

CHAPTER 2 CRYPTOGRAPHIC TOOLS

ANSWERS TO QUESTIONS

- 2.1** Plaintext, encryption algorithm, secret key, ciphertext, decryption algorithm.
- 2.2** One secret key.
- 2.3** (1) a strong encryption algorithm; (2) Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.
- 2.4** Message encryption, message authentication code, hash function.
- 2.5** An authenticator that is a cryptographic function of both the data to be authenticated and a secret key.
- 2.6** **(a)** A hash code is computed from the source message, encrypted using symmetric encryption and a secret key, and appended to the message. At the receiver, the same hash code is computed. The incoming code is decrypted using the same key and compared with the computed hash code. **(b)** This is the same procedure as in (a) except that public-key encryption is used; the sender encrypts the hash code with the sender's private key, and the receiver decrypts the hash code with the sender's public key. **(c)** A secret value is appended to a message and then a hash code is calculated using the message plus secret value as input. Then the message (without the secret value) and the hash code are transmitted. The receiver appends the same secret value to the message and computes the hash value over the message plus secret value. This is then compared to the received hash code.
- 2.7**
- 1.** H can be applied to a block of data of any size.
 - 2.** H produces a fixed-length output.
 - 3.** $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.
 - 4.** For any given value h , it is computationally infeasible to find x such that $H(x) = h$.
 - 5.** For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$.

6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

2.8 Plaintext: This is the readable message or data that is fed into the algorithm as input. **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext. **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input. **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts. **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

2.9 Encryption/decryption: The sender encrypts a message with the recipient's public key. **Digital signature:** The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message. **Key exchange:** Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

2.10 The key used in conventional encryption is typically referred to as a **secret key**. The two keys used for public-key encryption are referred to as the **public key** and the **private key**.

2.11 A **digital signature** is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

2.12 A **public-key certificate** consists of a public key plus a User ID of the key owner, with the whole block signed by a trusted third party. Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution.

2.13 Several different approaches are possible, involving the private key(s) of one or both parties. One approach is Diffie-Hellman key exchange. Another approach is for the sender to encrypt a secret key with the recipient's public key.

ANSWERS TO PROBLEMS

2.1 Yes. The eavesdropper is left with two strings, one sent in each direction, and their XOR is the secret key.

2.2 a.

	2	8	10	7	9	6	3	1	4	5
C	R	Y	P	T	O	G	A	H	I	
B	E	A	T	T	H	E	T	H	I	
R	D	P	I	L	L	A	R	F	R	
O	M	T	H	E	L	E	F	T	O	
U	T	S	I	D	E	T	H	E	L	
Y	C	E	U	M	T	H	E	A	T	
R	E	T	O	N	I	G	H	T	A	
T	S	E	V	E	N	I	F	Y	O	
U	A	R	E	D	I	S	T	R	U	
S	T	F	U	L	B	R	I	N	G	
T	W	O	F	R	I	E	N	D	S	

	4	2	8	10	5	6	3	7	1	9
N	E	T	W	O	R	K	S	C	U	
T	R	F	H	E	H	F	T	I	N	
B	R	O	U	Y	R	T	U	S	T	
E	A	E	T	H	G	I	S	R	E	
H	F	T	E	A	T	Y	R	N	D	
I	R	O	L	T	A	O	U	G	S	
H	L	L	E	T	I	N	I	B	I	
T	I	H	I	U	O	V	E	U	F	
E	D	M	T	C	E	S	A	T	W	
T	L	E	D	M	N	E	D	L	R	
A	P	T	S	E	T	E	R	F	O	

ISRNG BUTLF RRAFR LIDL P FTIYO NVSEE TBEHI HTETA
 EYHAT TUCME HRGTA IOENT TUSRU IEADR FOETO LHMET
 NTEDS IFWRO HUTEL EITDS

- b.** The two matrices are used in reverse order. First, the ciphertext is laid out in columns in the second matrix, taking into account the order dictated by the second memory word. Then, the contents of the second matrix are read left to right, top to bottom and laid out in columns in the first matrix, taking into account the order dictated by the first memory word. The plaintext is then read left to right, top to bottom.
- c.** Although this is a weak method, it may have use with time-sensitive information and an adversary without immediate access to good cryptanalysis (e.g., tactical use). Plus it doesn't require anything more than paper and pencil, and can be easily remembered.

