[Part 3]
Integrity, Authenticity, and Key Management

Chapter 12

Cryptographic Hash Functions

Forouzan, B.A. Cryptography and Network Security (International Edition). United States: McGraw Hill, 2008.

1.1

Chapter 12 Objectives

- To introduce general ideas behind cryptographic hash functions.
- To discuss the structure of SHA-512 as an example of a cryptographic hash function.

1.2

Chapter 12 Contents

1.3

Chapter 11 Contents

1.4

Chapter 12

12.1 Introduction

- A cryptographic hash function takes a message of arbitrary length and creates a message digest of fixed length.
- There are two most promising cryptographic hash algorithms - *SHA-512* and *Whirlpool.*
- However, this chapter will discuss the details of the SHA-512 only.

1.5

Chapter 12

12.1 Introduction

Iterated Hash Function

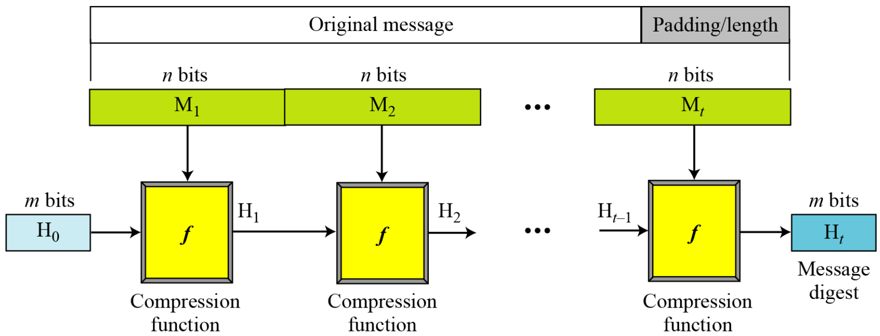
- All cryptographic hash functions need to create a *fixed-size digest* out of a variable-size message.
- The best way to create such function is using *iteration*, and used a necessary number of times.
- The fixed-size input function is referred to as a **compression function**.
  - Compresses an n-bit string to create an m-bit string, where $n > m$
- This scheme is referred to as an **iterated cryptographic hash function**.

1.6

Chapter 12 — 12.1 Introduction

***Merkle-Damgard* Scheme**

- This is an iterated hash function that is **collision resistant** if the compression function is collision resistant.
- This is the basis for many cryptographic hash functions today.



1.7

Chapter 12 — 12.1 Introduction

Groups of Compression Functions

- The design of a compression function must be **collision resistant** using the *Merkle-Damgard* scheme.
- There is a tendency to use two different approaches in designing a hash function.
  a) Compression function from *scratch*.
  b) A symmetric-key *block cipher* serves as a compression function.
- Figure 12.1 illustrates the approaches in general.

1.8

Chapter 12

12.1 Introduction

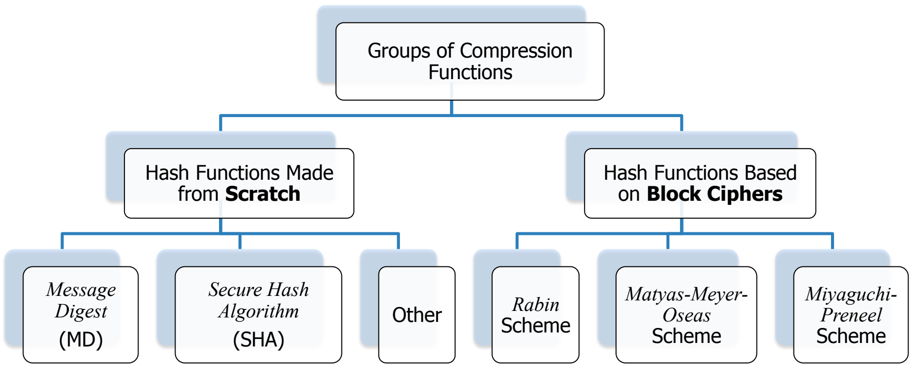


Figure12.1: Two groups of compression function in cryptographic hash function

MD2
MD4
MD5

SHA-1
SHA-224
SHA-256
SHA-384
SHA-512

Whirlpool

1.9

Chapter 12

12.1 Introduction

a) Hash Functions Made from Scratch

- Specifically designed for the purpose they serve.

