

Public Key Cryptography: Hash Functions

Additional Material from Textbook

COMP90043

Lecture 1

Public Key Cryptography: Diffie-Hellman and RSA



Lecture 1

- 1.1 Concept of Public Key
 - Limitations of Symmetric key system
 - Notations for Public key
- 1.2 Diffie-Hellman Protocol
 - Motivation
 - The protocol and Implications
 - Man in the Middle Attack
- 1.3 RSA Idea
 - Informal Idea
 - RSA Algorithm
 - Attacks on RSA

Secure Hash Algorithm (SHA)

- SHA was originally designed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993
- Was revised in 1995 as SHA-1
- Based on the hash function MD4 and its design closely models MD4
- Produces 160-bit hash values
- In 2002 NIST produced a revised version of the standard that defined three new versions of SHA with hash value lengths of 256, 384, and 512
 - Collectively known as SHA-2

Table 11.3

Comparison of SHA Parameters

Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Note: All sizes are measured in bits.

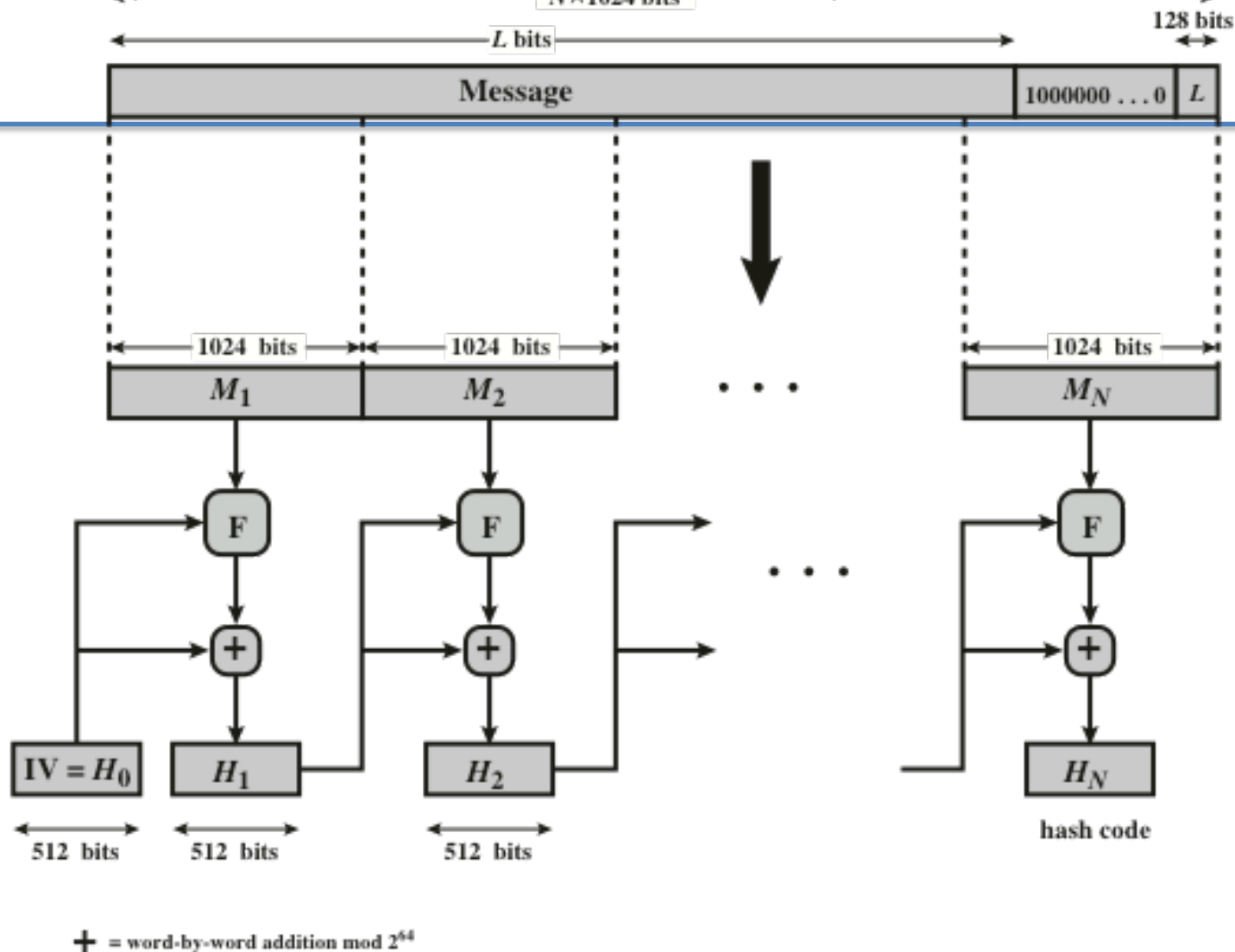


Figure 11.9 Message Digest Generation Using SHA-512

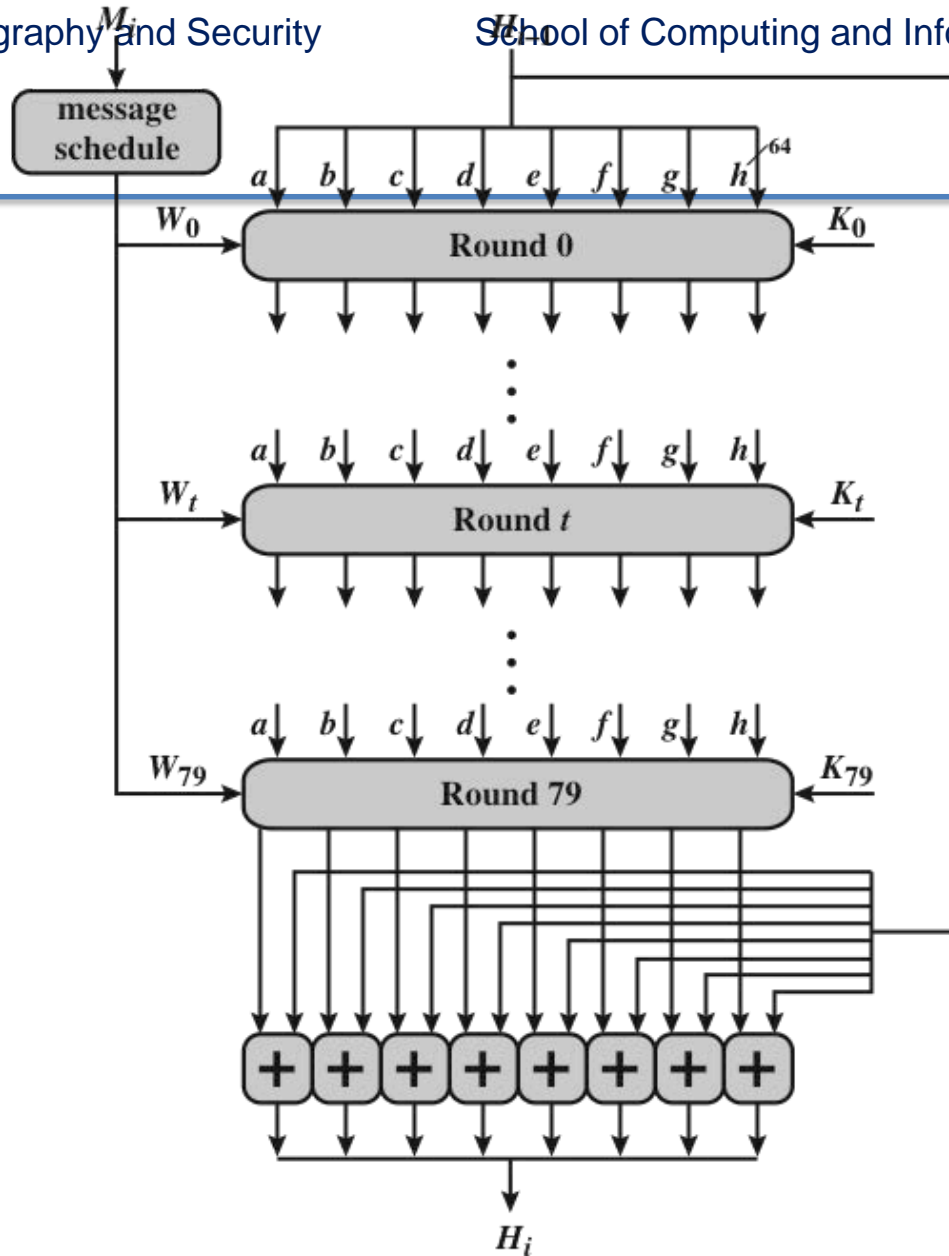


Figure 11.10 SHA-512 Processing of a Single 1024-Bit Block

Table 11.4 ---- SHA-512 Constants

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240calcc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbbd41fbd4	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
90beffffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceeaa26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c
4cc5d4becb3e42b6	597f299cfc657e2a	5fcb6fab3ad6faec	6c44198c4a475817