2.3 a. Let $-X$ be the additive inverse of X . That is $-X \oplus X = 0$. Then:

$$P = (C \oplus -K_1) \oplus K_0$$

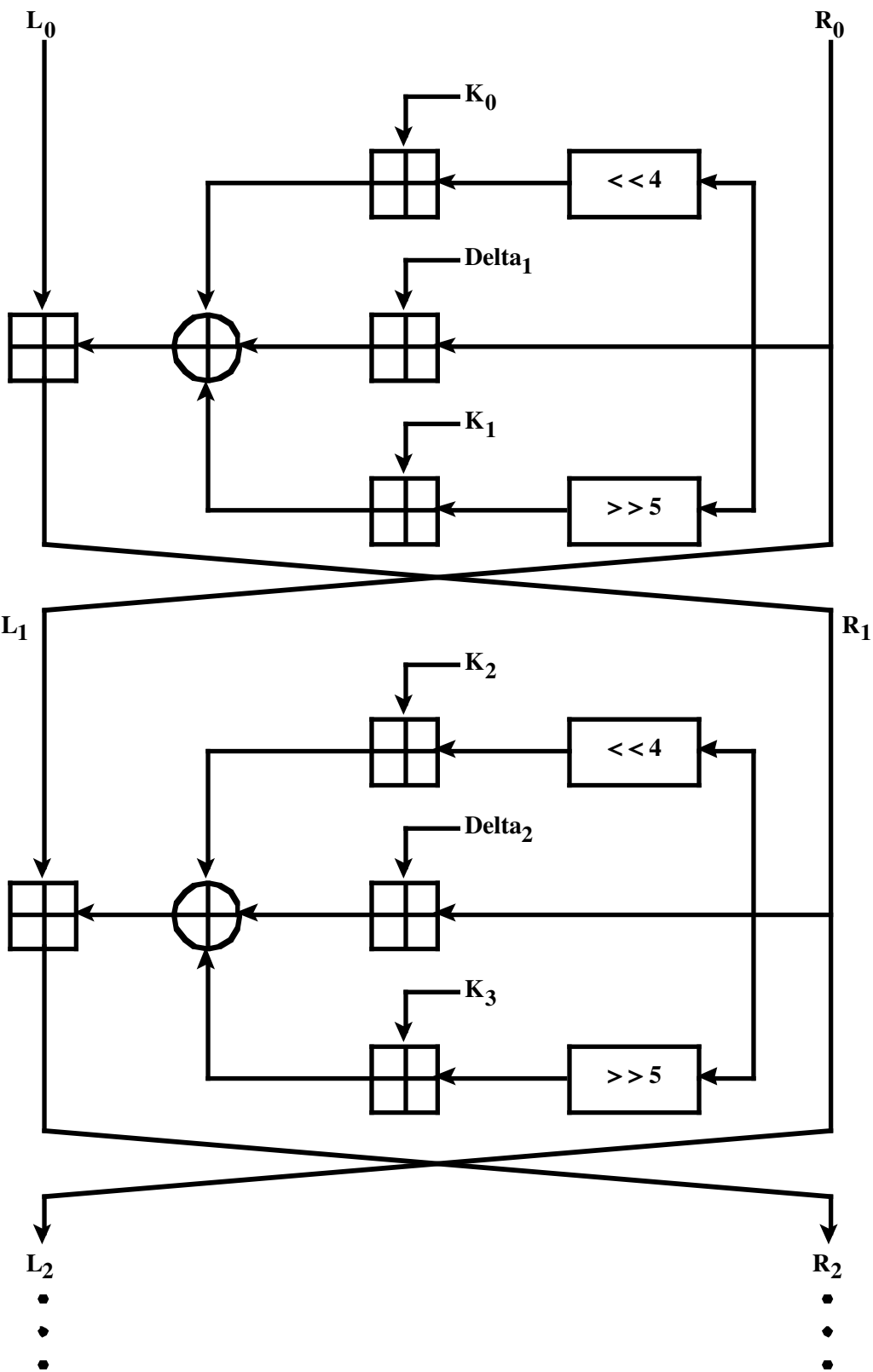
b. First, calculate $-C'$. Then $-C' = (P' \oplus K_0) \oplus (-K_1)$. We then have:

$$C \oplus -C' = (P \oplus K_0) \oplus (P' \oplus K_0)$$

However, the operations \oplus and \oplus are not associative or distributive with one another, so it is not possible to solve this equation for K_0 .

2.4 a. The constants ensure that encryption/decryption in each round is different.

b. First two rounds:



c. First, let's define the encryption process:

$$L_2 = L_0 \oplus [(R_0 \ll 4) \oplus K_0] \oplus [R_0 \oplus \delta_1] \oplus [(R_0 \gg 5) \oplus K_1]$$

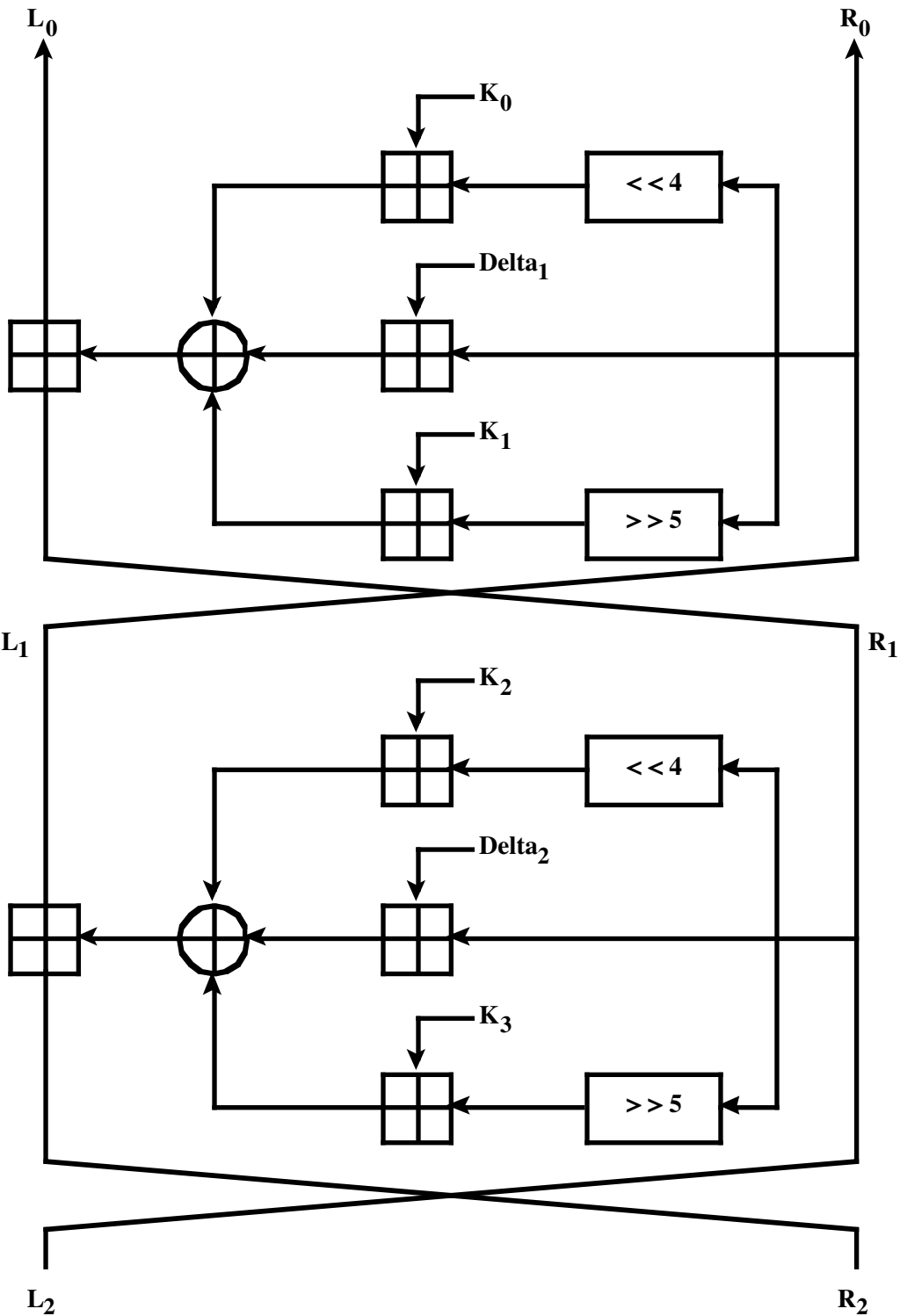
$$R_2 = R_0 \oplus [(L_2 \ll 4) \oplus K_2] \oplus [L_2 \oplus \delta_2] \oplus [(L_2 \gg 5) \oplus K_3]$$