Message Digest **(MD)**

- Several hash algorithms referred to as *MD2, MD4*, and *MD5* (latest version).
- *MD5* strengthened version of *MD4*:
  - divides message into blocks of 512 bits, and
  - creates a 128-bit digest.

The digest size of 128-bit is too small to resist collision attack.

1.10

Chapter 12 — 12.1 Introduction

***Secure Hash Algorithm* (SHA)**

- Sometimes referred to as *Secure Hash Standard* (SHS).
- The standard mostly based on *MD5*.
- 1st version: *SHA-1*.
- New versions: *SHA-224, SHA-256, SHA-384, SHA-512*.

Table 12.1 *Characteristics of Secure Hash Algorithms (SHAs)*

Characteristics	*SHA-1*	*SHA-224*	*SHA-256*	*SHA-384*	*SHA-512*
Maximum Message size	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{128} - 1$	$2^{128} - 1$
Block size	512	512	512	1024	1024
Message digest size	160	224	256	384	512
Number of rounds	80	64	64	80	80
Word size	32	32	32	64	64

1.11

Chapter 12 — 12.1 Introduction

b) Hash Functions Based on Block Cipher

- The whole idea is that there are several secure symmetric-key block ciphers (Triple DES, AES) can be used to make a one-way function instead of creating a new compression function.
- The block cipher in this case can only performs encryption.
- One of the most promising scheme: *Whirlpool*.

