zero** (krb5_keyblock *keyblock)
KRB5_LIB_FUNCTION void KRB5_LIB_CALL **krb5_free_keyblock_contents** (krb5_context context, krb5_keyblock *keyblock)
KRB5_LIB_FUNCTION void KRB5_LIB_CALL **krb5_free_keyblock** (krb5_context context, krb5_keyblock *keyblock)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_copy_keyblock_contents** (krb5_context context, const krb5_keyblock *inblock, krb5_keyblock *to)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_copy_keyblock** (krb5_context context, const krb5_keyblock *inblock, krb5_keyblock **to)
KRB5_LIB_FUNCTION krb5_enctype KRB5_LIB_CALL **krb5_keyblock_get_enctype** (const krb5_keyblock *block)
KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL **krb5_keyblock_init** (krb5_context context, krb5_enctype type, const void *data, size_t size, krb5_keyblock *key)

Detailed Description

Function Documentation

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_allow_weak_crypto (krb5_context context, krb5_boolean enable)

Enable or disable all weak encryption types

Parameters:

context Kerberos 5 context
enable true to enable, false to disable

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_cksumtype_to_enctype (krb5_context context, krb5_cksumtype ctype, krb5_enctype * etype)

Return the coresponding encryption type for a checksum type.

Parameters:

context Kerberos context
ctype The checksum type to get the result enctype for
etype The returned encryption, when the matching etype is not found, etype is set to ETYPE_NULL.

Returns:

Return an error code for an failure or 0 on success.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_copy_keyblock (krb5_context context, const krb5_keyblock * inblock, krb5_keyblock ** to)

Copy a keyblock, free the output keyblock with **krb5_free_keyblock()**.

Parameters:

context a Kerberos 5 context
inblock the key to copy
to the output key.

Returns:

0 on success or a Kerberos 5 error code

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_copy_keyblock_contents (krb5_context context, const krb5_keyblock * inblock, krb5_keyblock * to)

Copy a keyblock, free the output keyblock with **krb5_free_keyblock_contents()**.

Parameters:

context a Kerberos 5 context
inblock the key to copy
to the output key.

Returns:

0 on success or a Kerberos 5 error code

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_create_checksum_iov (krb5_context context, krb5_crypto crypto, unsigned usage, krb5_crypto_iov * data, unsigned int num_data, krb5_cksumtype * type)

Create a Kerberos message checksum.

Parameters:

context Kerberos context
crypto Kerberos crypto context
usage Key usage for this buffer
data array of buffers to process
num_data length of array
type output data

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_destroy (krb5_context context, krb5_crypto crypto)

Free a crypto context created by **krb5_crypto_init()**.

Parameters:

context Kerberos context
crypto crypto context to free

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_fx_cf2 (krb5_context context, const krb5_crypto crypto1, const krb5_crypto crypto2, krb5_data * pepper1, krb5_data * pepper2, krb5_enctype enctype, krb5_keyblock * res)

The FX-CF2 key derivation function, used in FAST and preauth framework.

Parameters:

context Kerberos 5 context
crypto1 first key to combine
crypto2 second key to combine
pepper1 factor to combine with first key to garante uniqueness
pepper2 factor to combine with second key to garante uniqueness
enctype the encryption type of the resulting key
res allocated key, free with **krb5_free_keyblock_contents()**

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_getblocksize (krb5_context context, krb5_crypto crypto, size_t * blocksize)

Return the blocksize used algorithm referenced by the crypto context

Parameters:

context Kerberos context
crypto crypto context to query
blocksize the resulting blocksize

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_getconfoundersize (krb5_context context, krb5_crypto crypto, size_t * confoundersize)

Return the confounder size used by the crypto context

Parameters:

context Kerberos context
crypto crypto context to query
confoundersize the returned confounder size

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_getenctype (krb5_context context, krb5_crypto crypto, krb5_enctype * enctype)

Return the encryption type used by the crypto context

Parameters:

context Kerberos context
crypto crypto context to query
enctype the resulting encryption type

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_getpadsize (krb5_context context, krb5_crypto crypto, size_t * padsize)

Return the padding size used by the crypto context

Parameters:

context Kerberos context
crypto crypto context to query
padsize the return padding size

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_crypto_init (krb5_context context,

const krb5_keyblock * key, krb5_enctype etype, krb5_crypto * crypto)
Create a crypto context used for all encryption and signature operation. The encryption type to use is taken from the key, but can be overridden with the enctype parameter. This can be useful for encryptions types which is compatiable (DES for example).

To free the crypto context, use **krb5_crypto_destroy()**.

Parameters:

context Kerberos context
key the key block information with all key data
etype the encryption type
crypto the resulting crypto context

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_decrypt_iov_ivec (krb5_context context, krb5_crypto crypto, unsigned usage, krb5_crypto_iov * data, unsigned int num_data, void * ivec)
Inline decrypt a Kerberos message.