Now the decryption process. The input is the ciphertext (L_2, R_2) , and the output is the plaintext (L_0, R_0) . Decryption is essentially the same as encryption, with the subkeys and delta values applied in reverse order. Also note that it is not necessary to use subtraction because there is an even number of additions in each equation.

$$R_0 = R_2 \oplus [(L_2 \ll 4) \oplus K_2] \oplus [L_2 \oplus \delta_2] \oplus [(L_2 \gg 5) \oplus K_3]$$

$$L_0 = L_2 \oplus [(R_0 \ll 4) \oplus K_0] \oplus [R_0 \oplus \delta_1] \oplus [(R_0 \gg 5) \oplus K_1]$$

d.

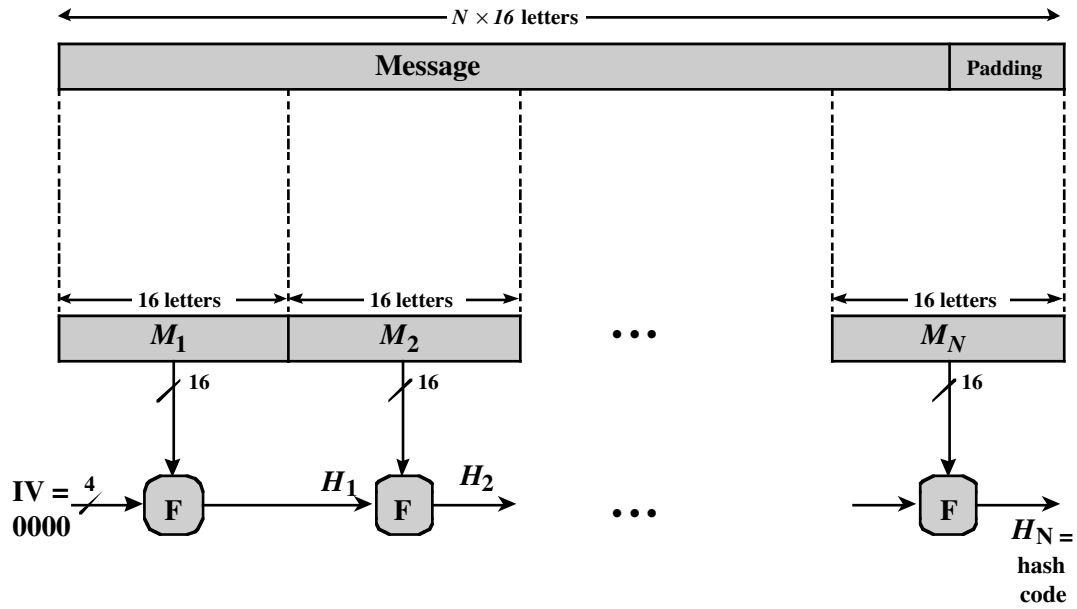


- 2.5 a.** Will be detected with both (i) DS and (ii) MAC.
b. Won't be detected by either (Remark: use timestamps).
c. (i) DS: Bob simply has to verify the message with the public key from both. Obviously, only Alice's public key results in a successful verification.