(Table
can be
found on
page 341
in

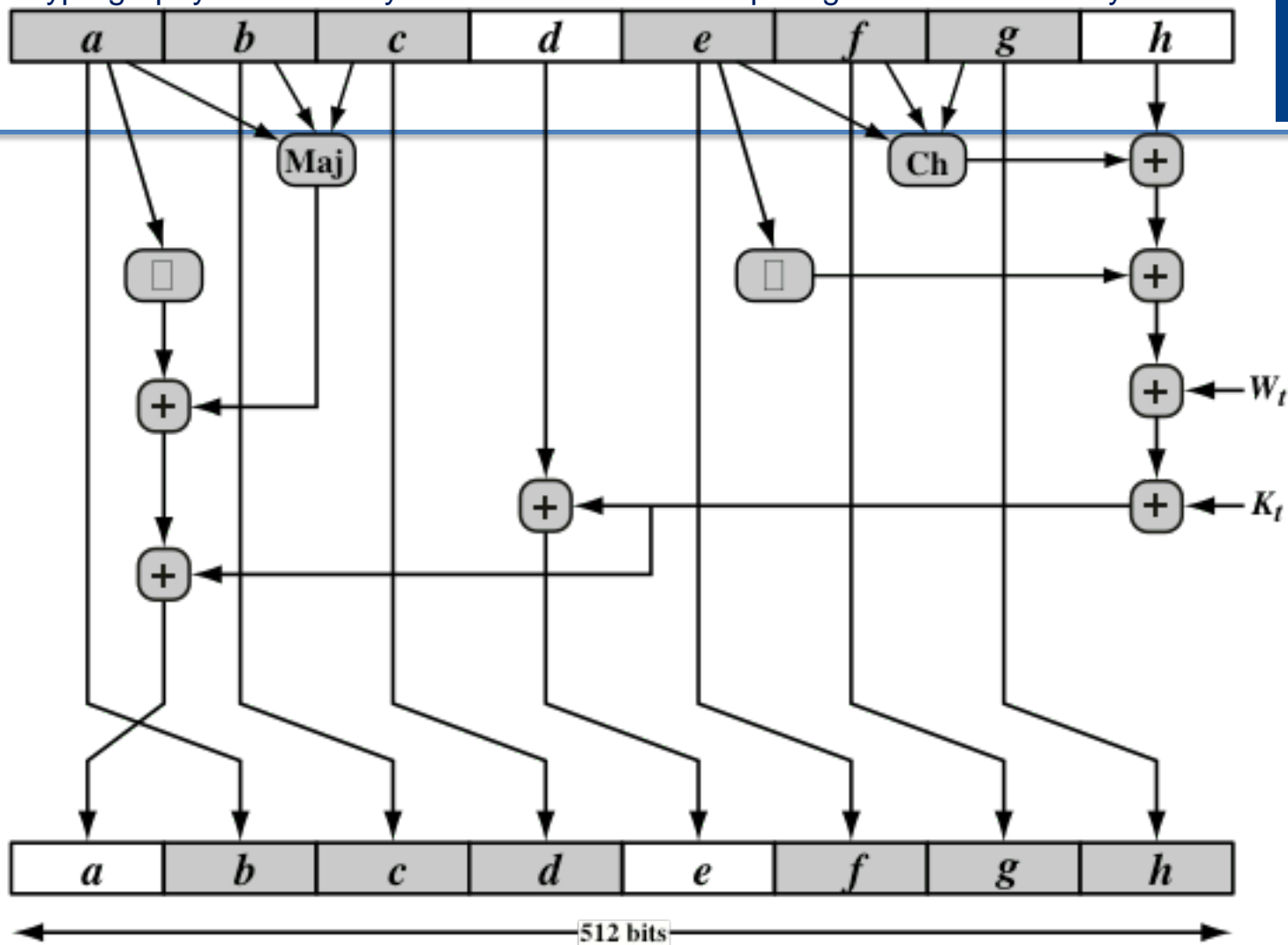


Figure 11.11 Elementary SHA-512 Operation (single round)