1.12

Chapter 12

12.1 Introduction

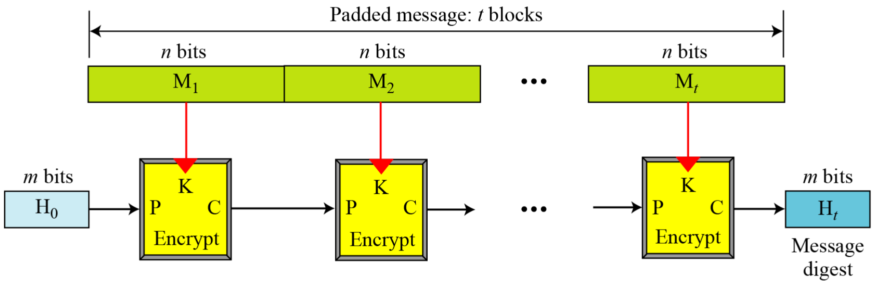


Figure 12.2: Rabin scheme

1.13

Chapter 12

12.1 Introduction

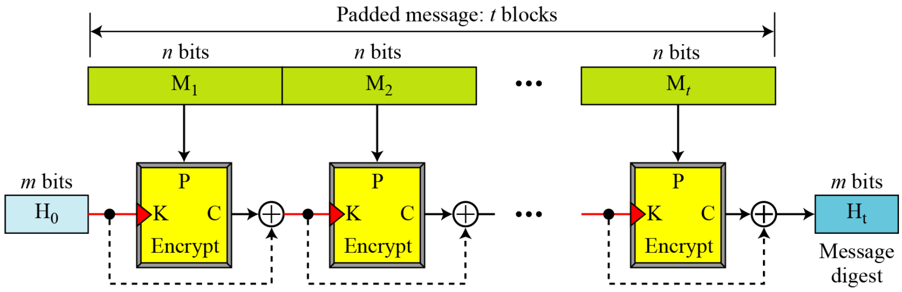


Figure 12.3: Matyas-Meyer-Oseas scheme

1.14

Chapter 12 — 12.1 Introduction

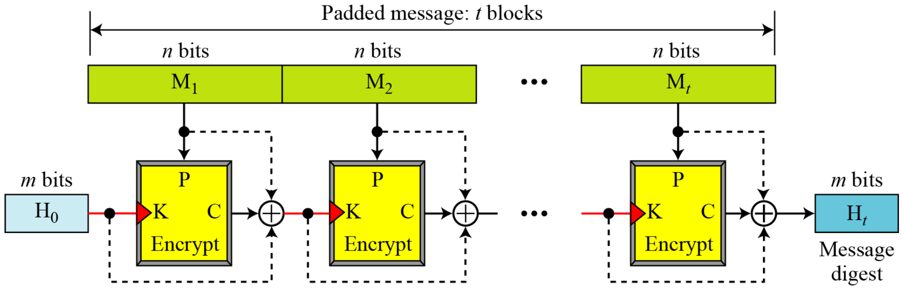


Figure 12.4: Miyaguchi-Preneel scheme

1.15

Chapter 12 — Contents

1.16

Chapter 12

12.2 SHA-512

Introduction

- SHA-512 is the version of SHA based on the *Merkle-Damgard* scheme.
- SHA-512 creates a digest of 512 bits from a multiple-block message.
- Each block is 1024 bits in length, as shown in Figure 12.5.

1.17

Chapter 12

12.2 SHA-512

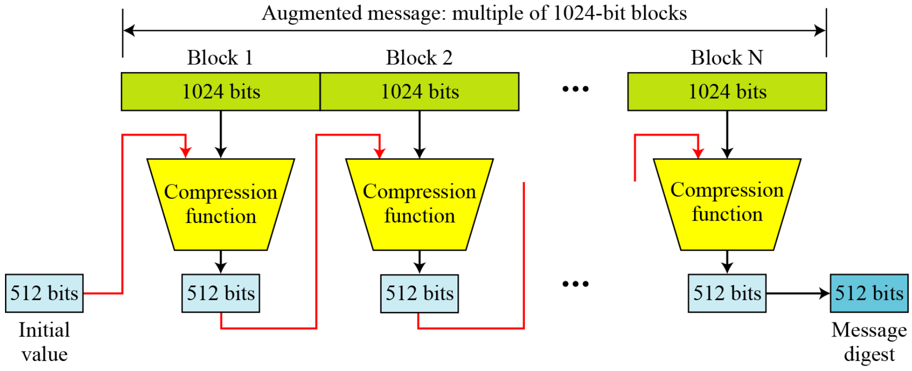


Figure 12.5: Message digest creation SHA-512

1.18

Chapter 12 | 12.2 SHA-512

Message Preparation

- SHA-512 insists that the length of the original message be less than 2128 bits.
- If the length of a message >= 2^{128}, it will not be processed.
- This is not usually a problem because is probably larger than the total storage capacity of any system.

1.19

Chapter 12 | 12.2 SHA-512

Example 12.1: This example shows that the message length limitation of SHA-512 is not a serious problem. Suppose we need to send a message that is 2^{128} bits in length.

How long does it take for a communications network with a data rate of 264 bits per second to send this message?

Solution 12.1: A communications network that can send 2^{64} bits per second is not yet available.

Even if it were, it would take many years to send this message.

This tells us that we do not need to worry about the SHA-512 message length restriction.

1.20

Chapter 12 — 12.2 SHA-512

Example 12.2: This example also concerns the message length in SHA-512.
How many pages are occupied by a message of 2^{128} bits?

Solution 12.2: Suppose that a character is 32, or 2^6, bits. Each page is less than 2048, or approximately 2^{12}, characters.

So 2^{128} bits need at least $2^{128} / 2^{18}$, or 2^{110}, pages.

This again shows that we need not worry about the message length restriction.

1.21

Chapter 12 — 12.2 SHA-512

Length Field and Padding

- Before the message digest can be created, SHA-512 requires the addition of a 128-bit unsigned-integer length field ($0 – (2^{128}-1)$) to the message that defines the length of the message in bits.
- This is the length of the original messages before padding.