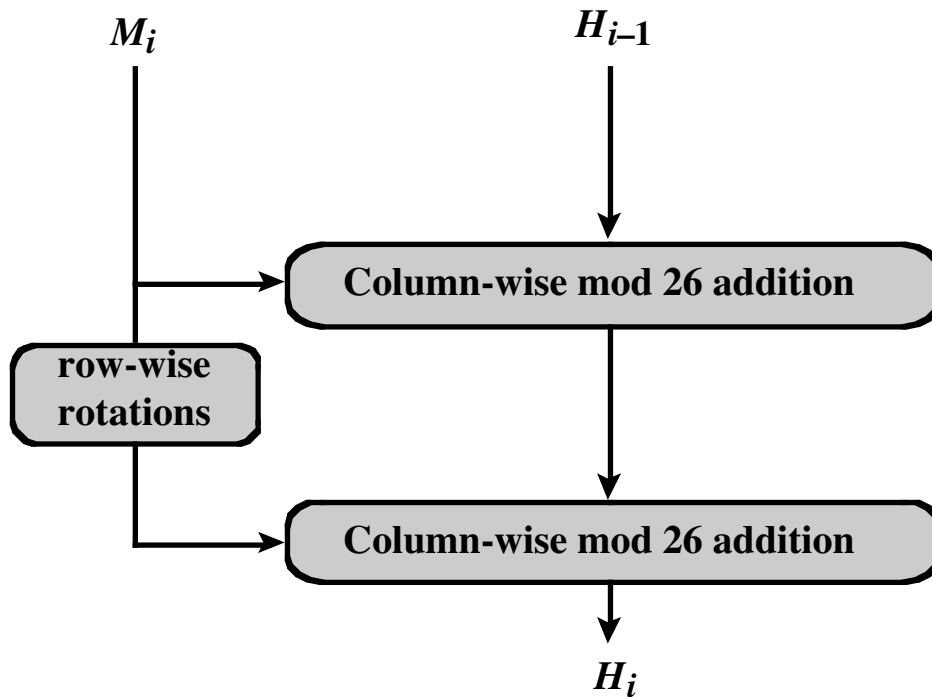
- (ii) MAC: Bob has to challenge both, Oscar and Bob, to reveal their secret key to him (which he knows anyway). Only Bob can do that.
- d.** (i) DS: Alice has to force Bob to prove his claim by sending her a copy of the message in question with the signature. Then Alice can show that message and signature can be verified with Bob's public key) Bob must have generated the message.
- (ii) MAC: No, Bob can claim that Alice generated this message.

2.6 The statement is false. Such a function cannot be one-to-one because the number of inputs to the function is of arbitrary, but the number of unique outputs is 2^n . Thus, there are multiple inputs that map into the same output.

2.7 a. Overall structure:



Compression function F :



b. BFQQ

c. Simple algebra is all you need to generate a result:

```

AYHGDAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
    
```

2.8

a. $M3 =$

5	2	1	4	5
1	4	3	2	2
3	1	2	5	3
4	3	4	1	4
2	5	5	3	1

- b. Assume a plaintext message p is to be encrypted by Alice and sent to Bob. Bob makes use of $M1$ and $M3$, and Alice makes use of $M2$. Bob chooses a random number, k , as his private key, and maps k by $M1$ to get x , which he sends as his public key to Alice. Alice uses x to encrypt p with $M2$ to get z , the ciphertext, which she sends to Bob. Bob uses k to decrypt z by means of $M3$, yielding the plaintext message p .
- c. If the numbers are large enough, and $M1$ and $M2$ are sufficiently random to make it impractical to work backwards, p cannot be found without knowing k .

2.9 We show the creation of a digital envelope:

