SHA1 C-Library with Test Class

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

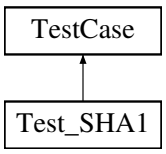
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 Test_SHA1 Class Reference

Inheritance diagram for Test_SHA1:



Public Member Functions

- **CPPUNIT_TEST_SUITE** (Test_SHA1)
- **CPPUNIT_TEST** (SHA1_Concat_test1)
- **CPPUNIT_TEST** (SHA1_Concat_test2)
- **CPPUNIT_TEST** (SHA1_Concat_test3)
- **CPPUNIT_TEST** (SHA1_test1)
- **CPPUNIT_TEST** (SHA1_File_test1)
- **CPPUNIT_TEST** (HMAC_SHA1_test1)
- **CPPUNIT_TEST** (HMAC_SHA1_test2)
- **CPPUNIT_TEST** (HMAC_SHA1_test3)
- **CPPUNIT_TEST_SUITE_END** ()
- void SHA1_Concat_test1 ()
- void SHA1_Concat_test2 ()
- void SHA1_Concat_test3 ()
- void SHA1_test1 ()
- void SHA1_File_test1 ()
- void HMAC_SHA1_test1 ()
- void HMAC_SHA1_test2 ()
- void HMAC_SHA1_test3 ()

4.1.1 Member Function Documentation

4.1.1.1 void Test_SHA1::HMAC_SHA1_test1 ( )

Test of HMAC_SHA1 with both text and key shorter than the block-size of 64 bytes

text: "what do ya want for nothing?"

key: "Jefe"

digest: 0xeffcdf6a, 0xe5eb2fa2, 0xd27416d5, 0xf184df9c, 0x259a7c79

4.1.1.2 void Test_SHA1::HMAC_SHA1_test2 ( )

Test of HMAC_SHA1 with both text and key longer than the block size of 64 bytes

text: "Test Using Larger Than Block-Size Key and Larger Than One Block-Size Data"

key: 0xaa repeated 80 times

digest: 0xe8e99d0f, 0x45237d78, 0x6d6bbaa7, 0x965c7808, 0xbbff1a91

4.1.1.3 void Test_SHA1::HMAC_SHA1_test3 ( )

Test of HMAC_SHA1 with both text and key defined by number sequences

text: 0xcd repeated 50 times

key: 0x 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19

digest: 0x4c9007f4, 0x026250c6, 0xbc8414f9, 0xbf50c86c, 0x2d7235da

4.1.1.4 void Test_SHA1::SHA1_Concat_test1 ( )

Test of SHA1_Concat with short text

text: "abc"

digest: 0xa9993e36, 0x4706816a, 0xba3e2571, 0x7850c26c, 0x9cd0d89d

4.1.1.5 void Test_SHA1::SHA1_Concat_test2 ( )

Test of SHA1_Concat with total text size shorter than the block-size of 64 bytes

text: "abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq"

digest: 0x84983e44, 0x1c3bd26e, 0xbaae4aa1, 0xf95129e5, 0xe54670f1

4.1.1.6 void Test_SHA1::SHA1_Concat_test3 ( )

Test of SHA1_Concat with total text size larger than the block-size of 64 bytes

text: "abcdefghbcdefghicdefghijdefghijkefghijklfghijklmghijklmnhijklmnoijklmnopjklmnopqklmnopqrlmnopqrsmnopqrstnopqrstu"

digest: 0xa49b2446, 0xa02c645b, 0xf419f995, 0xb6709125, 0x3a04a259

4.1.1.7 void Test_SHA1::SHA1_File_test1 ( )

Test of SHA1_File with file "testfile.txt" containing the plain text "Now is the winter of our discontent"

digest: 0x9b08c1d2, 0x42b9a4b2, 0x43b6df17, 0x3ffa21e8, 0x2868b119

4.1.1.8 void Test_SHA1::SHA1_test1 ( )

Test of SHA1 with long text

text: The letter 'a' repeated 1'000'000 times

digest: 0x34aa973c, 0xd4c4daa4, 0xf61eeb2b, 0xdbad2731, 0x6534016f

The documentation for this class was generated from the following files:

- test_sha1.h
- test_sha1.cpp

Chapter 5

File Documentation

5.1 sha1.h File Reference

Function prototypes for the SHA1 library.

Functions

- void **Load_Buffer** (struct sha1_word_pointer *p, uint32_t *W)
- void **Set_Zero** (struct sha1_word_pointer *p)
- void **Set_Pad** (struct sha1_word_pointer *p, unsigned char *pad, uint64_t text_byte_size)
- void **Conv_Int_To_Word** (uint32_t i, char *a)
- void **SHA1_Iterate_Hash** (struct sha1_word_pointer *p, uint32_t *H)
- void **SHA1_Compute** (struct sha1_word_pointer *p, uint32_t *hash)
- void SHA1_Concat (char **strings, uint64_t nr_of_strings, uint64_t *strings_byte_len, uint32_t *hash)

  Takes as an argument a collection of char arrays, performs a virtual concatenation of these arrays in their given order, implements the SHA1 algorithm on the concatenated array and stores the resulting hash.
- void SHA1 (char *text, uint64_t text_byte_size, uint32_t *hash)

  Takes as an argument a char array, implements the SHA1 algorithm on the array and stores the resulting hash.
- int SHA1_File (char *filename, uint32_t *hash)

  Takes as an argument a file name, implements the SHA1 algorithm on the file and stores the resulting hash.
- void HMAC_SHA1 (char *key, unsigned int key_size, char *text, uint64_t text_size, uint32_t *digest)

  Takes as an argument a string and a key, implements the HMAC-SHA1 algorithm and stores the resulting digest.

5.1.1 Detailed Description

Function prototypes for the SHA1 library.

Author

Anders Nordenfelt

5.1.2 Function Documentation

5.1.2.1 void HMAC_SHA1 (char * *key*, unsigned int *key_size*, char * *text*, uint64_t *text_size*, uint32_t * *digest*)

Takes as an argument a string and a key, implements the HMAC-SHA1 algorithm and stores the resulting digest.

Parameters

key	pointer to the char array containing the key
key_size	the key size in bytes
text	pointer to the char array containing the text
text_size	the text size in bytes
hash	pointer to the uint32_t array where the resulting hash is to be stored

Returns

void

5.1.2.2 void SHA1 (char * *text*, uint64_t *text_byte_size*, uint32_t * *hash*)

Takes as an argument a char array, implements the SHA1 algorithm on the array and stores the resulting hash.

Parameters

text	the pointer to the char array containing the text to be hashed
text_byte_size	the text size in bytes
hash	pointer to the uint32_t array where the resulting hash is to be stored

Returns

void

5.1.2.3 void SHA1_Concat (char ** *strings*, uint64_t *nr_of_strings*, uint64_t * *strings_byte_len*, uint32_t * *hash*)

Takes as an argument a collection of char arrays, performs a virtual concatenation of these arrays in their given order, implements the SHA1 algorithm on the concatenated array and stores the resulting hash.

Parameters

strings	the pointer to the char* array containing the pointers to the char arrays in the concatenation
nr_of_strings	the number of char arrays in the concatenation
strings_byte_size	pointer to the uint64_t array containing the size in bytes of each char array in the order they appear
hash	pointer to the uint32_t array where the resulting hash is to be stored

Returns

void

5.1.2.4 int SHA1_File (char * *filename*, uint32_t * *hash*)

Takes as an argument a file name, implements the SHA1 algorithm on the file and stores the resulting hash.

Parameters

filename	the pointer to the char array containing the file name
hash	pointer to the uint32_t array where the resulting hash is to be stored

Returns

int

Index