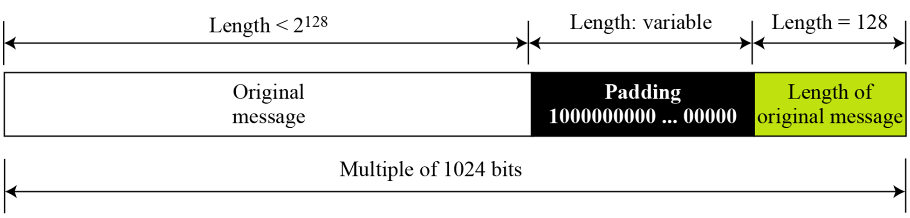


Figure 12.6: Padding and length field in SHA-512

1.22

Chapter 12 — 12.2 SHA-512

- Before the addition of the length field, we need to pad the original message to make the length a multiple of 1024.
- The length of the padding field can be calculated as follows:

$$(|M| + |P| + 128) = 0 \bmod 1024$$

$$\rightarrow |P| = (-|M| - 128) \bmod 1024$$

- |M| the length of original message.
- |P| the length of the padding field.

1.23

Chapter 12 — 12.2 SHA-512

Example 12.3: What is the number of padding bits if the length of the original message is 2590 bits?

Solution 12.3: We can calculate the number of padding bits as follows:

$$|P| = (-2590 - 128) \bmod 1024$$
$$= -2718 \bmod 1024$$
$$= 354$$

The padding consists of one 1 followed by 353 0's.

1.24

Chapter 12 — 12.2 SHA-512

Example 12.4: Do we need padding if the length of the original message is already a multiple of 1024 bits?

Solution 12.4: Yes we do, because we need to add the length field.

So padding is needed to make the new block a multiple of 1024 bits.

1.25

Chapter 12 — 12.2 SHA-512

Example 12.5: What is the minimum and maximum number of padding bits that can be added to a message?

Solution 12.5: The minimum length of padding is 0 and it happens when $(-|M| - 128) \bmod 1024$ is 0.

This means that $|M| = -128 \bmod 1024 = 896 \bmod 1024$ bits.

In other words, the last block in the original message is 896 bits.

We add a 128-bit length field to make the block complete.

1.26

Chapter 12 **12.2 SHA-512**

Solution 12.5: The maximum length of padding is 1023 and it happens when $(-|M| - 128) = 1023 \bmod 1024$.

This means that the length of the original message is $|M| = (-128 - 1023) \bmod 1024$ or the length is $|M| = 897 \bmod 1024$.

In this case, we cannot just add the length field because the length of the last block exceeds one bit more than 1024.

So we need to add 897 bits to complete this block and create a second block of 896 bits.

Now the length can be added to make this block complete.

Chapter 12 **12.2 SHA-512**

Words

- SHA-512 operates on words of 64 bits.
- This means that after the padding and the length field are added to the message, each block of the message consists of sixteen 64-bits words.
- The message digest also made of 64-bit words, but it is only eight words that named as *A, B, C, D, E, F, G*, and *H*.

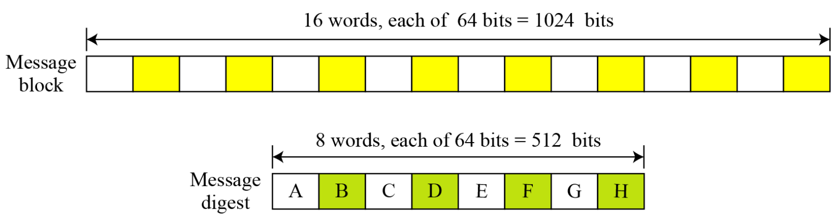


Figure 12.7: A message block and the digest as words

Chapter 12 — 12.2 SHA-512

Words Expansion

- Before processing, each message block must be expanded to 1024 bits, or sixteen 64-bit words.
- Figure 12.8 show the word-expansion process for 80 words.
- The 16-word blocks needs to be expanded to 80 words, from W_0 to W_{79}.
- The 1024-bit block become the first 16 words; the rest of the words come from already-made words according to the operation.