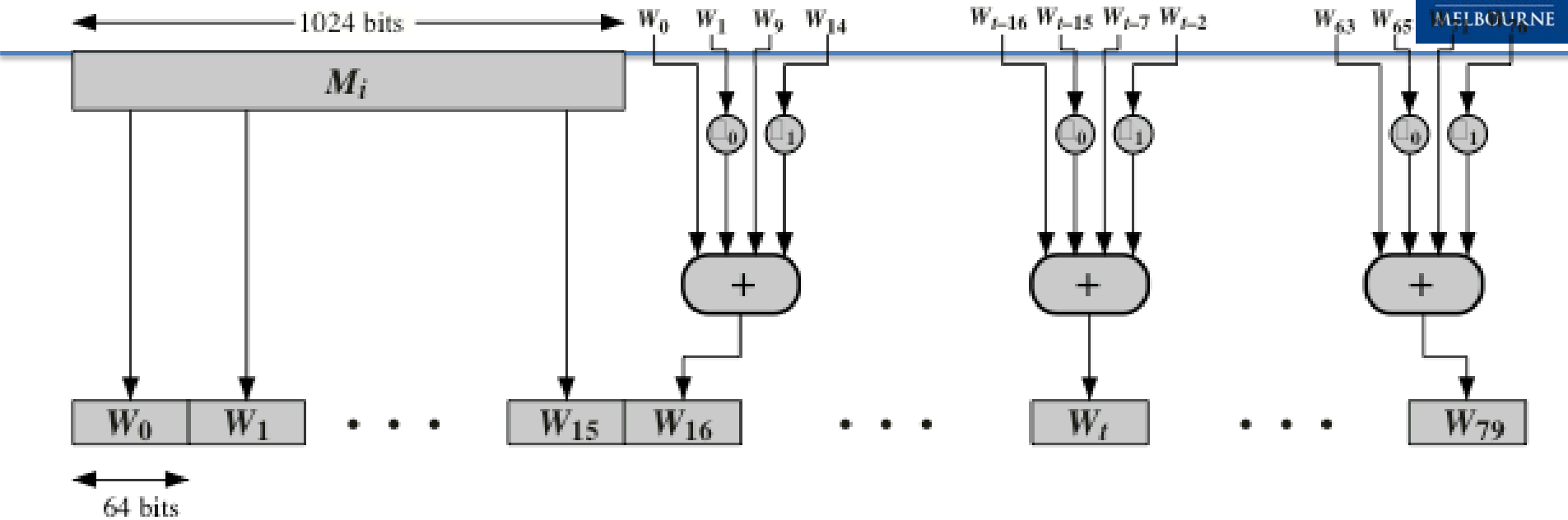


Figure 11.12 Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

The padded message consists blocks M_1, M_2, \dots, M_N . Each message block M_i consists of 16 64-bit words $M_{i,0}, M_{i,1} \dots M_{i,15}$. All addition is performed modulo 2^{64} .

$$\begin{array}{ll} H_{0,0} = 6A09E667F3BCC908 & H_{0,4} = 510E527FADE682D1 \\ H_{0,1} = BB67AE8584CAA73B & H_{0,5} = 9B05688C2B3E6C1F \\ H_{0,2} = 3C6EF372FE94F82B & H_{0,6} = 1F83D9ABFB41BD6B \\ H_{0,3} = A54FF53A5F1D36F1 & H_{0,7} = 5BE0CDI9137E2179 \end{array}$$

for $i = 1$ **to** N

1. Prepare the message schedule W :

for $t = 0$ **to** 15

$$W_t = M_{i,t}$$

for $t = 16$ **to** 79

$$W_t = \alpha^{512}(W_{t-2}) + W_{t-7} + \alpha^{512}(W_{t-15}) + W_{t-16}$$

2. Initialize the working variables

$$a = H_{i-1,0} \quad e = H_{i-1,4}$$

$$b = H_{i-1,1} \quad f = H_{i-1,5}$$

$$c = H_{i-1,2} \quad g = H_{i-1,6}$$

$$d = H_{i-1,3} \quad h = H_{i-1,7}$$

3. Perform the main hash computation

for $t = 0$ **to** 79

$$T_1 = h + \text{Ch}(e, f, g) + \left(\sum_1^{512} e \right) + W_t + K_t$$

$$T_2 = \left(\sum_0^{512} a \right) + \text{Maj}(a, b, c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

4. Compute the intermediate hash value

$$H_{i,0} = a + H_{i-1,0} \quad H_{i,4} = e + H_{i-1,4}$$

$$H_{i,1} = b + H_{i-1,1} \quad H_{i,5} = f + H_{i-1,5}$$

$$H_{i,2} = c + H_{i-1,2} \quad H_{i,6} = g + H_{i-1,6}$$

$$H_{i,3} = d + H_{i-1,3} \quad H_{i,7} = h + H_{i-1,7}$$

return $\{H_{N,0} \parallel H_{N,1} \parallel H_{N,2} \parallel H_{N,3} \parallel H_{N,4} \parallel H_{N,5} \parallel H_{N,6} \parallel H_{N,7}\}$

(Figure can be found
on page 345 in
textbook)

Figure 11.13 SHA-512 Logic

SHA-3

SHA-1 has not yet been "broken"

- No one has demonstrated a technique for producing collisions in a practical amount of time
- Considered to be insecure and has been phased out for SHA-2



NIST announced in 2007 a competition for the SHA-3 next generation NIST hash function

- Winning design was announced by NIST in October 2012
- SHA-3 is a cryptographic hash function that is intended to complement SHA-2 as the approved standard for a wide range of applications

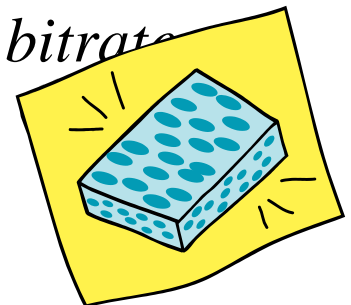
SHA-2 shares the same structure and mathematical operations as its predecessors so this is a cause for concern