Parameters:

context Kerberos context
crypto Kerberos crypto context
usage Key usage for this buffer
data array of buffers to process
num_data length of array
ivec initial cbc/cts vector

Returns:

Return an error code or 0.

1. KRB5_CRYPTO_TYPE_HEADER 2. one KRB5_CRYPTO_TYPE_DATA and array [0,...] of KRB5_CRYPTO_TYPE_SIGN_ONLY in any order, however the receiver have to aware of the order. KRB5_CRYPTO_TYPE_SIGN_ONLY is commonly used unencrypoted protocol headers and trailers. The output data will be of same size as the input data or shorter.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_encrypt_iov_ivec (krb5_context context, krb5_crypto crypto, unsigned usage, krb5_crypto_iov * data, int num_data, void * ivec)
Inline encrypt a kerberos message

Parameters:

context Kerberos context
crypto Kerberos crypto context
usage Key usage for this buffer
data array of buffers to process
num_data length of array
ivec initial cbc/cts vector

Returns:

Return an error code or 0.

Kerberos encrypted data look like this:

1. KRB5_CRYPTO_TYPE_HEADER 2. array [1,...] KRB5_CRYPTO_TYPE_DATA and array [0,...] KRB5_CRYPTO_TYPE_SIGN_ONLY in any order, however the receiver have to aware of the order. KRB5_CRYPTO_TYPE_SIGN_ONLY is commonly used headers and trailers. 3. KRB5_CRYPTO_TYPE_PADDING, at least on padsize long if padsize > 1 4. KRB5_CRYPTO_TYPE_TRAILER

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_enctype_disable (krb5_context context, krb5_enctype enctype)

Disable encryption type

Parameters:

context Kerberos 5 context
enctype encryption type to disable

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_enctype_enable (krb5_context context, krb5_enctype enctype)

Enable encryption type

Parameters:

context Kerberos 5 context
enctype encryption type to enable

Returns:

Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_enctype_valid (krb5_context context, krb5_enctype etype)

Check if a enctype is valid, return 0 if it is.

Parameters:

context Kerberos context
etype enctype to check if its valid or not

Returns:

Return an error code for an failure or 0 on success (enctype valid).

KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_free_keyblock (krb5_context context, krb5_keyblock * keyblock)

Free a keyblock, also zero out the content of the keyblock, uses **krb5_free_keyblock_contents()** to free the content.

Parameters:

context a Kerberos 5 context
keyblock keyblock to free, NULL is valid argument

KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_free_keyblock_contents (krb5_context context, krb5_keyblock * keyblock)

Free a keyblock's content, also zero out the content of the keyblock.

Parameters:

context a Kerberos 5 context
keyblock keyblock content to free, NULL is valid argument

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_generate_subkey_extended (krb5_context context, const krb5_keyblock * key, krb5_enctype etype, krb5_keyblock ** subkey)

Generate subkey, from keyblock

Parameters:

context kerberos context
key session key
etype encryption type of subkey, if ETYPE_NULL, use key's enctype
subkey returned new, free with **krb5_free_keyblock()**.

Returns:

0 on success or a Kerberos 5 error code

KRB5_LIB_FUNCTION krb5_enctype KRB5_LIB_CALL krb5_keyblock_get_enctype (const krb5_keyblock * block)
Get encryption type of a keyblock.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_keyblock_init (krb5_context context, krb5_enctype type, const void * data, size_t size, krb5_keyblock * key)
Fill in ‘key’ with key data of type ‘enctype’ from ‘data’ of length ‘size’. Key should be freed using **krb5_free_keyblock_contents()**.

Returns:
0 on success or a Kerberos 5 error code

KRB5_LIB_FUNCTION void KRB5_LIB_CALL krb5_keyblock_zero (krb5_keyblock * keyblock)
Zero out a keyblock

Parameters:
keyblock keyblock to zero out

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_random_to_key (krb5_context context, krb5_enctype type, const void * data, size_t size, krb5_keyblock * key)
Converts the random bytestring to a protocol key according to Kerberos crypto frame work. It may be assumed that all the bits of the input string are equally random, even though the entropy present in the random source may be limited.

Parameters:
context Kerberos 5 context
type the enctype resulting key will be of
data input random data to convert to a key
size size of input random data, at least krb5_enctype_keysize() long
key key, output key, free with **krb5_free_keyblock_contents()**

Returns:
Return an error code or 0.

KRB5_LIB_FUNCTION krb5_error_code KRB5_LIB_CALL krb5_verify_checksum_iov (krb5_context context, krb5_crypto crypto, unsigned usage, krb5_crypto_iov * data, unsigned int num_data, krb5_cksumtype * type)
Verify a Kerberos message checksum.

Parameters:

context Kerberos context
crypto Kerberos crypto context
usage Key usage for this buffer
data array of buffers to process
num_data length of array
type return checksum type if not NULL

Returns:
Return an error code or 0.