1.29

Chapter 12 — 12.2 SHA-512

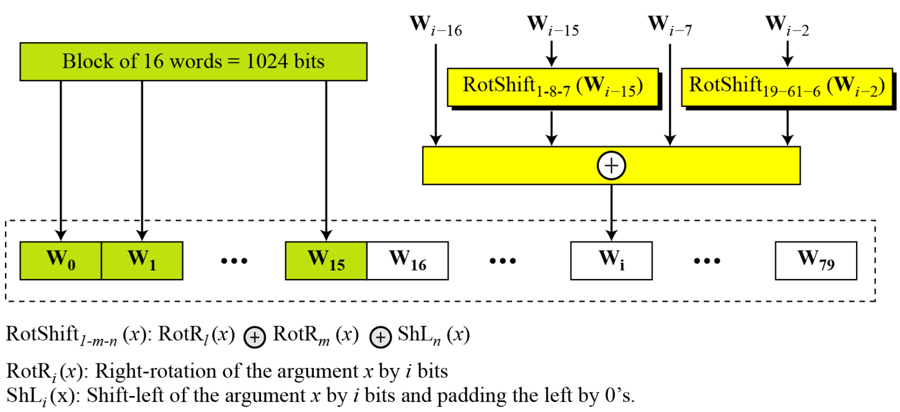


Figure 12.8: Words expansion in SHA-512

1.30

Chapter 12 — 12.2 SHA-512

Example 12.6: Show how W_{60} is made.

Solution 12.6: Each word in the range W_{16} to W_{79} is made from four previously-made words.

W_{60} is made as:

$$W_{60} = W_{44} \oplus RotShift_{1-8-7}(W_{45}) \oplus W_{53} \oplus RotShift_{19-61-6}(W_{58})$$

1.31

Chapter 12 — 12.2 SHA-512

Message Digest Initialization

- The algorithm uses eight constant for message digest initialization.
- We call these constants A_0 to H_0 to match the word naming used for the digest.

Table 12.2 *Values of constants in message digest initialization of SHA-512*

Buffer	*Value (in hexadecimal)*	*Buffer*	*Value (in hexadecimal)*
A_0	6A09E667F3BCC908	E_0	510E527FADE682D1
B_0	BB67AE8584CAA73B	F_0	9B05688C2B3E6C1F
C_0	3C6EF372FE94F828	G_0	1F83D9ABFB41BD6B
D_0	A54FE53A5F1D36F1	H_0	5BE0CD19137E2179

1.32

Chapter 12

12.2 SHA-512

Compression Function

- SHA-512 creates a 512-bit (eight 64-bit words) message digest from a multiple-block message where each block is 1024 bits.
- The processing of each block of data in SHA-512 involves 80 rounds.
- Figure 12.9 shows the general outline for the compression function.

1.33

Chapter 12

12.2 SHA-512

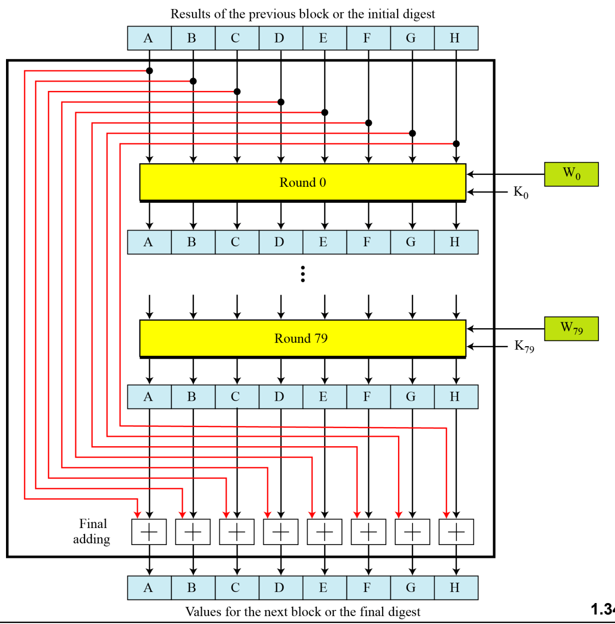


Figure 12.9: Compression function in SHA-512

1.34

Chapter 12

12.2 SHA-512

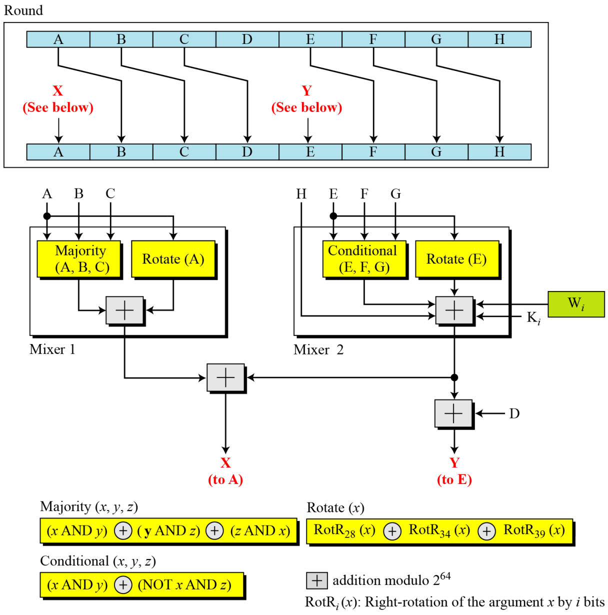


Figure 12.10: Structure of each round in SHA-512

1.35

Chapter 12

12.2 SHA-512

Majority Function:

$$(A_j \text{ AND } B_j) \oplus (B_j \text{ AND } C_j) \oplus (C_j \text{ AND } A_j)$$

Conditional Function:

$$(E_j \text{ AND } F_j) \oplus (\text{NOT } E_j \text{ AND } G_j)$$

Rotate Functions:

$$\text{Rotate (A)}: RotR_{28}(A) \oplus RotR_{34}(A) \oplus RotR_{29}(A)$$

$$\text{Rotate (E)}: RotR_{28}(E) \oplus RotR_{34}(E) \oplus RotR_{29}(E)$$

1.36

Chapter 12 — 12.2 SHA-512

Constants

- There are 80 constants, K_0 to K_{79}, each of 64 bits.
- Similar to the initial values for the eight digest buffer, these values are calculated from the first 80 prime numbers (2, 3, …, 409).
- For **example**: The 80th prime is 409, with the cubic root (409)1/3 = 7.42291412044. Converting this number to binary with only 64 bits in the fraction part, we get:

$$(111.0110\ 1100\ 0100\ 0100 \ldots 0111)_2 \rightarrow (7.6C44198C4A475817)_{16}$$

The fraction part: $(6C44198C4A475817)_{16}$

1.37

Chapter 12 — 12.2 SHA-512

Table 12.3 *Eighty constants used for eighty rounds in SHA-512*

428A2F98D728AE22	7137449123EF65CD	B5C0FBCFEC4D3B2F	E9B5DBA58189DBBC
3956C25BF348B538	59F111F1B605D019	923F82A4AF194F9B	AB1C5ED5DA6D8118
D807AA98A3030242	12835B0145706FBE	243185BE4EE4B28C	550C7DC3D5FFB4E2
72BE5D74F27B896F	80DEB1FE3B1696B1	9BDC06A725C71235	C19BF174CF692694
E49B69C19EF14AD2	EFBE4786384F25E3	0FC19DC68B8CD5B5	240CA1CC77AC9C65
2DE92C6F592B0275	4A7484AA6EA6E483	5CB0A9DCBD41FBD4	76F988DA831153B5
983E5152EE66DFAB	A831C66D2DB43210	B00327C898FB213F	BF597FC7BEEF0EE4
C6E00BF33DA88FC2	D5A79147930AA725	06CA6351E003826F	142929670A0E6E70
27B70A8546D22FFC	2E1B21385C26C926	4D2C6DFC5AC42AED	53380D139D95B3DF
650A73548BAF63DE	766A0ABB3C77B2A8	81C2C92E47EDAEE6	92722C851482353B
A2BFE8A14CF10364	A81A664BBC423001	C24B8B70D0F89791	C76C51A30654BE30
D192E819D6EF5218	D69906245565A910	F40E35855771202A	106AA07032BBD1B8
19A4C116B8D2D0C8	1E376C085141AB53	2748774CDF8EEB99	34B0BCB5E19B48A8
391C0CB3C5C95A63	4ED8AA4AE3418ACB	5B9CCA4F7763E373	682E6FF3D6B2B8A3
748F82EE5DEFB2FC	78A5636F43172F60	84C87814A1F0AB72	8CC702081A6439EC
90BEFFFA23631E28	A4506CEBDE82BDE9	BEF9A3F7B2C67915	C67178F2E372532B
CA273ECEEA26619C	D186B8C721C0C207	EADA7DD6CDE0EB1E	F57D4F7FEE6ED178
06F067AA72176FBA	0A637DC5A2C898A6	113F9804BEF90DAE	1B710B35131C471B
28DB77F523047D84	32CAAB7B40C72493	3C9EBE0A15C9BEBC	431D67C49C100D4C
4CC5D4BECB3E42B6	4597F299CFC657E2	5FCB6FAB3AD6FAEC	6C44198C4A475817

1.38

Chapter 12 — 12.2 SHA-512

Example 12.7: We apply the Majority function on buffers *A*, *B*, and *C*. If the leftmost hexadecimal digits of these buffers are `0x7`, `0xA`, and `0xE`, respectively, what is the leftmost digit of the result?

Solution 12.7: The digits in binary are `0111`, `1010`, and `1110`.

a. The first bits are 0, 1, and 1. The majority is 1.

b. The second bits are 1, 0, and 1. The majority is 1.

c. The third bits are 1, 1, and 1. The majority is 1.

d. The fourth bits are 1, 0, and 0. The majority is 0.

The result is `1110`, or `0xE` in hexadecimal.

1.39

Chapter 12 — 12.2 SHA-512

Example 12.8: We apply the Conditional function on E, F, and G buffers. If the leftmost hexadecimal digits of these buffers are `0x9`, `0xA`, and `0xF` respectively, what is the leftmost digit of the result?

Solution 12.8: The digits in binary are `1001`, `1010`, and `1111`.

a. The first bits are 1, 1, and 1. The result is F_1, which is 1.

b. The second bits are 0, 0, and 1. The result is G_2, which is 1.

c. The third bits are 0, 1, and 1. The result is G_3, which is 1.

d. The fourth bits are 1, 0, and 1. The result is F_4, which is 0.

The result is `1110`, or `0xE` in hexadecimal.

1.40

Chapter 12

12.2 SHA-512

Analysis

- With a message digest of 512 bits, SHA-512 expected to be resistant to all attacks, including collision attacks.
- It has been claimed that this version's improved design makes it more efficient and more secure than the previous versions.
- However, more research and testing are needed to confirm this claim.

1.41

Chapter 12

Contents

1.42

Chapter 12

12.3 Summary

- All cryptographic hash functions must create a **fixed-size digest** out of a variable-size message.
  - Creating such a function is best accomplished using **iteration**.
  - A compression function is repeatedly used to create the digest → **iterated hash function**.

- The *Merkle-Damgard* scheme is an iterated cryptographic hash function that is **collision resistant** if the compression function is collision resistant.
  - Basis for many cryptographic hash functions today.

1.43

Chapter 12

12.3 Summary

- There is a tendency to use two different approaches in designing the compression function that:
  - made from scratch.
  - based on block-cipher.

- A set of cryptographic hash functions uses compression functions that are made from scratch such as *Message Digest* (MD) group and *Secure Hash Algorithm* (SHA) group.

- SHA-512 is one of the promisingcryptographic hash function with 512-bit message digest that based on the *Merkle-Damgard* scheme.

1.44

Chapter 12 Exercises

Exercise 12.1: In SHA-512, show the value of the length field in hexadecimal for the following message lengths:

a) 1000 bits.
b) 10,000 bits.
c) 1,000,000 bits.

Forouzan,B.A. Cryptography and Network Security (International Edition). Singapore: McGraw-Hill, 2008. (page 386)

Chapter 12 Exercises

Exercise 12.2: What is the padding for SHA-512 if the length of the message is:

a) 5120 bits.
b) 5121 bits.
c) 6143 bits.

Forouzan,B.A. Cryptography and Network Security (International Edition). Singapore: McGraw-Hill, 2008. (page 386)