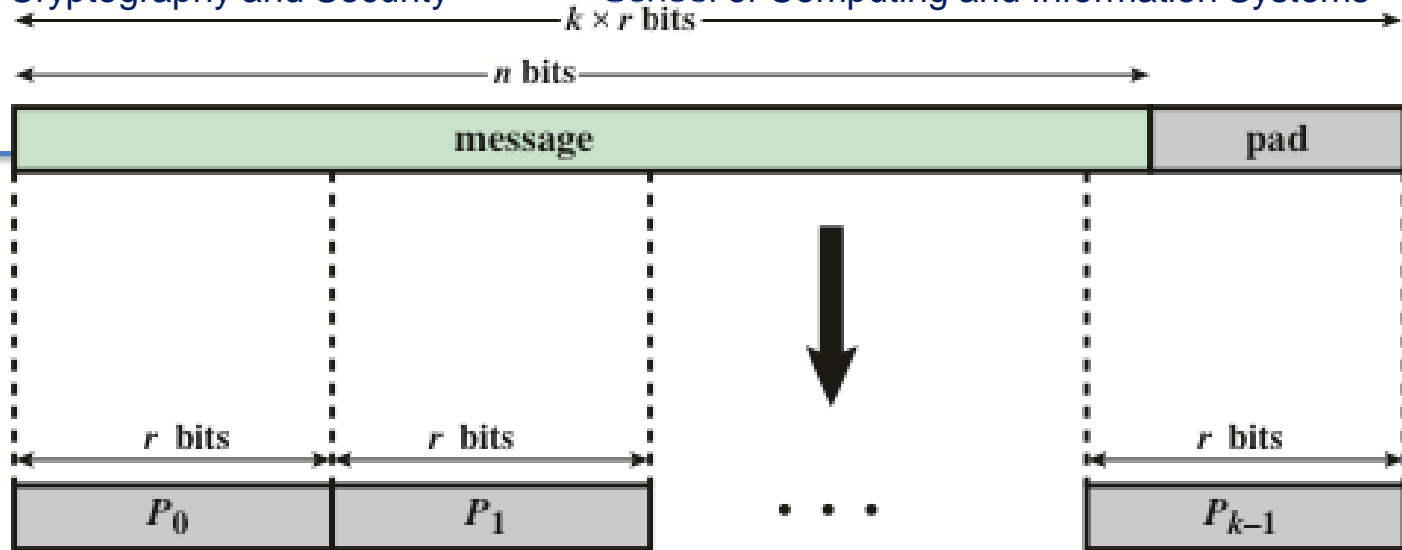
- Because it will take years to find a suitable replacement for SHA-2 should it become vulnerable, NIST decided to begin the process of developing a new hash standard



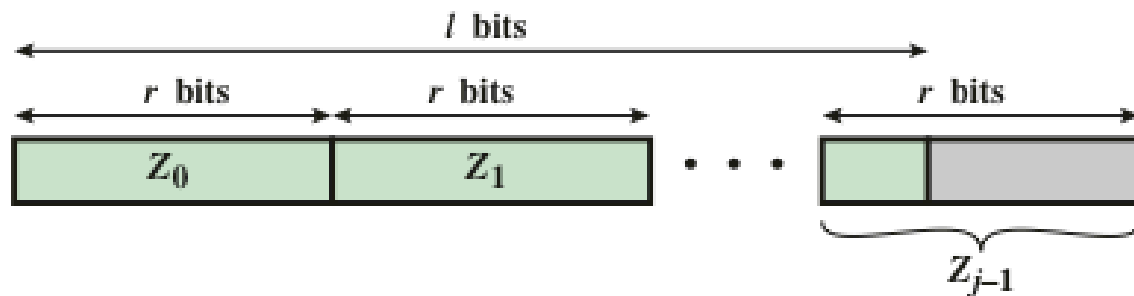
The Sponge Construction

- Underlying structure of SHA-3 is a scheme referred to by its designers as a *sponge construction*
- Takes an input message and partitions it into fixed-size blocks
- Each block is processed in turn with the output of each iteration fed into the next iteration, finally producing an output block
- The sponge function is defined by three parameters:
 - f = the internal function used to process each input block
 - r = the size in bits of the input blocks, called the *bitrate*
 - pad = the padding algorithm





(a) Input



(b) Output

Figure 11.14 Sponge Function Input and Output

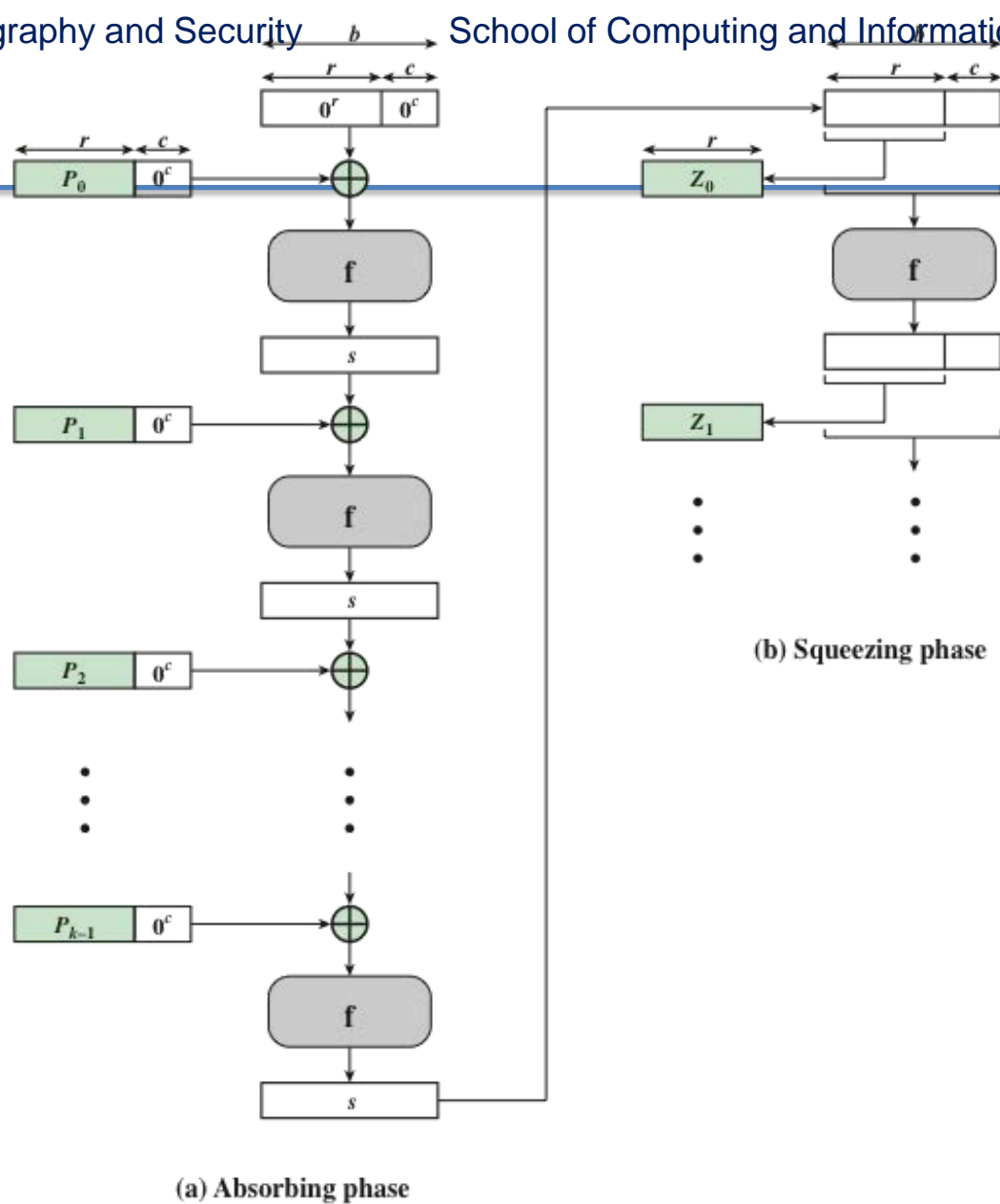


Figure 11.15 Sponge Construction

Table 11.5

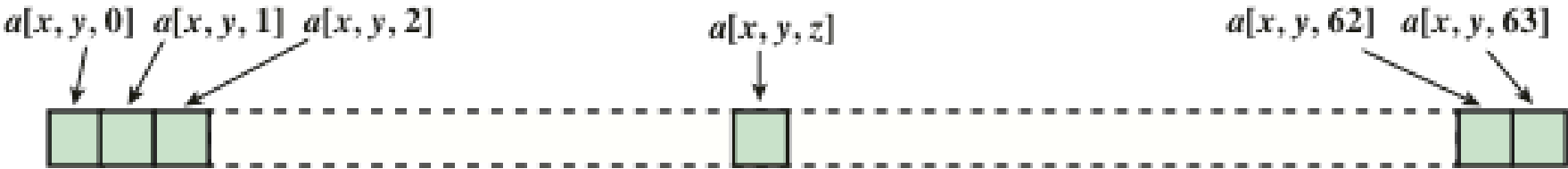
SHA-3 Parameters

Message Digest Size	224	256	384	512
Message Size	no maximum	no maximum	no maximum	no maximum
Block Size (bitrate r)	1152	1088	832	576
Word Size	64	64	64	64
Number of Rounds	24	24	24	24
Capacity c	448	512	768	1024
Collision resistance	2^{112}	2^{128}	2^{192}	2^{256}
Second preimage resistance	2^{224}	2^{256}	2^{384}	2^{512}



	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$
$y = 4$	$L[0, 4]$	$L[1, 4]$	$L[2, 4]$	$L[3, 4]$	$L[4, 4]$
$y = 3$	$L[0, 3]$	$L[1, 3]$	$L[2, 3]$	$L[3, 3]$	$L[4, 3]$
$y = 2$	$L[0, 2]$	$L[1, 2]$	$L[2, 2]$	$L[3, 2]$	$L[4, 2]$
$y = 1$	$L[0, 1]$	$L[1, 1]$	$L[2, 1]$	$L[3, 1]$	$L[4, 1]$
$y = 0$	$L[0, 0]$	$L[1, 0]$	$L[2, 0]$	$L[3, 0]$	$L[4, 0]$

(a) State variable as 5 × 5 matrix A of 64-bit words



(b) Bit labeling of 64-bit words

Figure 11.16 SHA-3 State Matrix

SHA-3 Iteration Function f

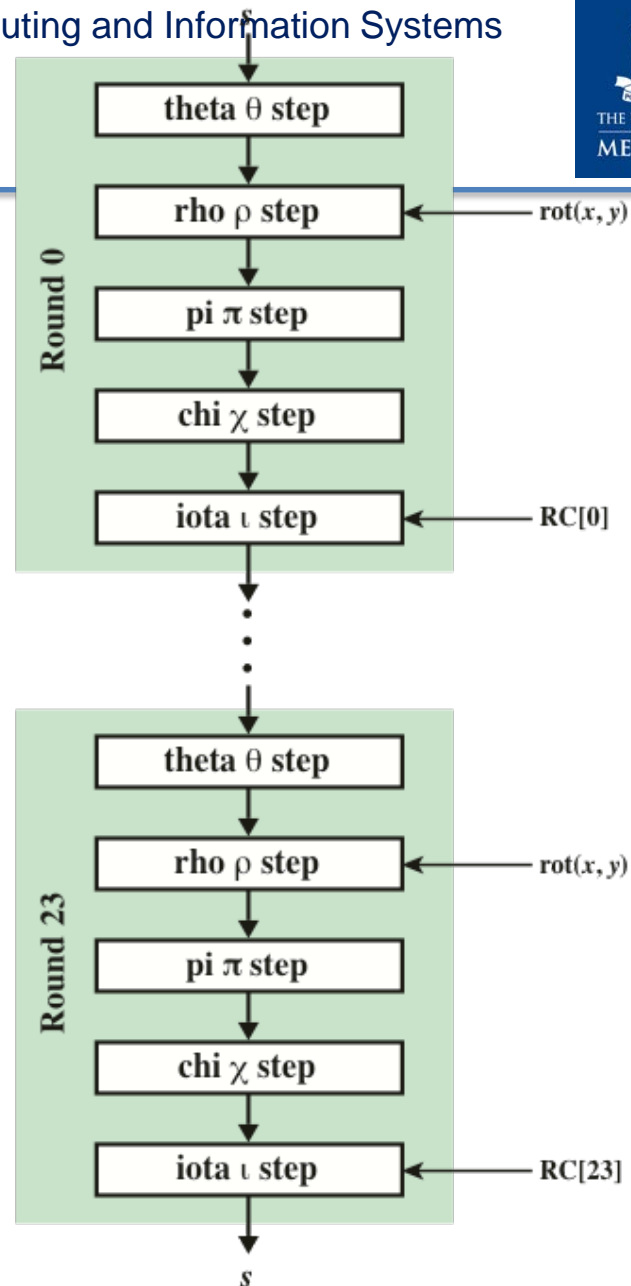


Figure 11.17 SHA-3 Iteration Function f

Table 11.6

Step Functions in SHA-3

Function	Type	Description
θ	Substitution	New value of each bit in each word depends its current value and on one bit in each word of preceding column and one bit of each word in succeeding column.
ρ	Permutation	The bits of each word are permuted using a circular bit shift. $W[0, 0]$ is not affected.
π	Permutation	Words are permuted in the 5×5 matrix. $W[0, 0]$ is not affected.
χ	Substitution	New value of each bit in each word depends on its current value and on one bit in next word in the same row and one bit in the second next word in the same row.
ι	Substitution	$W[0, 0]$ is updated by XOR with a round constant.

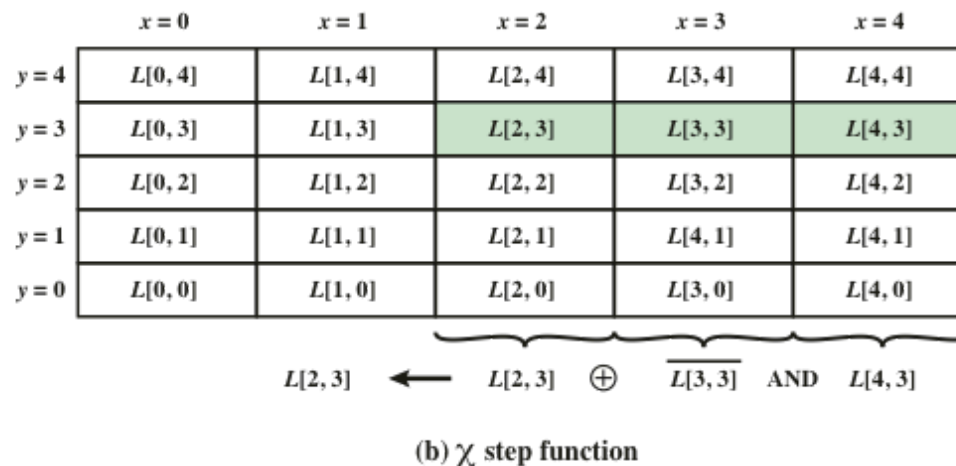
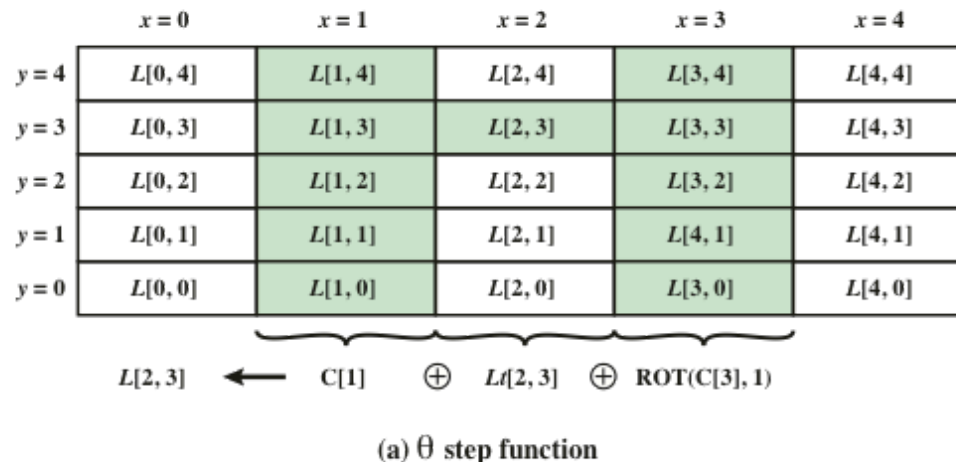
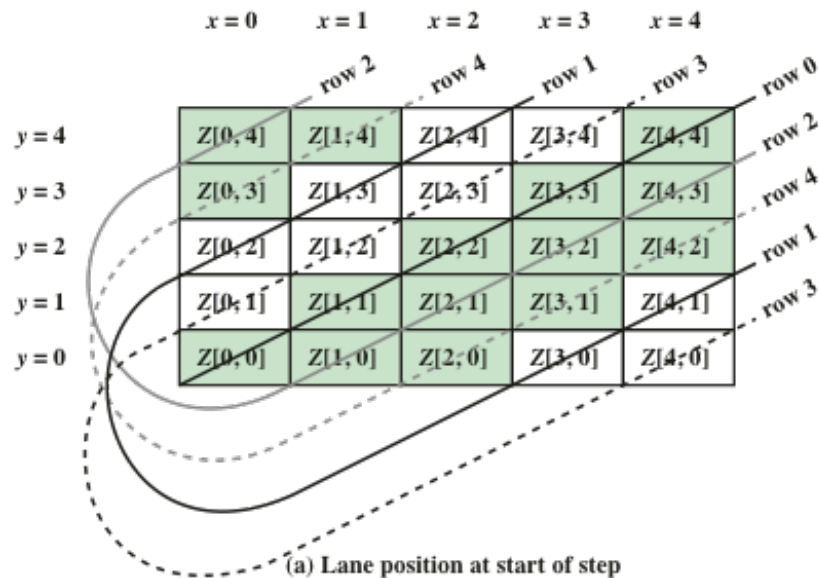


Figure 11.18 Theta and Chi Step Functions



	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$
$y = 4$	$Z[2, 0]$	$Z[3, 1]$	$Z[4, 2]$	$Z[0, 3]$	$Z[1, 4]$
$y = 3$	$Z[4, 0]$	$Z[0, 1]$	$Z[1, 2]$	$Z[2, 3]$	$Z[3, 4]$
$y = 2$	$Z[1, 0]$	$Z[2, 1]$	$Z[3, 2]$	$Z[4, 3]$	$Z[0, 4]$
$y = 1$	$Z[3, 0]$	$Z[4, 1]$	$Z[0, 2]$	$Z[1, 3]$	$Z[2, 4]$
$y = 0$	$Z[0, 0]$	$Z[1, 1]$	$Z[2, 2]$	$Z[3, 3]$	$Z[4, 4]$

(b) Lane position after permutation

Figure 11.19 Pi Step Function

Table 11.8

Round Constants in SHA-3

Round	Constant (hexadecimal)	Number of 1 bits
12	0000000008000808B	6
13	8000000000000008B	5
14	80000000000008089	5
15	80000000000008003	4
16	80000000000008002	3
17	80000000000000080	2
18	0000000000000800A	3
19	8000000008000000A	4
20	80000000080008081	5
21	80000000000008080	3
22	00000000080000001	2
23	80000000080008008	4